

**Online students:** *There is an assignment based on this module and the "Web and HCI" module. Your assignment is due on Friday February 2nd. There is a lecture video that will soon be available for this module -- you may watch the video, or read through the content.*

**On campus students:** *There are lecture videos available for this module -- the material will be on your in class quiz for "XHTML/Web Development".*

## **XHTML**

XHTML stands for "Extensible HyperText Markup Language" - it is the language used to create webpages. This module introduces you to the structure of XHTML files. It demonstrates how to use several of the XHTML elements (tags) that are used to format the content of webpages, and shows how to create hyperlinks - the component that actually links the pages of the World Wide Web. Techniques to organize content and make pages more appealing will be explored; these techniques include lists and tables. The module also explains how to create forms on your webpages, which when combined with JavaScript will get information from your users.

Finally, the module explains some of the common errors that people make when creating XHTML files and how to go about correcting these errors.

By the end of this module, you will be able to create your own simple website, if you choose to do so.

## **Nvu Editor**

Nvu is a WYSIWYG (What You See Is What You Get) Web Page editor. In it, you begin with a blank document where you can add text, images, tables, lists and other XHTML elements just like you would to a word document. Nvu will then generate XHTML code from your document you created so that you will have a page that is ready to be viewed on the web.

This lesson will show you how to install Nvu and then describes the different features of the editor. You may not understand all the XHTML or programming terms at first, but you can refer back to the lesson later as you work through the module.

## **Installing Nvu**

1. Download Nvu for Windows from [here and save this file](#) to a location on your computer you will remember.

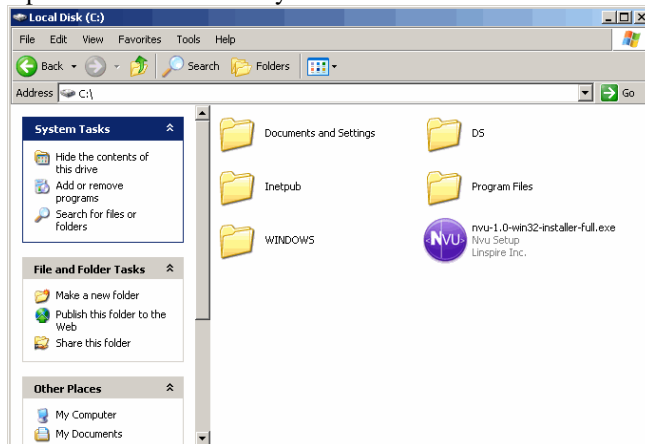
For different operating systems, visit [Nvu's Download Page](#).

If you have a CMPT 100 Online Course cd, Nvu is also available on the course cd (for both Windows and MacOSX), so you do not have to download it from the web.

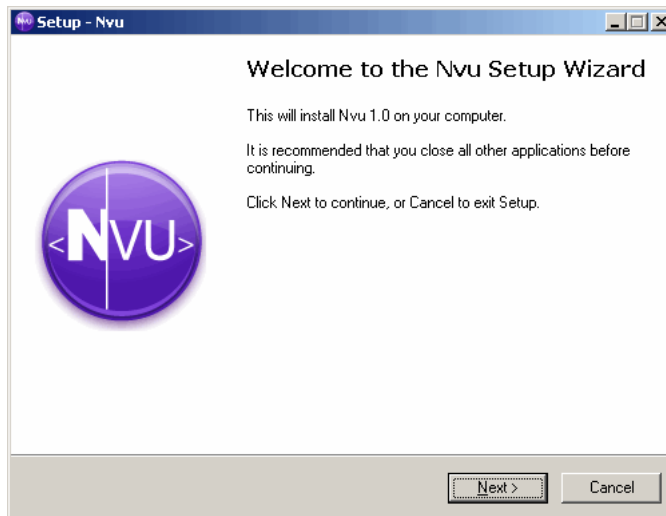
In Windows, right click on the cd drive in Windows Explorer and choose "explore". Open the software folder, and run the file **nvu-1.0-win32-installer-full.exe**. Then, skip to step 3.

For instructions on installing Nvu on MacOSX from the course cd, please contact your instructor.

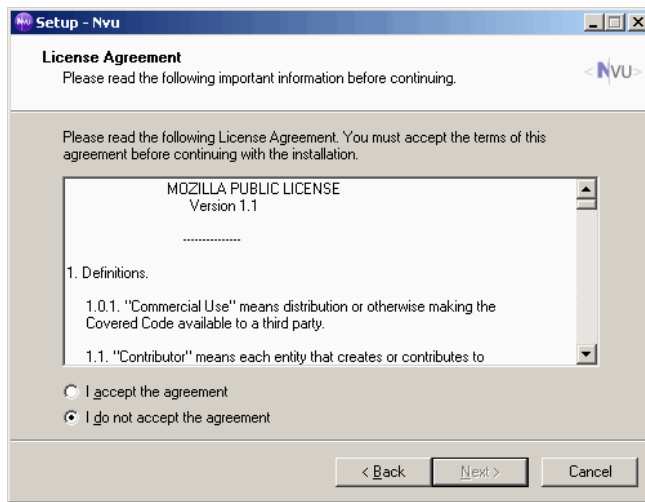
2. Open the location where you saved the file.



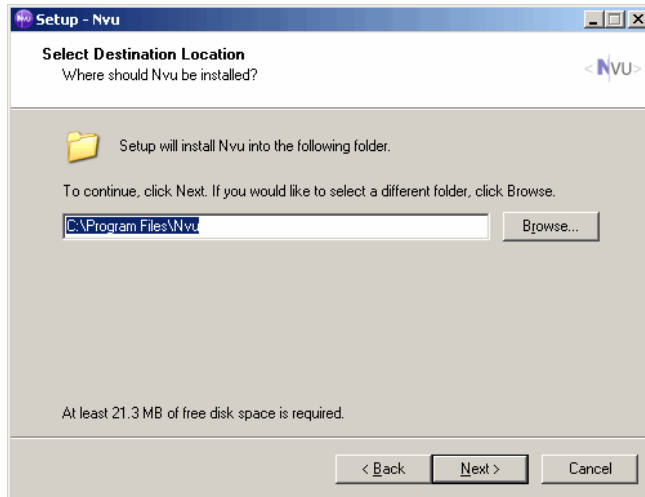
3. Run the installation program by double-clicking on the **nvu-1.0-win32-installer-full.exe** icon.



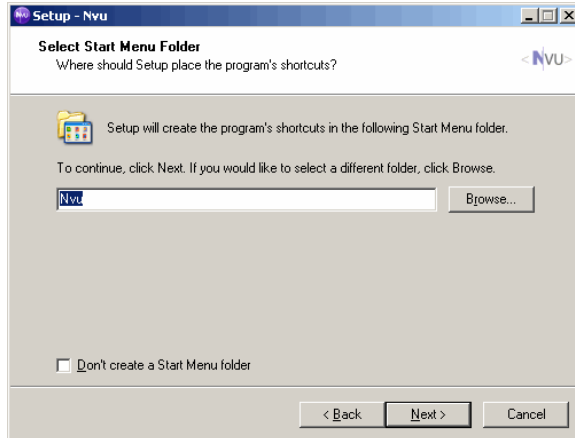
4. Click **Next** to continue with the installation.



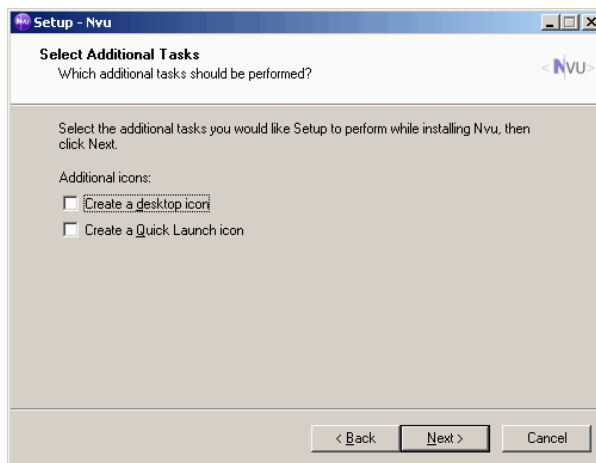
5. Read the License Agreement, if you want, then click **I accept the agreement** and click **Next**.



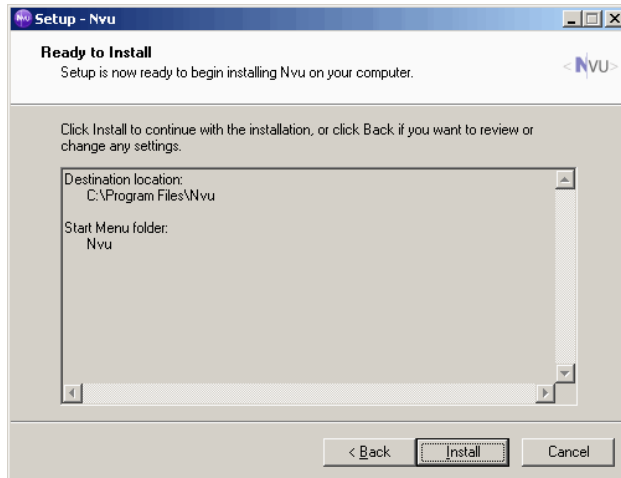
6. In the text box enter **C:\Program Files\Nvu**, if it has not done so already, and click **Next**.



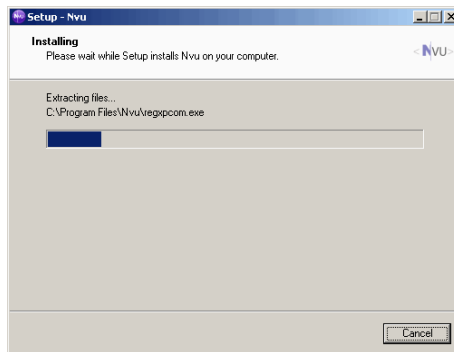
7. In the text box enter **Nvu**, if it has not done so already, and click **Next**.



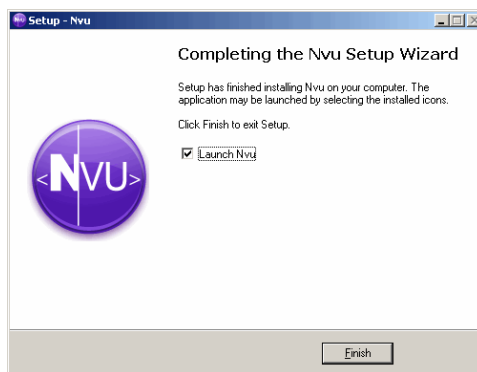
8. In the next window, check the **Create a desktop icon** and/or **Create a Quick Launch icon** if you want the icons to be created. Then click **Next**.



9. Click the **Install** button and your installation should begin.



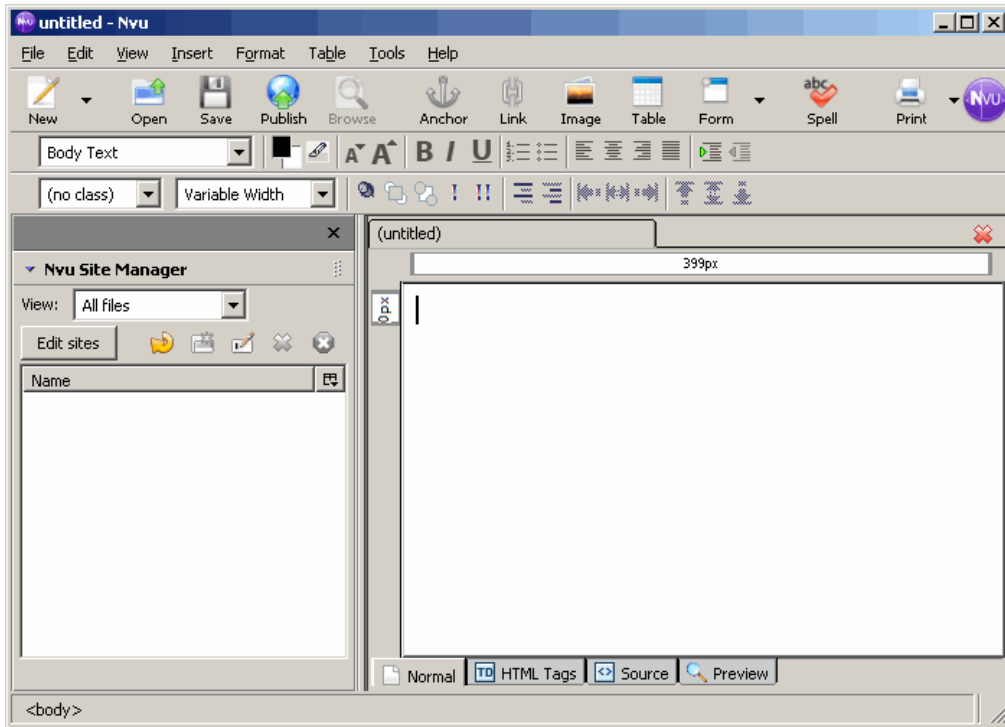
10. The installation should run, when it is finished the following screen will appear:



11. Leave the **Launch Nvu** box checked if you want to run Nvu right away and click **Finish**. If not, just uncheck the box and click **Finish**
12. You should now be able to open Nvu through the **Start Menu -> Program Files -> Nvu -> Nvu**.

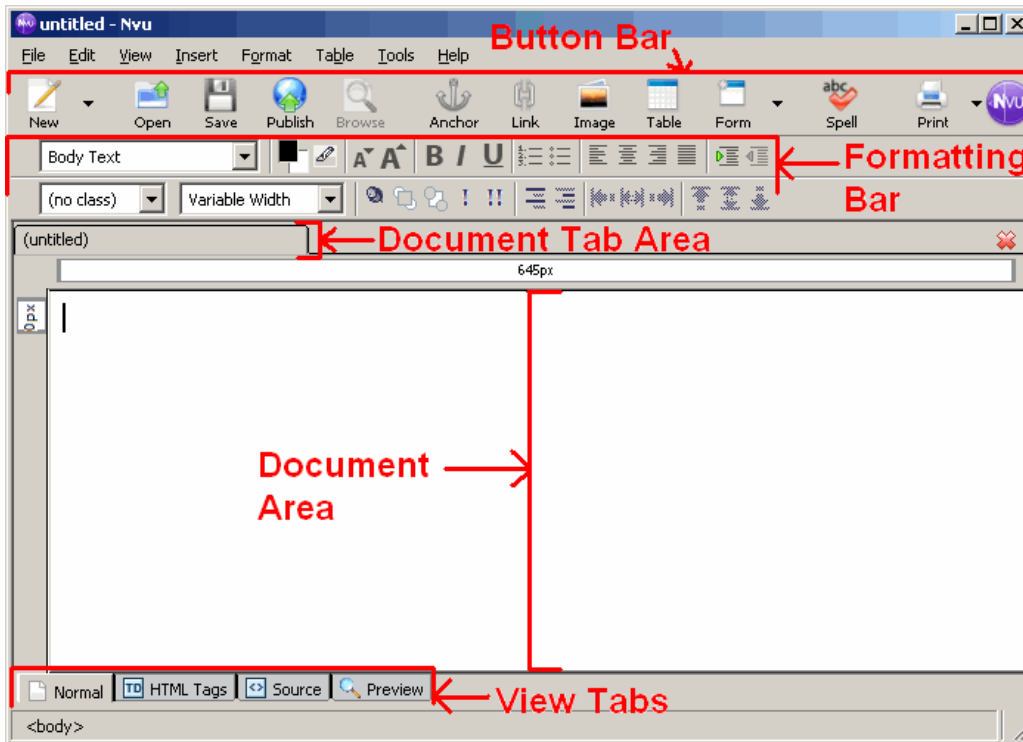
## Nvu's Interface

When you first open up Nvu you will get a screen that looks like this:



You will notice on the left hand side of the screen the Site Manager. Click the X in the top right portion of the Site Manager to close it, you will not need it for this class

Looking at Nvu we can divide it up into 5 parts:



The **Button Bar** contains shortcuts to commonly used actions or items for your webpage

The **Formatting Bar** contains the formatting options for alignment, font size, etc.

The **Document Tab Area** shows you the various pages you have opened. It allows for easy navigation between pages, just by clicking on the tab

The **Document Area** is where you develop your webpage. This is where you add text, images, links, etc. It works very similar to a word processing application such as Microsoft Word

The **View Tabs** allows you four different views of your web page:

- Normal - Is the view where you will edit your web page, much like a Word document
- HTML Tags - Shows you a view with the HTML tag labels beside the content of your page
- Source - Lets you view the HTML code that Nvu produces as you create your webpage. You can also edit this code
- Preview - Lets you view your webpage as it would appear in a browser

## Introduction to XHTML

XHTML stands for "Extensible HyperText Markup Language" - it is the language used to create webpages. This lesson introduces you to XHTML - what it is, what it's used for, and it's basic structure.

The lesson then shows you how to create XHTML code using Nvu. It will explain the steps needed for constructing and viewing webpages with Nvu.

## ***Lesson Objectives***

- Understand the purpose of XHTML
- Know the basic structure of an XHTML page
- To learn how to create a basic webpage in Nvu

## **Keywords**

- XHTML
- Elements
- Tags
- Attributes
- Cascading Style Sheets
- Container Elements
- Empty Elements
- Block-level Elements
- Inline Elements
- Button Bar
- Formatting Bar
- Document Tab Area
- Document Area
- View Tabs
- Source Code

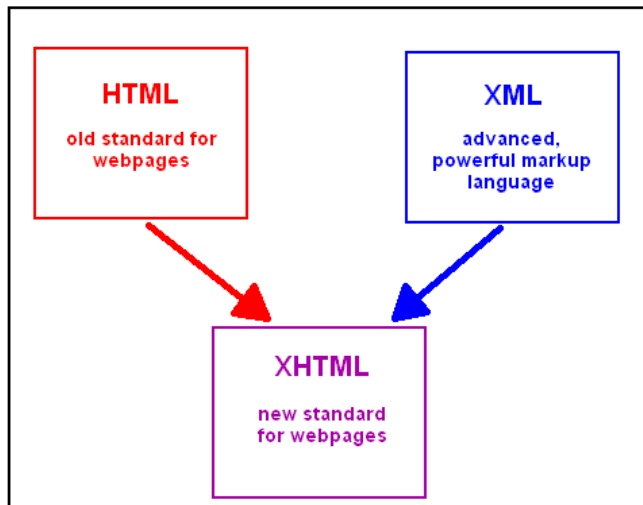
## **What Is XHTML?**

XHTML stands for Extensible HyperText Markup Language. It is a language used to produce webpages.

Previously, authors used a language called HTML (HyperText Markup Language) to create webpages, but this language is being phased out and replaced with XHTML. The old HTML concerns itself only with how things are displayed on a page, it doesn't worry about how any of the items are related or what their meaning is. It has a fairly limited set of elements, so authors must use what is available. In recent years, browsers have become very tolerant and will attempt to interpret HTML code even if it is incorrectly written. This means that there are tons of badly written webpages on the web. However, new devices for accessing the web - like handheld computers and cellphones do not necessarily understand these incorrect pages.

To deal with the limitations of HTML, web developers created a language called XML - Extensible Markup Language. This language is very flexible and customizable. Authors can define their own elements and the exact interpretation of each element. XML defines the structure of the data in a way that describes the different parts of the data rather than simply how it will be displayed. Because of this extreme flexibility, writing XML documents is more complicated than writing HTML webpages.

From these two sources, XHTML was created. It takes many of the same ideas as HTML and combines them with the flexibility of XML. It is a very structured language but gives much more control to the author of the webpage. An XHTML file contains the data (the content displayed by the web browser) as well as the tags instructing the web browser what to do with this data (e.g., how to format the data). It is becoming the new standard for creating webpages.



What you see when you look at a webpage using a browser is only half of the picture. It is the "interpreted" version of the webpage, meaning that the browser looked at the file, followed its instructions, and displayed the results. When you look at XHTML using a text editor, you see **ALL** of its contents (it's a bit like looking at it through an x-ray machine, you get to see how it's put together from the inside).

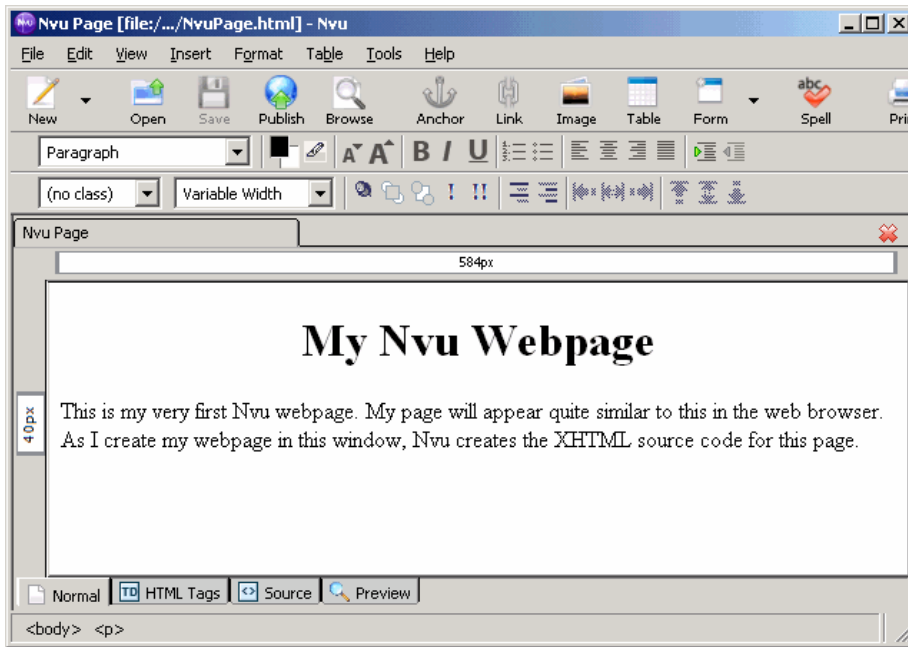
An XHTML file includes instructions for the web browser so that it can be displayed as the author intended. For example, the file might include tags that say "this paragraph should be written in bold font" and "display an image, here's the file containing the image" - of course, these tags are written using the particular commands within XHTML, not regular English.

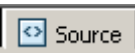
### **WYSIWYG Editors**

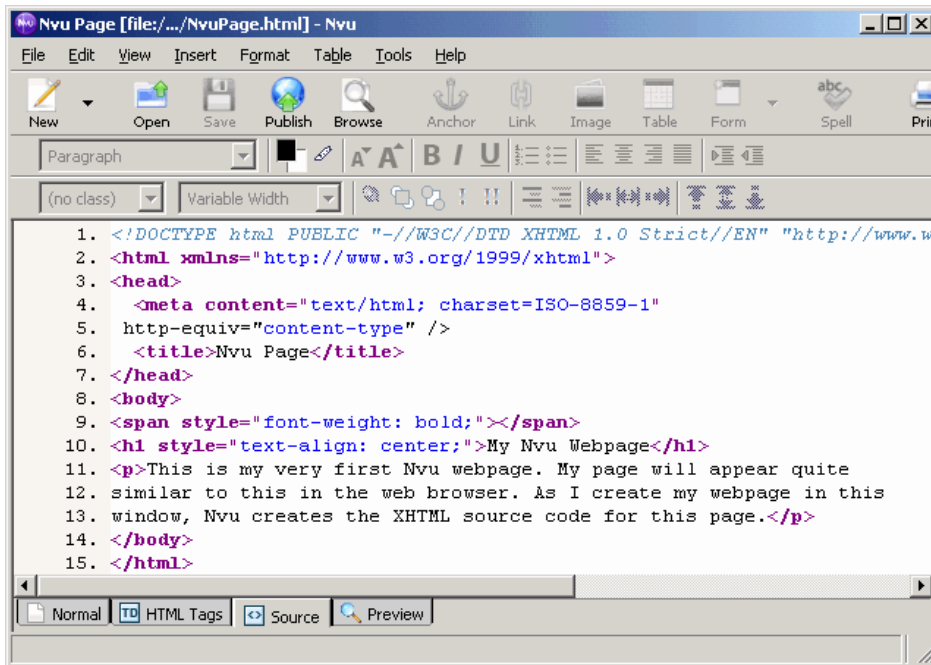
**WYSIWYG editors** (WYSIWYG stands for **What You See Is What You Get**) are programs where you can create a webpage almost exactly how you want it to appear on a webpage. These editors provide you with a blank webpage where you can add elements such as paragraphs, images, tables, etc. to your webpage. As each element is added, the editor creates the XHTML code for your webpage automatically for you, in the background.

There are a variety of XHTML WYSIWYG Editors, the most popular being Macromedia's Dreamweaver and Microsoft Frontpage. However these editors cost upwards of \$200. There are many open source editors available for free such as Amaya, Mozilla Composer, and Nvu. The editor we will be using in this class is Nvu. Here is an example of a webpage created with Nvu:





By clicking on the  Source tab we can see the corresponding XHTML code that Nvu creates for our webpage:



When we view our page in our browser, we see that it closely resembles the page we created in Nvu.



It is important to remember that all three of the above images are displaying the **same** file. The file just gets displayed differently depending on which program you view it in. When you edit your XHTML file, you'll use Nvu (the first image). To view your XHTML file, you'll look at it using a browser (third image). Remember that the browser **interprets** the underlying XHTML code that Nvu produces for you in the source code (the second image).

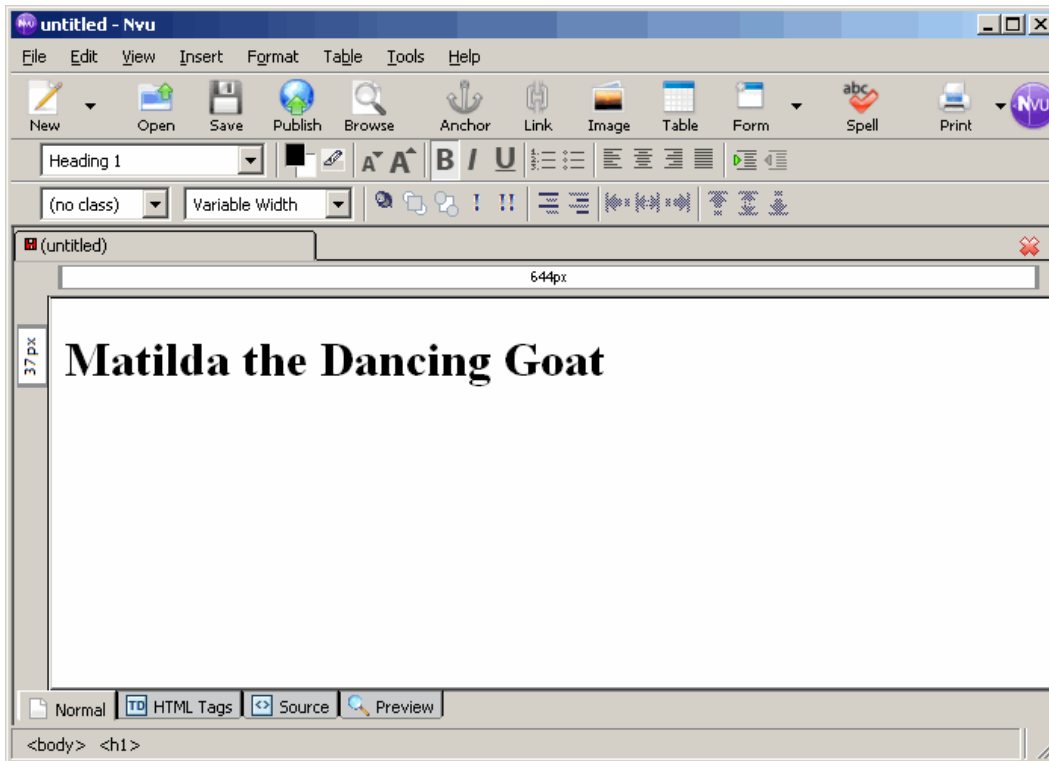
## **XHTML Keywords**

Before learning to create XHTML pages, you must first become familiar with a few terms since these terms are used in any description of XHTML.

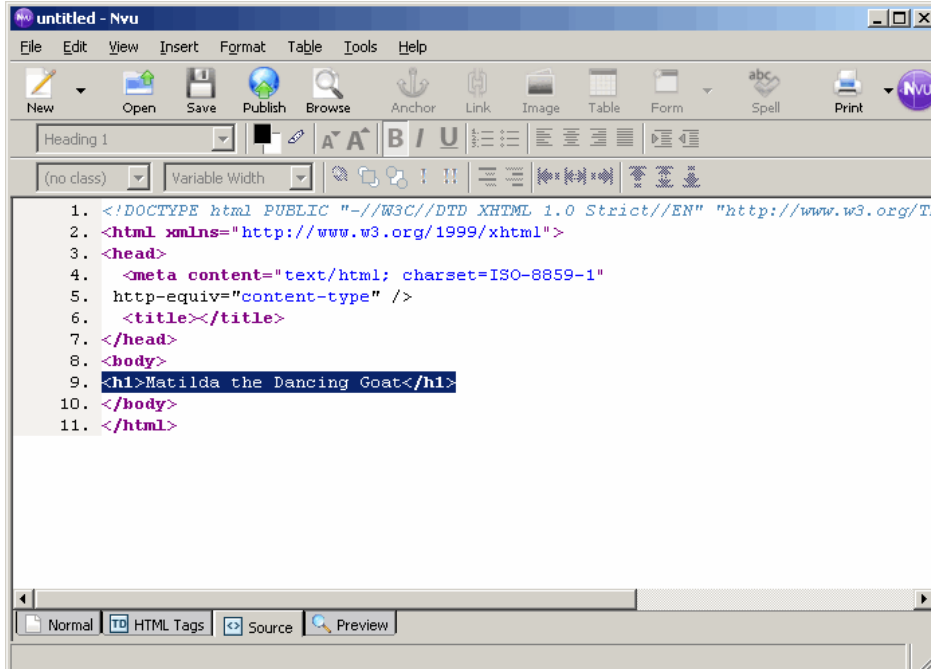
### *Elements*

Elements mark off regions of a text file to indicate that some processing is to be applied to that section of text. Every item you add to your page is an element. Whether a heading, paragraph, image, hyperlink, list, table, or others, all of these are considered individual elements.

For example, the following screenshot shows a heading of size 1, this is an element.



Creating the above element will generate the following source code in Nvu:

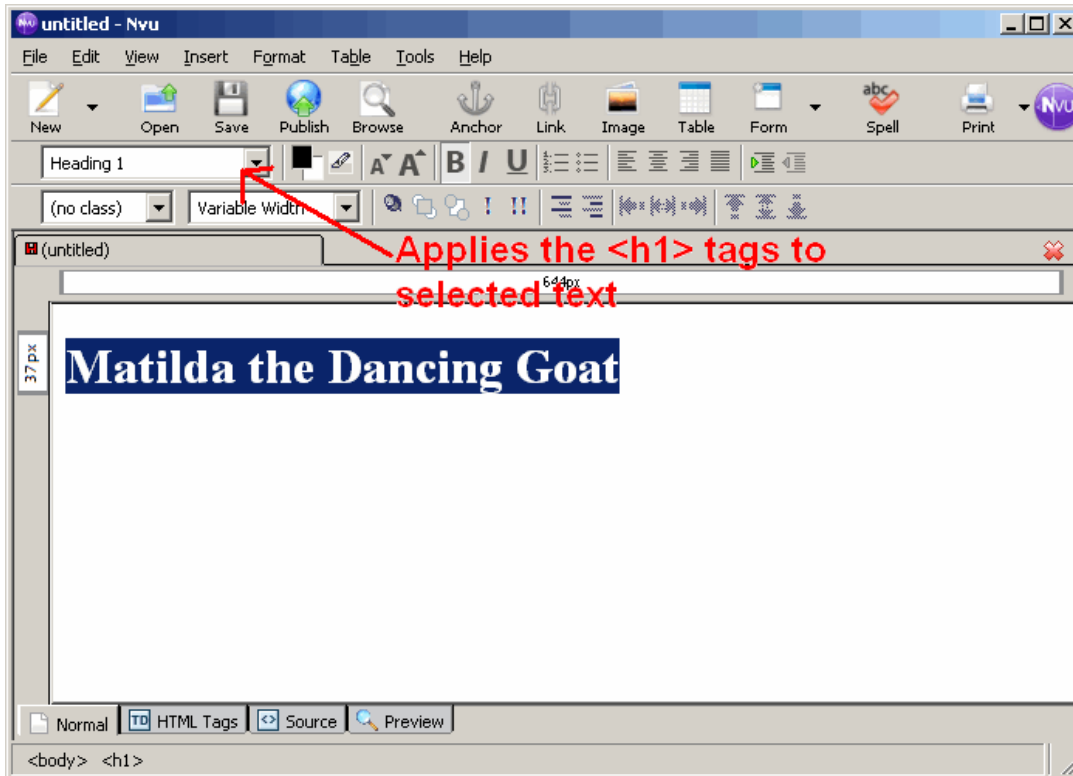


The highlighted text is our element. Elements generally consist of three parts: the start tag which marks the beginning of the element, the content, and an end tag which marks the end of the element. In our example above, <h1> is our start tag, "Matilda the Dancing Goat" is our content, and </h1> is our end tag.

## Tags

Tags indicate the start and end points of elements. They contain the instructions for the browser and are always enclosed in triangular brackets <>.

For example, by selecting "Heading 1" from the formatting bar, Nvu will place <h1></h1> tags around the selected text.

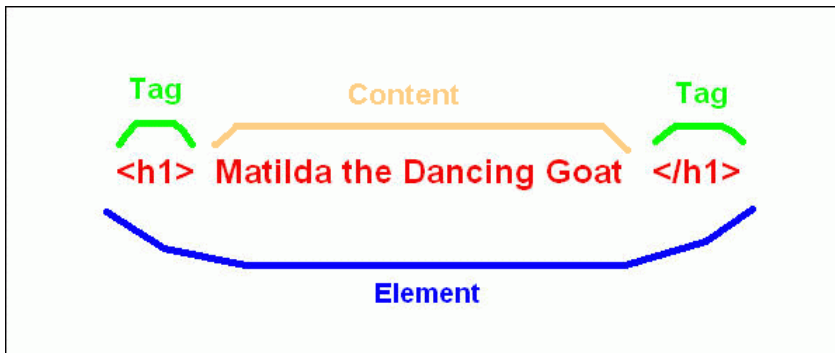


We can see this by looking at the source code.

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title></title>
7. </head>
8. <body>
9.   <h1>Matilda the Dancing Goat</h1>
10. </body>
11. </html>
```

By having the `<h1></h1>` tags around it, the browser knows to display "Matilda the Dancing Goat" as a heading of size 1.

Here, `<h1>` and `</h1>` are tags. `<h1>` is the start (or "opening") tag because it marks the start of an element. `</h1>` is the closing (or "end") tag because it marks the end of an element. Notice that closing tags always have a / (backslash) as their first character. Start tags never include a backslash.



### *Container Elements*

Most elements are considered "container elements" because they have a start and a closing tag with some data contained between them.

```
<h1>Matilda the Dancing Goat</h1>
```

is called a "container element" because "Matilda the Dancing Goat" is contained between the start and close tags. This is the case for most XHTML elements.

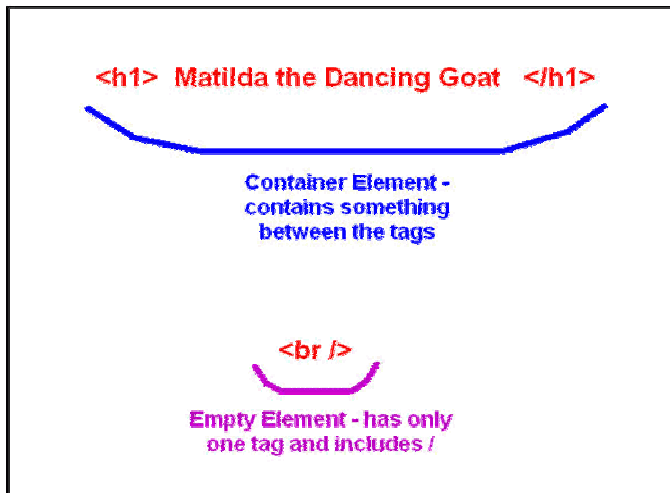
### *Empty Elements*

Empty elements are those that do not have an end tag. There are few empty elements and these are generally "exceptions to the rule". Most elements require a closing tag.

An example of an empty element is:

```
<br />
```

This simply means to "break" -- go to the next line. It is similar to pressing the Enter key in a word processor, the text stops and moves to the next line. Notice that empty elements include a / at the end. This is to indicate to the browser that it doesn't have to search for the corresponding closing tag. Empty elements always end with " />" (space - slash - closing bracket). The space is important, browsers won't recognize the tag otherwise.



### *Attributes*

Attributes are used inside of start tags to specify extra details about the instruction. Attributes are always added inside of a tag, they cannot stand on their own.

An example of an attribute:

```
<p id="First_Paragraph">This is a paragraph. The browser will add  
blank lines  
before and after it.</p>
```

Notice that `id="First_Paragraph"` appears inside of the starting paragraph tag, but does not appear in the closing `</p>` tag. The `id="First_Paragraph"` portion is an attribute of the `p` tag. It gives an ID (name) to the paragraph to distinguish it from others. We will then be able to apply special formatting to this particular paragraph by saying things like "I want the First\_Paragraph to be displayed in red and bold".



### *Block-level vs. Inline Elements*

All XHTML elements are grouped into two categories: **Block-level** and **Inline** elements.

Block-level elements generally describe larger structures. They can contain inline elements and other block-level elements. Things like headings and paragraphs are considered block-level. Block-level elements usually automatically start on a new line and add a line break at the end.

Inline elements can contain only text or other inline elements. They are not automatically surrounded by any white space. Examples of inline elements are hyperlinks and emphasis. Inline elements cannot stand on their own. They must be enclosed within a block-level element.

You will typically be told whether an element is block-level or inline. If you accidentally use an inline element without containing it within a block-level element, you will get an error when you try to validate your XHTML page.

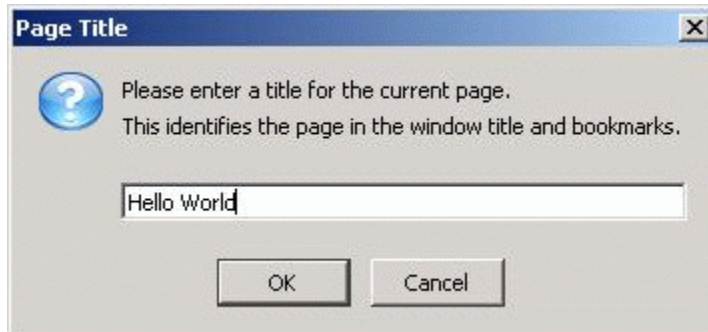
### *Cascading Style Sheets*

Cascading Style Sheets tell the browser how to format a page. We use CSS to apply "styles" to the different portions of our webpage. These can indicate things like colors, margin widths, border types, font types, and alignment.

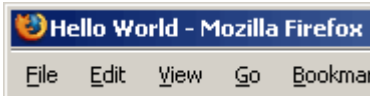
Nvu has a built in CSS Editor. Much like it creates the tags for the XHTML code, it automatically generates the code for the Cascading Style Sheets as you create rules within the CSS Editor.

### **Creating a Basic Web Page**

1. The first step to creating a web page in Nvu is to create a new XHTML Strict document.
  - To do this click on the **down arrow** ▼ beside the **New** icon in the Button Bar.
  - Select "**More Options...**"
  - In this window check the **create a XHTML document** box and the **Strict DTD** box then click **Create**
  - In this class we want our pages to be XHTML Strict. XHTML has a much broader range than HTML and is expected to eventually replace it. But most importantly, using XHTML strict requires the code to be more accurate, more efficient and cleaner.
2. Now that you have a blank document prepared you want to save your web page
  - click on the **Save** icon in the Button Bar
  - Nvu will prompt you to enter a title



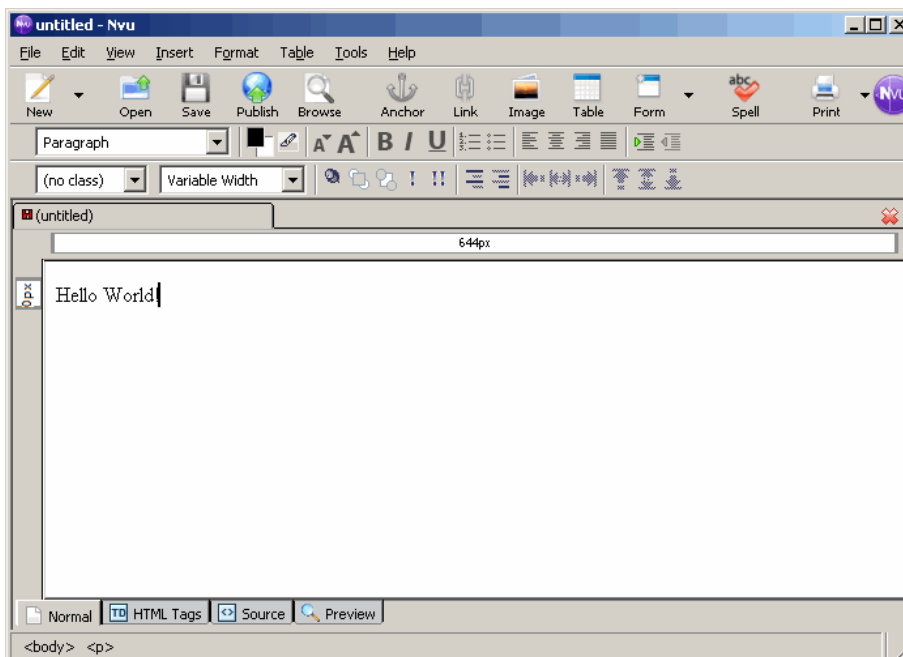
- When viewing your page, the title will appear in the top left corner of the browser in the Title Bar



- After entering your title, select where you would like to save your file and give it a name. When naming your files be sure of a few things:
  - Be certain to include the **.html** filename extension
  - Filenames should not contain spaces. If you want to break up words, use the underscore or capital letters instead (ie. hello\_world.html OR helloWorld.html)
  - Filenames can only contain numbers and letters. No special characters such as %, &, \*, etc. are allowed.

3. Now that you've saved your file you can begin editing it.

- Add a paragraph to your page by selecting **Paragraph** from the select list in the formatting bar
- Type in your text



- 4. Make sure you save your changes by clicking on the **Save** icon
- 5. To view your web page in a browser click on the **Browse** icon in the Button bar
  - You can also view your web page in a browser by opening Mozilla and clicking on **File - > Open File**

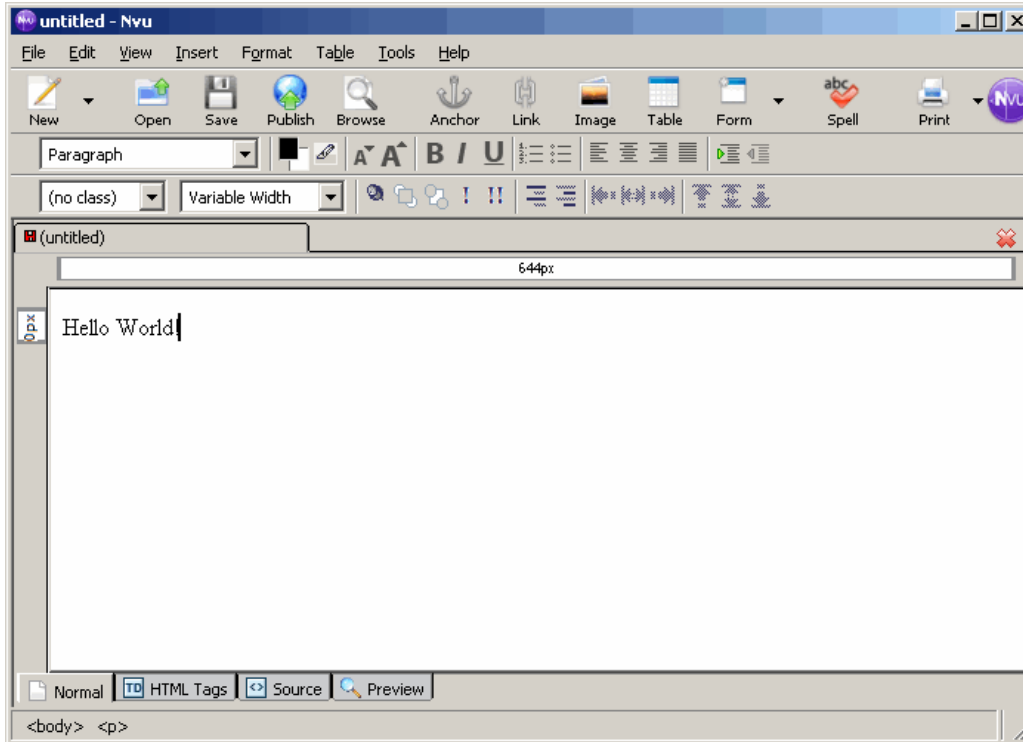


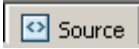
- Browse to where you saved your file, select it and click open
6. As you make changes to your web page make sure you save your file. To see the changes take place in the browser, just hit the **Reload** button

## XHTML Source Code

As you add items to your web page, Nvu produces the corresponding XHTML code for the item. To look at this you can click on the Source tab in the View Tabs area.

Looking at our Basic Web Page example,



We can view its source code by clicking on the  Source tab

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.     http-equiv="content-type" />
6.   <title>Hello World</title>
7. </head>
8. <body>
9.   <p>Hello World!</p>
10. </body>
11. </html>
```

The first line of code is simply to tell the browser what type of file we are creating. This will appear at the start of every page you create in Nvu. You do not have to worry about what it means other than it tells the browser you are creating an XHTML Strict document.

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.     http-equiv="content-type" />
6.   <title>Hello World</title>
7. </head>
8. <body>
9.   <p>Hello World!</p>
10. </body>
11. </html>
```

Annotations in the image:

- A yellow box highlights the opening `<html>` tag on line 2 and the closing `</html>` tag on line 11, with the label "html" to the right.
- A green box highlights the `<head>` tag on line 3, the `<meta>` tag on line 4, the `<title>` tag on line 6, and the `</head>` tag on line 7, with the label "head" to the right.
- A blue box highlights the `<body>` tag on line 8, the `<p>` tag on line 9, and the `</body>` tag on line 10, with the label "body" to the right.

After the first line we see our opening `<html>` tags. All the tags for our page will be nested in between the opening and closing `<html>` tags -- the html tags tell the browser "this is an HTML document".

Following that we see our `<head>` tags. The head is the area where we put information that does not directly appear on the webpage, but gives extra information to the browser about how to display the page. This section will eventually be where we put JavaScript code and CSS rules. It also contains the title of the webpage, which appears at the top of the browser window in the title bar.

Next are the <body> tags. The body section contains all the elements (paragraphs, tables, lists, etc) that will actually be displayed on the webpage. Anything that you want to appear on your page when you view it in a browser goes between the opening <body> and closing </body> tags.

You can also edit the source code within Nvu. If you find it easier to manipulate your webpage by coding the XHTML tags, this is where you would do it.

Another way to view the source code of a web page is to use a built in function in most browsers. If you open your web page in your browser you can "View Source Code" to see the code that results in the web page the browser is displaying. To do this in Firefox, you click on **View -> Page Source**. A window will pop up with the XHTML code for the web page.



```
view-source: - Source of: file:///C:/Documents%20and%20Settings/Andrew/My%20Documents...
File Edit View Help
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type" />
  <title>Hello World</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

## Comment Tags

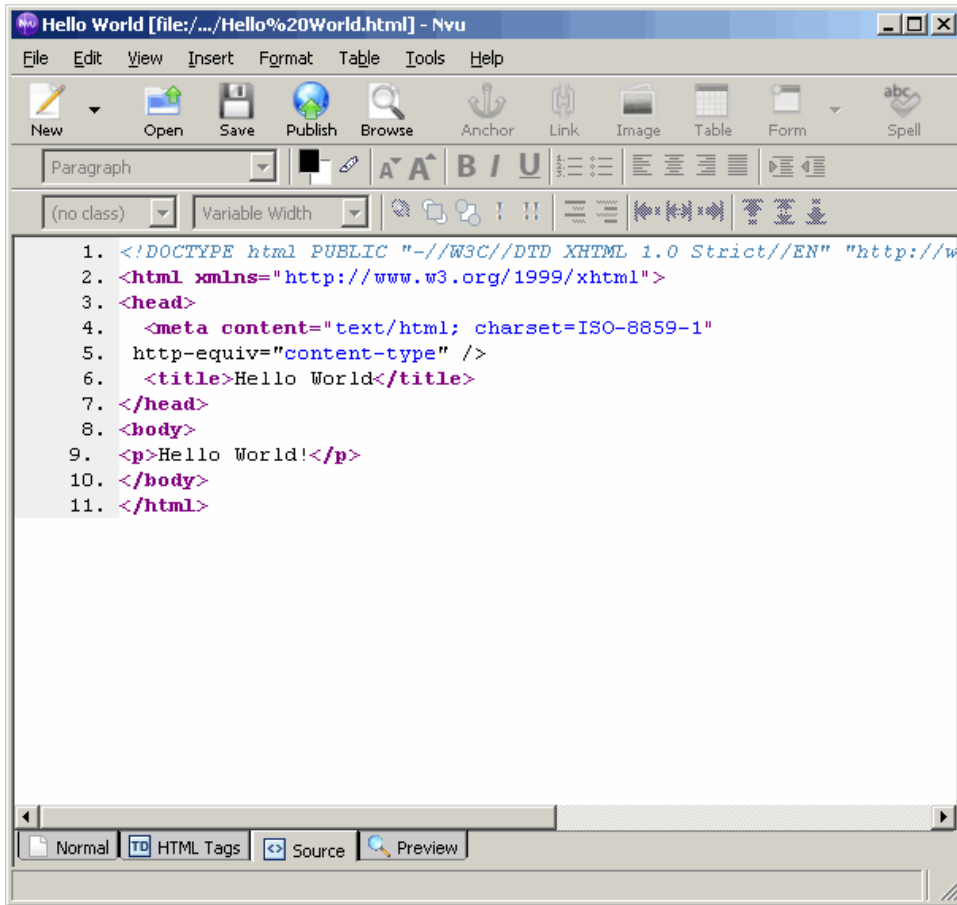
Comment tags are entirely for the person who is reading your code. The computer simply ignores them. The sole purpose of comment tags is to give information to the humans who read the code. They can be inserted anywhere within the document and will not appear on the webpage when it is displayed by the browser.

A XHTML comment tag looks like this:

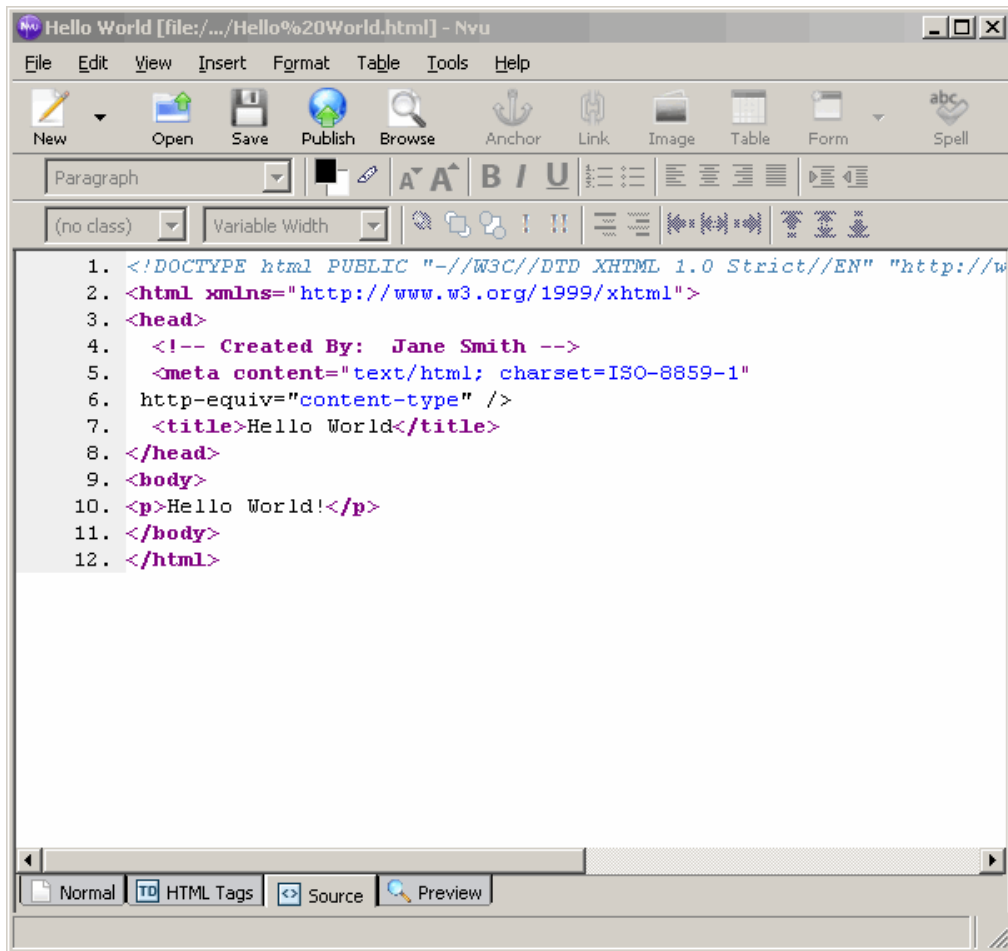
```
<!-- this is a comment -->
```

It starts with a "less than sign, exclamation mark, dash, dash" and then ends with "dash, dash, greater than sign". Notice that there is no exclamation mark at the end. Any text included between these symbols is considered a comment and will be ignored by the browser.

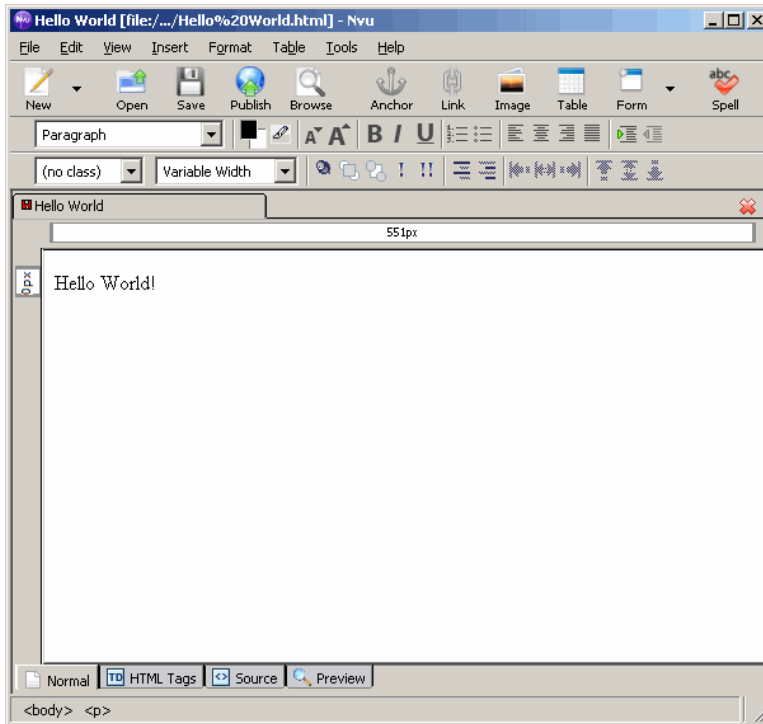
To add a comment to your page using Nvu, you'll have to use the "Source" view. Click the Source tab.



Find the place you want to add your comment. If you're adding identification information, this information usually goes right after the <head> tag.



After you've added your comment, click the "Normal" tab to return back to the editing mode. Notice that you can't see your comment on your page -- remember that the comment is only in the source code, and will not show up in the browser.



You should get into the habit of including comments within your code. They should be used to identify different sections of your page, to include identification info, and to note anything unusual or tricky. It makes it much easier to re-read your code if you've included comments that describe the code.

## Conclusion

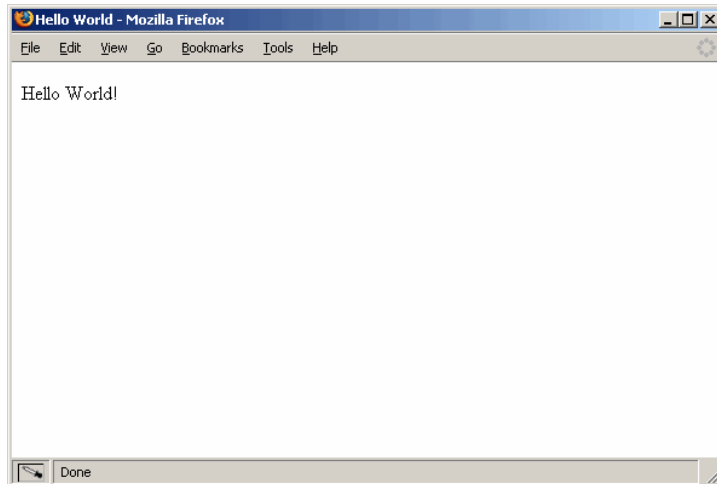
XHTML, Extensible Hypertext Markup Language, is a language used to make web pages. It uses a system of elements and tags to instruct the browser how content should be displayed.

You should be creating your webpages in a WYSIWYG editor called Nvu. Nvu is a program that allows you to create web pages much like you would a word processing document in a program such as MS Word. Nvu generates the source code for your webpage so you don't have to.

Finally, we looked at the generated source code to see XHTML pages always have the same basic structure. This structure includes a head section and a body section. Everything that is displayed on the webpage is included within the body of the page. The head section contains items that aren't directly displayed on the webpage. For simple pages, the head usually remains empty except for the title.

## Exercises

1. Define the keywords in this lesson
2. Create the following webpage in Nvu



3. In the source code of the page you created, find both the opening and closing `<head>`, `<body>`, and `<html>` tags

## Basic Elements

This lesson shows you the basic elements for creating a webpage - those used to format the appearance of text on your page.

### *Lesson Objectives*

- Create headings and paragraphs
- Create hyperlinks
- Insert images into your page

### Keywords

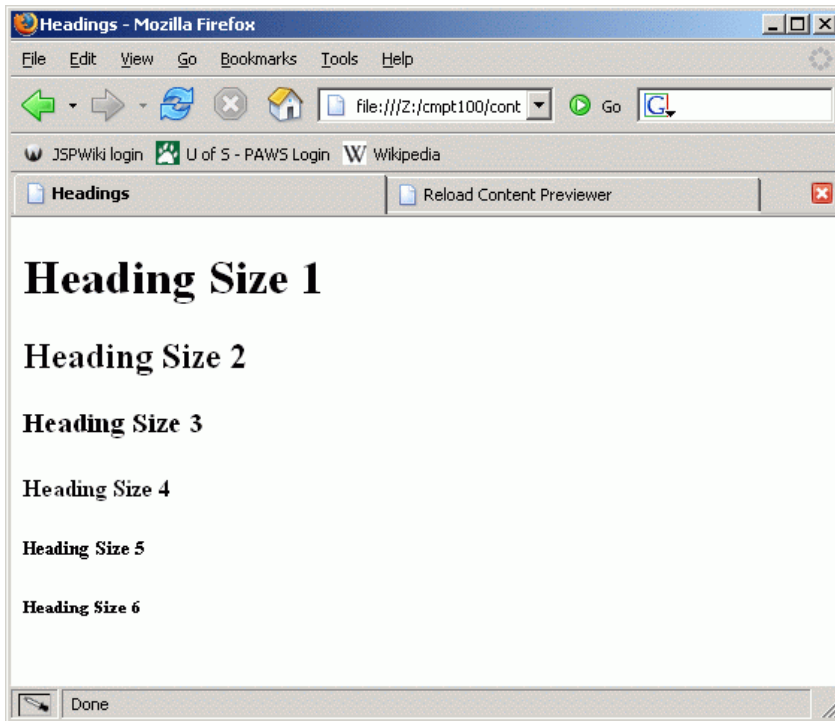
- headings
- paragraphs
- horizontal rule
- hyperlink
- absolute link
- relative link
- internal link
- mailto link
- images

## Headings and Paragraphs

Headings and Paragraphs allow you to clean up your page by dividing it into different sections.

### *Headings*

Most every webpage has a heading of some kind. XHTML provides six different headings - numbered 1 through 6. A heading of size 1 is the largest, while a heading of size 6 is the smallest.

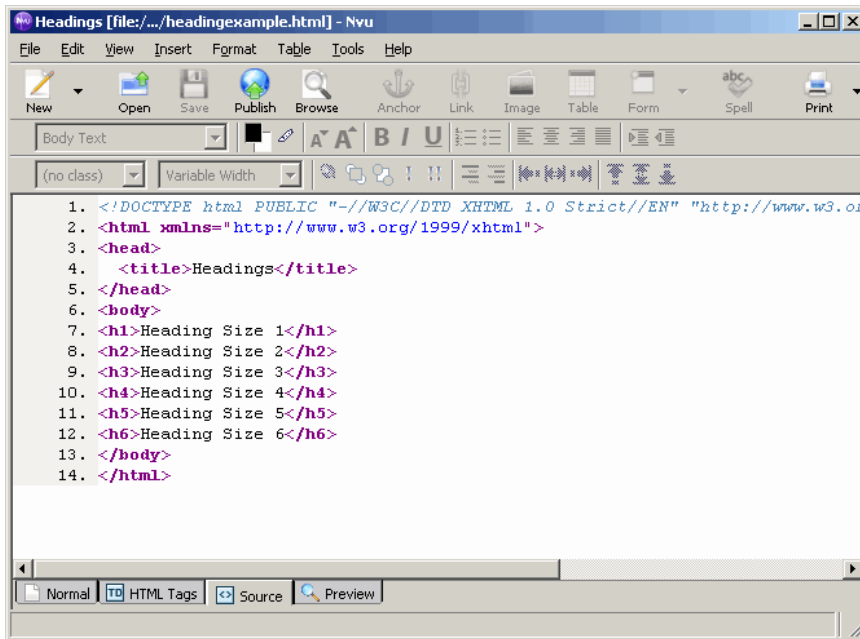


You can select any of the six sizes in Nvu by selecting it from the drop down menu in the Formatting Bar.



The code for the headings example above looks like this:





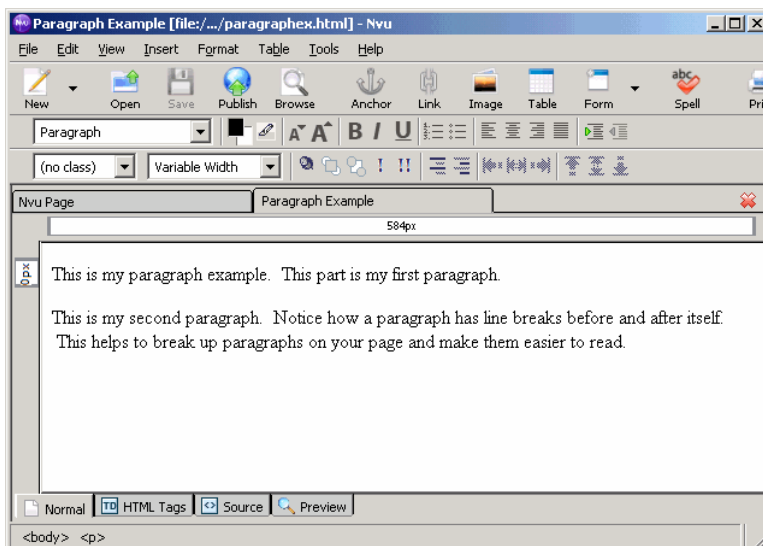
As you can see from the source code view, the heading tag starts with an h, followed by a number indicating its size.

Headings are considered "**block-level**" elements. This means they can contain other elements.

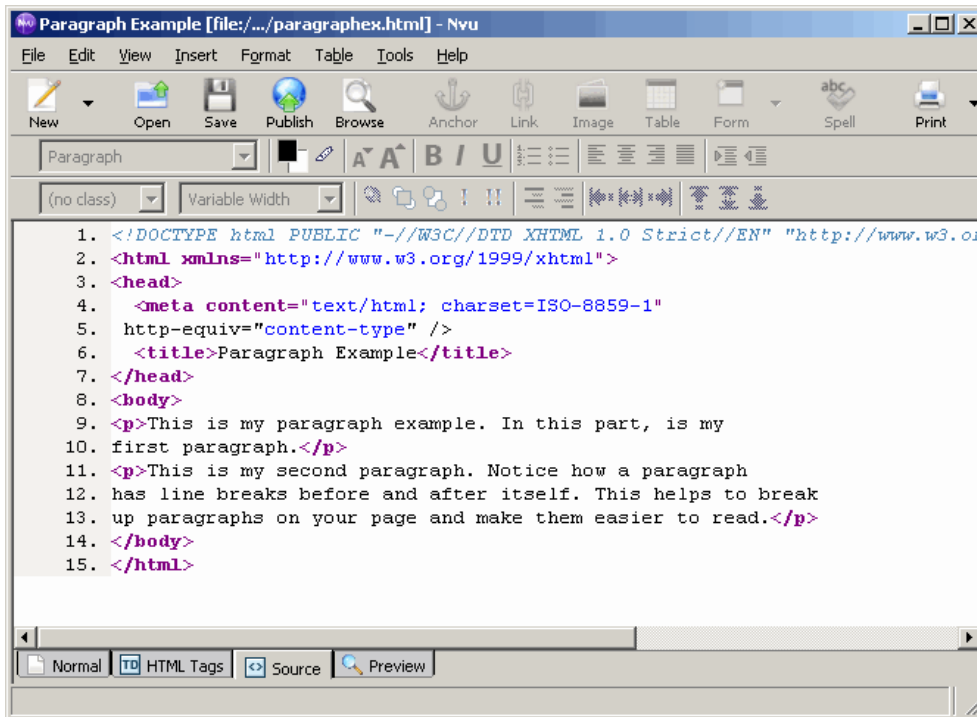
### *Paragraphs*

Paragraphs are a simple way to group text on your webpage. A line break is inserted before and after each paragraph on your page. You can choose the paragraph format by using the same drop down menu as headings.

Here is an example of two separate paragraphs:



And here is the resulting source code:



The screenshot shows the Nvu web editor interface. The title bar reads "Paragraph Example [file:///.../paragraphex.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, Spell, and Print. Below the toolbar is a "Paragraph" style dropdown and a "Variable Width" dropdown. The main editing area displays the following HTML code:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/1999/xhtml">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.     http-equiv="content-type" />
6.   <title>Paragraph Example</title>
7. </head>
8. <body>
9. <p>This is my paragraph example. In this part, is my
10. first paragraph.</p>
11. <p>This is my second paragraph. Notice how a paragraph
12. has line breaks before and after itself. This helps to break
13. up paragraphs on your page and make them easier to read.</p>
14. </body>
15. </html>
```

At the bottom of the editor, there are tabs for "Normal", "HTML Tags", "Source", and "Preview".

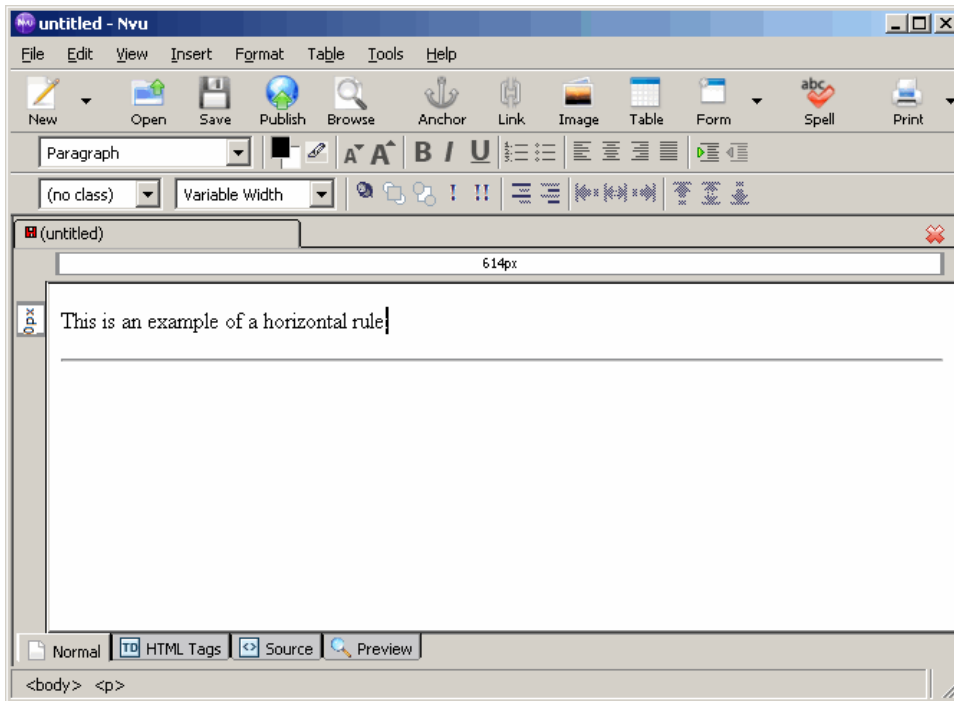
First, notice the paragraph tag that is created is simply `<p>`. Also notice when there is a break between the paragraphs on the webpage. When you look at the paragraphs in the code, there is no break. This is because the paragraph formatting includes a line break at the start and end.

Like headings, paragraphs are also block-level elements so they can contain other elements.

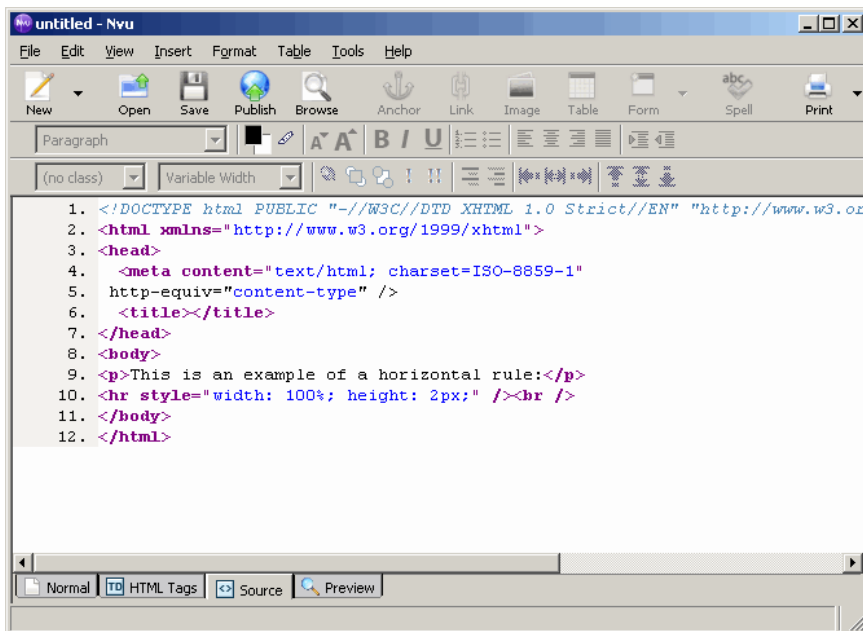
## Horizontal Rule and Break

### *Horizontal Rule*

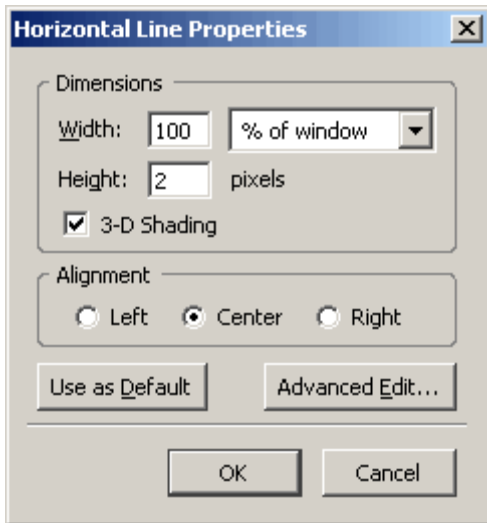
Horizontal rules are horizontal lines drawn across the page. They are usually used as a divider between page sections. The default horizontal rule creates a thin black line across the entire width of the webpage. To insert a horizontal rule into your webpage all you have to do is click "Insert -> Horizontal Line".



Nvu inserts a `<hr />` (Horizontal Rule) tag into your source code



You will notice in the source code the `<hr />` tag has a "style" attribute with it. We will look at what this means in a later lesson. But for now, to change the style of the horizontal rule, right-click on it and select **Horizontal Line Properties**. You will have a window pop up where you can determine the formatting of your horizontal line.



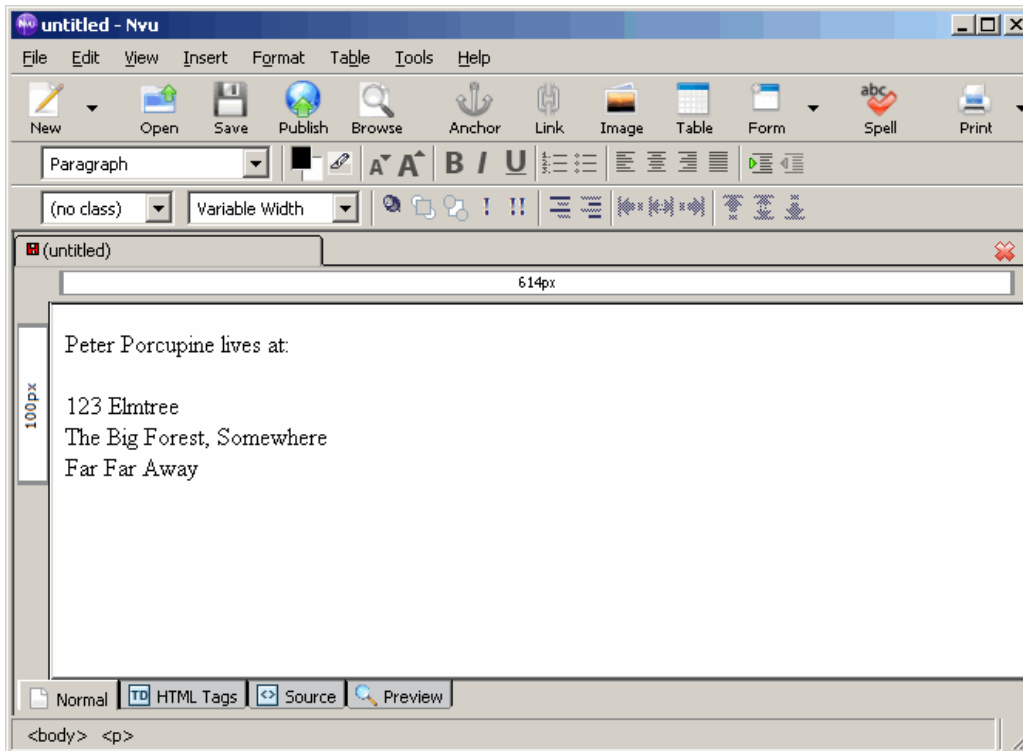
This element is special in two ways. First, it does not have a corresponding closing tag because it is an "empty element". Both the opening and closing tags are combined into one. Notice how the / is included at the end tag (with a space preceding it). This indicates to the browser that the element is complete and it should not look for a corresponding closing tag.

Secondly, it does not fall into either block-level or inline categories. It does not need to be contained and can stand on its own - like a block-level element. But, since it does not have a closing tag, it cannot contain any further elements (as block-level elements should). So, `<hr />` elements can be placed within block-level elements or on their own.

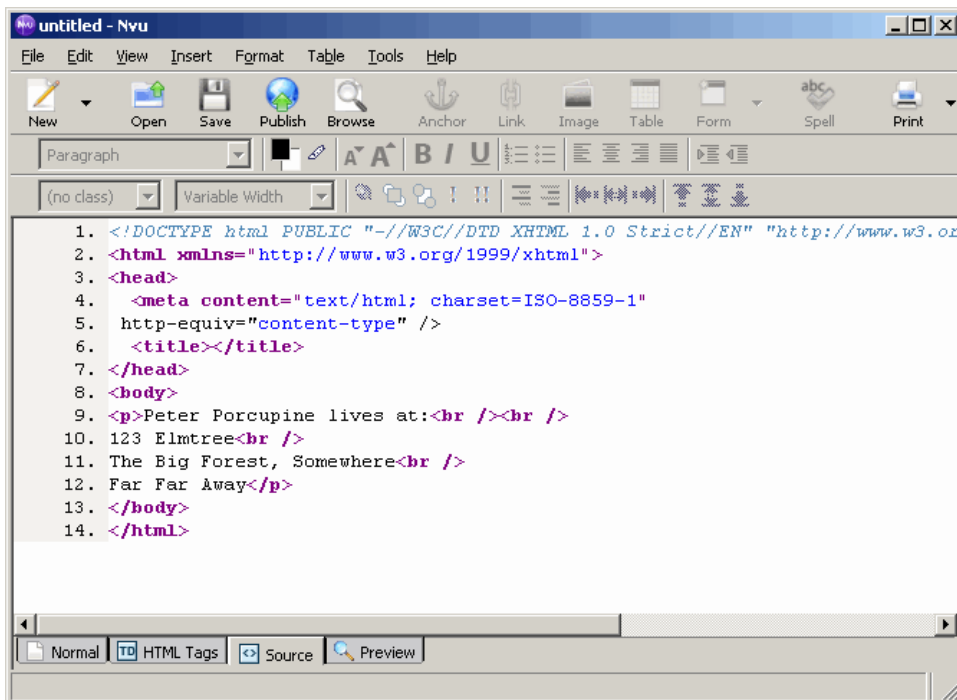
### ***Break***

One of the easiest ways to break up sections of information on a webpage is to add blank lines between the sections.

To add blank lines in Nvu, simply hit the **Enter** key on your keyboard. However, if you wish to insert a break inside a paragraph you have to hold **Shift + Enter**. Both methods will insert the break tag into your source code. For example,



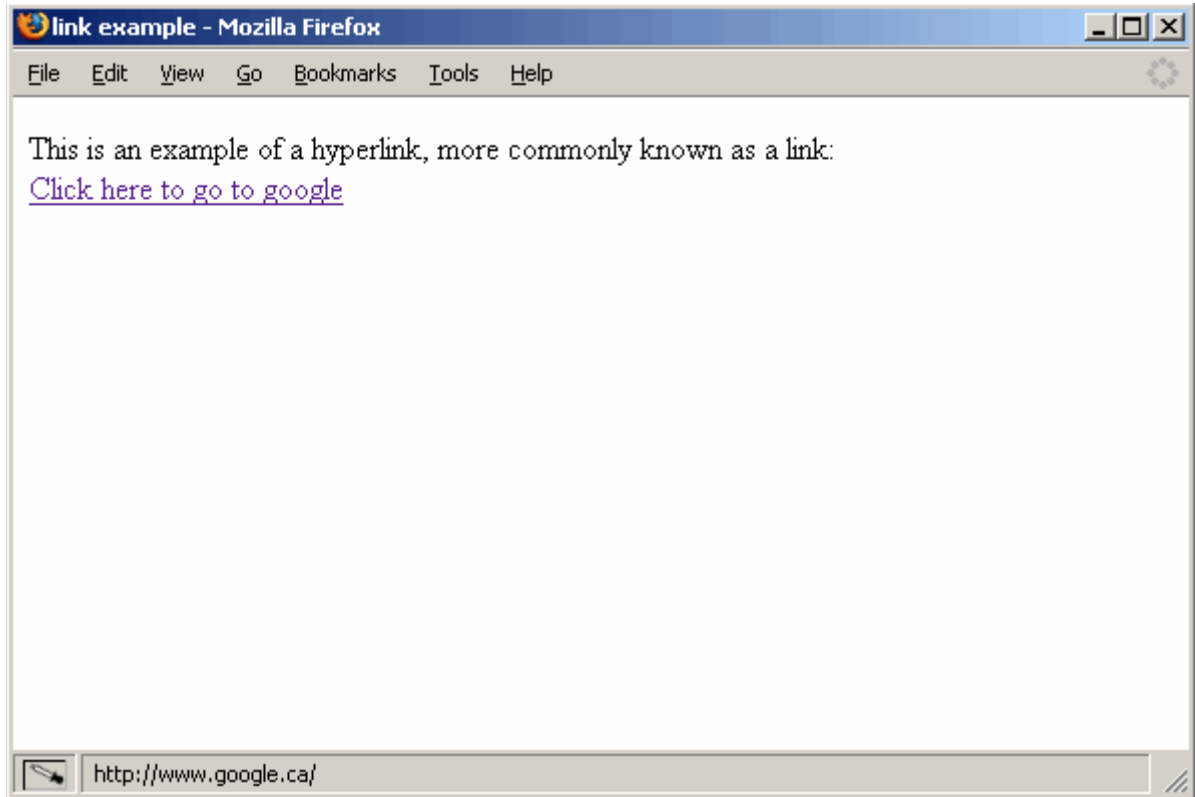
Results in the following source code. Notice the `<br />` (break) tags inserted.



The break element is an **inline element**, so it must be contained within another element. It is also an empty element and combines its opening and closing tags into one. Notice the `/` at the end of the tag.

## Hyperlinks

One of the most important features of webpages is the ability to create **hyperlinks**, which are more commonly known as "links". These links bring the user to different locations, such as a page on the internet, another link on your website, or even a link to another part of your page.



Hyperlinks are considered **inline** elements in XHTML. This means that they must be contained within a block-level element. For now, this typically means that your hyperlinks should be within a set of paragraph tags.

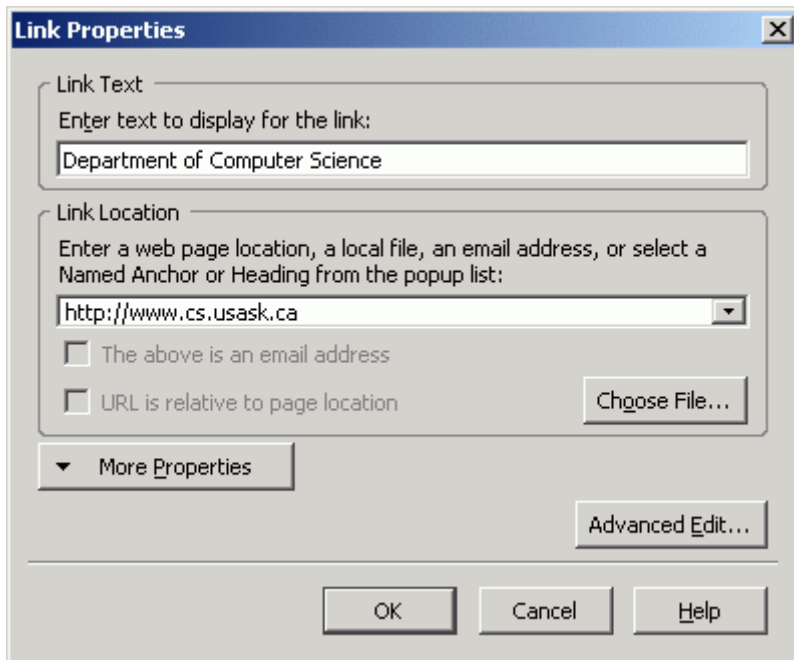
There are several types of hyperlinks:

### *Absolute*

Absolute links provide a way to jump to another webpage on the web. This can be any website, it does not need to be created by you. The term "absolute" signifies that when you create the link, you provide the entire URL where this new page is located.

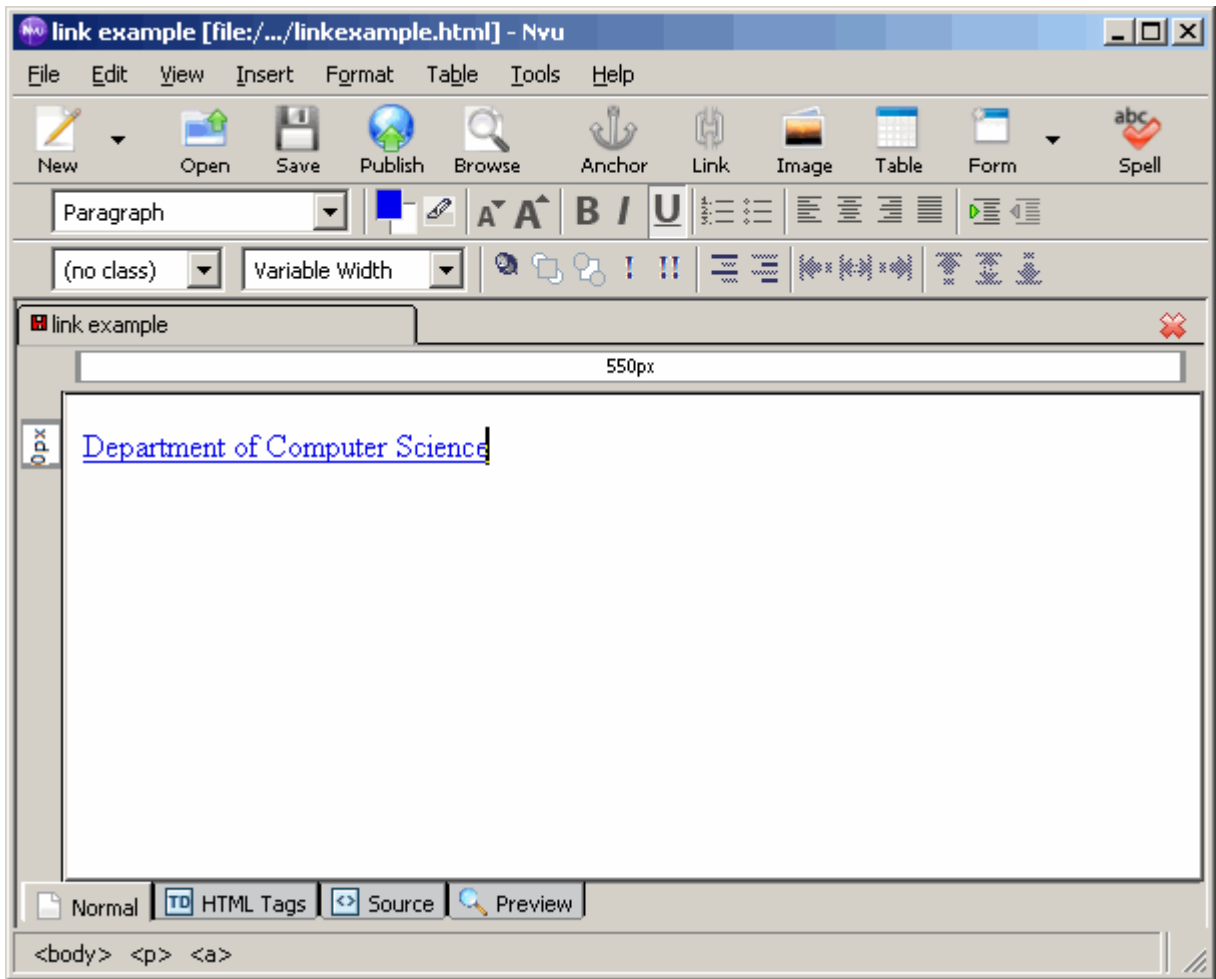
An absolute hyperlink is used on a webpage as a shortcut to another page. Clicking on one brings you directly to the specified webpage. An absolute hyperlink always uses the full address of the page it is linking to (ie. <http://www.cs.usask.ca>).

Absolute hyperlinks, like all other hyperlinks, are created using the Link button in Nvu's Button Bar. Once you click the Link button, a window will pop up. It first asks you for the Link Text. This is the text that will show up on the page for the user to click for the hyperlink. It also asks for the Link Location. This is where you put the URL of the webpage. This is likely what you would think of as a "normal" link because it gives the full address of the webpage.



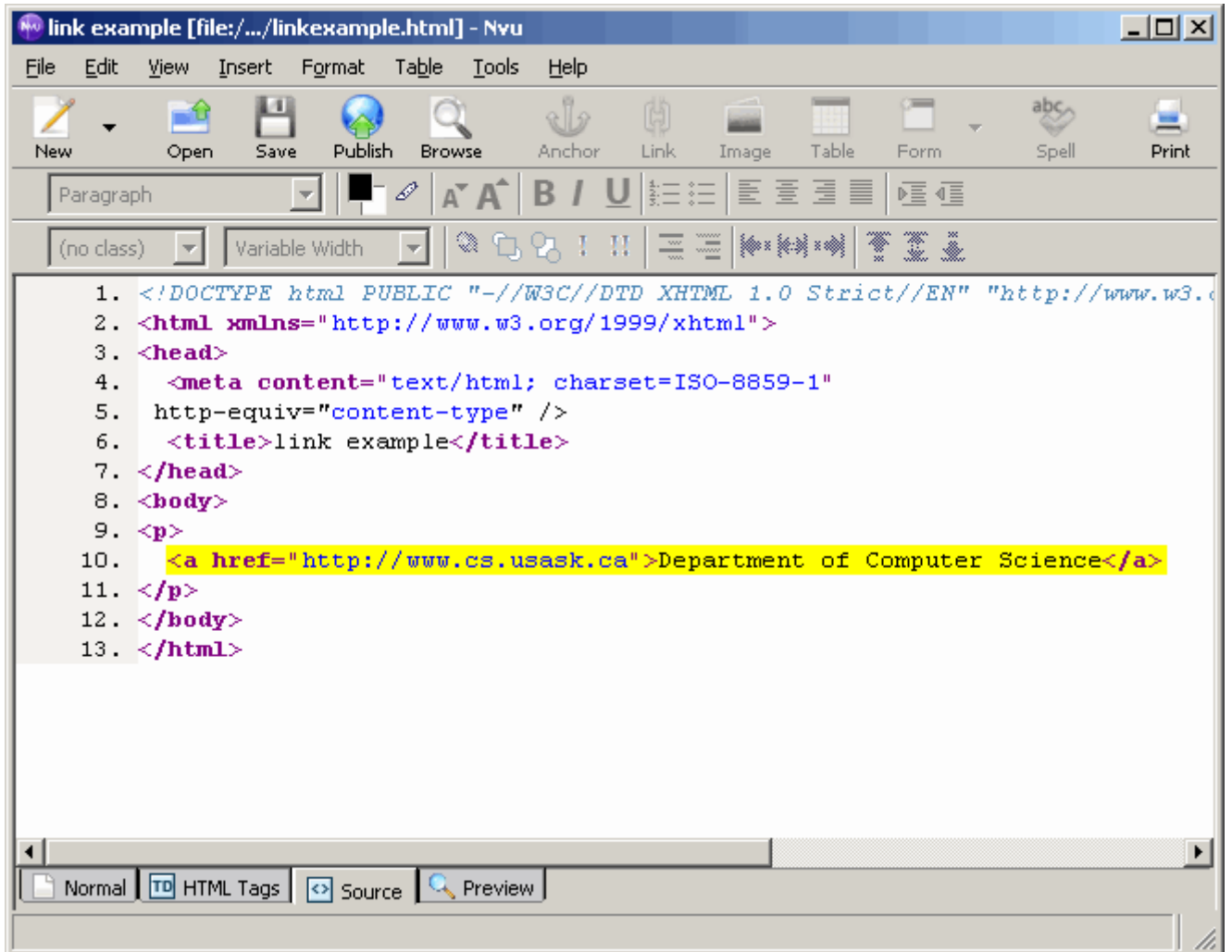
It is important to include the **http://** in front of the URL. Leaving it out will often result in a link that does not work because rather than going out to the internet to find the website, your browser will try to find a file on your computer by that name.

The link we created above will look like this:



And here is the source code that Nvu creates:



The image shows a screenshot of the Nvu web editor interface. The title bar reads "link example [file:///.../linkexample.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, Spell, and Print. Below the toolbar are two rows of formatting options, including Paragraph, text alignment, and list creation. The main editing area displays HTML source code with line numbers 1 through 13. Line 10 is highlighted in yellow and contains the code: `<a href="http://www.cs.usask.ca">Department of Computer Science</a>`. At the bottom, there are tabs for "Normal", "HTML Tags", "Source", and "Preview", with "Source" currently selected.

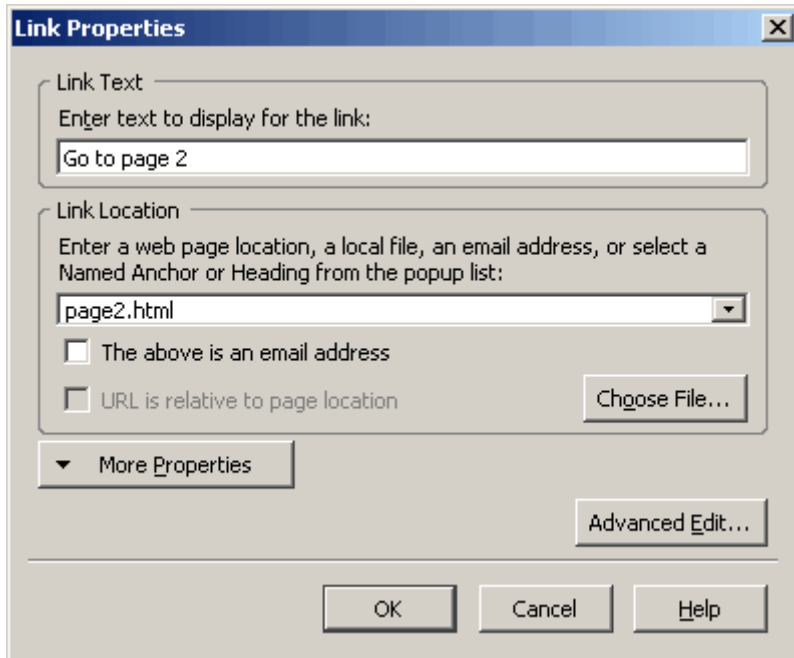
Looking at the source code, we can see our link created. Like all other hyperlinks, absolute hyperlinks are created using the "Anchor" element. Its tags are `<a>` and `</a>`. To create a link, the href attribute must be included. **href** stands for "**H**yperlink **RE**ference", meaning that it contains the address of the page where the link should go to when you click on it.

### *Relative*

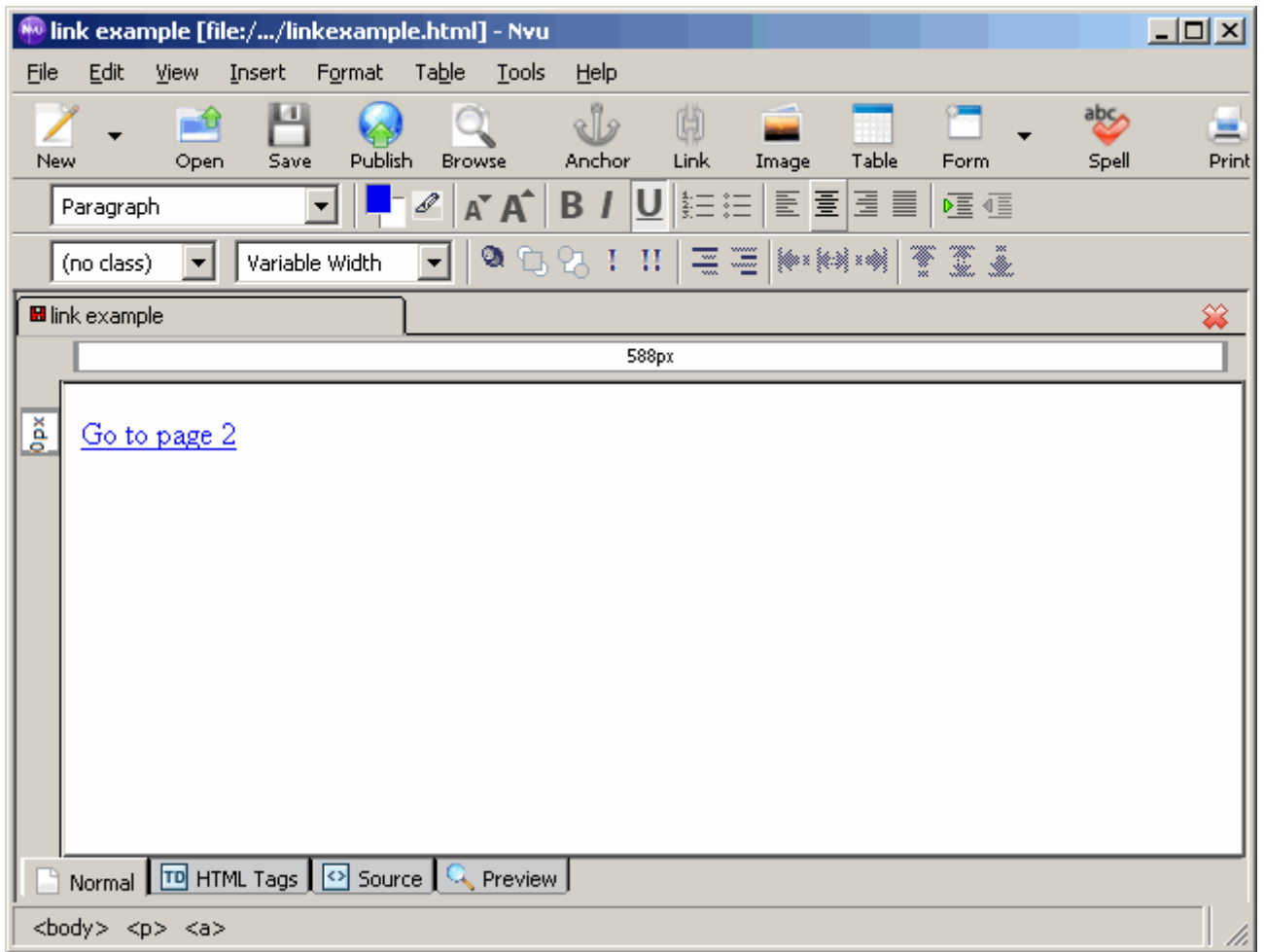
Relative links are used to join pages within a website. All of the webpages are stored in the same location and you only need to specify the filename of the page you want to reach. There is no need to provide the entire URL.

Contrary to absolute links, relative hyperlinks do not specify the entire address of the webpage they point to. Relative links are used for webpages that are stored on the same web server (typically in the same folder even). They give instructions on where to find the page based on your current location.

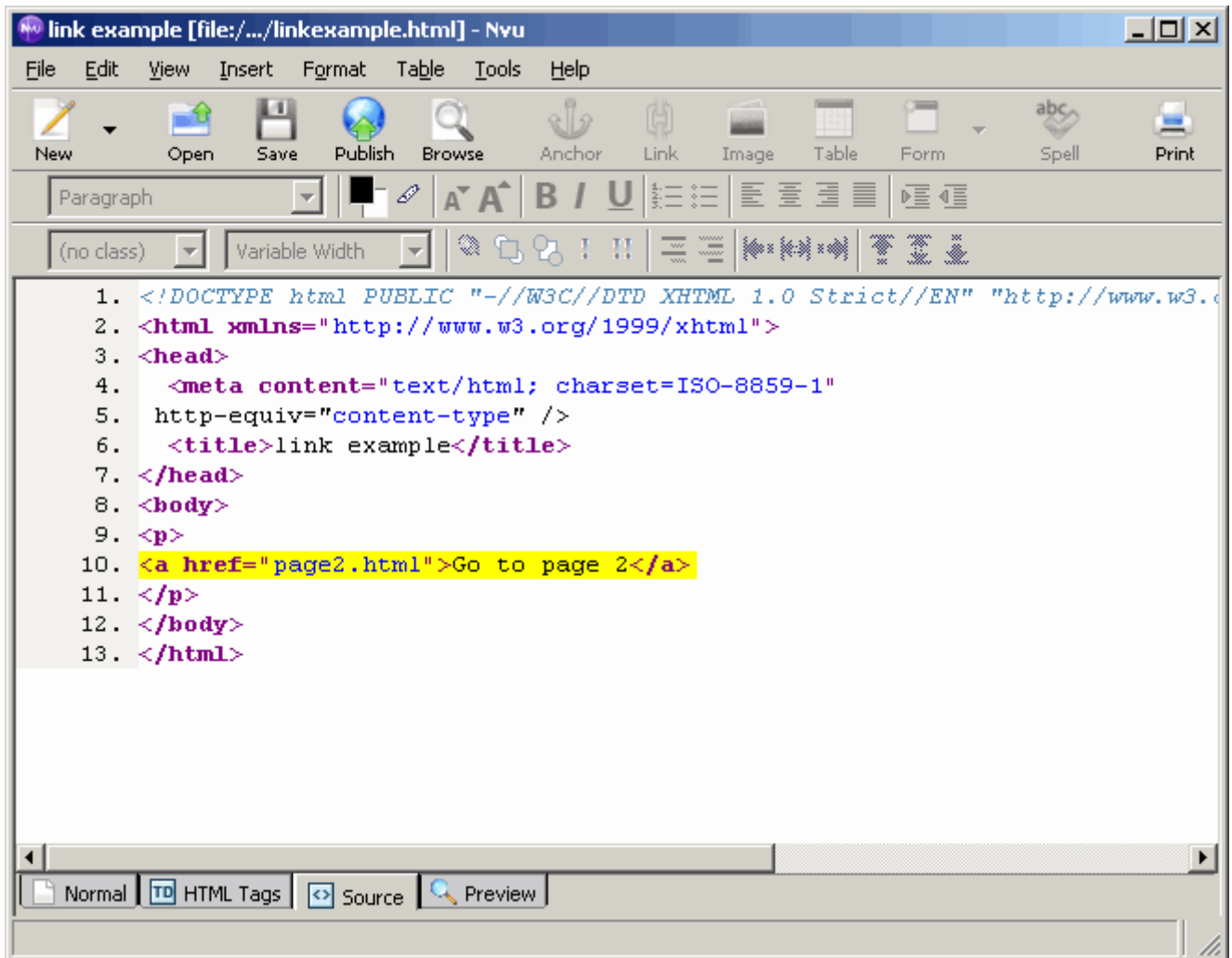
A Relative Link is created the same way as an absolute link, or any other link in Nvu. However the difference is in what you put for the Link Location.



Notice that the value of the Link Location is different than that for an absolute link. The Link Location only contains the name of the file with no extra location information. When the browser sees a reference like this, rather than going out to the Internet to find the webpage, it looks in the current folder. Our relative link created above will appear on our webpage much like the absolute link.



However our source code will be different



When you create a set of webpages that are all stored together, you can use relative links to navigate between them. Only filename and directory location relative to the current page is needed rather than the complete address of the page that includes the webserver.

Make sure you do not use absolute paths for files that are stored together (like `c:\cmpt100\myfile.html`) because once your webpages are moved to a web server or some other location (like submitted to E-Handin), the link will become invalid. Other people cannot access your `C:\` drive (or any other specific drive), so you should only use relative links between your own pages.

This may sound confusing but it's really quite simple once you have an analogy for it. Let's pretend that you and I are in the same room. There is a CMPT100 textbook on the table that we can both see. If I say to you "Take the CMPT100 book please", you assume that I mean "the CMPT100 book that's on this table, in this room, in this building". Since we are both in the same location, "the CMPT100 book" is enough information for you to understand which book I'm talking about. This is the same as a relative link - if the files are in the same location, we can assume that they know about each other and we do not have to give the full address of the file.

However, if I am talking to you on the phone - I don't know exactly where you are and you don't know exactly where I am. If I say "Take the CMPT100 book please", meaning the one that's on the table in front of me, you will not know which book I'm referring to. To make sure you understand, I would have to say

"Take the CMPT100 book that's on the table, in room Arts265, on the University of Saskatchewan campus, in Saskatoon, in Saskatchewan, in Canada." With these directions, presumably you'd be able to find the exact book I'm referring to. This is the same as an absolute link.

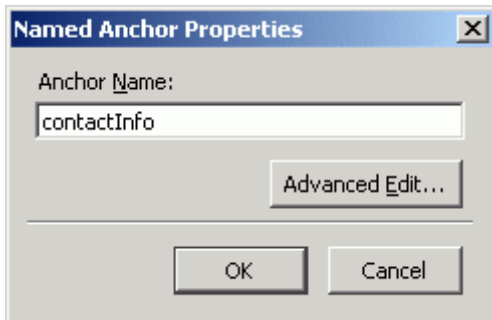
### ***Internal***

Internal links allow users to jump from one location to another on the same webpage. This is mainly used when you have a long webpage - you might provide users with a link to return to the top of the page, or to jump to the different sections of the page without having to manually scroll through the page.

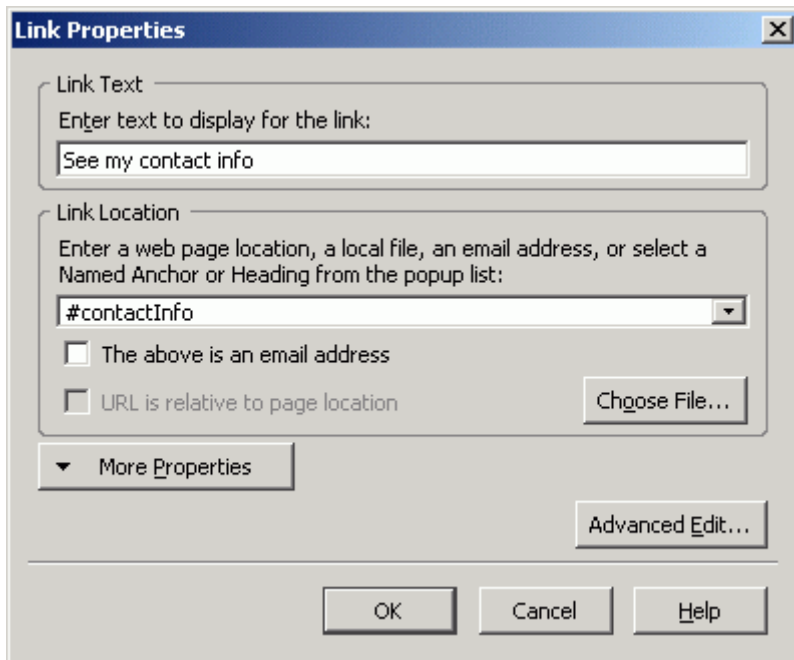
Sometimes our webpages are longer than what fits on the computer screen. A viewer can manually scroll down and find the different sections of the page, but this might get tedious if it's a really long page. We can create hyperlinks that go from one location on a page to another location on that same page. This type of hyperlink is called an internal link.

When we create links to different webpages, we use the name of the file or the website's URL to identify where we want to go. When we link to different areas on the same page, simply giving the filename is not enough since we need to go to a particular spot inside that file.

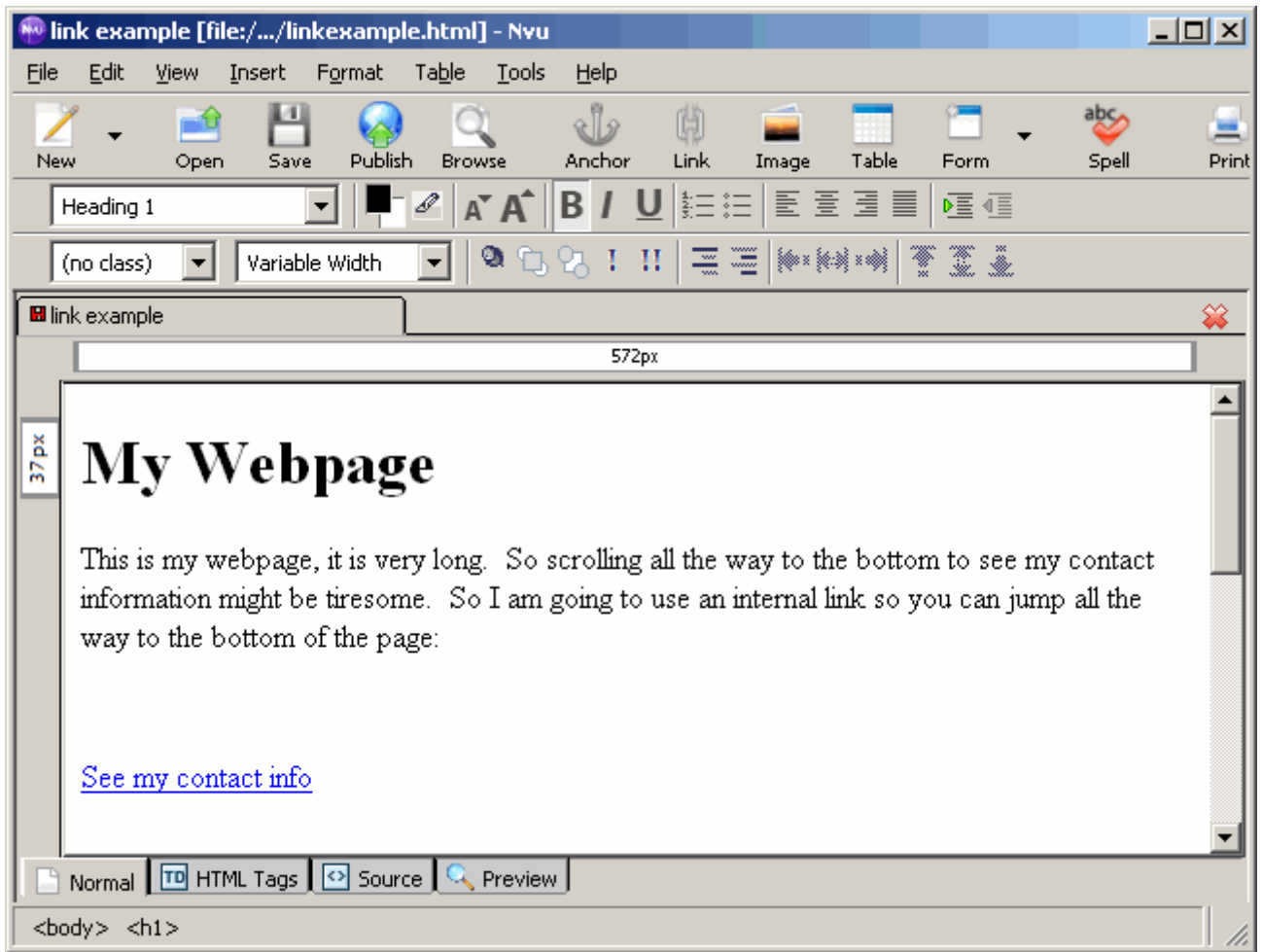
To create an internal link, the first thing we need to do is insert an anchor. To do this in Nvu, all you have to do is select the location you want to link to and click the **Anchor** button in the Button Bar. A window will pop up asking for the **Anchor Name**. The Anchor name can be anything the webpage creator decides, but it should be descriptive of the location.



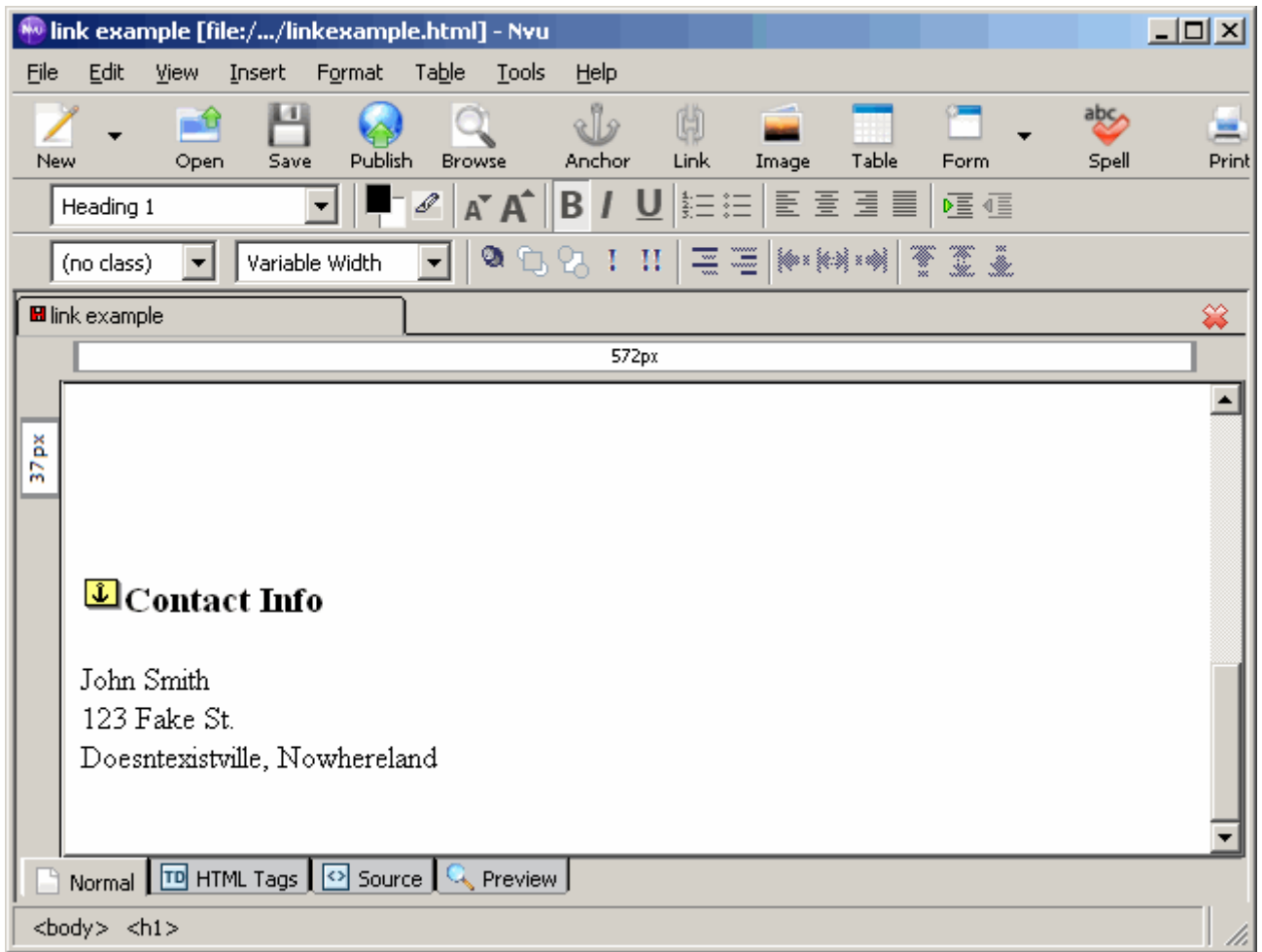
Now you can create the link wherever you want on the page just like Relative or Absolute links. The only difference here is that for the Link Location you need to enter a # sign followed by the name you gave to the anchor. The # sign tells the browser to look for the link within the current page. You can either manually type this in, or if you already have created the anchor, you can select it from the Link Location drop down menu.



The above internal link would look like this,



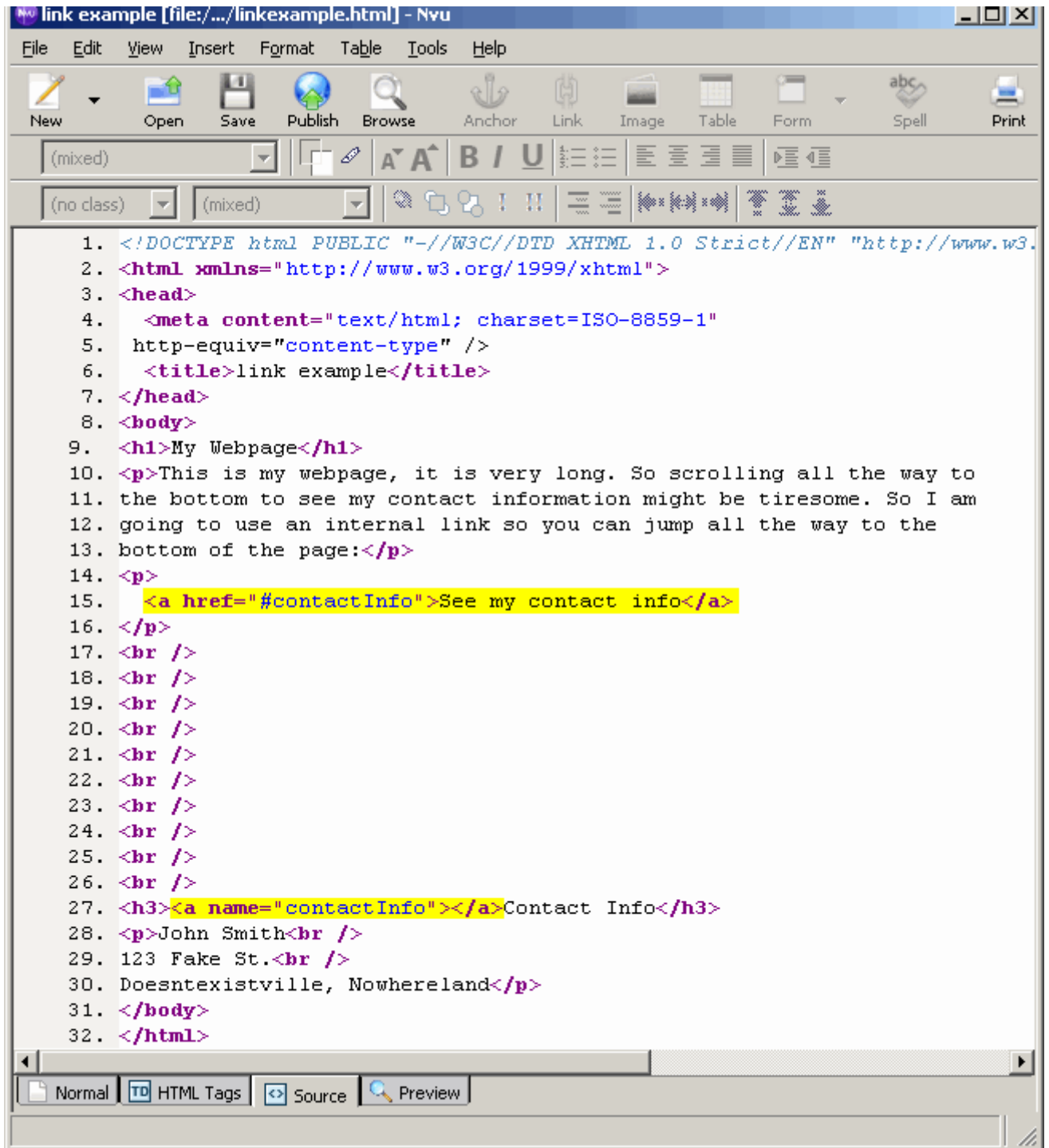
By clicking the above link you will be taken to the bottom of the page, which is where you put the anchor.



Notice the anchor symbol in Nvu. This will not appear on your webpage, it's just there to show you during editing that there is an anchor there.

The source code for our internal link looks like this





The screenshot shows the Nvu web editor interface. The title bar reads "link example [file:///.../linkexample.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, Spell, and Print. Below the toolbar are two style selection boxes, both set to "(mixed)". The main editing area displays HTML source code with line numbers 1 through 32. Line 15 contains the anchor tag `<a href="#contactInfo">See my contact info</a>`, which is highlighted in yellow. Line 27 contains the anchor tag `<a name="contactInfo"></a>`, also highlighted in yellow. The code defines a document type, meta charset, title, and body content including a paragraph with an internal link and a contact information section.

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title>link example</title>
7. </head>
8. <body>
9. <h1>My Webpage</h1>
10. <p>This is my webpage, it is very long. So scrolling all the way to
11. the bottom to see my contact information might be tiresome. So I am
12. going to use an internal link so you can jump all the way to the
13. bottom of the page:</p>
14. <p>
15.   <a href="#contactInfo">See my contact info</a>
16. </p>
17. <br />
18. <br />
19. <br />
20. <br />
21. <br />
22. <br />
23. <br />
24. <br />
25. <br />
26. <br />
27. <h3><a name="contactInfo"></a>Contact Info</h3>
28. <p>John Smith<br />
29. 123 Fake St.<br />
30. Doesntexistville, Nowhereland</p>
31. </body>
32. </html>
```

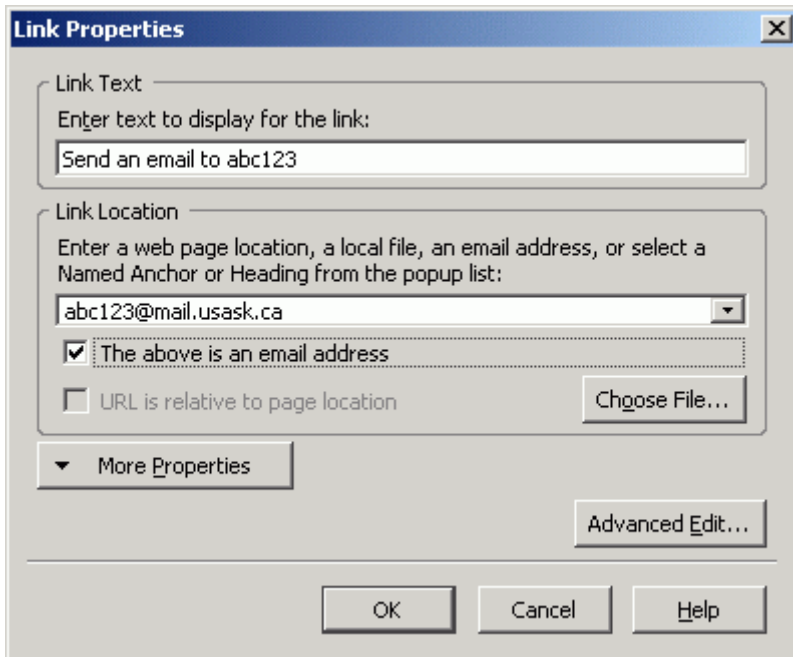
We can see our anchor is simply the anchor tag with the **name** attribute to label the spot we want to link to. Our link still contains the href attribute.

### *Mailto*

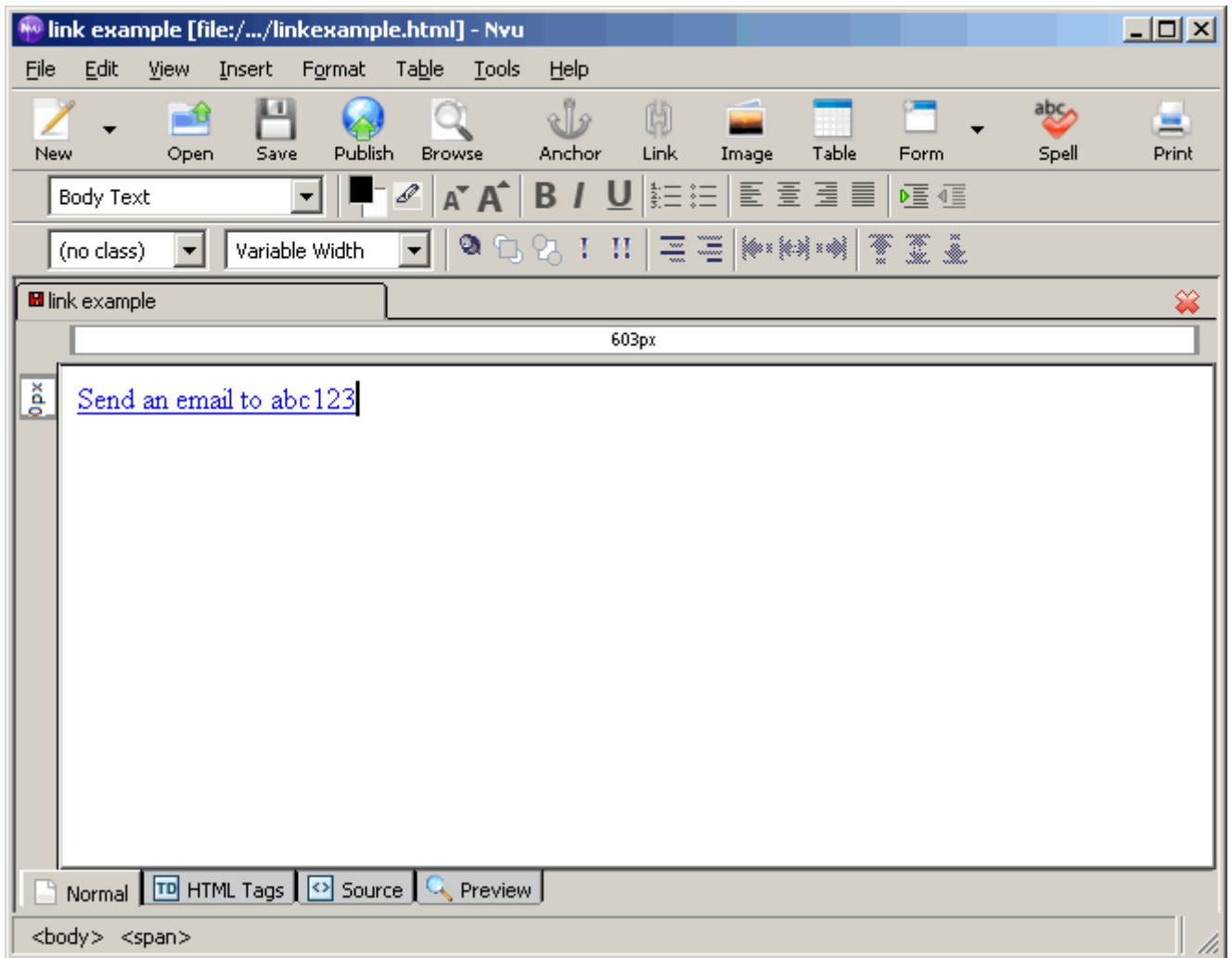
Mailto links look like regular hyperlinks, but instead of bringing you to a new location on the web, they try to open up your email program and start an email to the specified address. The browser will automatically fill in the email address in the "To" field so that you do not need to type it in.

The Mailto link looks just like any other link on a webpage but it behaves differently. Mailto opens up a new window to send an email to the recipient specified.

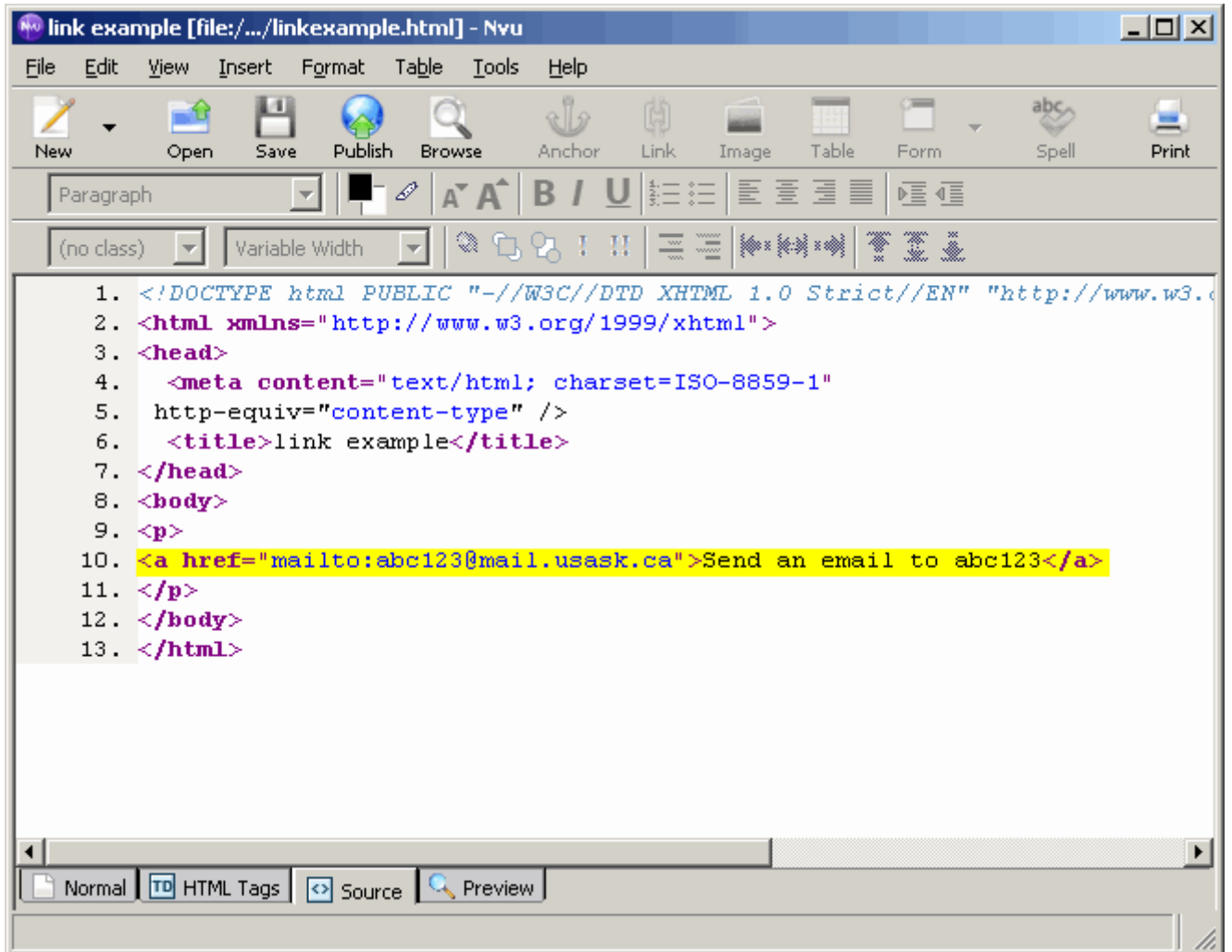
Mailto links are created like any other link in Nvu. However in the Link Location text box rather than putting a link, this is where you put the email address you want the link to send to. Before you click OK, check the box that says "The above is an e-mail address" and your Mailto link will be created.



The above link will appear like this on a webpage:



Nvu creates the following source code



Notice the value for the href attribute is "mailto:" followed by the e-mail address to mail to. This tells the browser to open up the default e-mail program on the computer and create a new message with the specified e-mail address in the **To:** field.

All hyperlinks are created by using Nvu's Link button in the Button bar. Depending on the type of hyperlink created, the "Link Location" will be different. Also remember that none of your hyperlinks will work within Nvu. To see the functionality of a link, you must view your page in a browser.

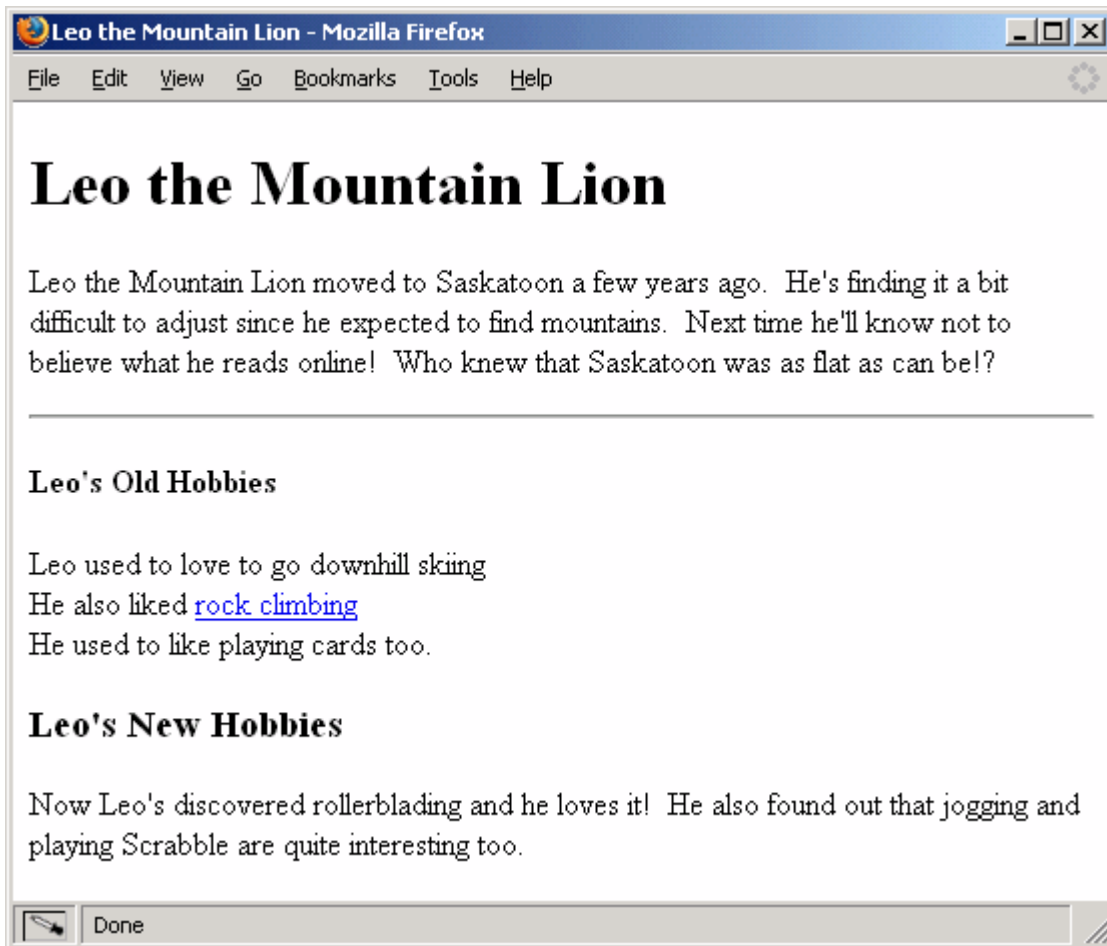
## Conclusion

In this section we saw there are several elements available to make your webpage appear exactly as you want it to. We saw how headings and paragraphs are used to organize a webpage and make it look tidy. We saw how we can use horizontal rules to divide a webpage into sections. We also learned how to create four different types of hyperlinks to connect our webpage to others on the World Wide Web.

## Exercises

1. Define the keywords in this lesson.
2. Create the following webpage in Nvu. Do not worry about exact sizing for the headings, approximate size is good enough.

3. The link used is an external link to [www.rockclimbing.com](http://www.rockclimbing.com)



## Images

Most webpages contain images as well as text. This lesson demonstrates how to insert images into your webpages as well as describes the formatting options available for images.

### *Objectives*

- Use images in a webpage
- Change the dimensions of an image in a webpage
- Use images as hyperlinks
- Use images as a webpage's background

### **Keywords**

- `img` tag
- background
- width
- height
- border

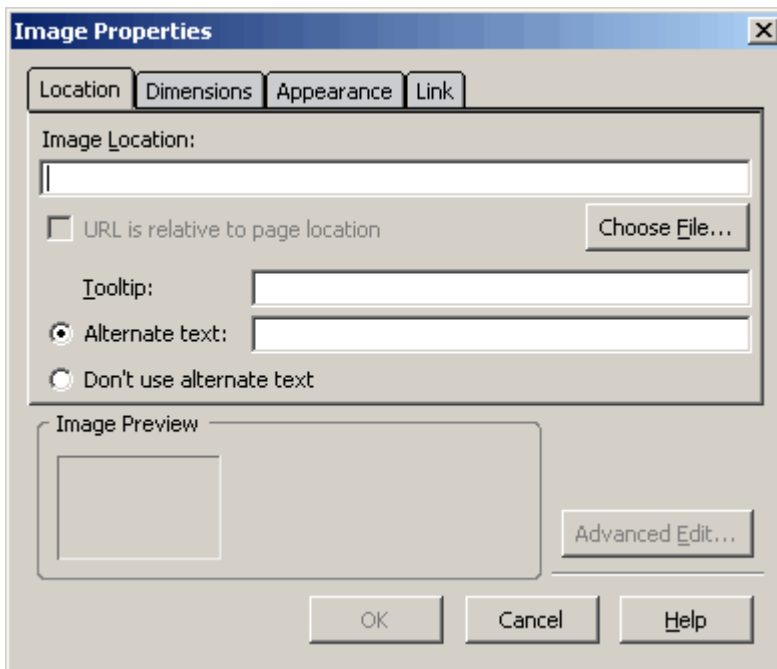
- alt

## Adding Images

Whenever you see a webpage that contains images, you'll find that these images are generally stored in separate files and then referred to within the XHTML code. Images used on webpages should be of specific file types - they can be either .gif or .jpg (pronounced "jif" and "J-peg").

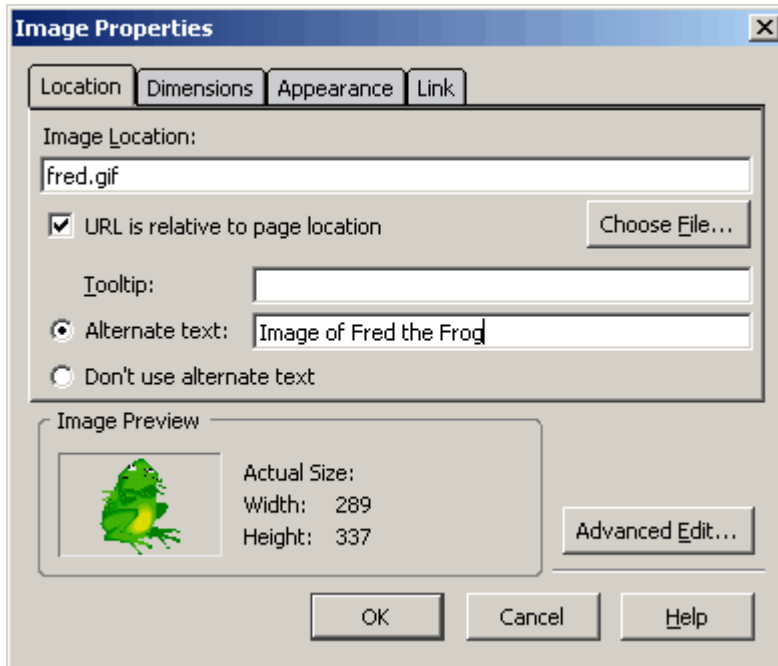
Button and icon images such as a "Home" button provide a simple way to spice up webpages. In most cases they are relatively small files that can be downloaded quickly - but beware that every image causes an additional trip to the server, so lots of icons can weigh your page down and make it slow to load. You should be careful that any image file you use is relatively small - images taken from digital cameras or scanned into the computer are typically very large. You may need to resize your images before using them on your page.

Images are inserted in Nvu by using its **Image** button in the button bar. When you click on the image button the following window will pop up,

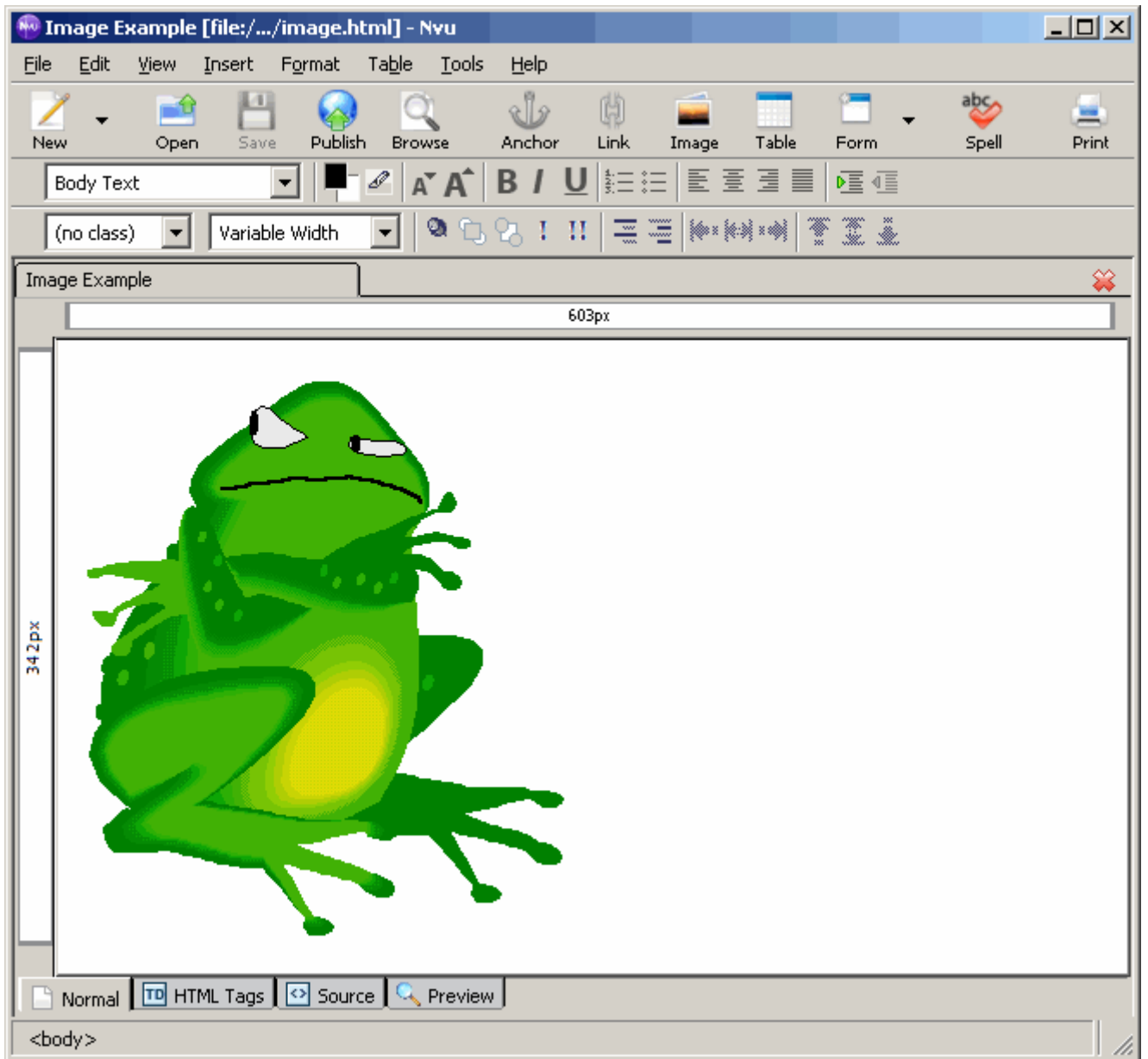


The next step is to include the name of your image file in the **Image Location** text box. You can do this one of two ways:

1. Simply type the name of the file into the textbox
2. Click on the **Choose File...** button and select your image file



Make sure you include a brief description of your picture in the **Alternate text** text box. You must include this for your page to be XHTML Strict. Once you have completed that click OK and your picture will appear on your page.



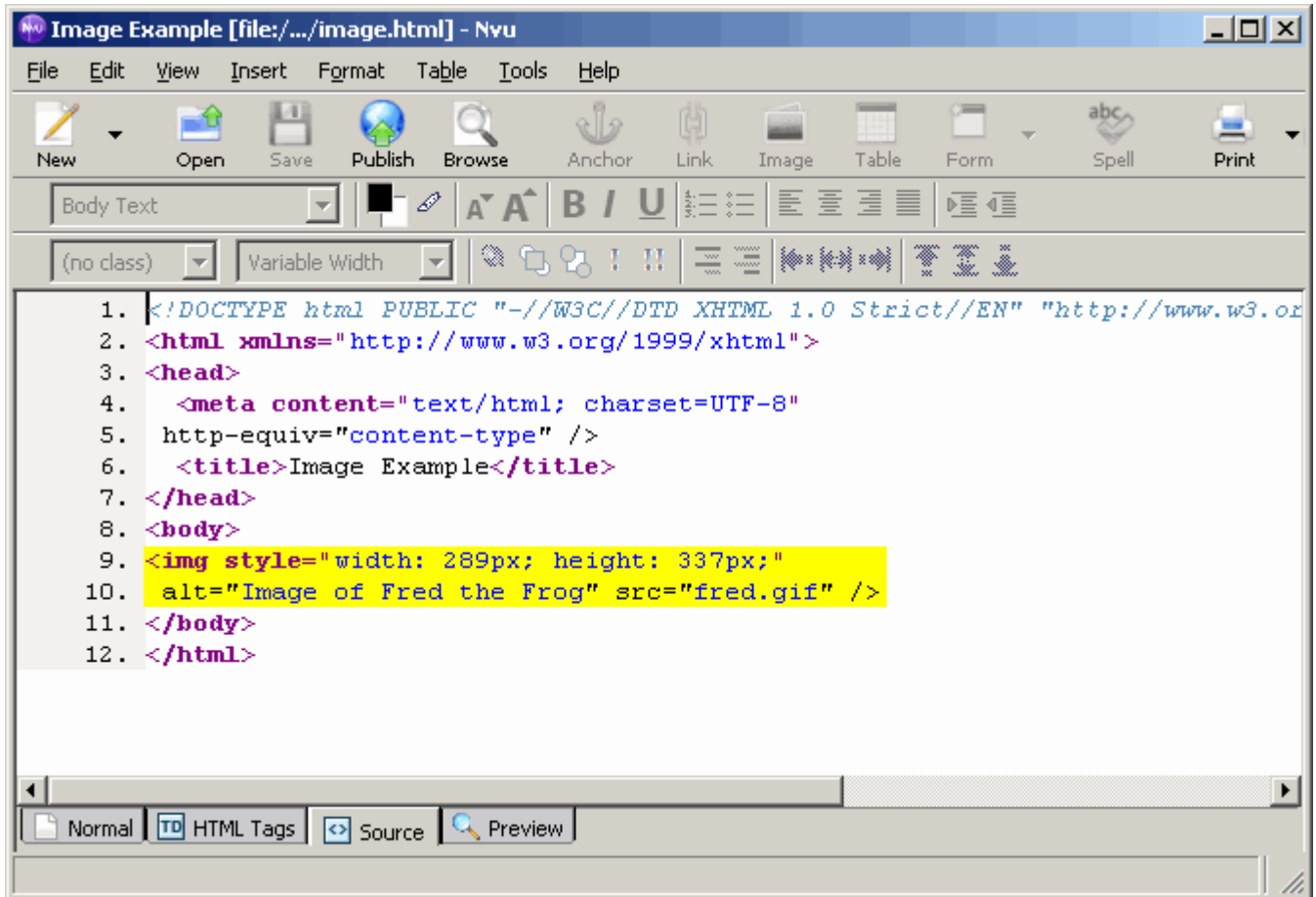
If the image cannot be found then the Alternate text will be displayed. For example:





*It is very important to save your image files to the same folder that your webpage file is in. Otherwise images will become more difficult to manage and problems may arise when submitting assignments to E-Handin.*

When you insert an image into your webpage, Nvu creates the `<img />` tag in your source code. The `<img />` tag is an empty element because it does not have a corresponding closing tag. Both the opening and closing tags are combined into one. Notice how the `/` is included at the end of the tag (with a space preceding it). This indicates to the browser that the element is complete and it should not look for a corresponding closing tag.



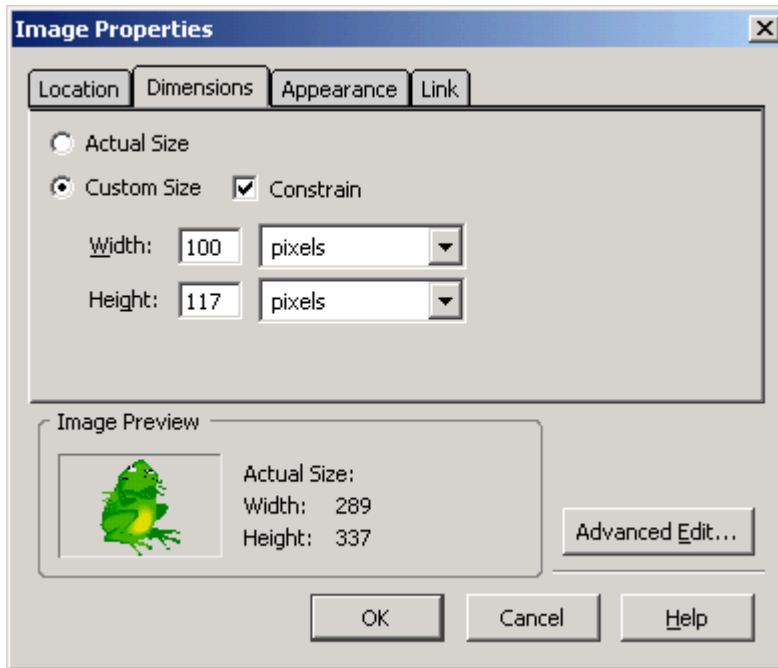
You can see the **src** attribute in the code signifies what file to use for the image. The **alt** attribute contains the Alternate text. Here we see the **style** attribute as well which sets the width and height of the image. We will look at this more in the next lesson.

## Resizing, Aligning and Borders

### *Resizing*

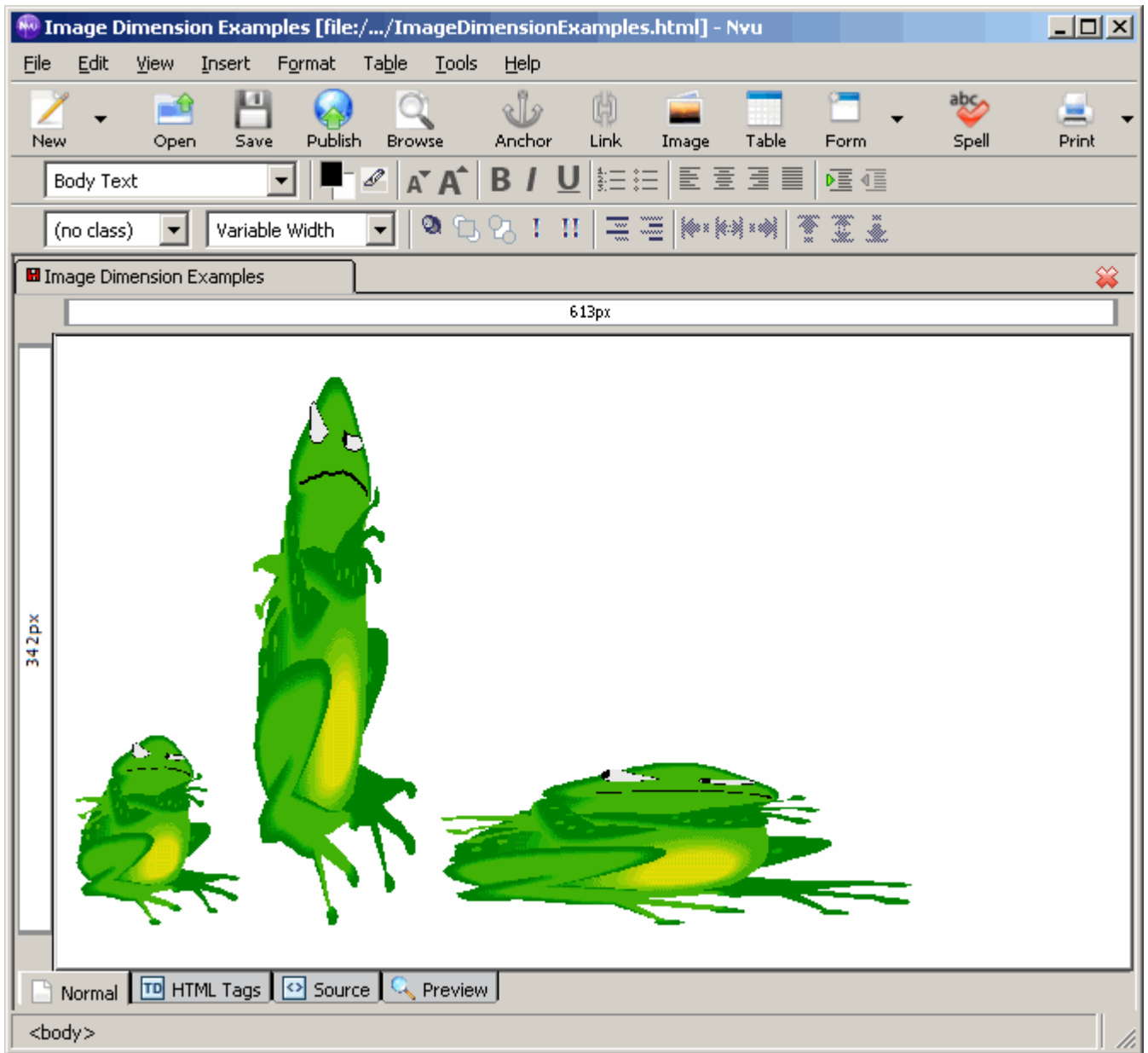
Several properties are available for the image tag, detailing how the image should be displayed. The first two affect the size of the image. Without them, the image will be displayed in its original size. You can set **height** and/or **width** properties to specify what size the image should be on the page. The height and width are specified in pixels or as a percentage relative to the size of the window.

You can set the height and/or width in Nvu in the **Image Properties** window. This is the window that pops up when you first insert an image. It is also accessible by right-clicking on an existing image and selecting **Image Properties...** To change the dimensions click on the **Dimensions** tab in this window. Here you have the option of keeping the Actual Size of the picture, or setting the custom size yourself. By clicking the **Custom Size** radio button you can specify the height and width for your image.

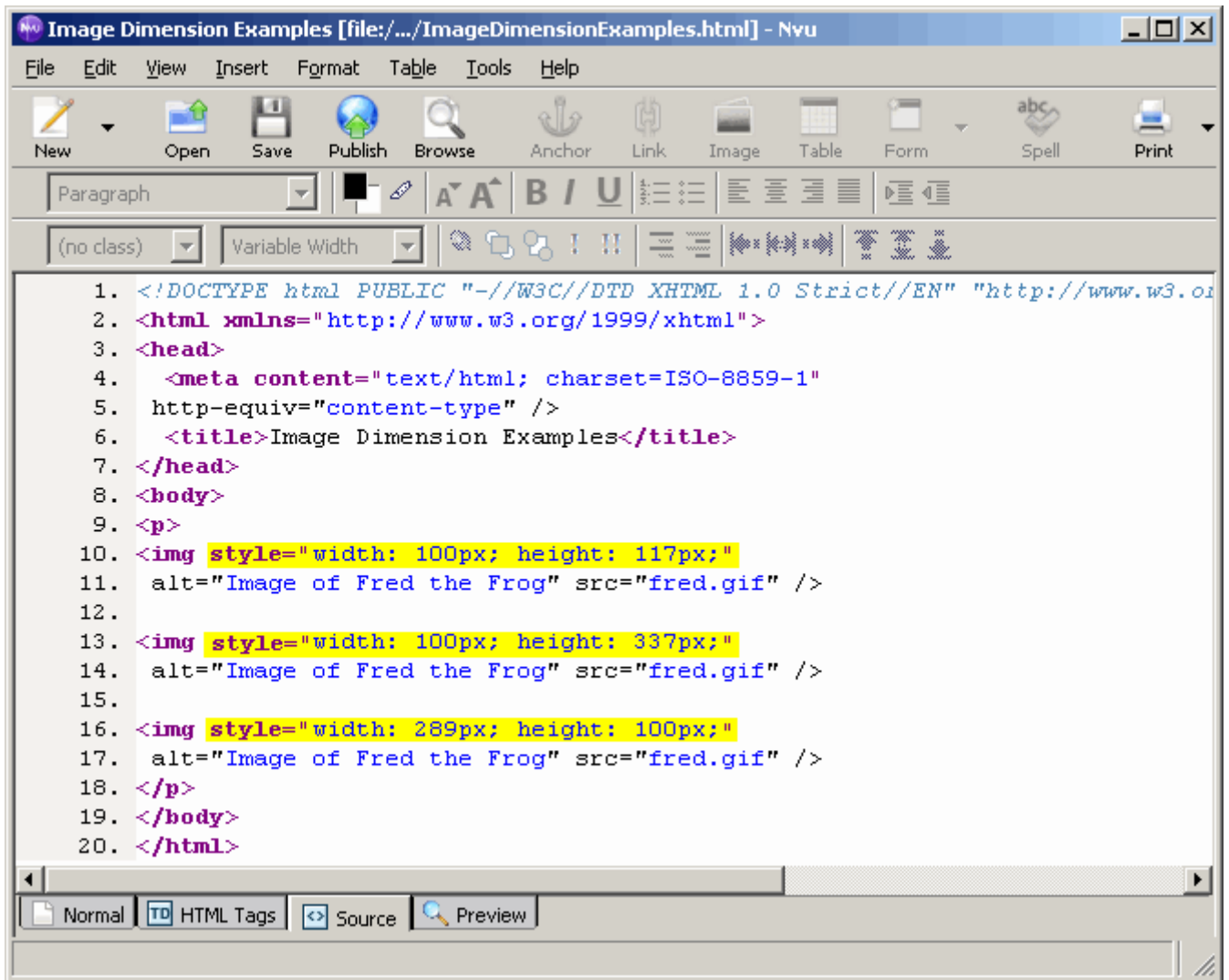


Normally, you would only specify either the height or the width, not both. When only one is specified, the image is resized proportionally, as long as the **Constrain** checkbox is checked. If you wish to stretch a picture vertically or horizontally, you can uncheck the **Constrain** checkbox and set the values for both height and width.

Here are a few examples,



The first image has its width set to 100px and its dimensions constrained. The "px" stands for "pixels". Pixels are small dots on your computer screen that take on a specified color. They combine to form the image you see on your monitor. The second image has its width set to 100px, but its dimensions are not constrained. Finally, the third picture has its height set to 100px and its dimensions are not constrained. If we look at the source code in Nvu we can see that it is the same image that is being used as the source, but with different values for height and width.



You can see all the examples are almost exactly the same except for the height and width values in the "style" attribute. As we change our height and width, these values are changed. The "style" attribute lets us set various formatting styles for an element. The basic syntax for styles is always:

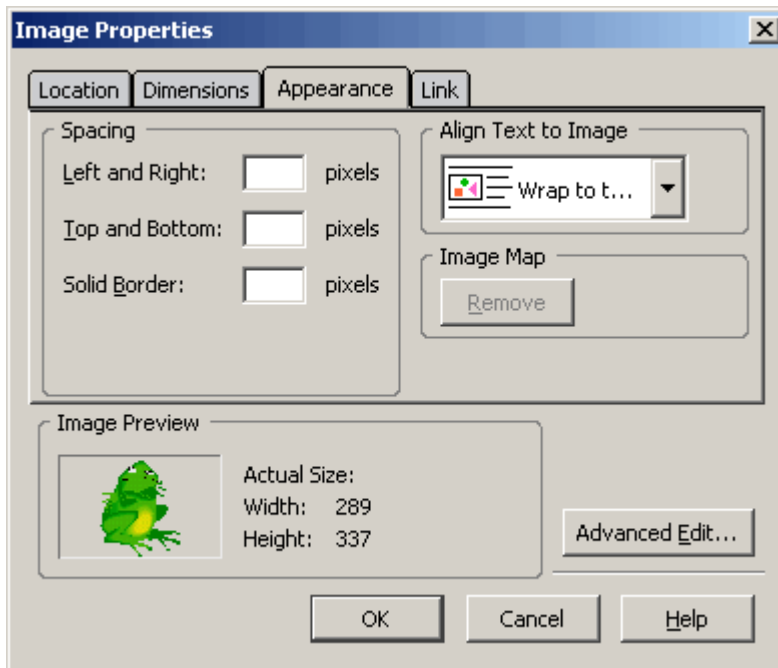
```
style="property: value;"
```

You can add as many styles as you wish, as long as they are separated by a semi-colon. We will look at **style** more closely in the formatting lesson.

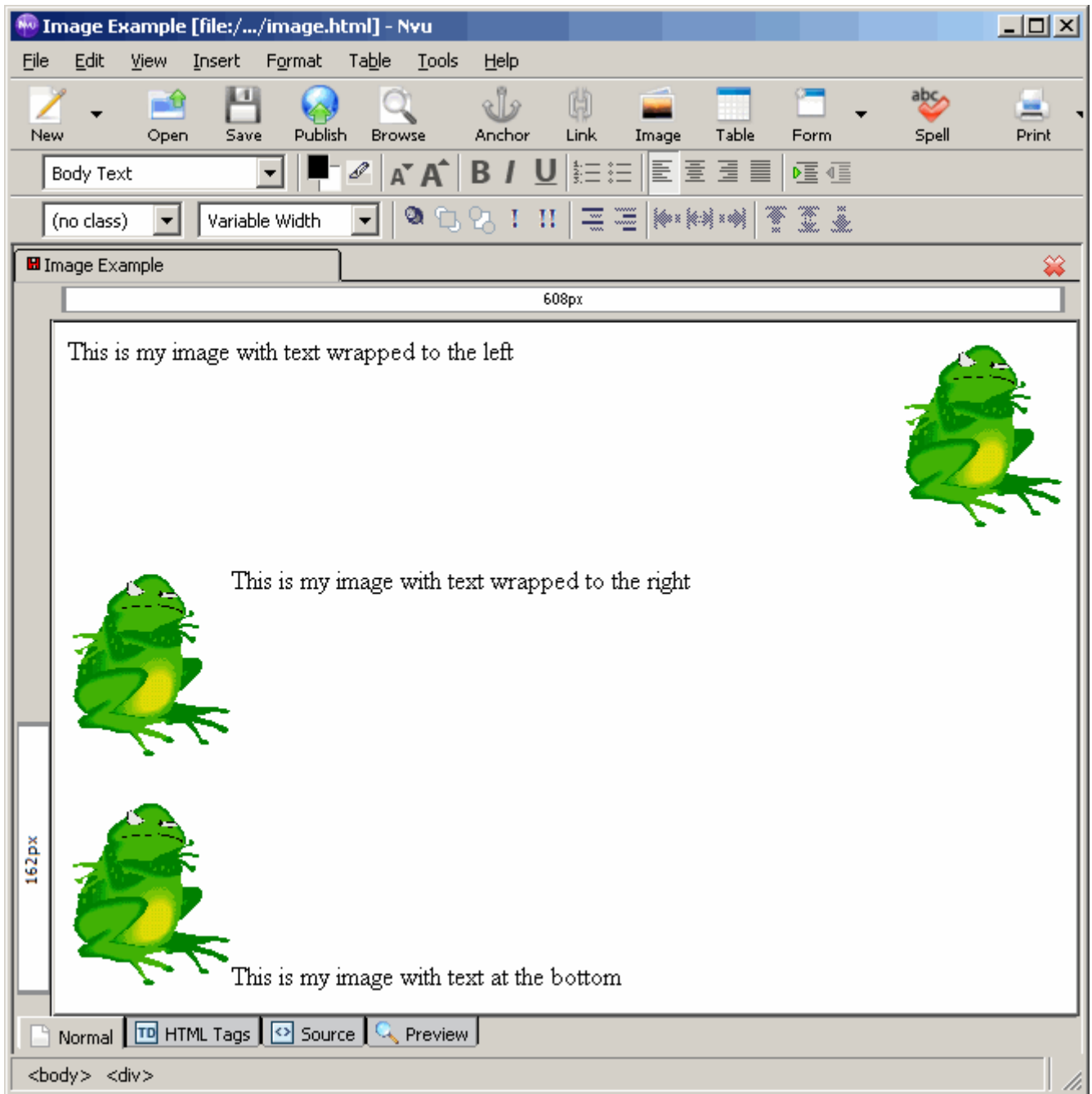
### *Aligning*

To align a picture in Nvu, select the picture and click on whichever alignment you wish (left, right, center) in the formatting bar, just like you would for a paragraph.

Nvu also allows you to choose how you want text to align relative to your image. If you bring up the **Image Properties** window you can click on the **Appearance** you will see an **Align Text to Image** select list.

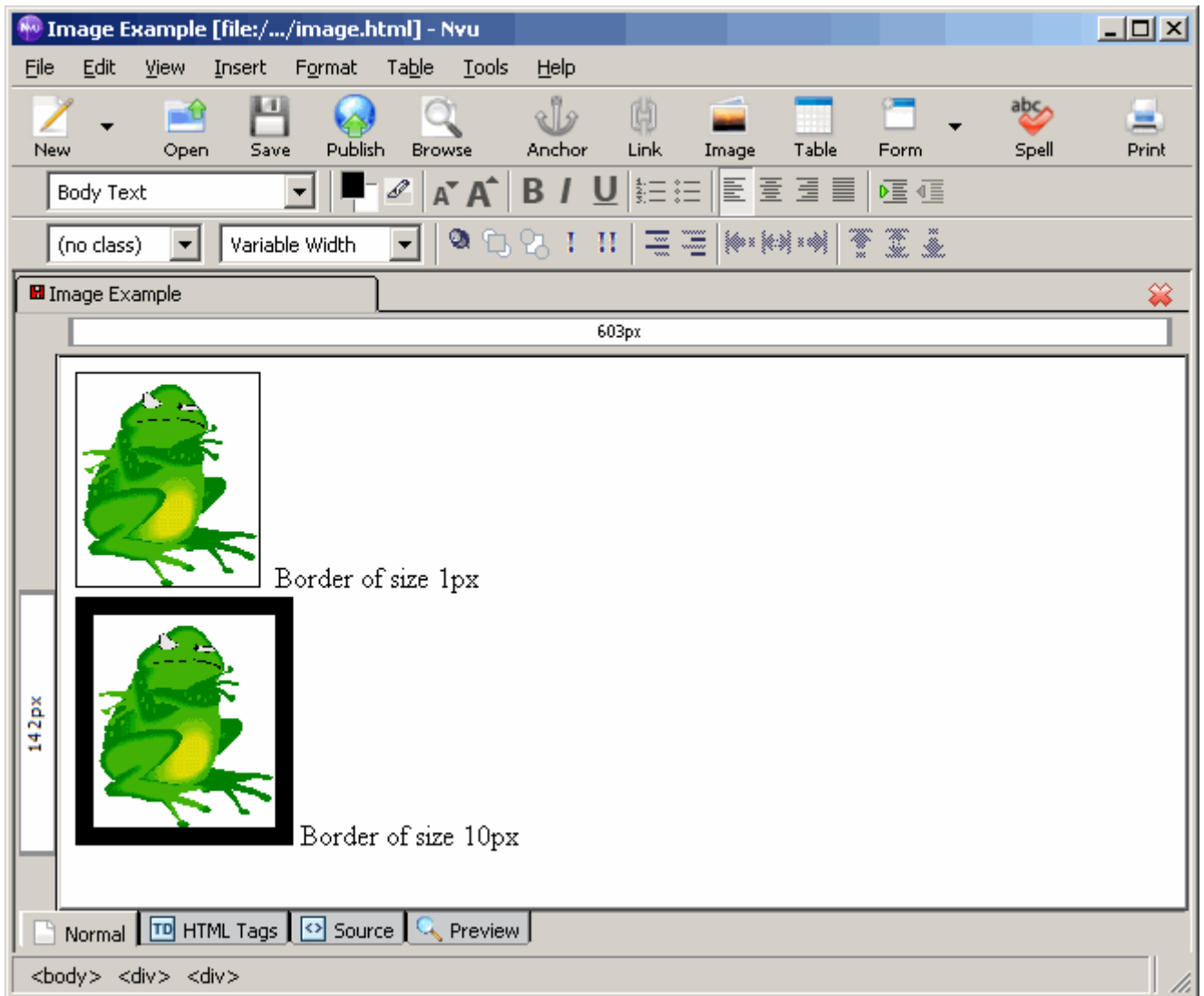


Here you will have three options: At the bottom, Wrap to the left, and Wrap to the right.



### ***Borders***

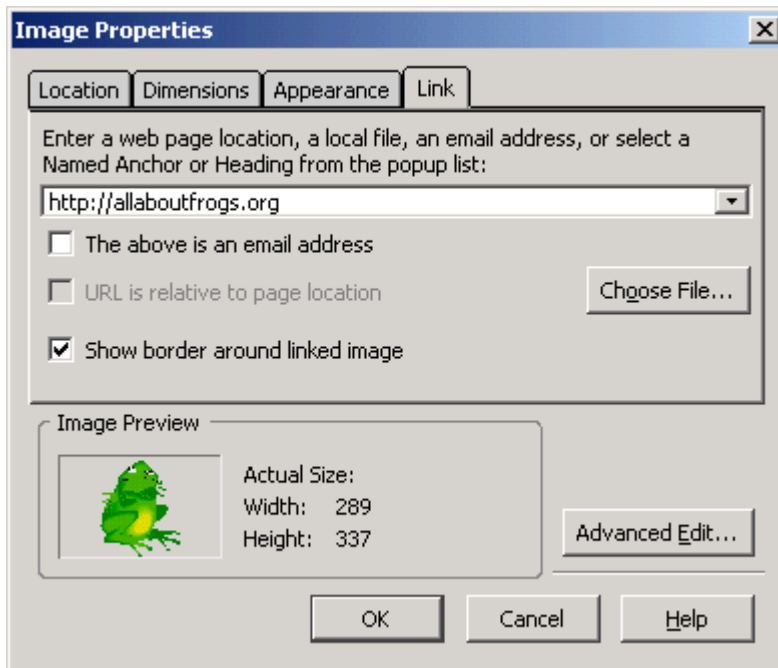
In the **Appearance** tab of the **Image Properties** window, you also have the option to set the border for an image. Simply enter the size (in pixels) that you want your border to be in the **Solid Border** text box. The smaller the number, the smaller the border. By default, there is no border around images.



### Images as Hyperlinks

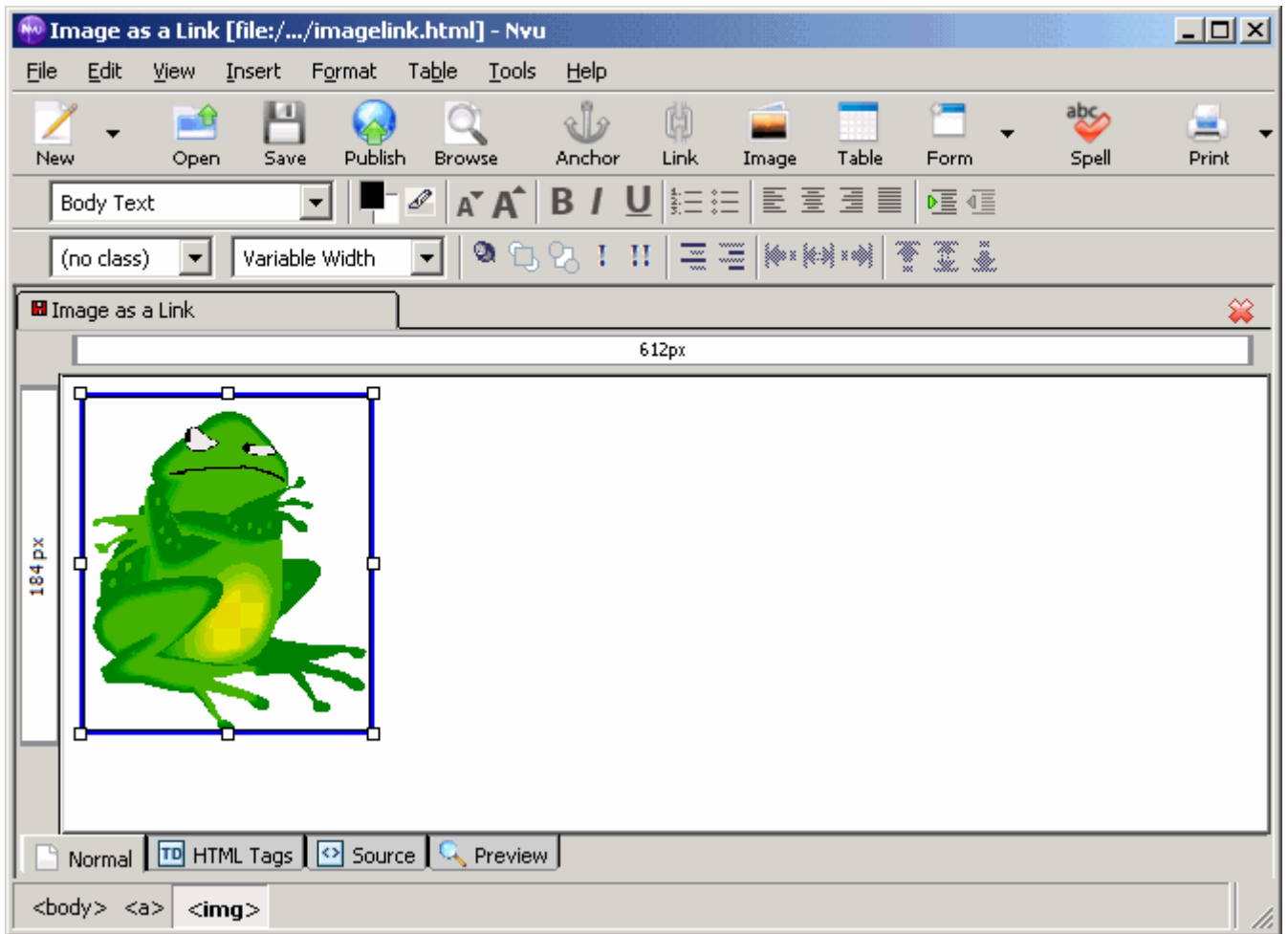
To make an image a hyperlink in Nvu (so that when the user clicks on the image they go to another page or anchor within your page), open up the images **Image Properties** window and click on the **Link** tab.





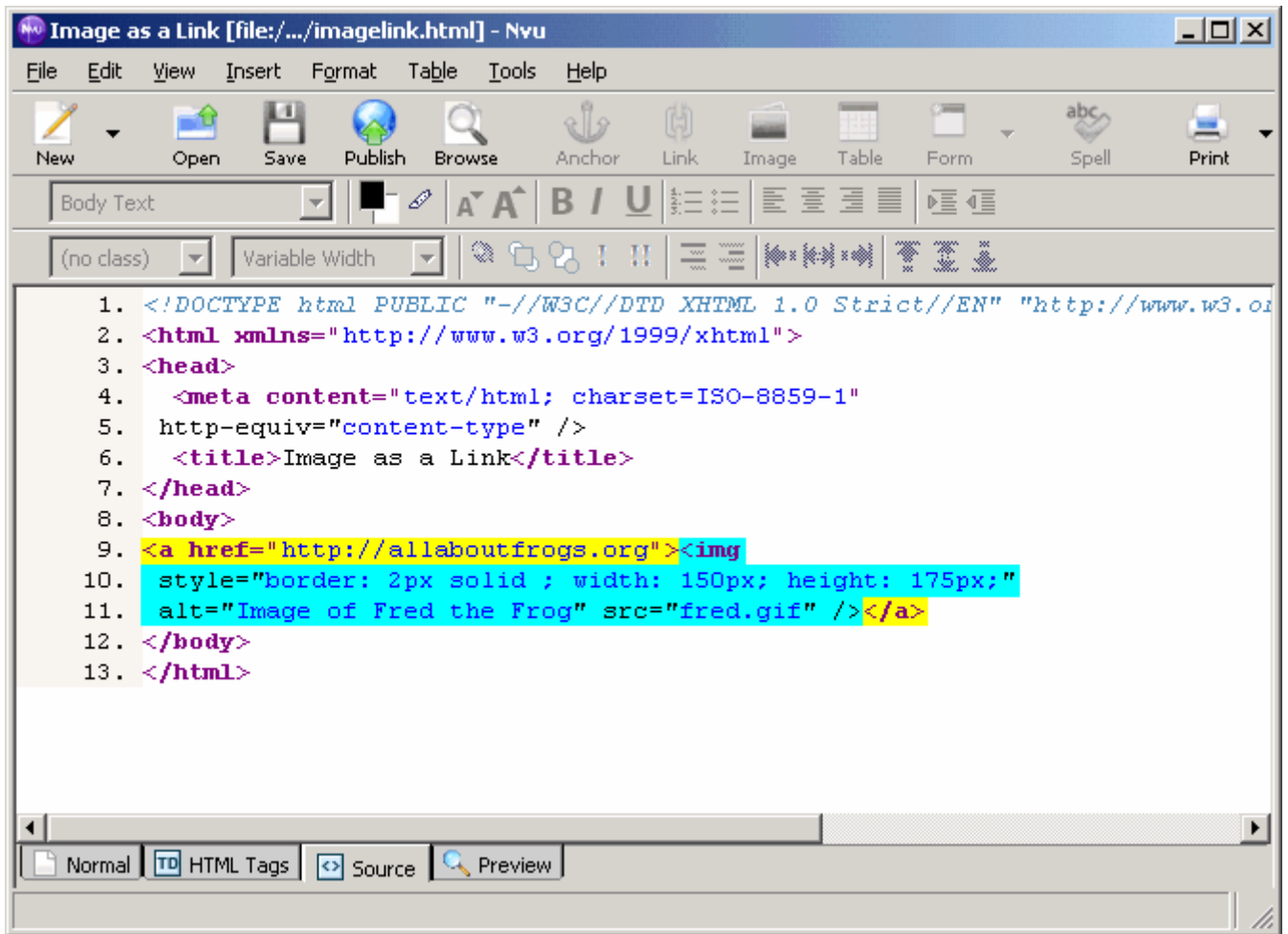
In the text area where it asks you to "Enter a web location" you can insert the URL for the location you wish to link to. This can be any of the four types we saw in the hyperlinks lesson (absolute, relative, internal or mailto).

If you wish to put a border around the image, you can check the **Show border around linked image** checkbox. The border around your image will be the default color for your hyperlinks. This is often a good idea, as it lets users know that the image is a link.



The link will look like the image above, when it is clicked on it will take you to <http://allaboutfrogs.org>.

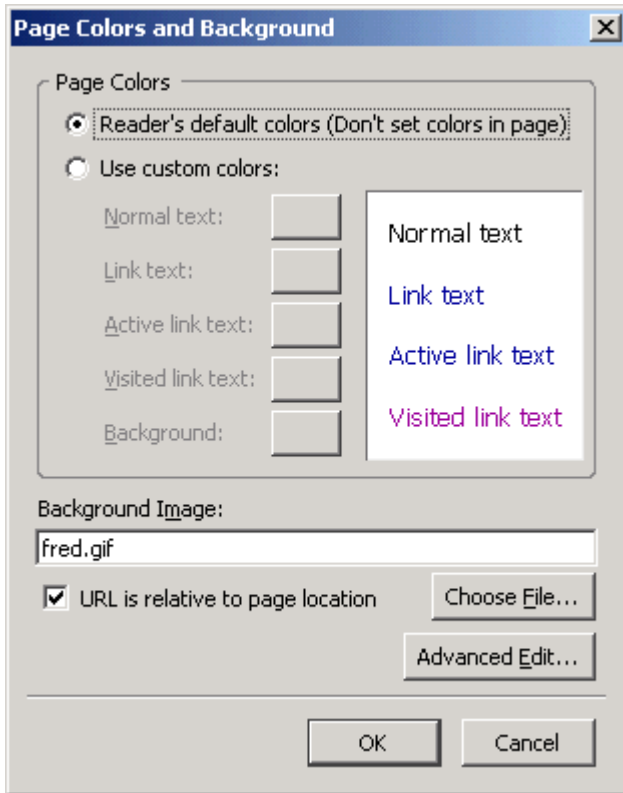
If we look at the source code that Nvu creates



Notice that the `<img />` tag is nested with the `<a><a>` tags where our linked text would usually go. This is what causes the image to be the clickable link, rather than text.

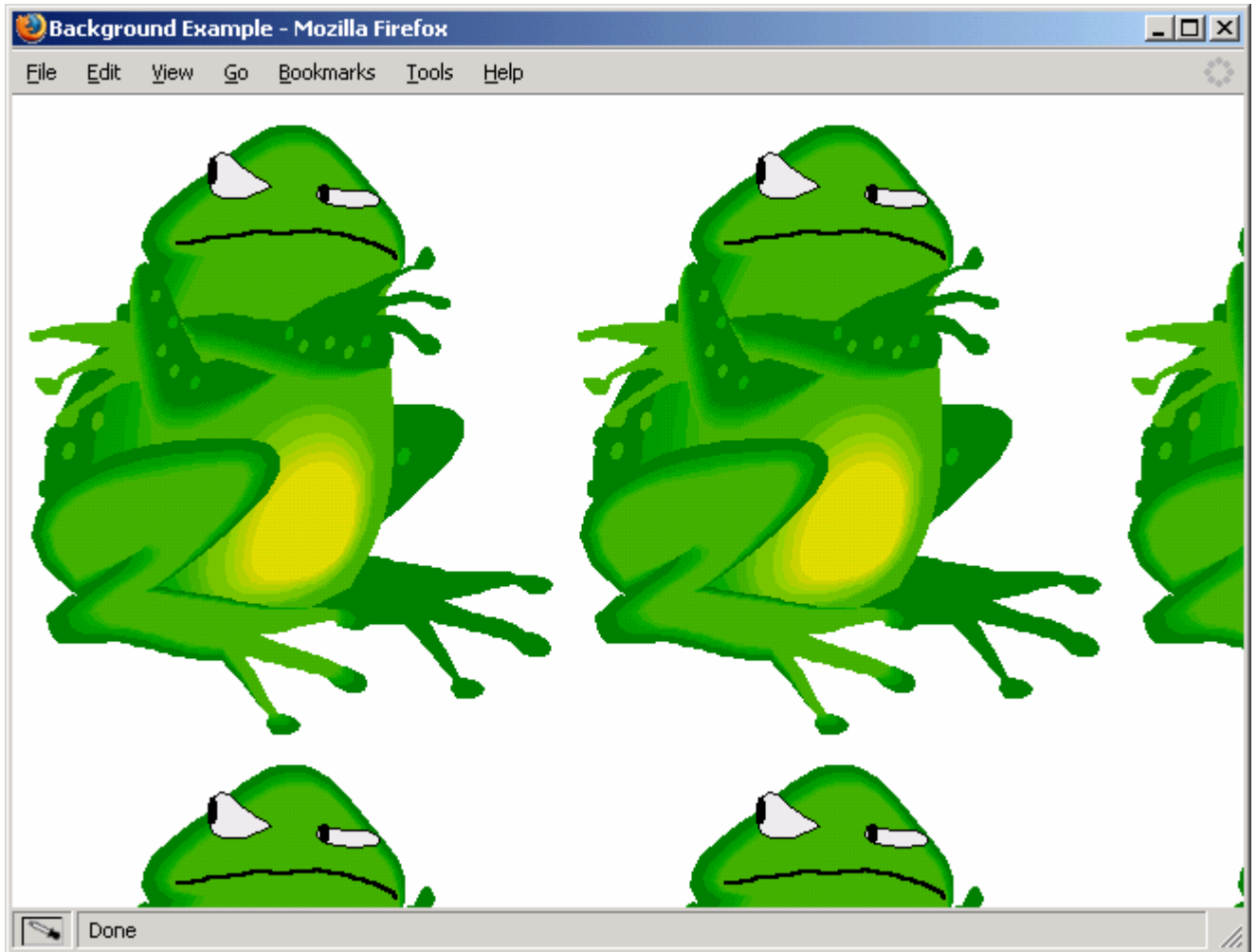
### Images as Background

Nvu allows you to set an image as the background of your webpage. To do this click on **Format -> Page Colors and Properties**.



In this window, type the name of the image file into the **Background Image** textbox or click on the **Choose File** button to select your file. Just like images on your webpage, make sure your background image is in the same folder as your webpage.

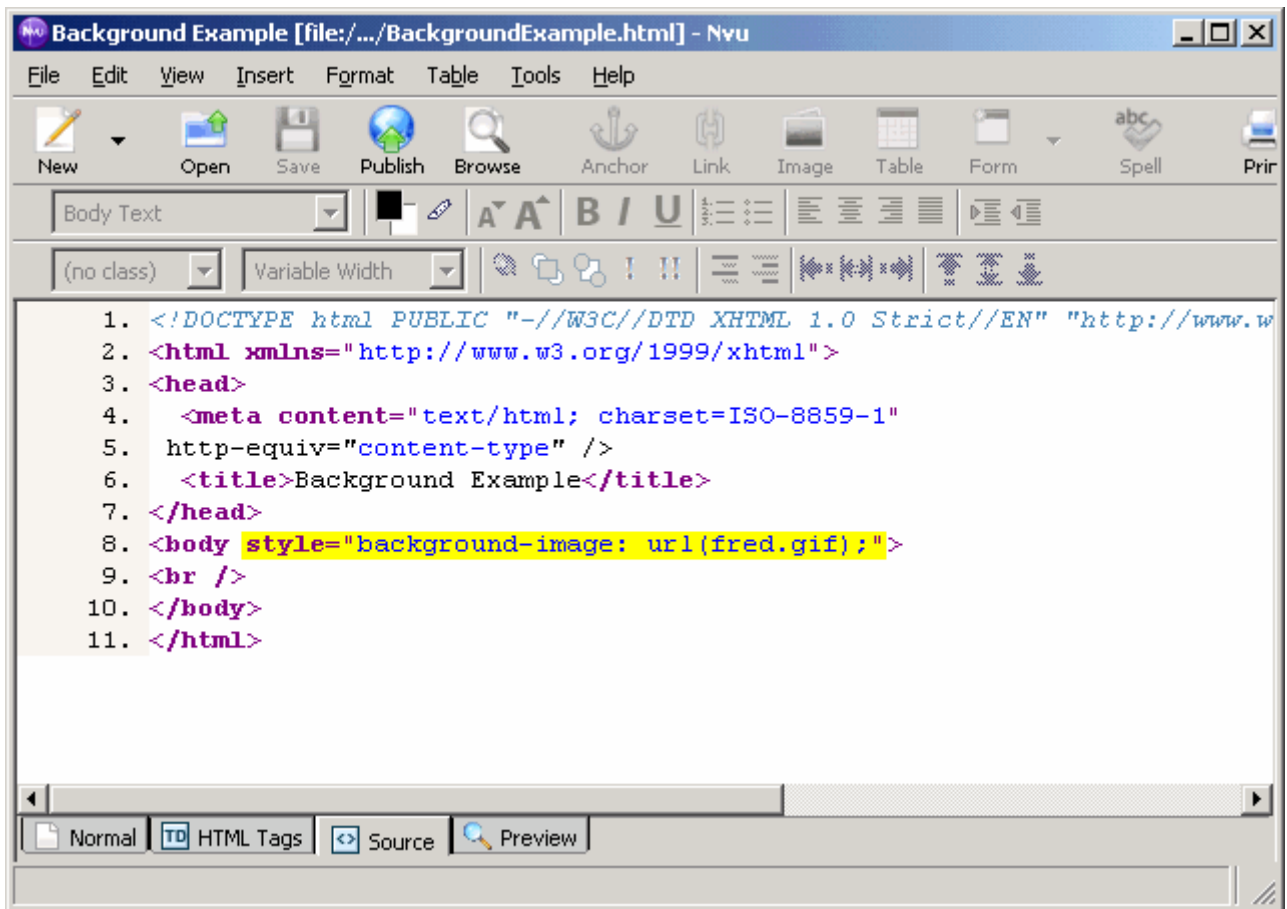
When you click **OK** you may not see the background appear in Nvu. This is the case if you do not have anything on your page yet. To see your background you can either begin to add content to your webpage, or view your webpage in a browser. The above example viewed in a browser looks like this



Notice that the image is "tiled" - meaning that it is repeated as many times as necessary to cover the entire background of the webpage. Typically background images are much less "busy" and bright than this one, but this made it easier to show exactly what is happening.

Background images act like a layer behind your webpage. Any text or other images that are included within the body of your page will appear on top of the background image.

Looking at the source code, we see that Nvu adds the background image by placing the **style** attribute in the opening `<body>` tag. Because everything on our webpage is inside our `<body></body>` tags, the specified image gets set as the background for everything.



## Conclusion

This lesson showed how images can be used to visually enhance a webpage. We saw how to modify an image's properties such as its dimensions and border. We also learned how to use images as backgrounds and as hyperlinks.

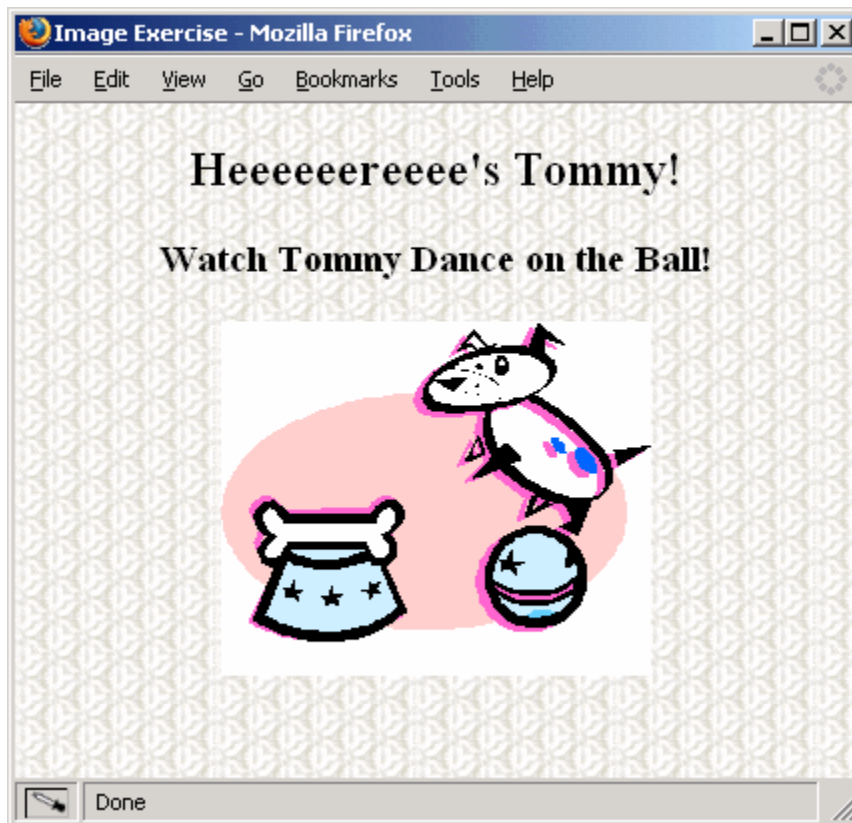
## Exercises

1. Define the keywords in this lesson.
2. Right-click and save the following images to your working directory. Note their filenames as you save the files because you will need to use them later on.





Using Nvu, create a webpage that looks as follows:



Create a link from the image to ensure that when the user clicks the picture, it leads to the URL <http://www.i-love-dogs.com>

### **Advanced Elements**

This module looks at page layout techniques to organize content and make pages more visually appealing; these techniques include lists and tables.

#### *Objectives*

- Create unordered, ordered and nested lists
- Create tables

- Merge cells within a table
- Create nested tables

### Keywords

- Ordered list
- Unordered list
- Nested list
- List Item
- Table
- Cell
- Row
- Column
- Nested Table

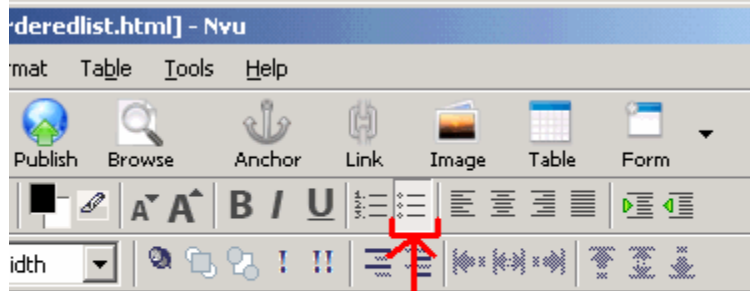
### Lists

There are two types of lists in XHTML: Unordered lists and Ordered lists. Unordered lists are a bulleted list while Ordered lists are numbered.

#### *Unordered List*

To create an unordered list in Nvu is much like creating one in a word processing application like MS Word.

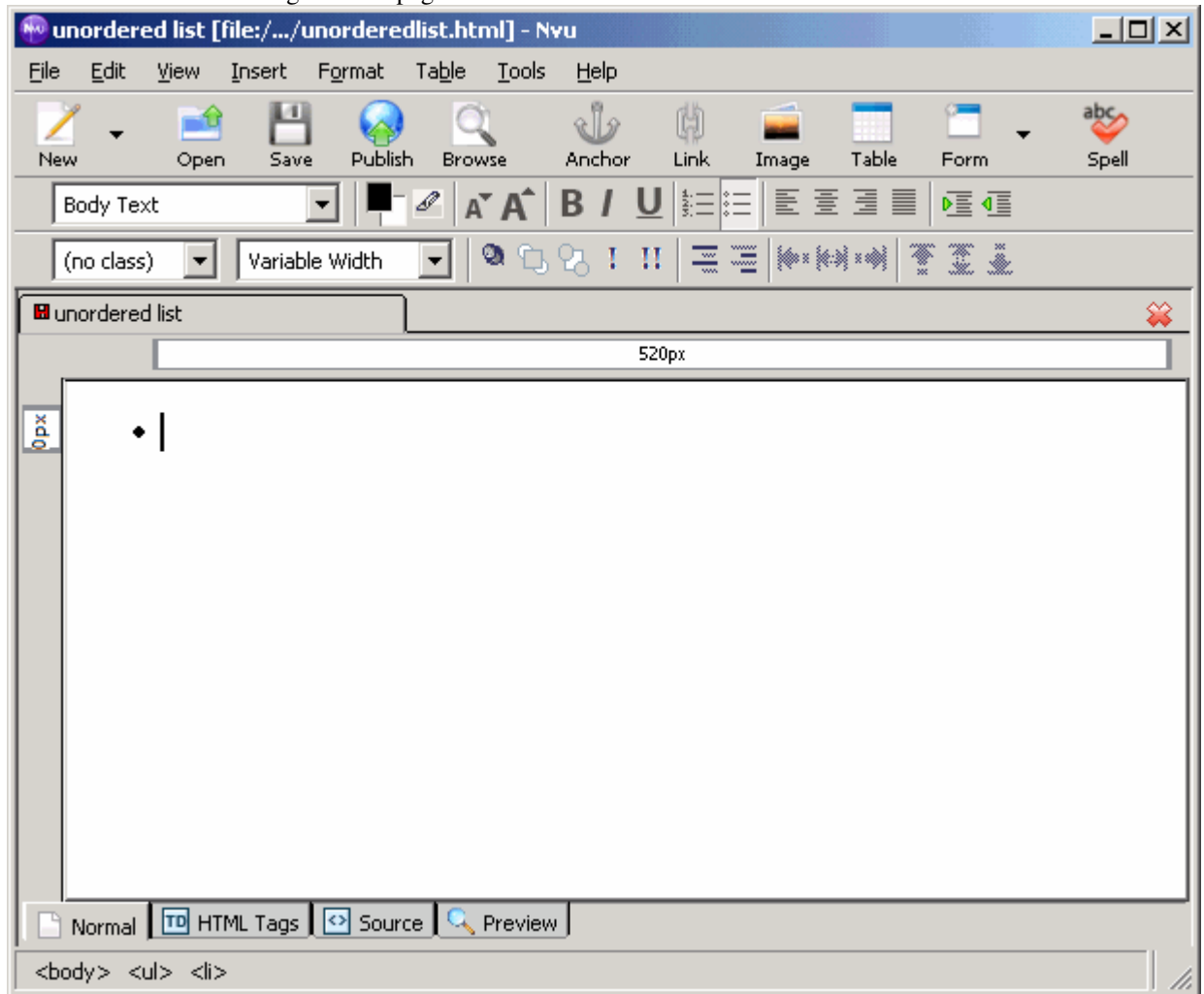
1. In Nvu's formatting bar, click on the Unordered List button



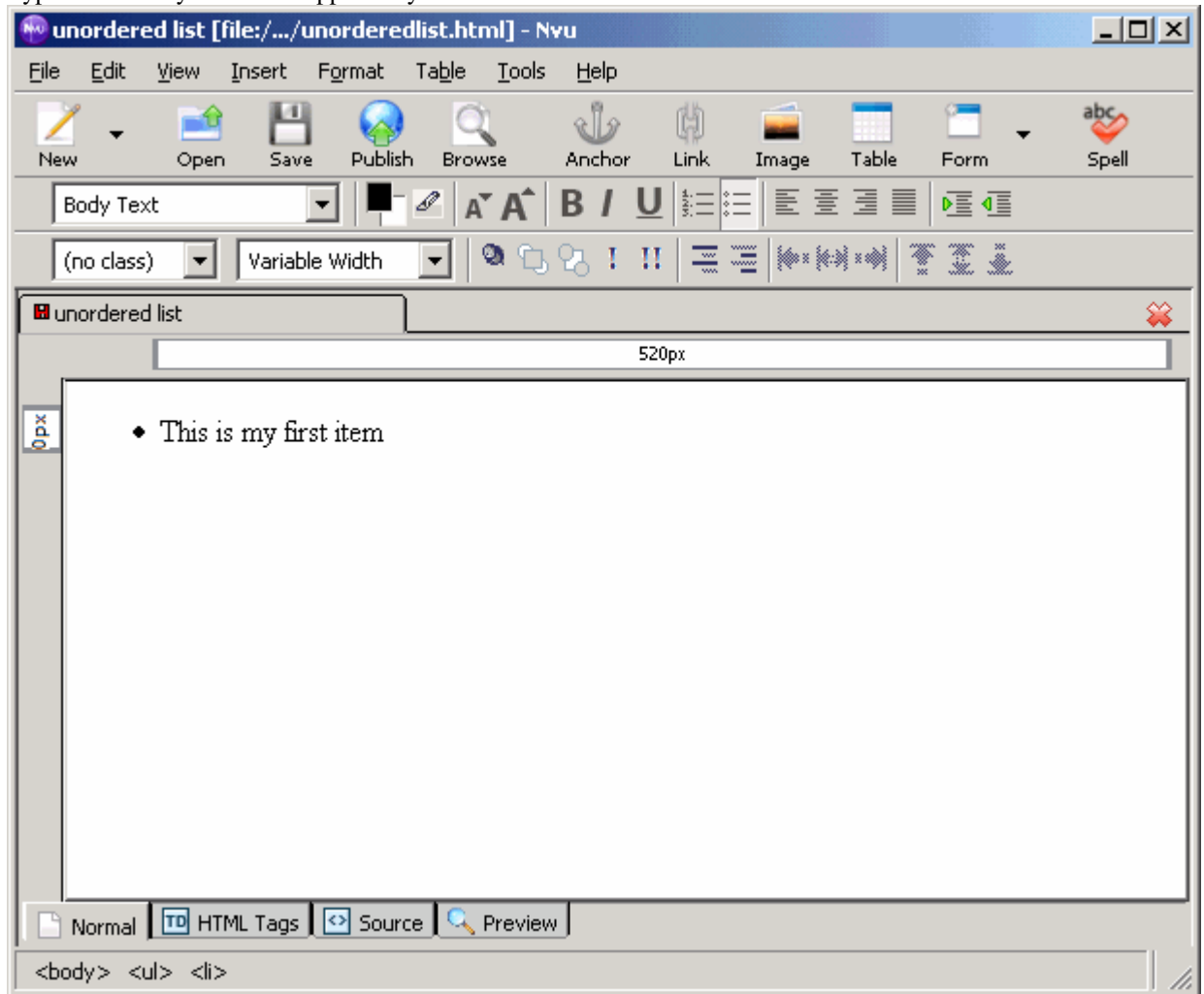
**Unordered List Button**



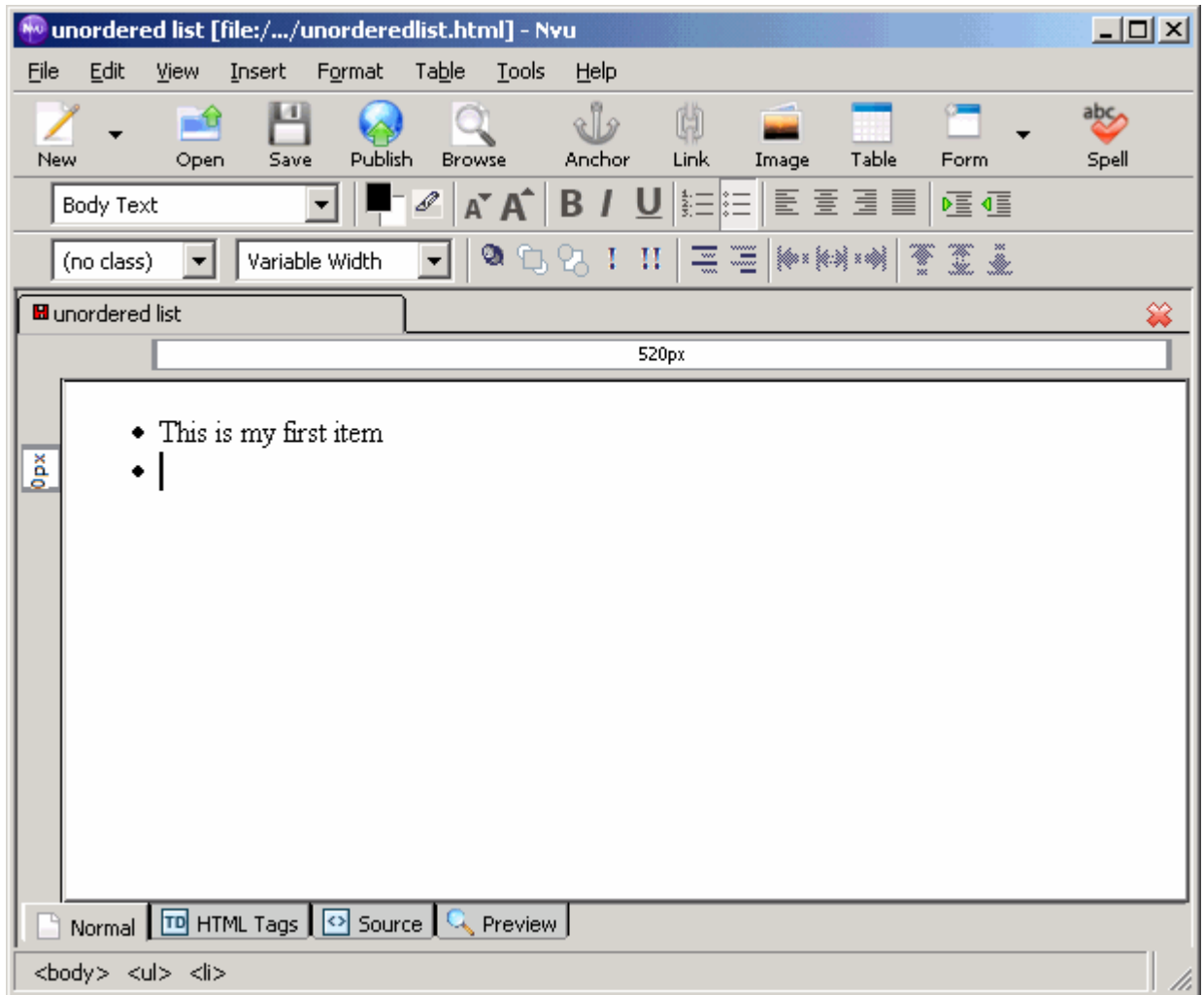
2. Your unordered list will begin on the page



3. Type in the text you want to appear in your list

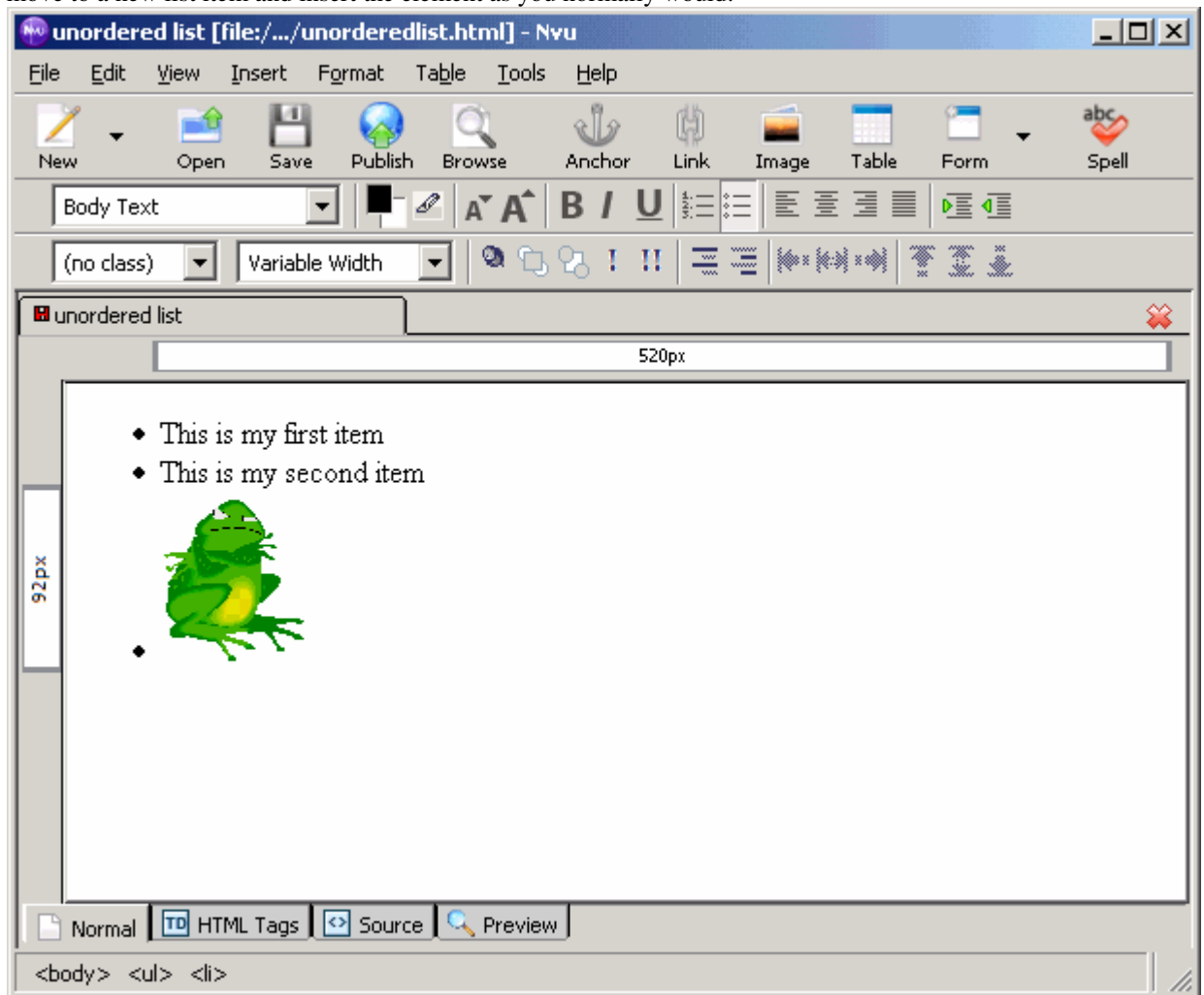


4. Hit **Enter** to move to the next item in the list



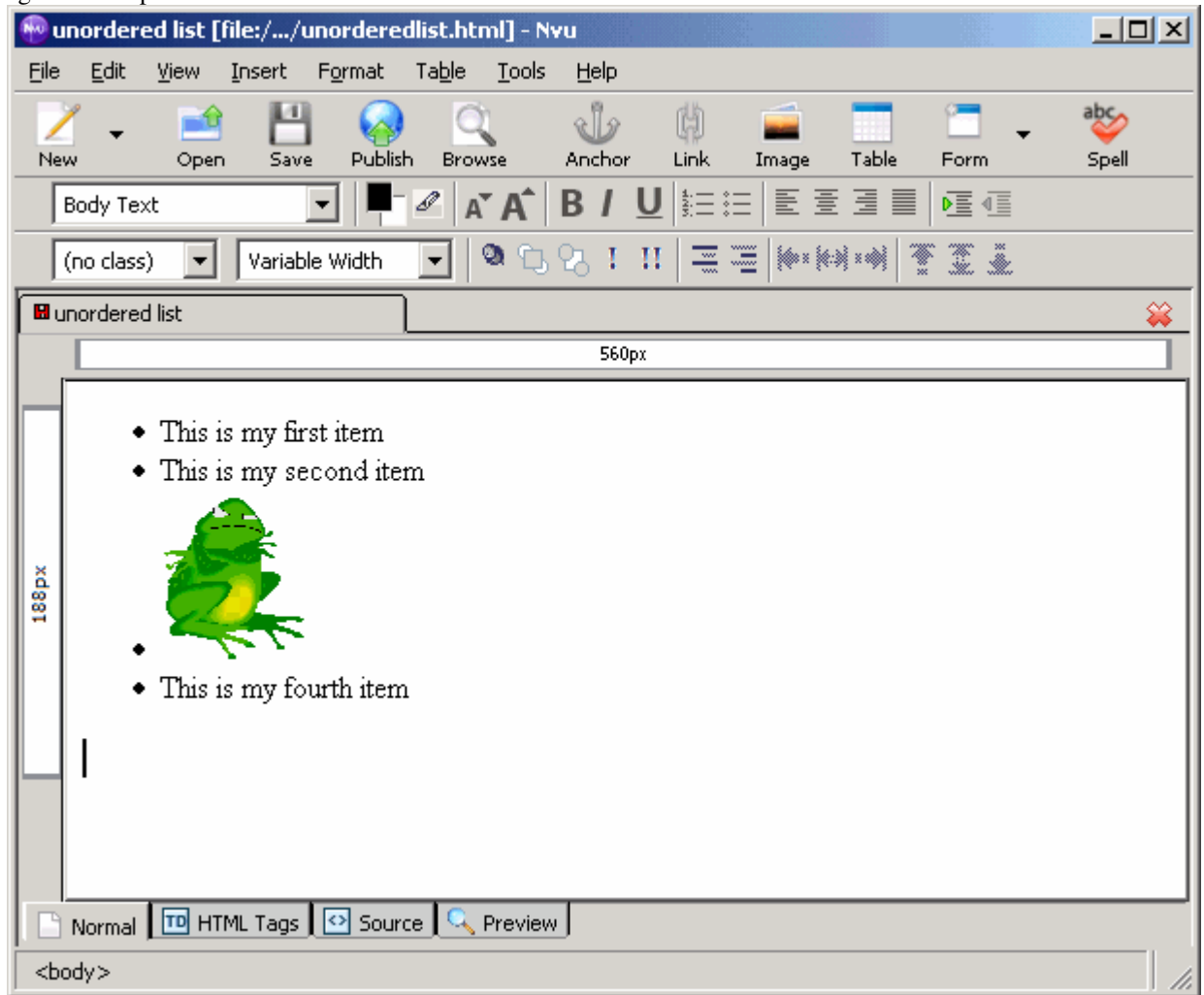
5. Repeat the previous two steps to fill out your list.

6. You can insert various elements into a list item, such as text, an image, a link, a table. To do this, move to a new list item and insert the element as you normally would.



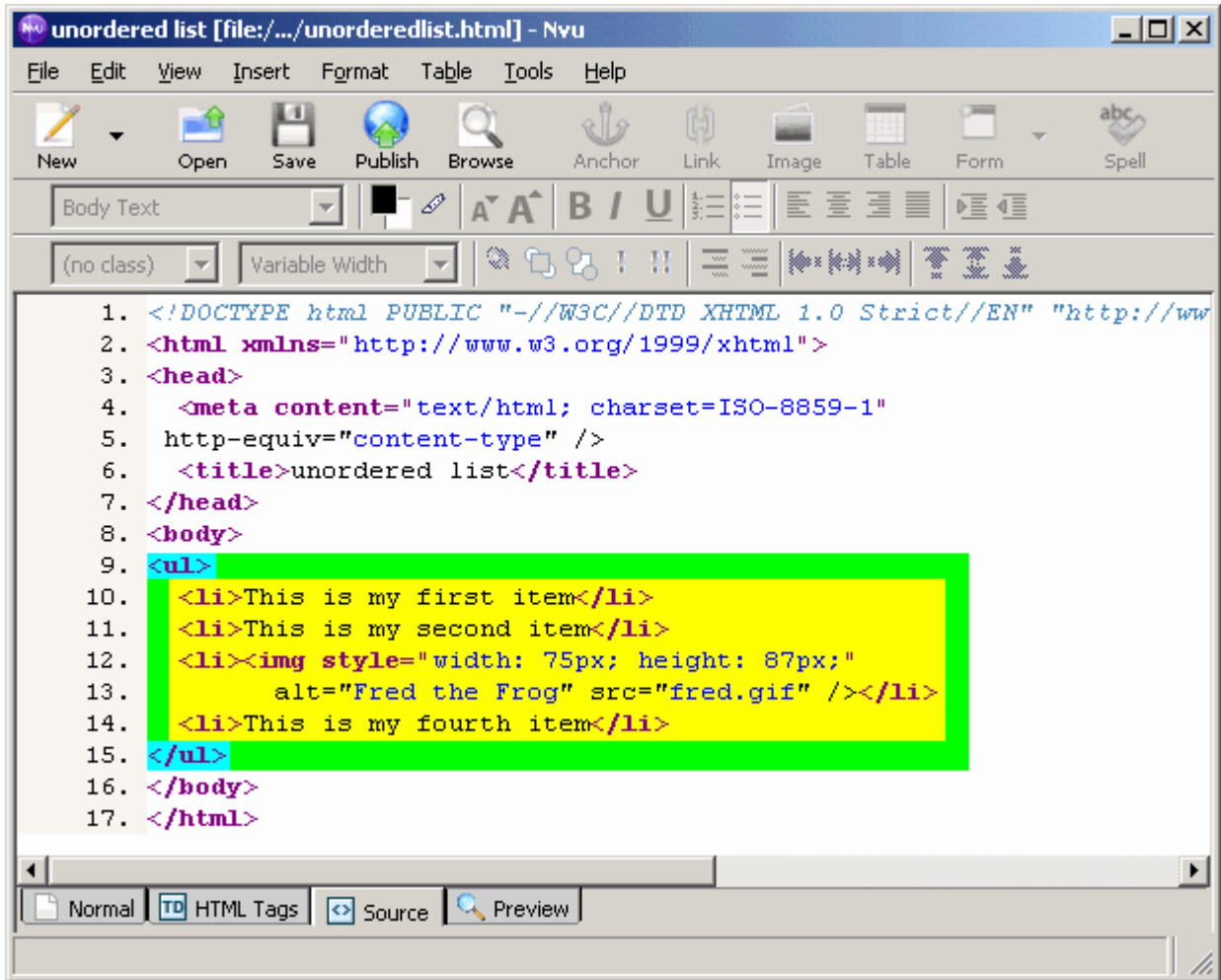
The screenshot shows the Nvu web editor interface. The title bar reads "unordered list [file:/.../unorderedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. The formatting toolbar includes options for text color, background color, font size, bold, italic, underline, bulleted list, numbered list, indent, and outdent. The main editing area shows an unordered list with three items: "This is my first item", "This is my second item", and an image of a green frog. The image is positioned to the right of the bullet point. The status bar at the bottom shows the current HTML tags: <body> <ul> <li>.

7. Once you are finished filling in your list, hit **Enter** to move to a new item in the list and hit **Enter** again to complete the list.



The screenshot shows the Nvu web editor interface. The title bar reads "unordered list [file:///.../unorderedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. The formatting toolbar shows "Body Text" as the selected style, along with options for bold, italic, underline, bulleted list, numbered list, indent, and outdent. The main editing area displays an unordered list with four items: "This is my first item", "This is my second item", a frog image, and "This is my fourth item". The editor's status bar at the bottom shows the current HTML tag as "<body>".

By looking at the source code for this unordered list, we can see how the list is created.



The screenshot shows the Nvu web editor interface. The title bar reads "unordered list [file:/.../unorderedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. Below the toolbar are two rows of formatting options, including text color, background color, font size, bold, italic, underline, bulleted list, numbered list, indent, and alignment. The main editing area displays HTML source code for an unordered list. The code is as follows:

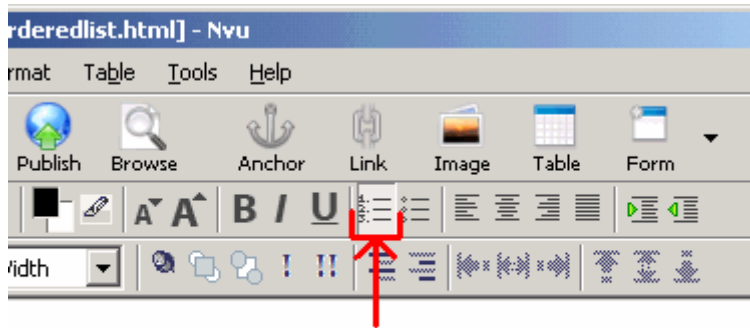
```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://ww
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title>unordered list</title>
7. </head>
8. <body>
9. <ul>
10.  <li>This is my first item</li>
11.  <li>This is my second item</li>
12.  <li></li>
14.  <li>This is my fourth item</li>
15. </ul>
16. </body>
17. </html>
```

The code is displayed in a list format with line numbers 1 through 17. The opening and closing `<ul>` and `</ul>` tags are highlighted in blue. The list items are highlighted in yellow. The `<img>` tag in the third item is also highlighted in yellow. At the bottom of the editor, there are tabs for "Normal", "HTML Tags", "Source", and "Preview".

We can see that an unordered list has two parts. First are the opening and closing `<ul></ul>` tags. The "ul" stands for **Unordered List**. This tells the browser what type of list we are creating. Next we see the `<li></li>` tags. "li" stands for **List Item**. Each set of List Item tags represent a new item in the list. And as we can see from the source code the element you want to appear for each point is included between the `<li></li>` tags. Together these tags create our list.

### ***Ordered List***

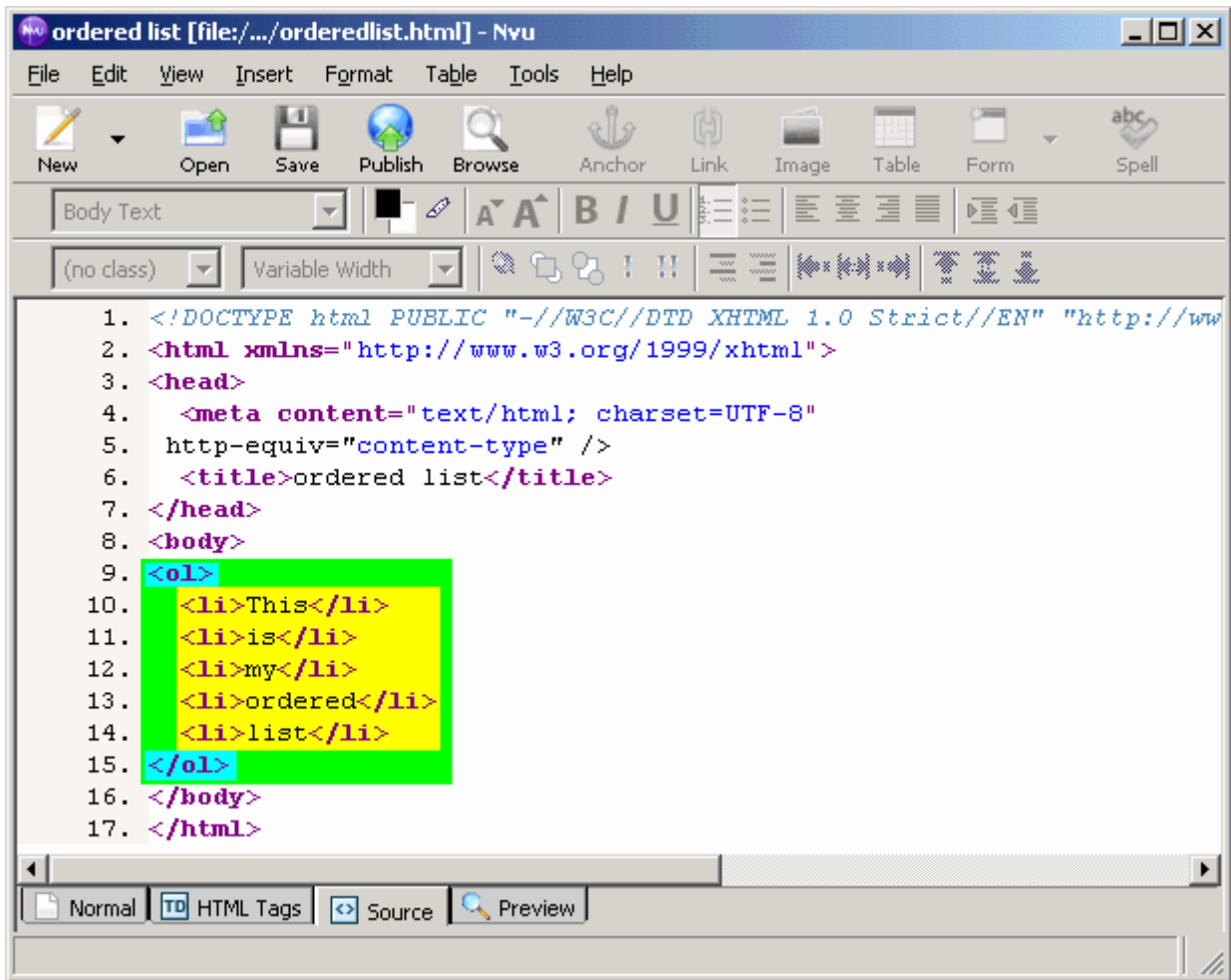
Creating an ordered list is much like creating an unordered list, but instead you click the ordered list button.



## Ordered List Button

By looking at an example and its source code, we will see the small difference between an ordered list and unordered list.

A screenshot of the Nvu web editor interface. The main window displays a preview of an ordered list with five items: "1. This", "2. is", "3. my", "4. ordered", and "5. list". The editor's menu bar includes 'File', 'Edit', 'View', 'Insert', 'Format', 'Table', 'Tools', and 'Help'. The toolbar contains icons for 'New', 'Open', 'Save', 'Publish', 'Browse', 'Anchor', 'Link', 'Image', 'Table', 'Form', and 'Spell'. The status bar at the bottom shows the current HTML tags: `<body>`, `<ol>`, and `<li>`. The preview window title is "ordered list" and it shows a width of 520px.



The screenshot shows the Nvu web editor interface. The title bar reads "ordered list [file:///.../orderedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. Below the toolbar are two rows of formatting options: "Body Text" and "(no class) Variable Width". The main editing area displays HTML source code for an ordered list. The code is as follows:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=UTF-8"
5.   http-equiv="content-type" />
6.   <title>ordered list</title>
7. </head>
8. <body>
9.   <ol>
10.    <li>This</li>
11.    <li>is</li>
12.    <li>my</li>
13.    <li>ordered</li>
14.    <li>list</li>
15.  </ol>
16. </body>
17. </html>
```

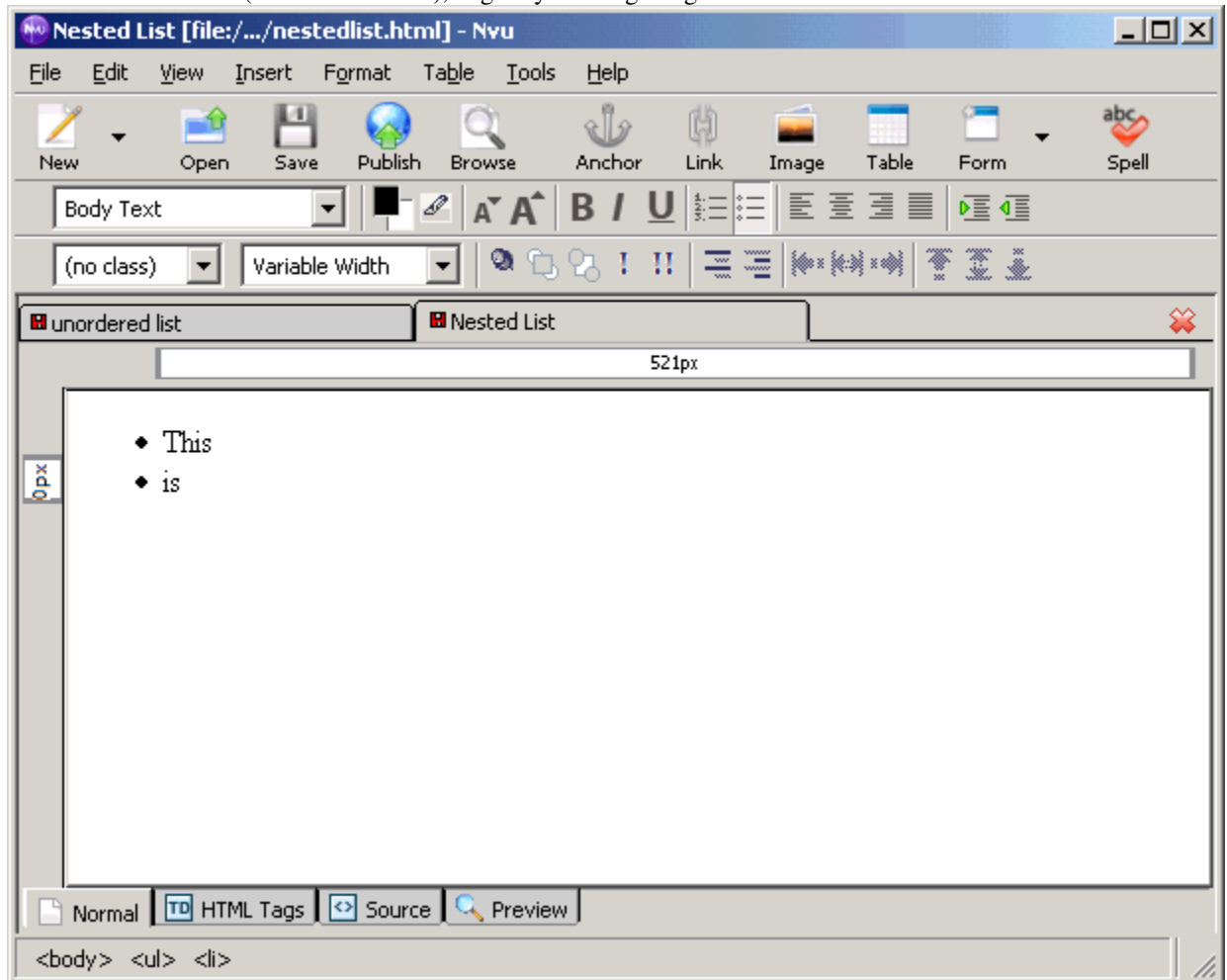
The code is displayed in a source view, with a tab at the bottom labeled "HTML Tags". The list items in the code are "This", "is", "my", "ordered", and "list".

We can see two differences between an ordered list and an unordered list. The first being the items are numbered in an ordered list, while they are bulleted in an unordered list. The second difference is seen when we look at the source code. An ordered list is created with `<ol></ol>` tags, which stand for **Ordered List**. These tell the browser to number the list items, rather than bullet them. The `<li></li>` tags work the same. Notice that we didn't have to include numbers (such as 1., 2., etc) ourselves in the source code when we created our ordered list. The browser knows to add the numbers in order just by using the `<li>` tag.

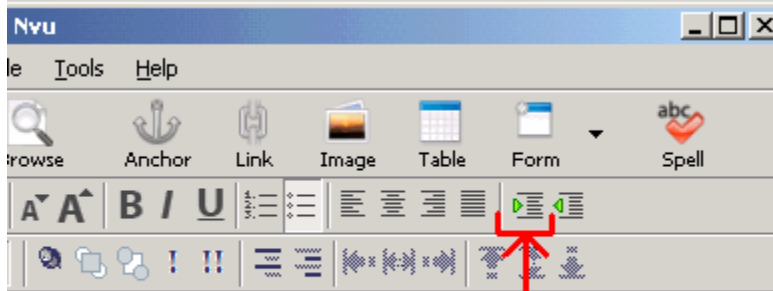
### *Nested Lists*



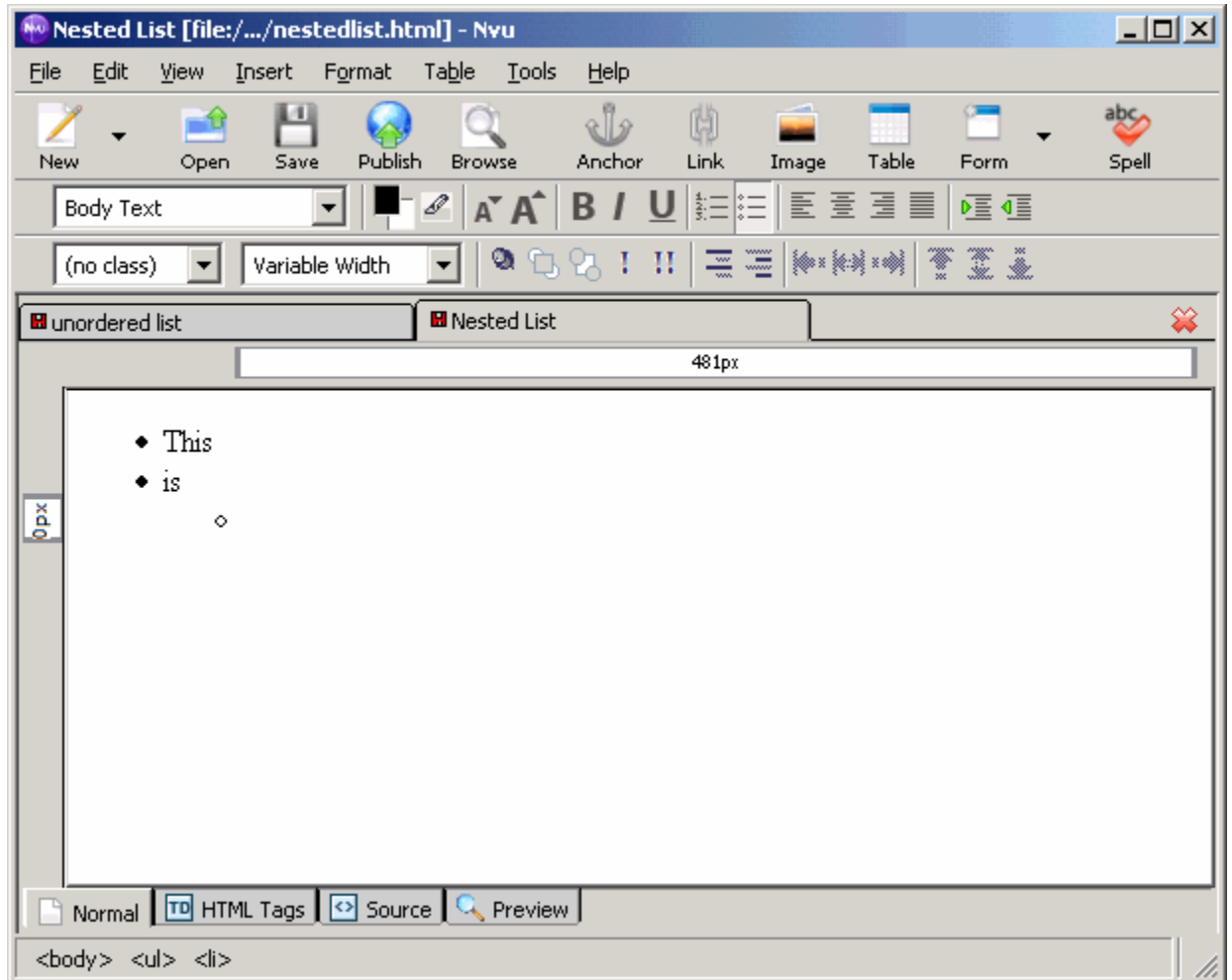
1. To create a nested list (a list within a list), begin by creating a regular list.



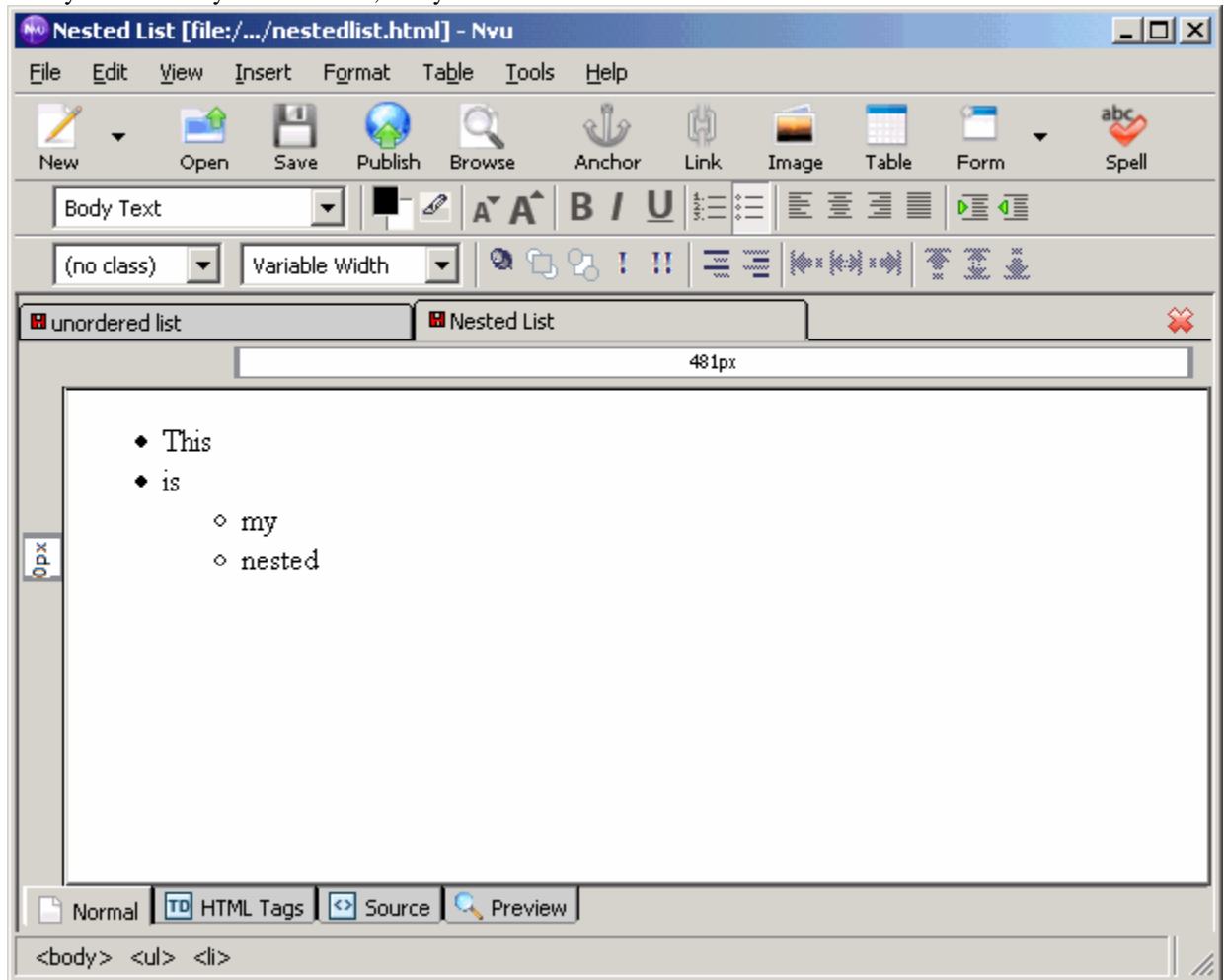
2. When you come to the item you wish to indent you can either hit the **Tab** key on the keyboard, or click the **Indent Button** in the formatting bar.



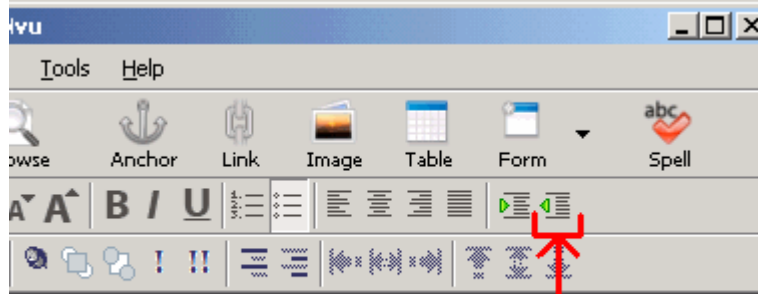
Indent Button



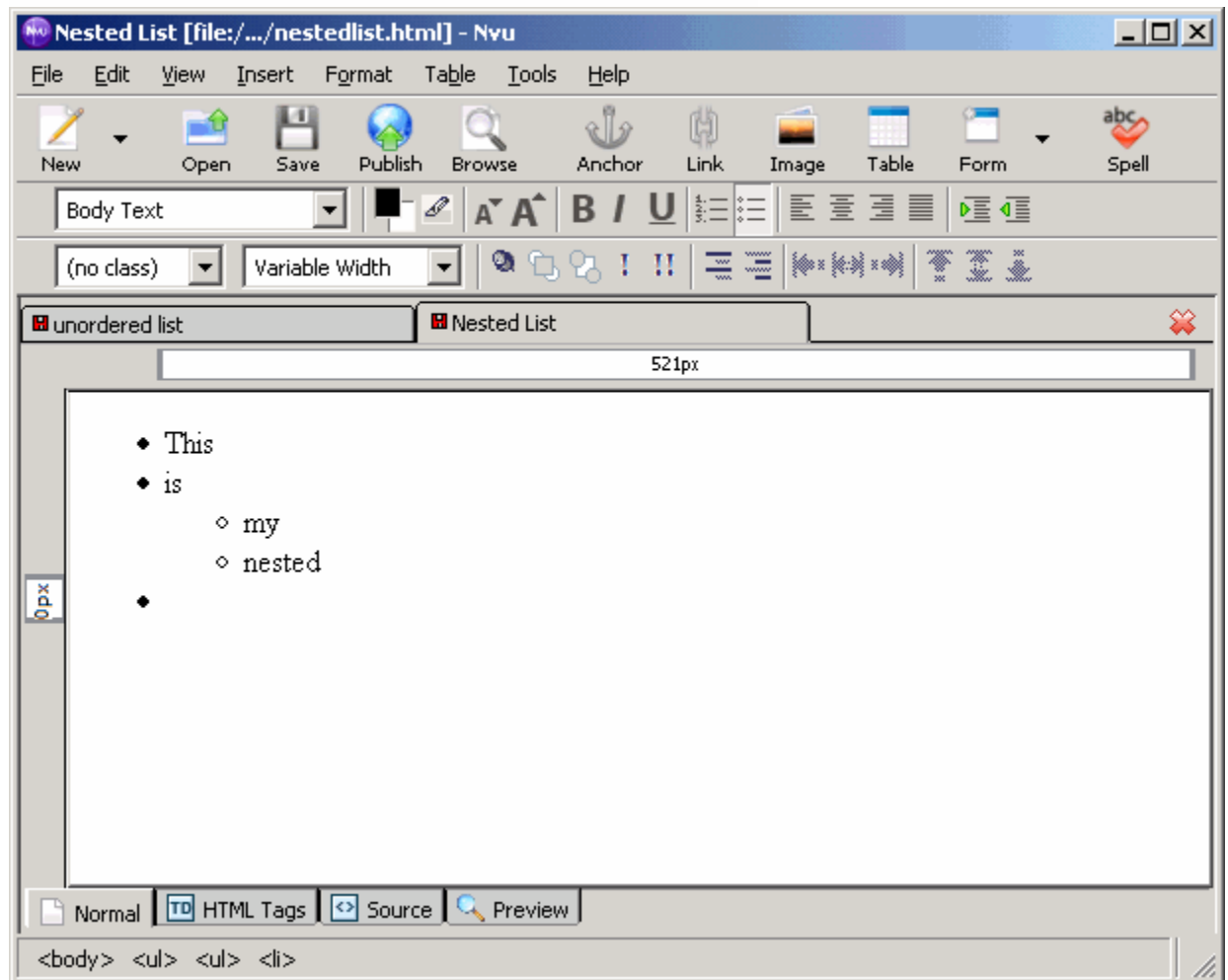
3. Now you can enter your nested list, like you would a normal list.



4. When wish to return to your original list (ie. break out of the nested list) you can either hit **Enter** or click the **Outdent Button** in the formatting bar.

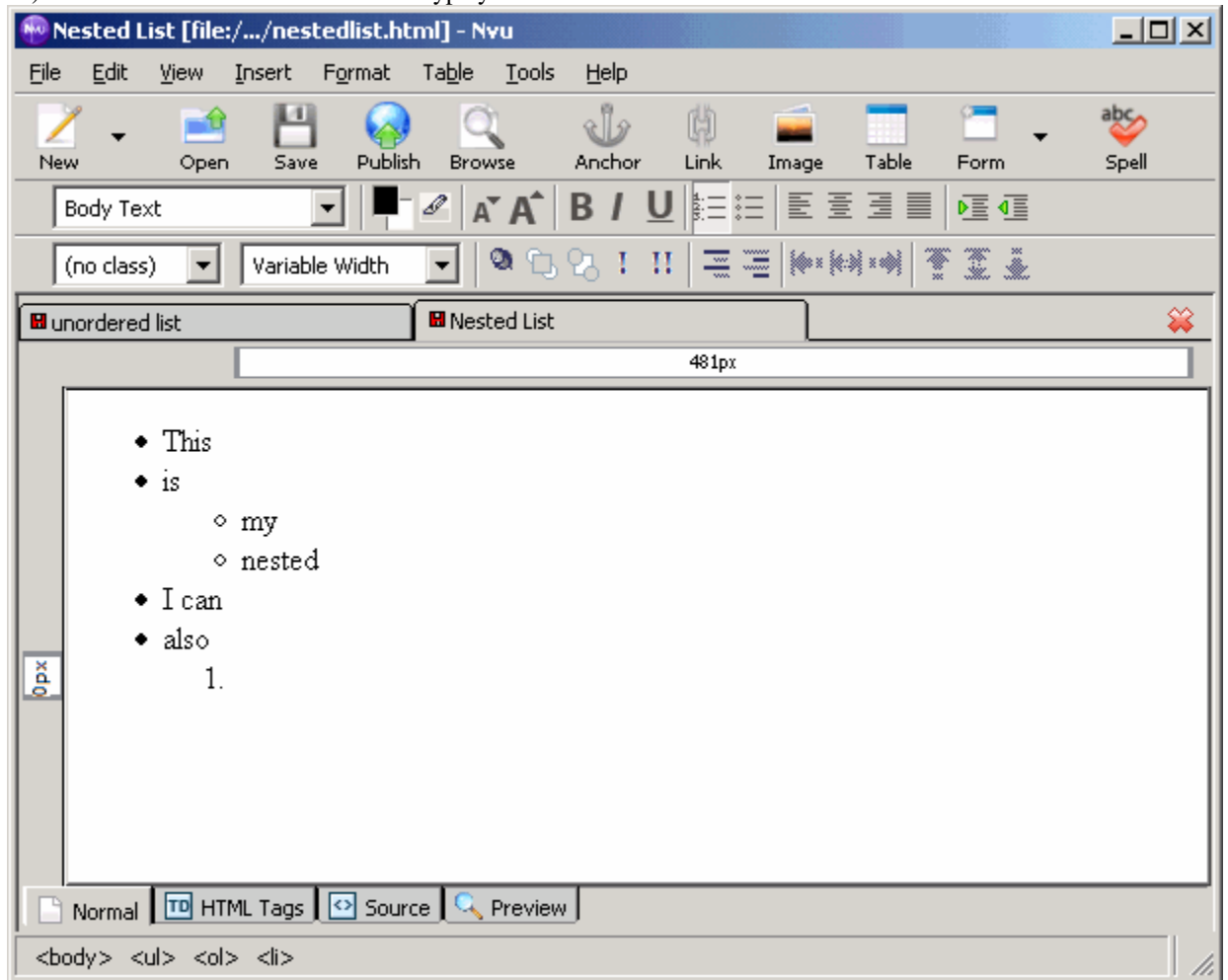


Outdent Button



5. You can nest either type of list within each other. To nest an ordered list in an unordered list, or vice versa, after indenting, click the Ordered List button (or Unordered List button if the case may

be) and the nested list will turn into the type you choose.

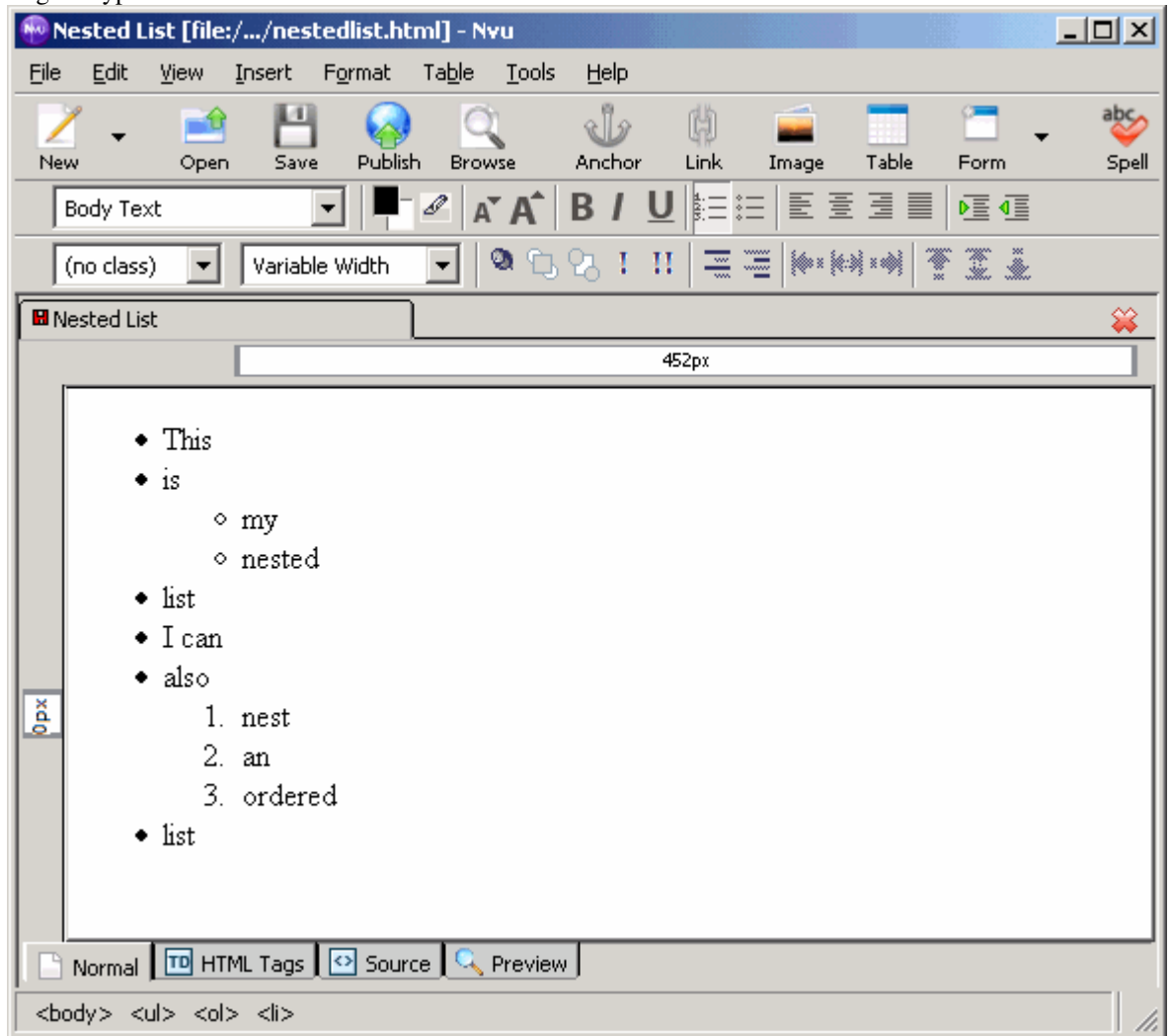


The screenshot shows the Nvu web editor interface. The title bar reads "Nested List [file:///.../nestedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. Below the toolbar are two rows of formatting options: the first row includes a text style dropdown (set to "Body Text"), a background color selector, and text formatting buttons (A^A, B, I, U); the second row includes a class dropdown (set to "(no class)"), a width dropdown (set to "Variable Width"), and various alignment and list creation icons. The main workspace is divided into two tabs: "unordered list" and "Nested List". The "Nested List" tab is active, showing a preview of a list structure with a width of 481px. The list content is as follows:

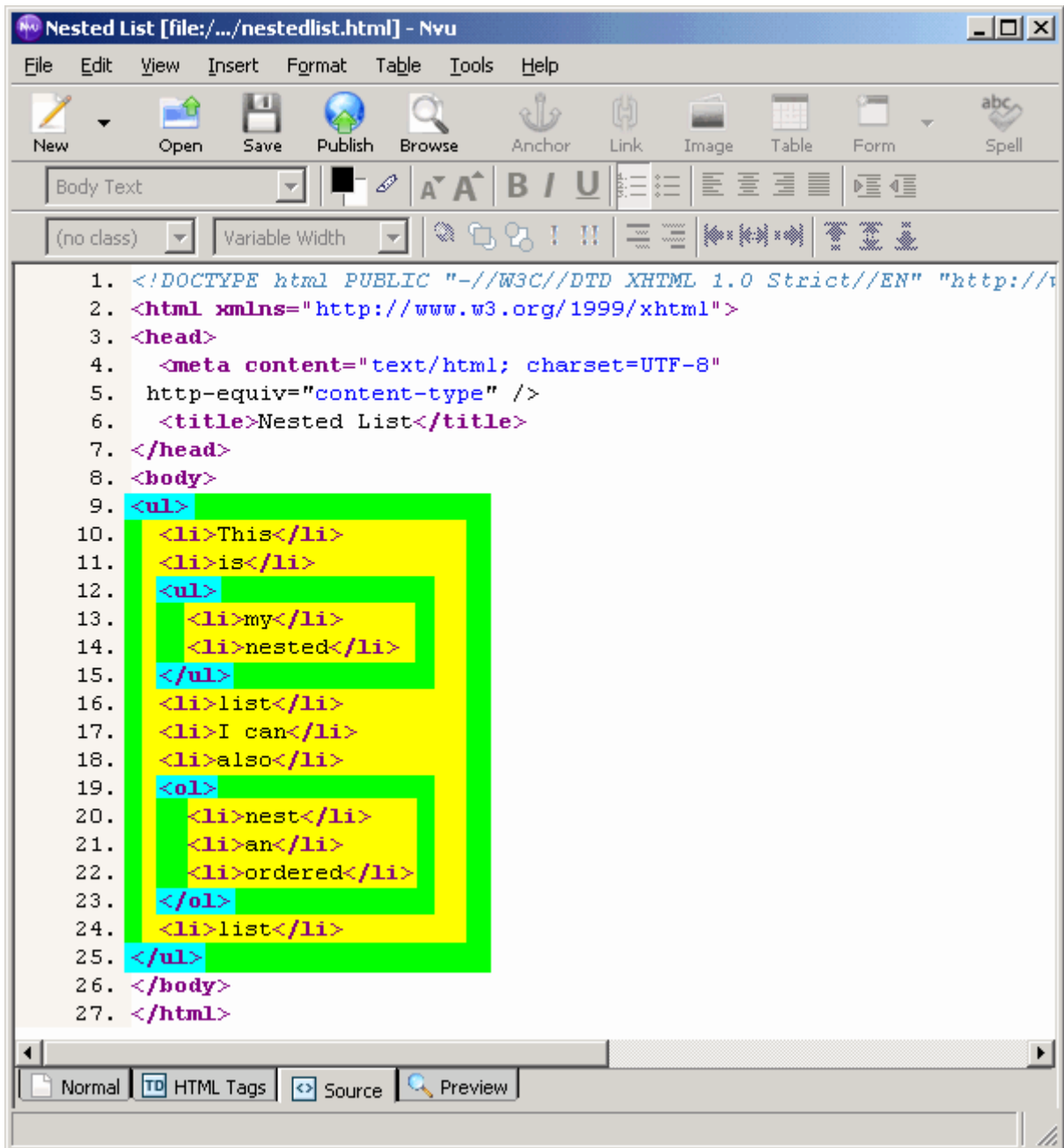
- ◆ This
  - ◆ is
    - ◇ my
    - ◇ nested
  - ◆ I can
  - ◆ also
    - 1.

At the bottom of the editor, there are tabs for "Normal", "HTML Tags", "Source", and "Preview". The "HTML Tags" tab is active, showing the code: `<body> <ul> <ol> <li>`.

6. When you are finished your nested list and return to your original, the list will switch back to the original type.



By looking at the source code we can see how these lists are nested within one another



The screenshot shows the Nvu web editor interface. The title bar reads "Nested List [file:///.../nestedlist.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. Below the toolbar are two rows of formatting options: "Body Text" and "(no class) Variable Width". The main text area displays the following HTML code:

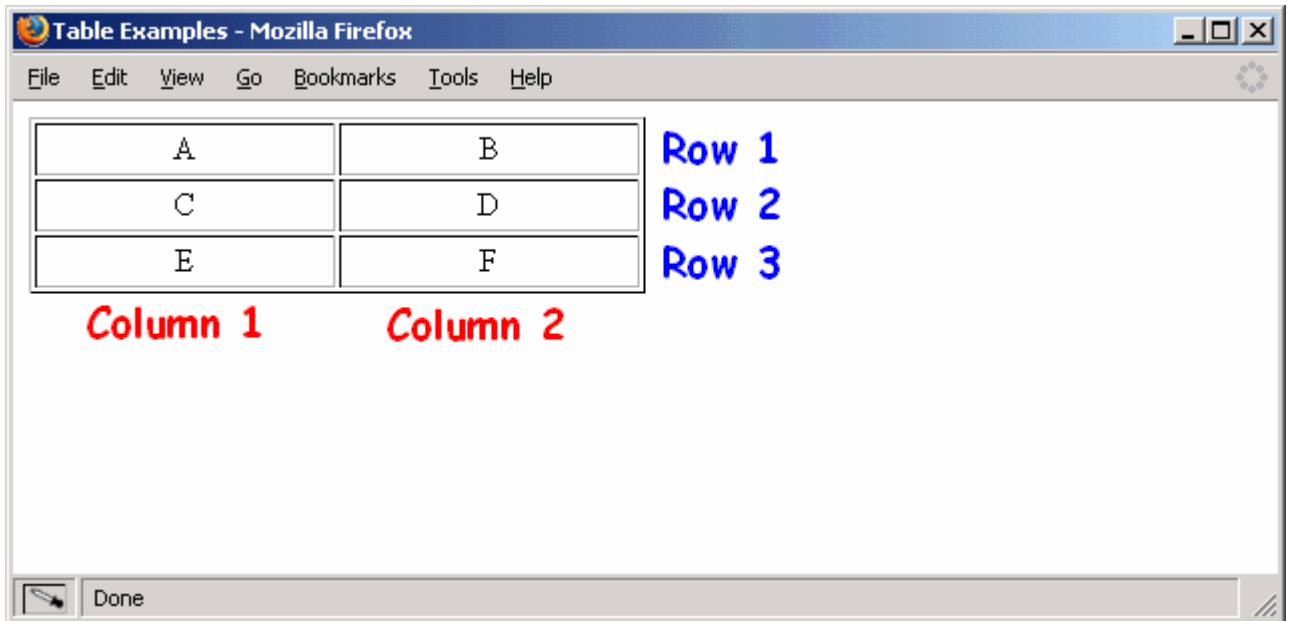
```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=UTF-8"
5.   http-equiv="content-type" />
6.   <title>Nested List</title>
7. </head>
8. <body>
9.   <ul>
10.    <li>This</li>
11.    <li>is</li>
12.    <ul>
13.     <li>my</li>
14.     <li>nested</li>
15.    </ul>
16.    <li>list</li>
17.    <li>I can</li>
18.    <li>also</li>
19.    <ol>
20.     <li>nest</li>
21.     <li>an</li>
22.     <li>ordered</li>
23.    </ol>
24.    <li>list</li>
25.  </ul>
26. </body>
27. </html>
```

At the bottom of the editor, there are four tabs: "Normal", "HTML Tags", "Source", and "Preview".

As we can see, the nested lists are lists nested inside the opening and closing tags of another list.

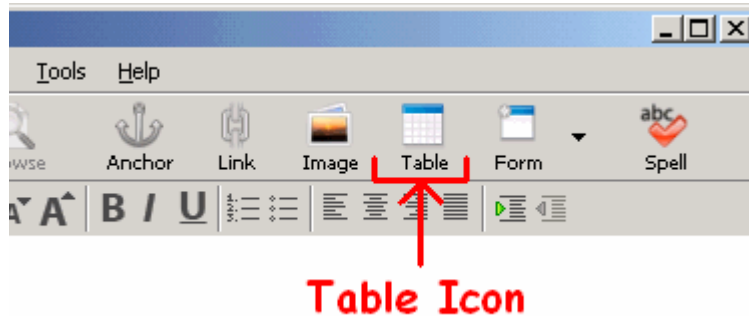
## Tables

One of the most useful ways to organize information on a webpage is by using tables. A table is made up of **cells**, where each cell contains some XHTML element(s). The number of cells in a table is determined by the number of rows and columns in the table. Columns are the vertical cells of the table, while rows are the horizontal cells of the table.

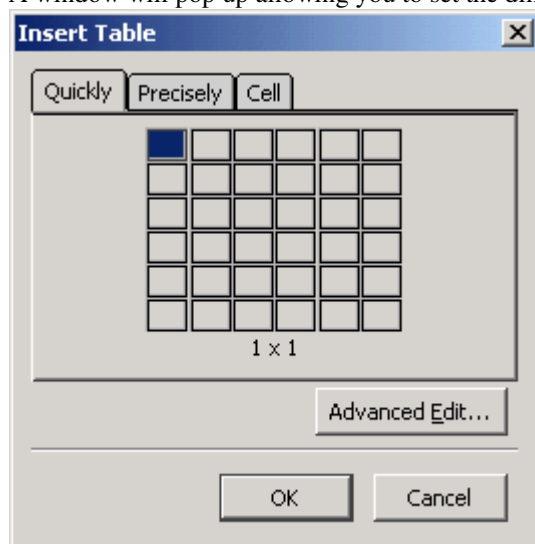


### Creating Your Table

1. To create a table in Nvu, place your cursor at the location you wish to insert your table and click the **Table** icon in the button bar.

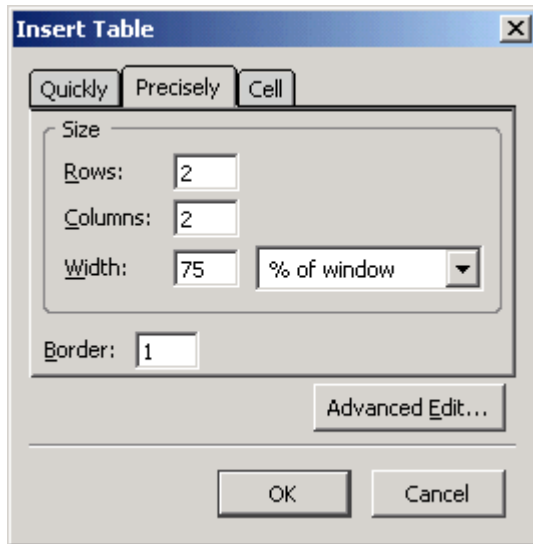


2. A window will pop up allowing you to set the dimensions of the table

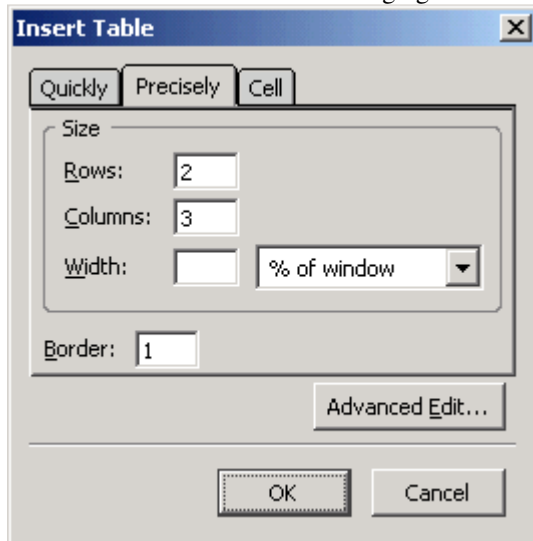


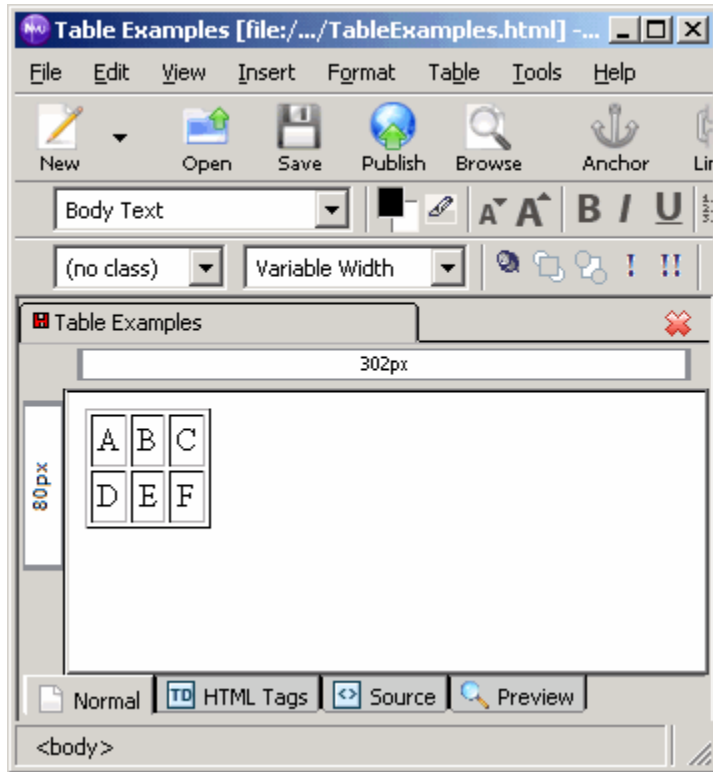


3. A more useful way to set the table is to click on the **Precisely** tab. This allows you to specify the number of rows and columns as well as the width and border for the table.

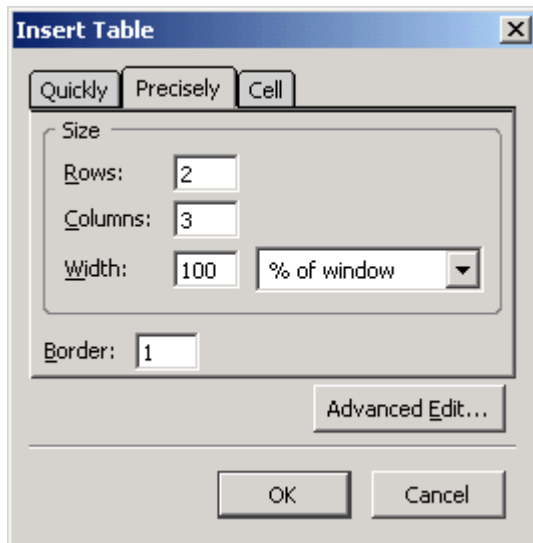


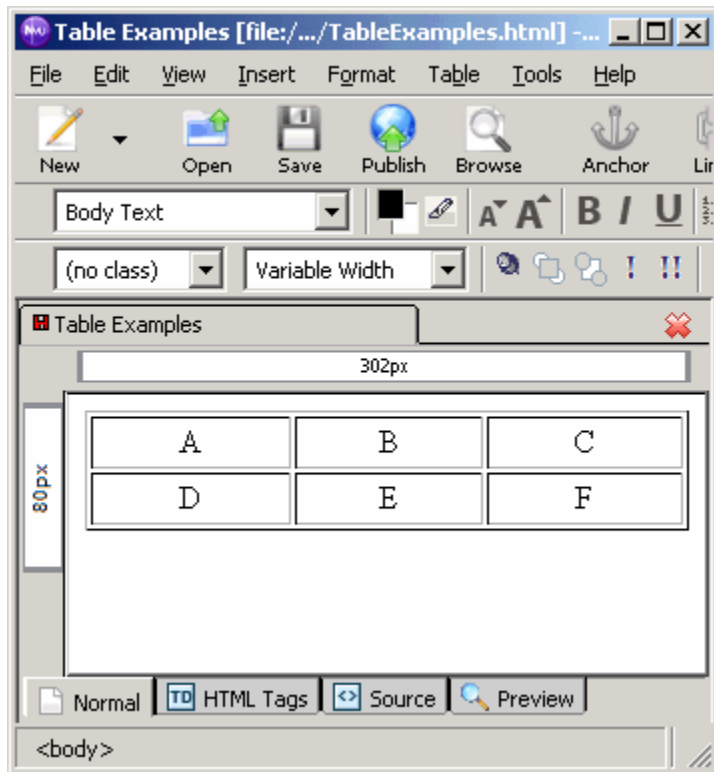
Here are some examples of tables, note how the different numbers for **rows** and **columns** effect the table. Also notice the how changing the **width** attribute changes the table



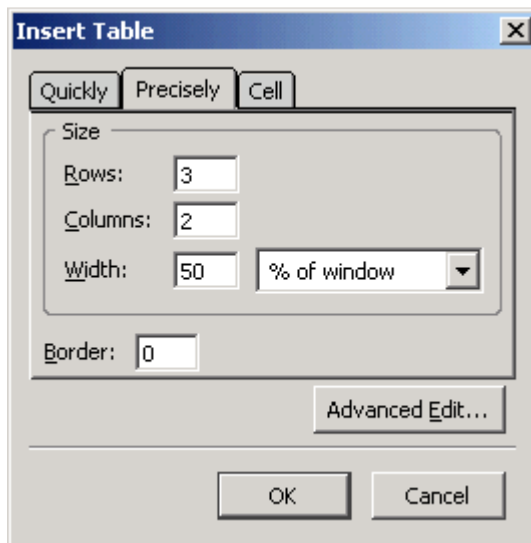


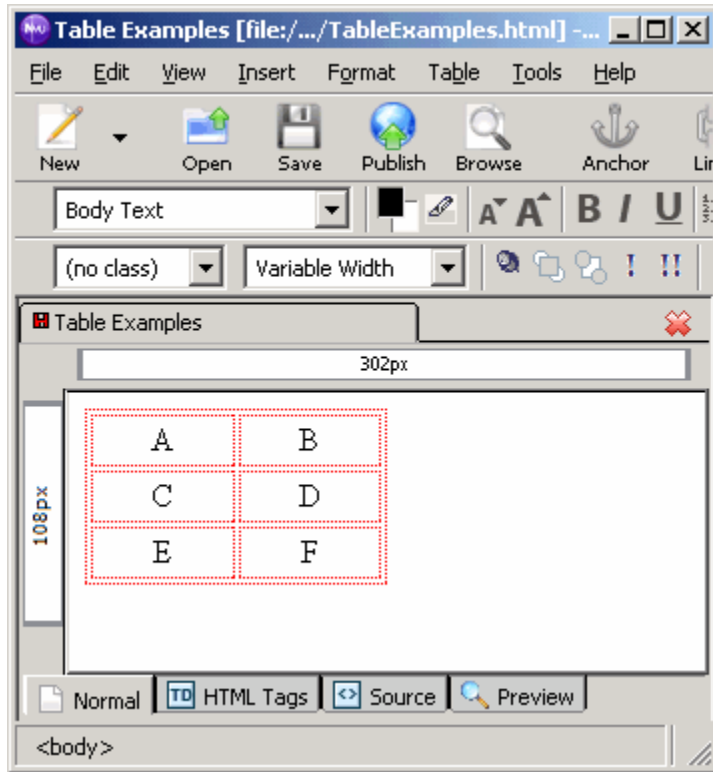
By leaving the width blank, the table adjusts itself to be just big enough to fit the elements inside.



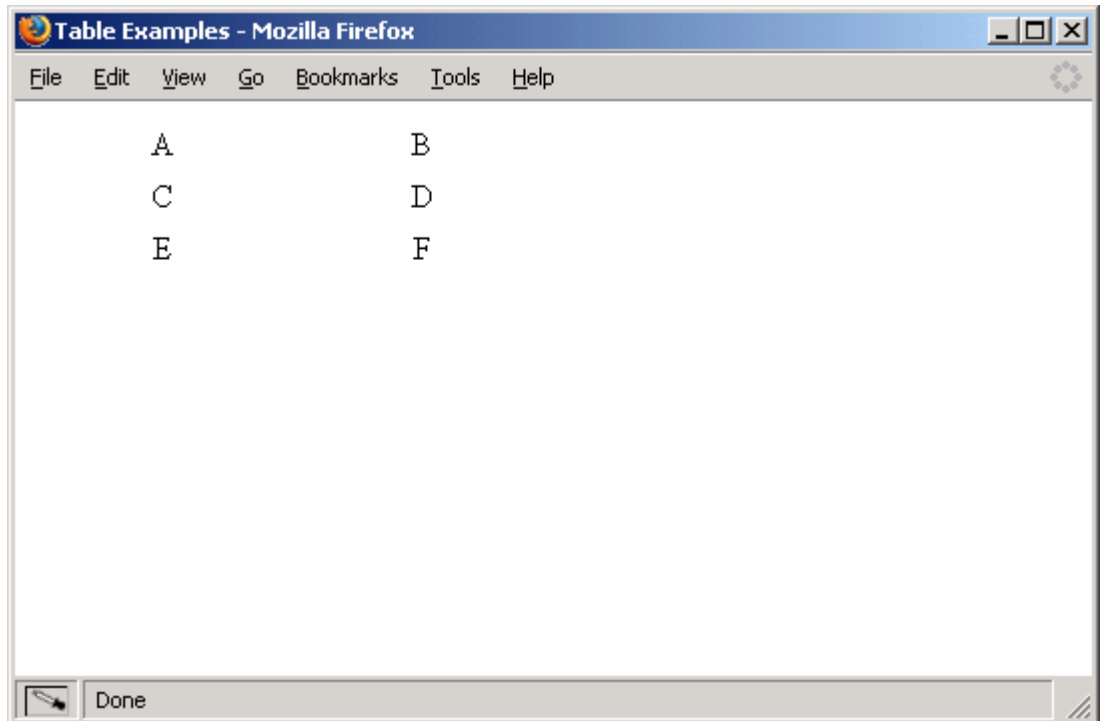


Here we have the table width set to 100% of the window. This causes the table to be stretched across the entire window.

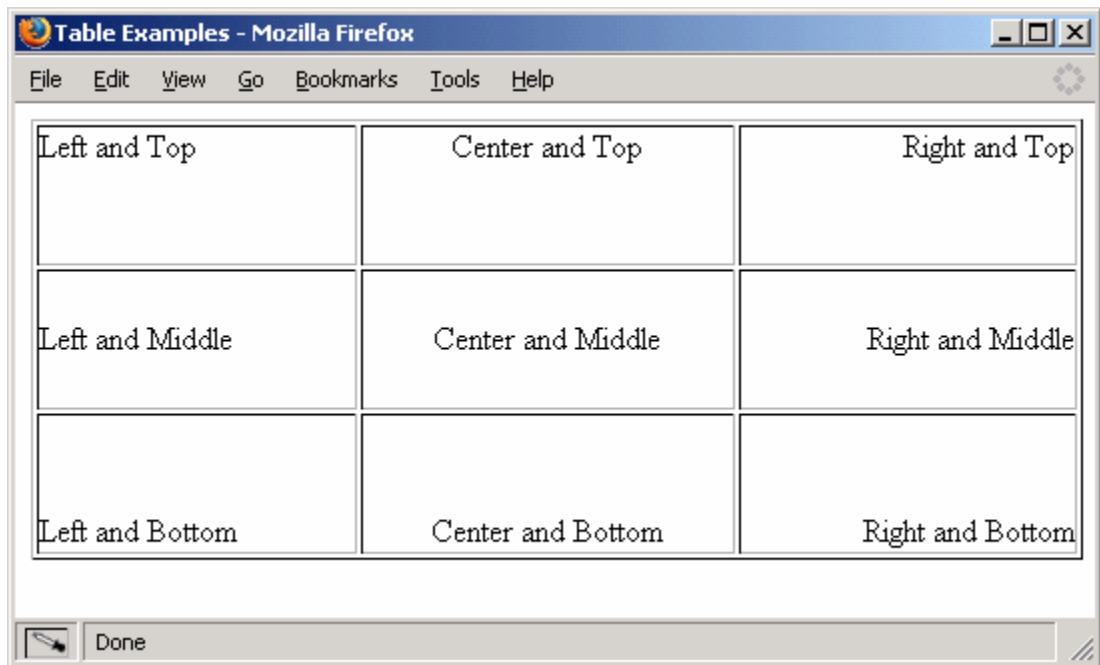
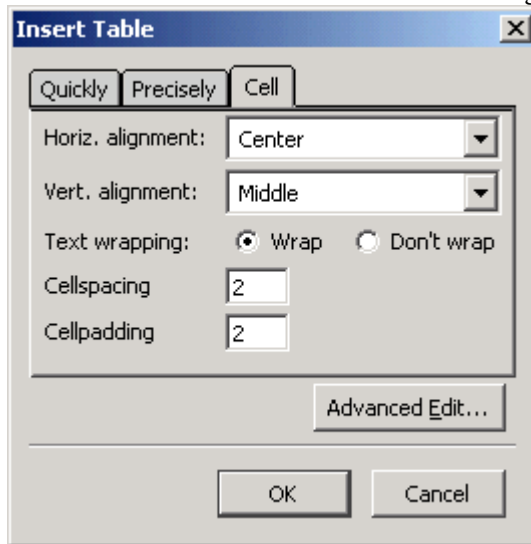




In the above example, we switch the number of rows and columns, causing our table to look completely different. We also set the width of the table to be 50% of the window, so it stretches across half our browser. Finally, we set the border to 0. Nvu puts in red dashed lines to make it easy to see each individual cell in the table. These disappear when we look at the page in a browser:

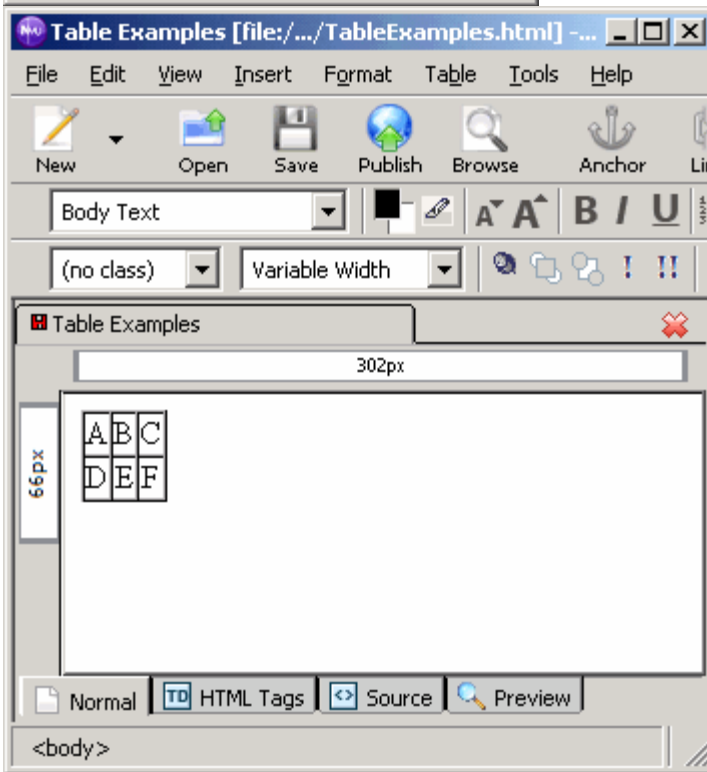
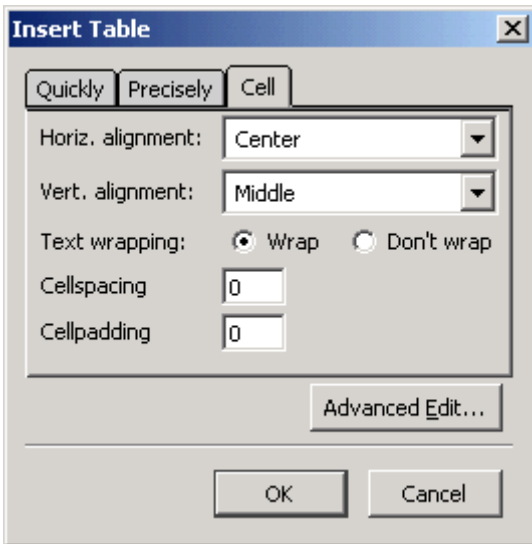


- By clicking on the cell tab you can set the horizontal alignment (left, center, right) for the contents of each cell in the table as well as the vertical alignment (top, middle, bottom).

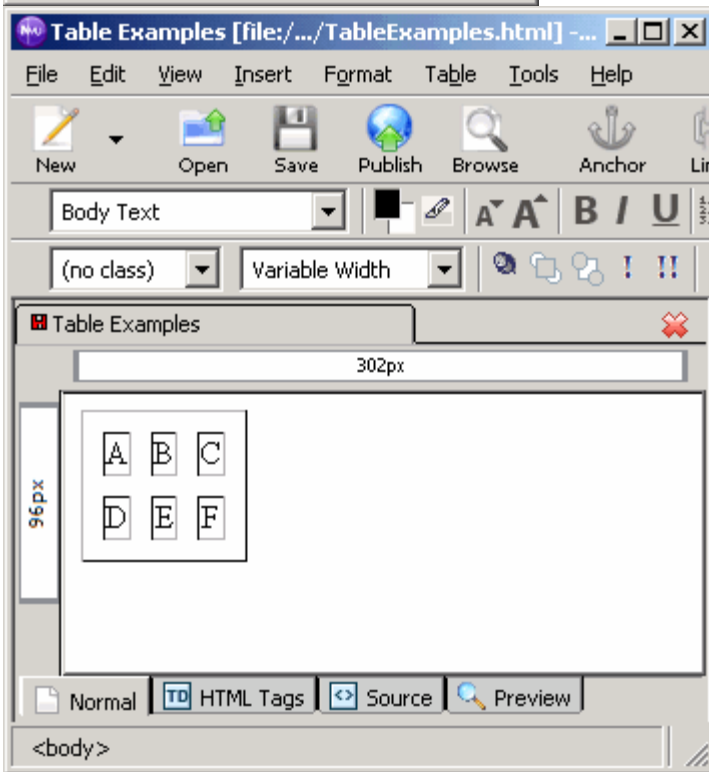
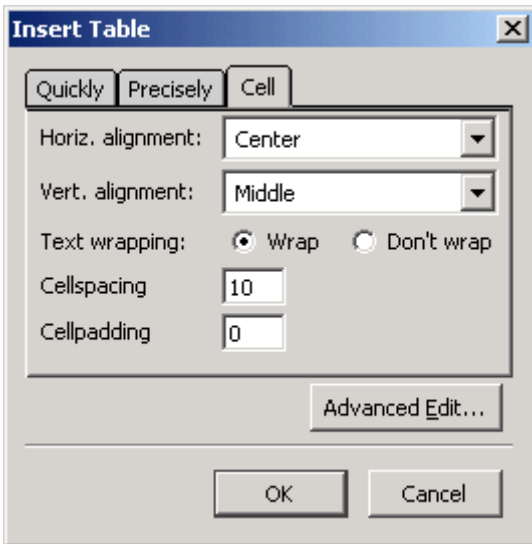


- You can also set the cellspacing, which is the amount of area between cells, and the cellpadding, which is the area between the edge of the cell and the content within the cell.

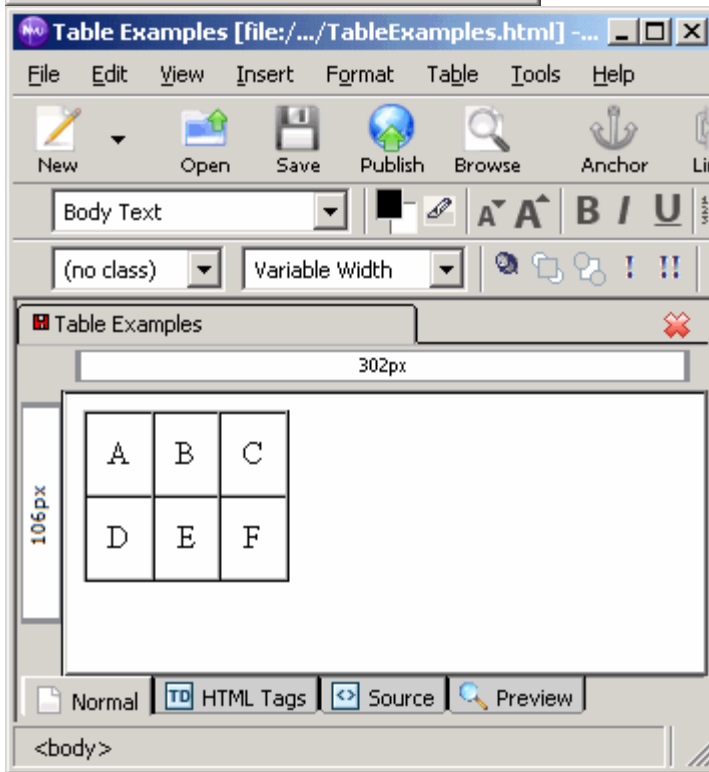
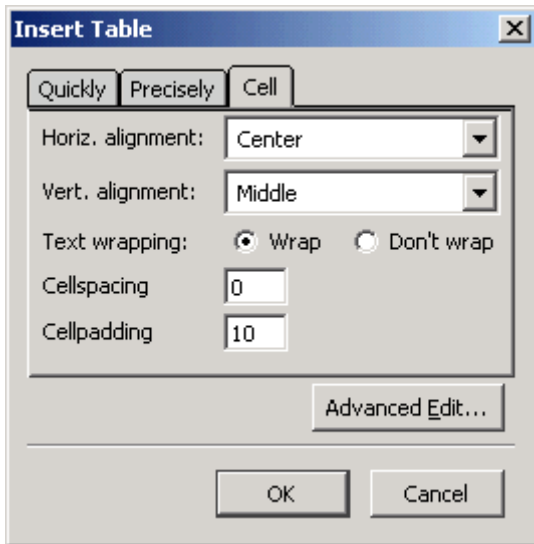
Here are some examples of cellspacing and cellpadding, note the differences in the properties window and how these effect the table.



With no cellpadding or cellspacing, our table appears cluttered.



By increasing the cellspacing, we can see the cells are no longer touching, and have space between them.

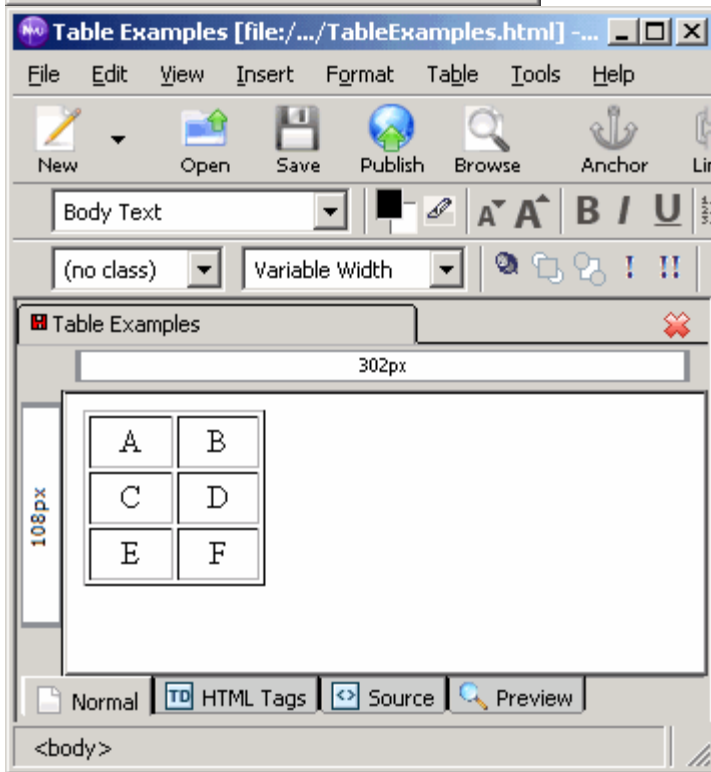
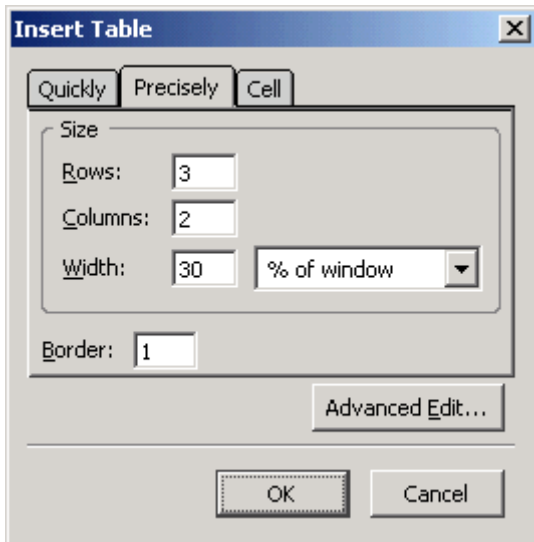


Increasing cellpadding gives the content within the cell more distance from the edge of the cell.

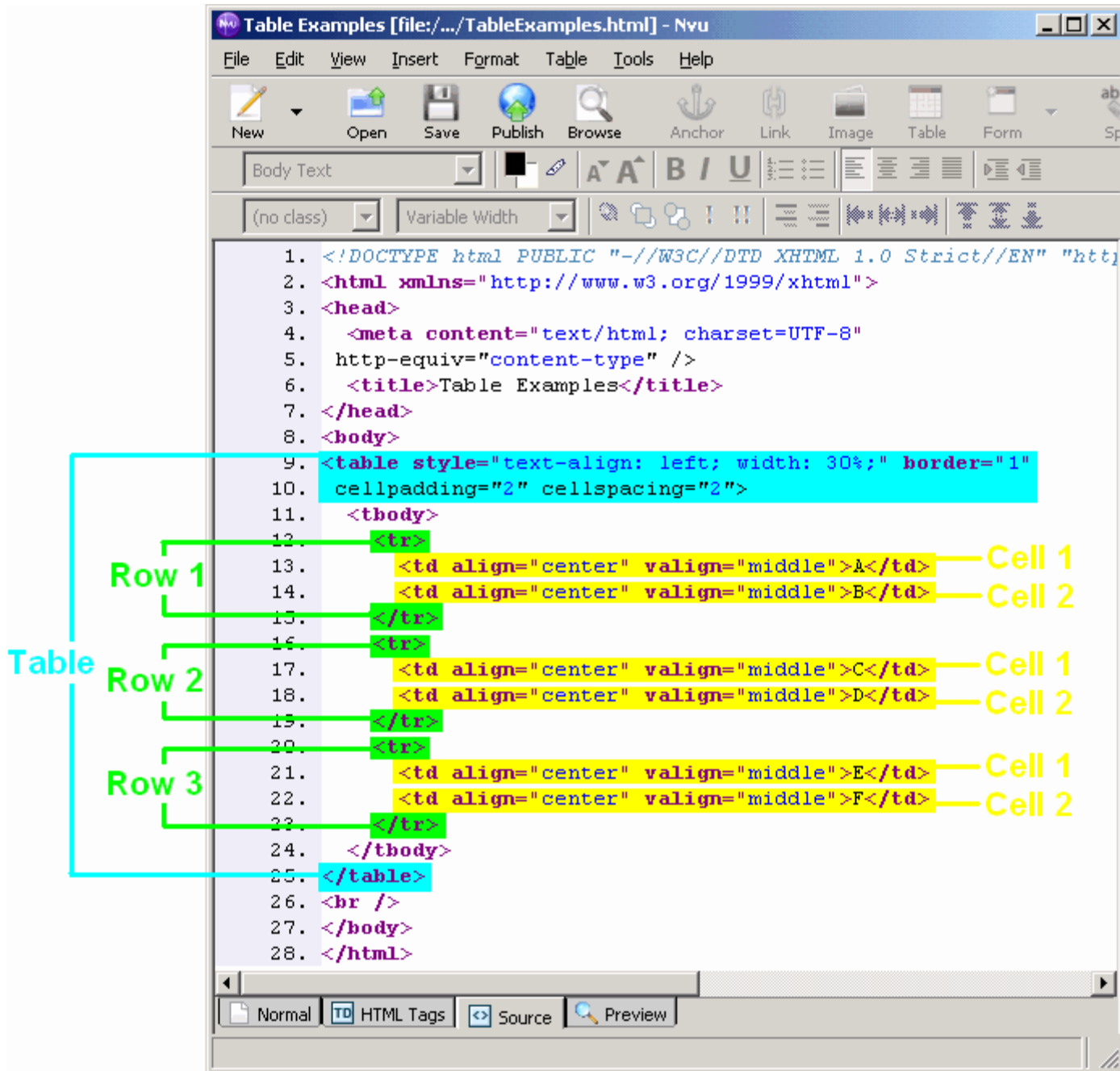
### Source Code

To further understand tables, let's look at the following example and its source code.





And here is the source code Nvu generates:



Our table begins with the opening `<table>` tag. Everything within our table must be between this and our closing `</table>` tag. Within the opening `<table>` tag we see the **style** attribute. As we can see, this is where the table properties are set.

Next we see `<tbody></tbody>` tags. These are just a way to divide our tables into sections. There are also `<thead></thead>` and `<tfoot></tfoot>` tags which allow you to set formatting for different sections of your table.

Next we see our first set of `<tr></tr>` tags. Each set of tr tags indicate a table row. Notice that column cells (`<td></td>`) are nested within the table row tags. Each set of td tags indicates a column in the row. Whatever elements we want in the cell, get nested between the td tags.

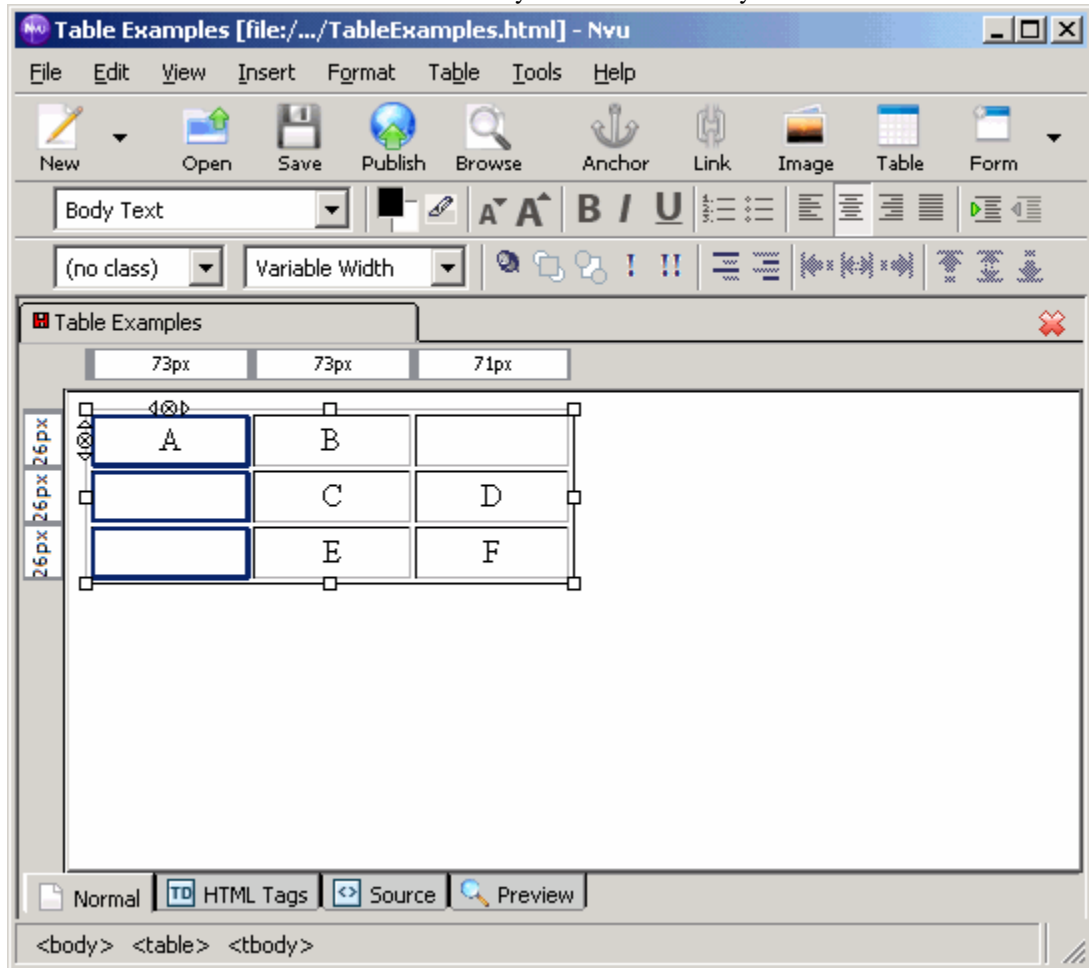
Looking at this table we see it is 3 rows by 2 columns. Therefore we have 3 sets of <tr> tags, each with 2 <td> tags.

Also notice inside each of the opening td tags we see the **style** attribute again. Here it is setting the alignment for within the individual cells.

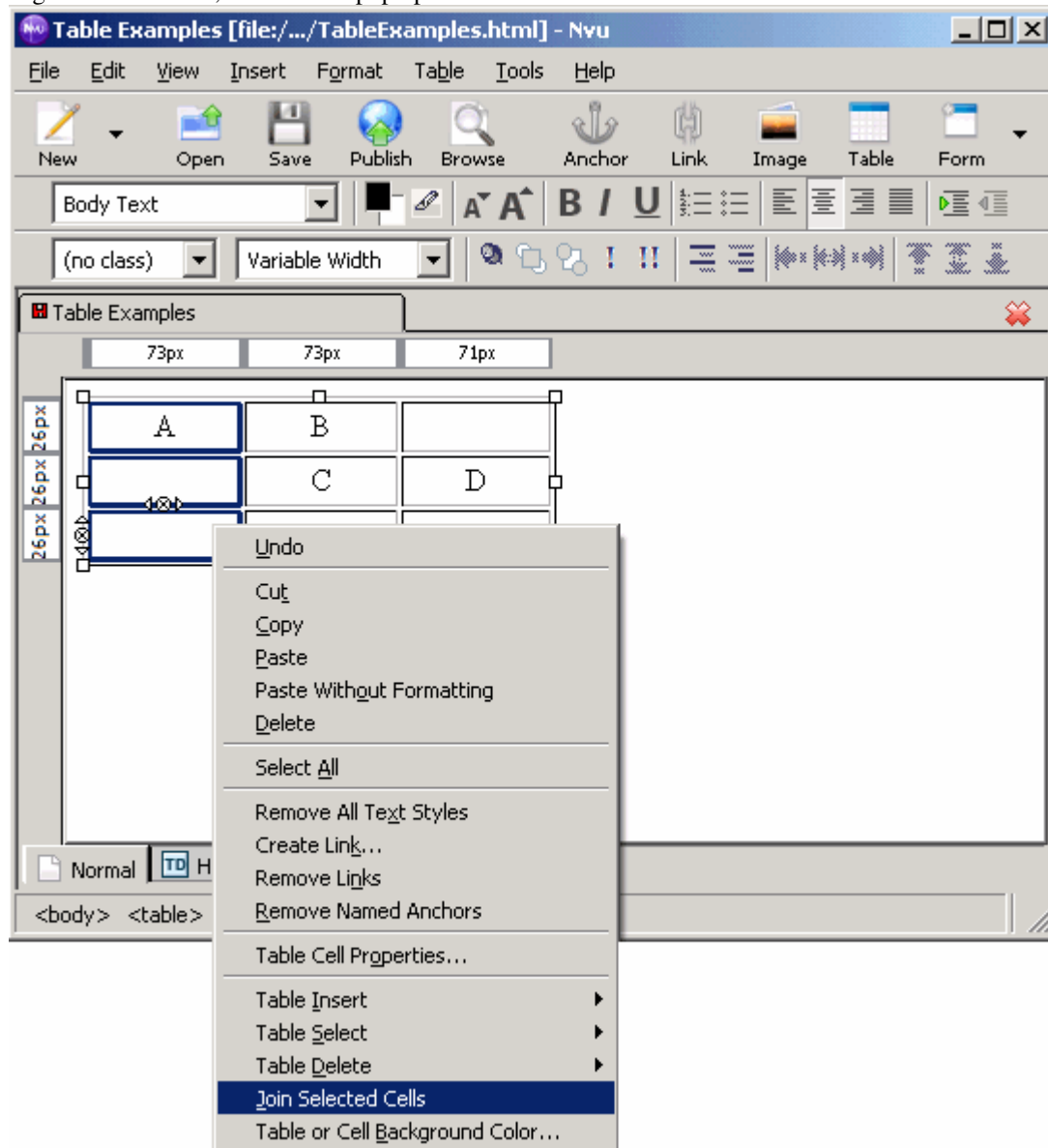
### Merging Cells

Sometimes you will want cells to span across multiple in other rows or columns.

1. To merge cells together, first select the cells you wish to merge by clicking and dragging your mouse over them. You can also hold the **ctrl** key and click each cell you wish to select.



2. Right-click on them, a menu will pop up.



3. Select **Join Selected Cells** and your cells will merge.

The screenshot shows the Nvu web editor interface. The title bar reads "Table Examples [file:///.../TableExamples.html] - Nvu". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Table", "Tools", and "Help". The toolbar contains icons for "New", "Open", "Save", "Publish", "Browse", "Anchor", "Link", "Image", "Table", and "Form". Below the toolbar are two rows of formatting options, including text alignment, bulleted lists, numbered lists, and indentation. The main workspace shows a table with the following structure:

A	B	
	C	D
	E	F

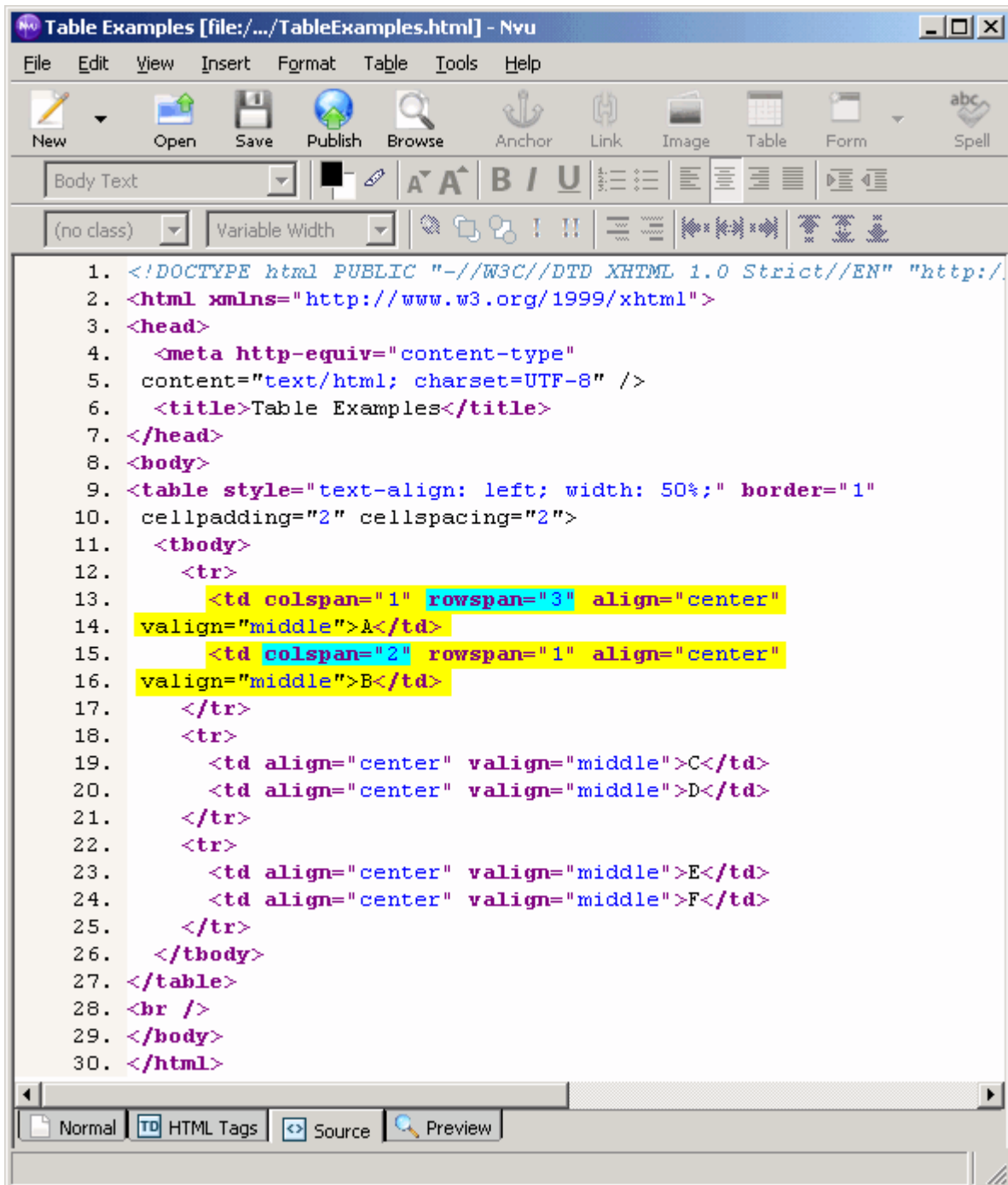
The table dimensions are indicated as 489px wide and 108px high. At the bottom, there are tabs for "Normal", "HTML Tags", "Source", and "Preview". The status bar shows the code "<body>".

Here is a slightly more detailed example and its source code:

The screenshot shows the Nvu web editor interface. The title bar reads "Table Examples [file:///.../TableExamples.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, and Form. Below the toolbar are two rows of formatting options, including text color, background color, font size, bold, italic, underline, bulleted list, numbered list, indent, and alignment. The main workspace shows a table with a width of 489px and a height of 128px. The table structure is as follows:

A	B	
	C	D
	E	F

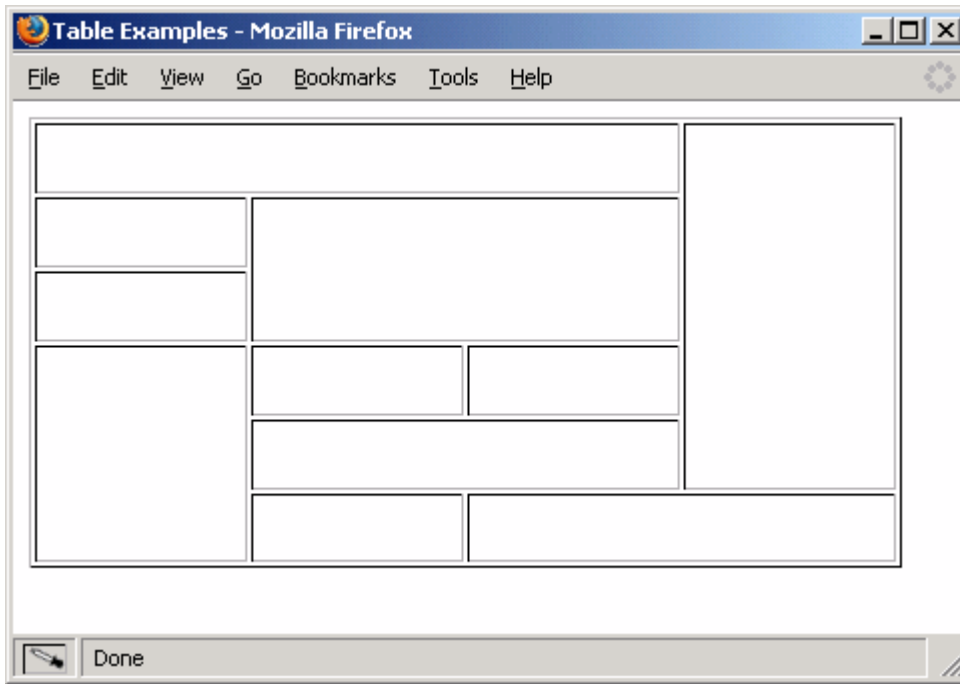
At the bottom, the source code view is active, showing the opening `<body>` tag.



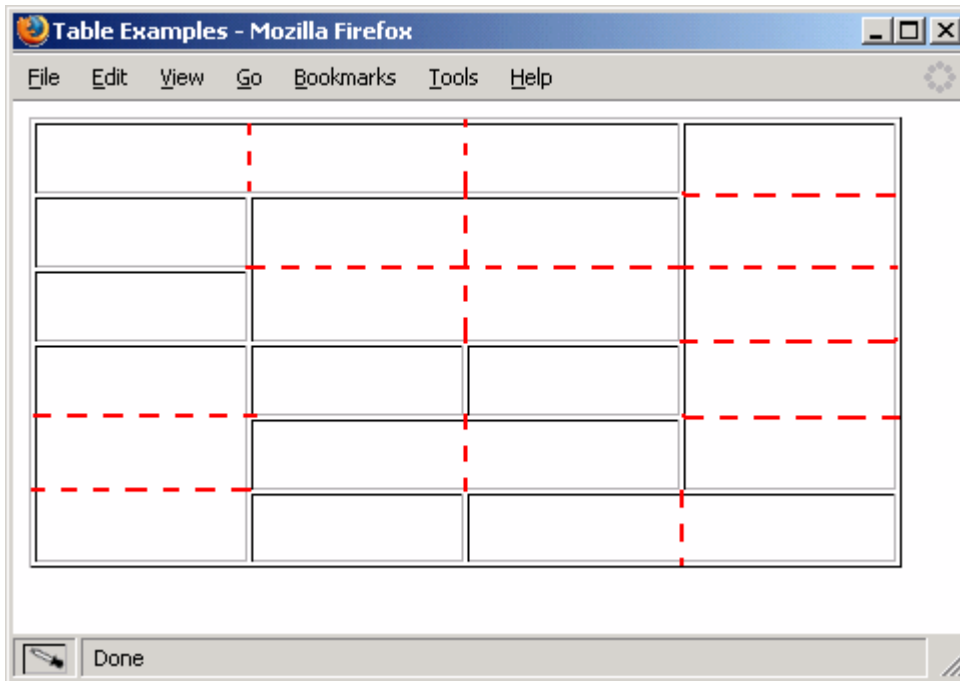
```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta http-equiv="content-type"
5.   content="text/html; charset=UTF-8" />
6.   <title>Table Examples</title>
7. </head>
8. <body>
9. <table style="text-align: left; width: 50%;" border="1"
10. cellpadding="2" cellspacing="2">
11.   <tbody>
12.     <tr>
13.       <td colspan="1" rowspan="3" align="center"
14.       valign="middle">A</td>
15.       <td colspan="2" rowspan="1" align="center"
16.       valign="middle">B</td>
17.     </tr>
18.     <tr>
19.       <td align="center" valign="middle">C</td>
20.       <td align="center" valign="middle">D</td>
21.     </tr>
22.     <tr>
23.       <td align="center" valign="middle">E</td>
24.       <td align="center" valign="middle">F</td>
25.     </tr>
26.   </tbody>
27. </table>
28. <br />
29. </body>
30. </html>
```

The above table started as a 3 X 3 table. As we can see the first cell in row 1, "A", spans vertically across three cells. Looking at the code, we see that Nvu adds the **rowspan** attribute to the cell's opening `<td>` tag. The value for this attribute is set to the number of rows the cell is to span across (in this case 3). Looking at the second cell in row 1, "B", this cell spans horizontally across two cells. Instead of **rowspan**, Nvu adds the **colspan** attribute to the cell's opening `<td>` tag. The value is set to the number of columns the cell is to span (in this case 2). Nvu also automatically generates the other "span" (ie. `colspan="1"` or `rowspan="1"`) attribute when merging cells, however this is redundant. By omitting the **colspan** and/or **rowspan** attributes the default is 1.

When trying to create a table that will contain merged cells, it is sometimes difficult to determine the dimensions of the original table. If we look at the following example, it is hard to say what the table dimensions are.



The easiest way to determine a table's dimensions is to find every line that does not go all the way across the table, and lightly join it to the edges of the table. The example above will look as follows.



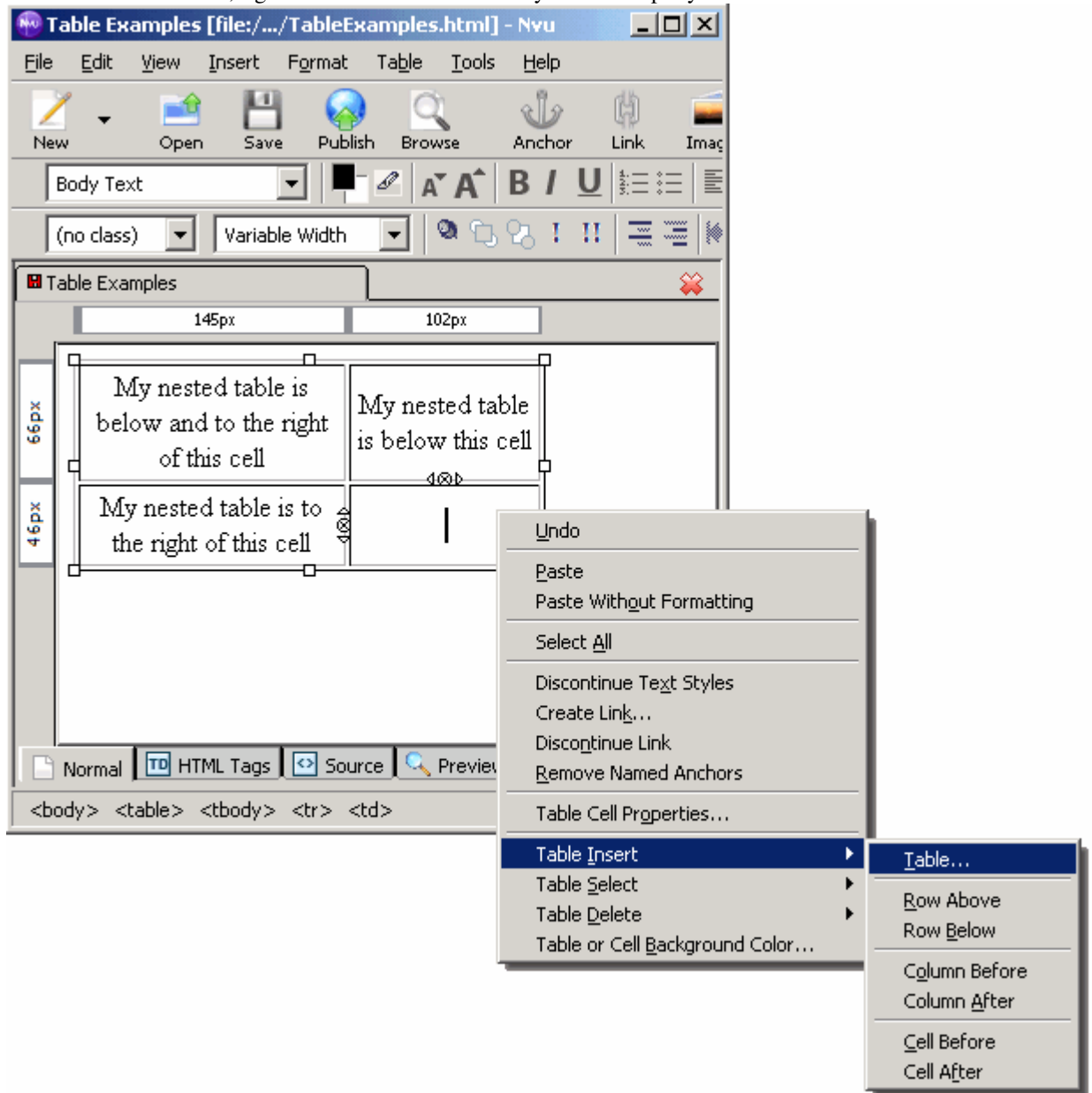
Notice that we did not create any "new" lines, we just joined existing ones to the edges of the table. If you create new lines, you are adding unnecessary cells to your table. Once all of our lines are joined, we can see that the table is a 6 row X 4 column table.



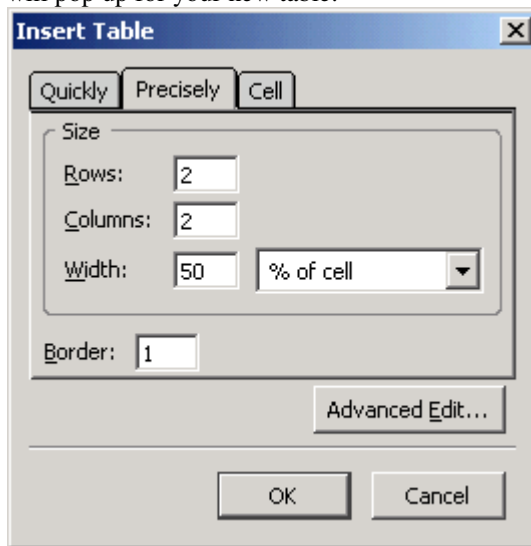
## Nested Tables

Nested Tables are tables that are within a cell of another table.

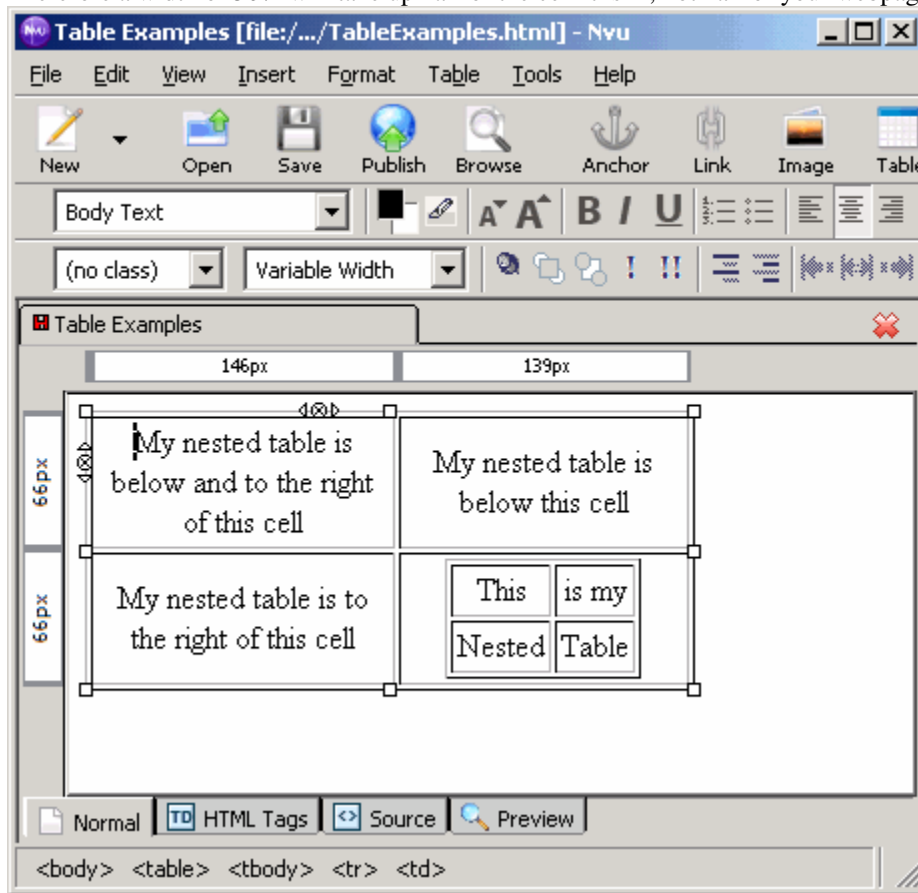
1. To create a nested table, right click on the cell in which you want to put your nested table.



- The pop up menu will appear. Select **Table Insert -> Table...** and the table properties window will pop up for your new table.



- Remember when you set the width for a nested table, you are setting it relative to the cell it is in. Therefore a width of 50% will take up half of the cell it is in, not half of your webpage.



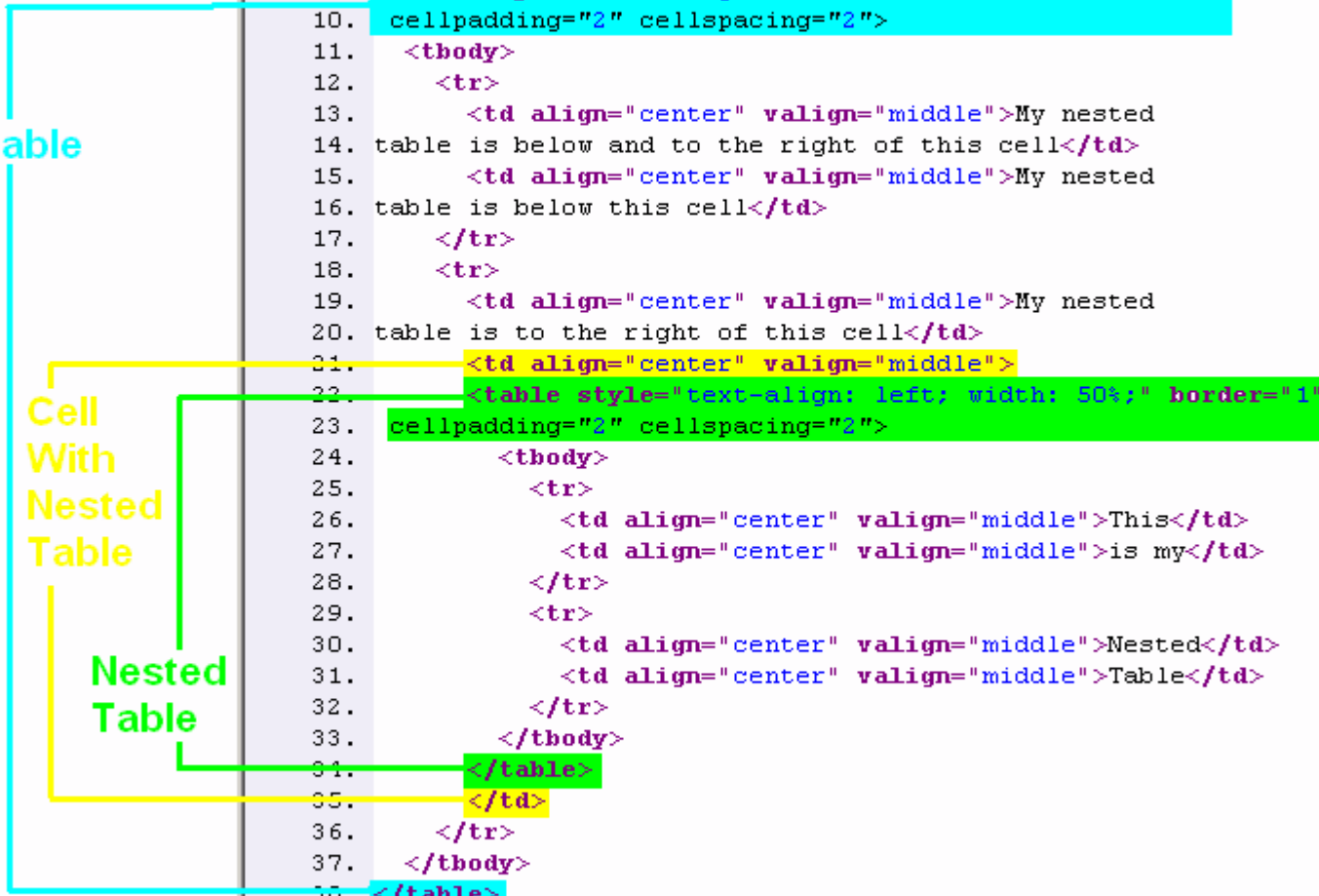
Taking a look at the source code for the above example, notice that the code for the inner table is fully inside the `<td>` tag of the outer table. Remember that *Nested Tables* are tables that are inside a cell of another table.

```
8. <body>
9. <table style="text-align: left; width: 75%;" border="1"
10. cellpadding="2" cellspacing="2">
11. <tbody>
12. <tr>
13. <td align="center" valign="middle">My nested
14. table is below and to the right of this cell</td>
15. <td align="center" valign="middle">My nested
16. table is below this cell</td>
17. </tr>
18. <tr>
19. <td align="center" valign="middle">My nested
20. table is to the right of this cell</td>
21. <td align="center" valign="middle">
22. <table style="text-align: left; width: 50%;" border="1"
23. cellpadding="2" cellspacing="2">
24. <tbody>
25. <tr>
26. <td align="center" valign="middle">This</td>
27. <td align="center" valign="middle">is my</td>
28. </tr>
29. <tr>
30. <td align="center" valign="middle">Nested</td>
31. <td align="center" valign="middle">Table</td>
32. </tr>
33. </tbody>
34. </table>
35. </td>
36. </tr>
37. </tbody>
38. </table>
39. <br />
40. </body>
41. </html>
```

Table

Cell With Nested Table

Nested Table



## Using Tables for Page Layout

It is common for websites to use tables for specifying a page's layout. If you look at some of the busier websites out there like [www.amazon.com](http://www.amazon.com) or [www.ebay.com](http://www.ebay.com), you will discover that they use tables to control page layout.

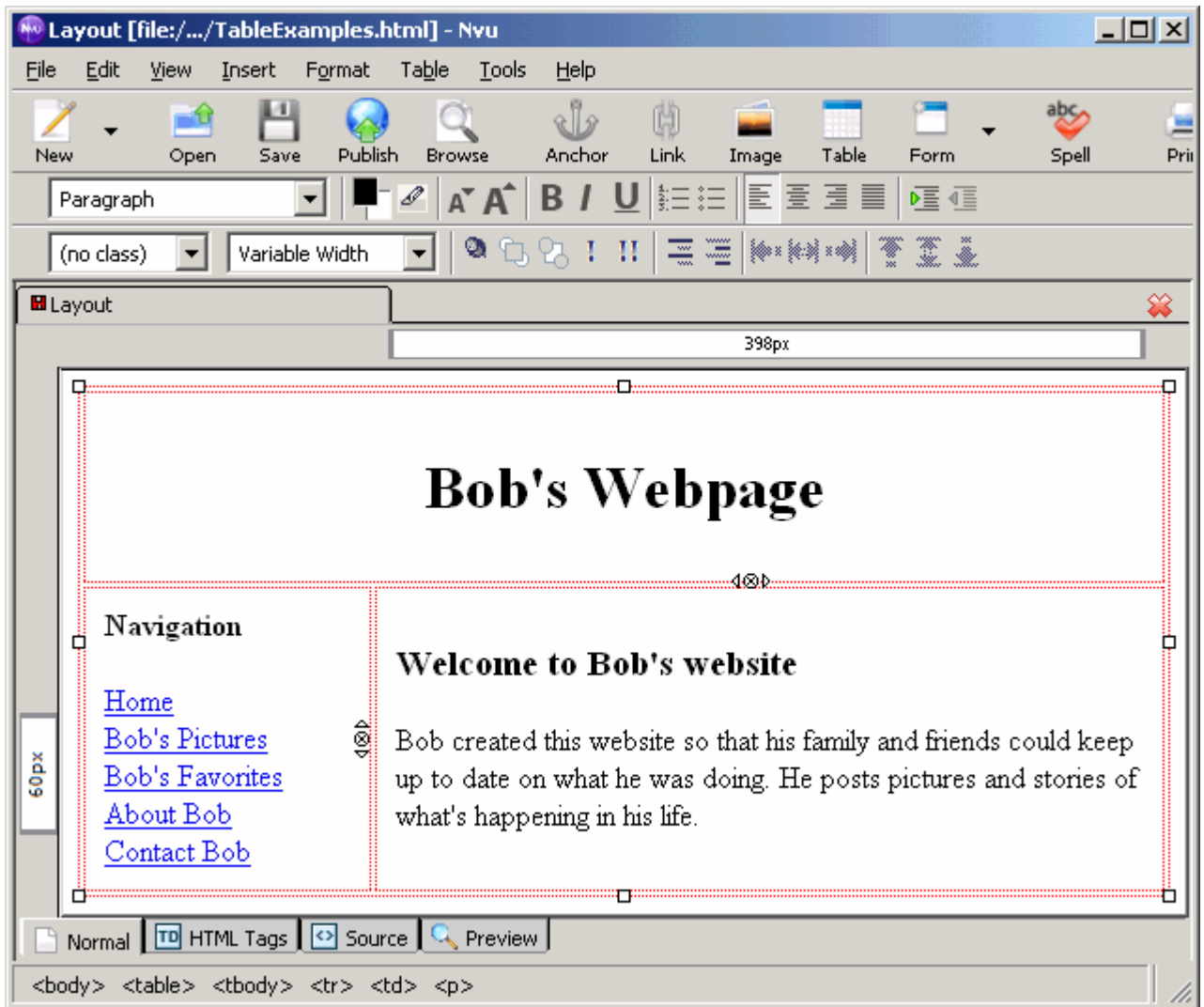
To control your page's layout using tables you must first divide the page into sections. You should sketch the page out on paper. It will allow you to look at several options side by side without having to code them. It's even better if you sketch it using grid paper as this will help you with the table proportions when you put the page into XHTML. Once you have sketched the page, create a table with a cell for each section of the page that you drew.

One of the more common and simplest page layout design using tables is a table with the first row spanning across the entire page and two columns directly below it. In this layout the first row holds the webpage's title. The second row holds two cells. The first is the navigation pane and the second holds the page's content.

The navigation pane is there to help visitors get from one part of the site to another more easily. It contains links to other parts of your website. The content pane is much wider than the navigation pane because it is the main source of information for the website. You can insert images, text, and other media.

<b>Title</b>	
<b>Links</b>	<b>Content</b>

Let's take a look at an Nvu page that can generate this simple layout.



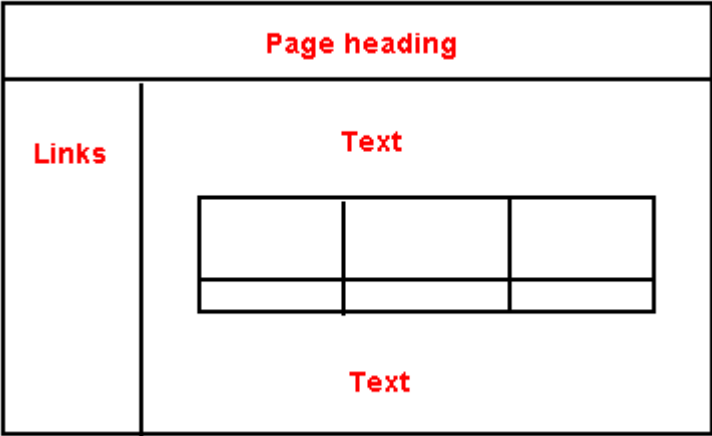
The above Nvu page generates the page displayed below. By having the borders set to 0 for our table, the page looks as though it is neatly organized.



Any of the content that you would normally use on your page can be used inside of a table's cell. You can even use tables within a table cell to further organize your page.

Images and other content can be easily organized within a table. Here is an example of a nested table used to organize a set of images. The same outer table is used as in the above example. The only changes are made to the "content" cell.

The sketch of the page looks like this:



And the Nvu page looks like this:

Layout [file:///.../TableExamples.html] - Nvu

File Edit View Insert Format Table Tools Help

New Open Save Publish Browse Anchor Link Image Table Form Spell Print

Paragraph Variable Width

Layout 476px




# Bob's Webpage

Navigation

- [Home](#)
- [Bob's Pictures](#)
- [Bob's Favorites](#)
- [About Bob](#)
- [Contact Bob](#)

## Welcome to Bob's website

Bob recently went on vacation... here are some of the highlights:

		
A ride on a camel.	A quiet town.	A tour of a castle.

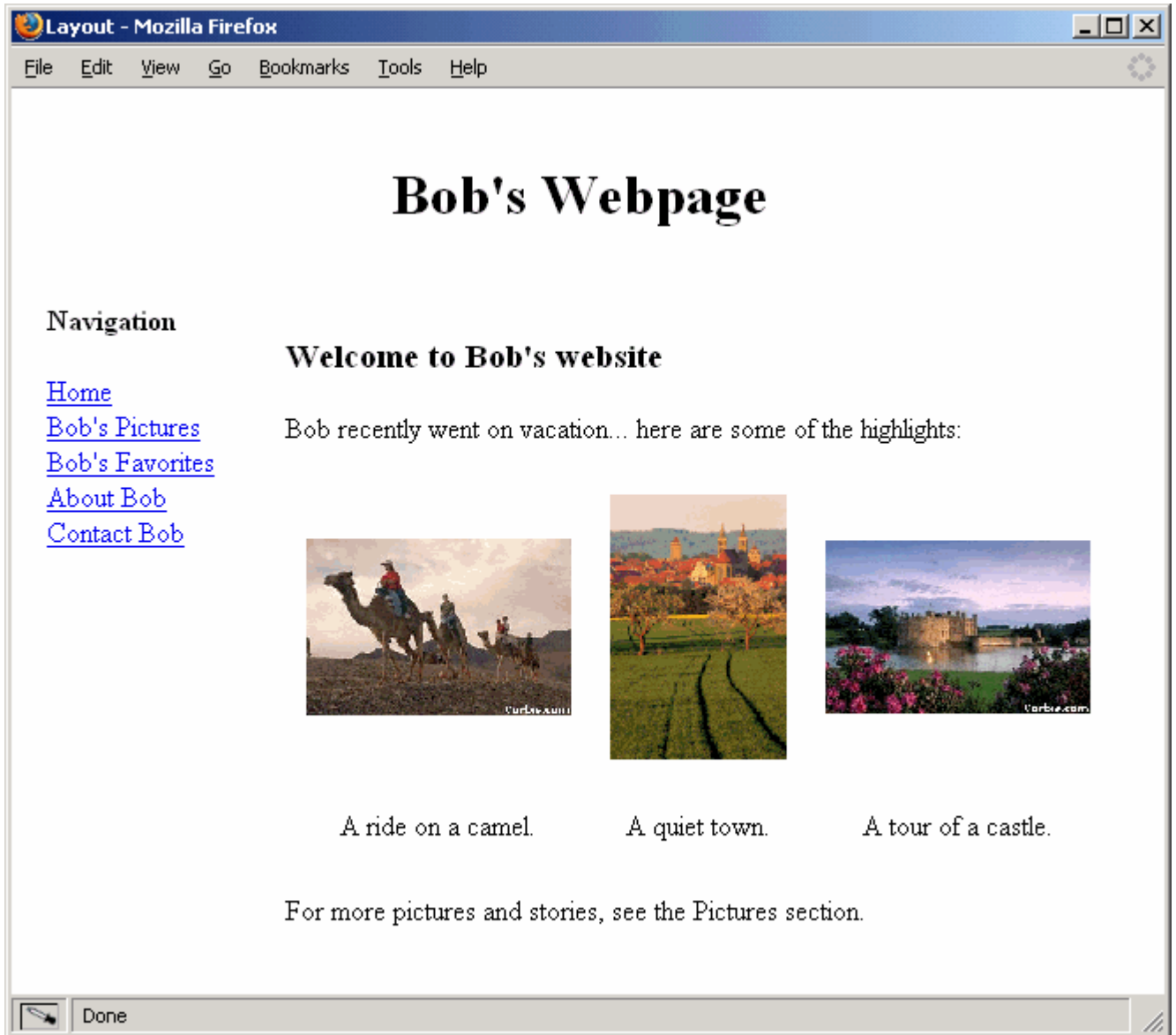
For more pictures and stories, see the Pictures section.

Normal HTML Tags Source Preview

<body> <table> <tbody> <tr> <td> <p>

Viewing the page in the browser, we see a very clean, organized page.





Tables can even be used to ensure that text appears around an image. This can be a little tricky. You'll have to play around with the text-align and vertical-align property values in order to get the exact look that you want.

A sketch of the page we're creating looks like this:

<b>Content</b>	
<b>Image</b>	<b>Content</b>
<b>Content</b>	

The Nvu page looks like this:

layout [file:///.../TableExamples.html] - Nvu

File Edit View Insert Format Table Tools Help

New Open Save Publish Browse Anchor Link Image Table Form Spell Print


Heading 1 Variable Width

623px

37px

# Our day at the beach

Our first day of vacation, we went to this lovely little beach. It was so quiet and peaceful, for most of the day we were the only ones there. We had packed a picnic lunch, which we ate sitting on a blanket.



In the afternoon, it was low tide, so we walked along the water's edge, looking for seashells. We also watched the hermit crabs walk along in the shallow water and built a sand castle. Next, we walked into the water to reach the sandbars and stood on our own little island in the middle of the water.

Once the tide started rising, washing away our castle, we decided it was time to go swimming. We dove into the waves, trying to jump at exactly the right time so that we would get the longest ride.

Normal HTML Tags Source Preview

<body> <table> <tbody> <tr> <td> <h1>

In a browser, the above Nvu page looks like the following:



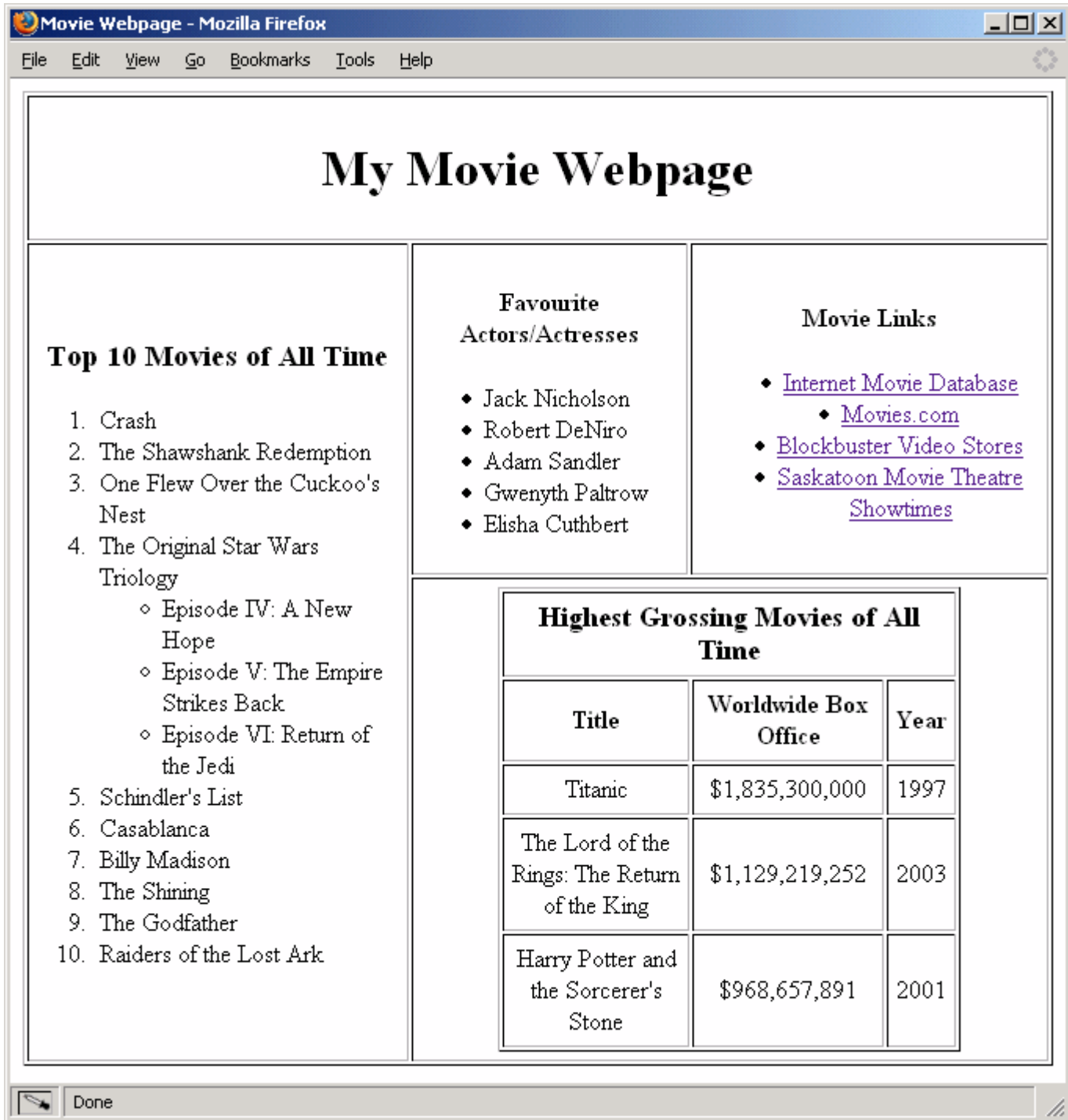
## Conclusion

This lesson showed how lists can be used to visually organize information on a webpage. It covered both types of lists, ordered and unordered, and the differences between them. It also showed how lists can be nested within one another.

Tables are very useful for organizing content on a webpage. The advanced table manipulation techniques presented here can help create complex page layouts. Nesting tables and playing with the table properties can lead to layouts well beyond what one might initially imagine when thinking of tables.

## Exercises

1. Define the keywords in this lesson
2. Create the webpage seen below



### Adding Style using Cascading Style Sheets

This lesson explores how to format your webpages using Cascading Style Sheets (CSS). Style sheets allow you to specify how your page should appear. You can adjust colors, margins, borders and so on by setting these properties within CSS Rules. We will take a look first at how CSS Rules work, and then how Nvu generates CSS Rules.

## *Objectives*

- Using CSS to format basic webpages
- Understand how style sheets are created and how styles can be applied
- Learn how Nvu generates CSS rules when you apply formatting to your page
- Learn how to use Nvu's CSS Editor to format your page

## **Keywords:**

- Cascading Style Sheets
- Styles
- Linked style sheets
- Global style sheets
- Inline style
- CSS Rule
- Class
- Selector
- Property
- Value
- Hexadecimal colors

## **What are Styles?**

**Cascading Style Sheets (CSS)** give us the ability to customize how our document will be displayed. For example, let's say that we want all headings of size 1 to be displayed in blue, italic, arial font of size 24. With regular HTML, you'd need to specify each of these details every time you create a new heading. If you suddenly decided that the headings should be red instead of blue, you'd have to go through your entire document and fix each instance of the heading 1. With CSS, you can pre-define what an heading should look like and have all of your headings follow that same format. So if you wanted to change from blue to red text, you'd only need to change it in one location and it would affect all of your headings of size 1 instantly.

Cascading Style Sheets are powerful because they can separate the content of the page from the formatting of that content. They also provide much more control over the exact formatting of the document. You can adjust borders, margins, colors, alignment for most elements, as well as have fine control over details specific to each type of element (like choosing what happens when you move your mouse over a hyperlink).

## **Location of Styles**

There are three places where you can define styles. You can have a "**linked style sheet**", a "**global style sheet**", or an "**inline style**". We will mostly be dealing with global and inline style definitions.

## *Linked Style Sheets*

A linked style sheet is created as a separate file from your XHTML document. Here, the content and the formatting are truly separate. Your XHTML page references the style sheet to find out how it should format the content of your webpage. This method allows you to create one style sheet and use it for several webpages. Web designers will do this when they create a website that has several pages with the same "look". Since all of the style information is stored in one place, making a change to how the website is presented becomes quite simple.

To link a style sheet to an XHTML document you would reference it in your XHTML file. You would then create the Cascading Style Sheet with the filename you referenced in the XHTML file. CSS files end with the .css filename extension and they contain all of the style information for your pages.

### ***Global Style Sheets***

Global styles are defined within the head of the XHTML document. All of the formatting information goes in the head of the document while all of the content of the page belongs within the body of the document.

### ***Inline Styles***

Inline styles are similar to old-fashioned HTML that used attributes within an element to define how it should be formatted. They apply to a single element only. To define an inline style, you use the **style** attribute within the tag for that particular element.

We will look at creating all these types of style sheets in a future lesson.

### ***Style Precedence***

A webpage can use all three types of style definitions at once, or any combination of them. When this happens however, there is a chance that two style definitions contradict each other. For example, a linked style sheet may say that paragraphs should be blue while the global style sheet says paragraphs should be green. How does the browser know which one to use? It follows these guidelines:

- inline styles override both global and linked styles
- global styles override linked styles

So, in the case where the linked style sheet says paragraphs are blue and the global style sheet says green, the paragraphs would in fact be green. If a particular paragraph on this webpage then had an inline style saying this paragraph is yellow, the browser would display it as yellow.

Think of it as "the closer the style definition is to the actual content, then the more influence it has". When the style definition is contained within the element, it always has precedence, followed by styles defined within the webpage (global), and lastly by styles defined in another file.

### **Style Terms and Format**

Each of the style definitions are called a "**CSS rule**". The rule defines what element you're talking about and what formatting should be applied to it. The following format is used for global and linked style sheets.

A CSS Rule looks like this:

```
selector
{
  property: value;
}
```

The **selector** identifies what element you're formatting. The **property** determines what aspect of the element you're formatting. There are specific properties available for each element. Examples of properties are color, font-size, font-family, and border-width. The **value** specifies how you want to set a particular property. For example, the color property may have values of red, green, or blue, while the font-family property may be assigned a value of arial or courier. Again, each property has a specific set of values that can be assigned to it.

It is also important to note the placement of the curly brackets, the colon, and the semi-colon. These must be included for the rule to be recognized by the browser. The curly brackets surround all of the property declarations, grouping them together. There is a colon after the name of each property and a semi-colon after each value.

Lets look at an example of a CSS rule:

```
Example
h1
{
  color: blue;
}
```

This rule sets all headings created with the <h1> tag to be displayed in blue.

It is possible to set several properties at a time. In fact, you can set as many properties as you want for one selector. The following example sets two properties for the <h1> element:



```
Example
h1
{
color: blue;
font-family: arial;
}
```

With this rule, all headings created with the <h1> tag would be blue and displayed using arial as the font.

You may want to set the same properties for several elements at once. For example, you might want headings and paragraphs to be displayed in blue and with the same font. It is possible to list more than one selector for a rule, as in the following example:

```
Example
p, h1, h2
{
color: blue;
font-family: arial;
}
```

Here, all paragraphs and headings created with either the <h1> or <h2> tag will be displayed in blue, using arial font. Notice that each selector is separated by a comma.

### Inline Style Format

Defining inline styles works much the same way as with global or linked style sheets, except the information is contained within the element it is describing.

The format of an inline style definition looks like this:

```
FORMAT
<element style="property: value;" >
```

We use the style attribute to set the properties for the particular element. The properties are placed within a set of quotation marks and still use a colon after the property name as well as a semi-colon after its value. Note that nothing extra appears in the closing tag for the elements - attributes (such as style) are only located in the opening tag.

Looking at an example:

```
Example  
  
<h2 style=" color: aqua; " >
```

Here, we are setting an h2's color property to aqua. Notice that this style only applies to this particular <h2> heading, not all <h2> headings on the page.

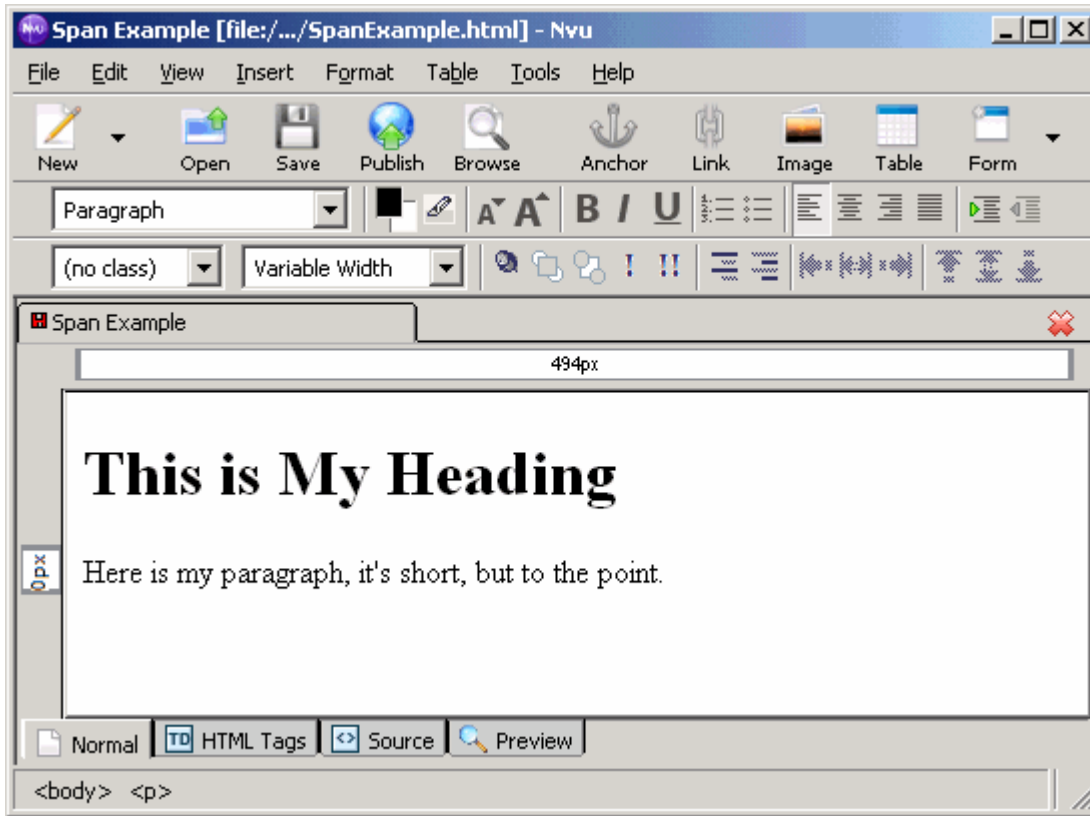
We can set several properties of an element at once, like this:

```
Example  
  
<p style=" color: aqua; font-family: arial; " >
```

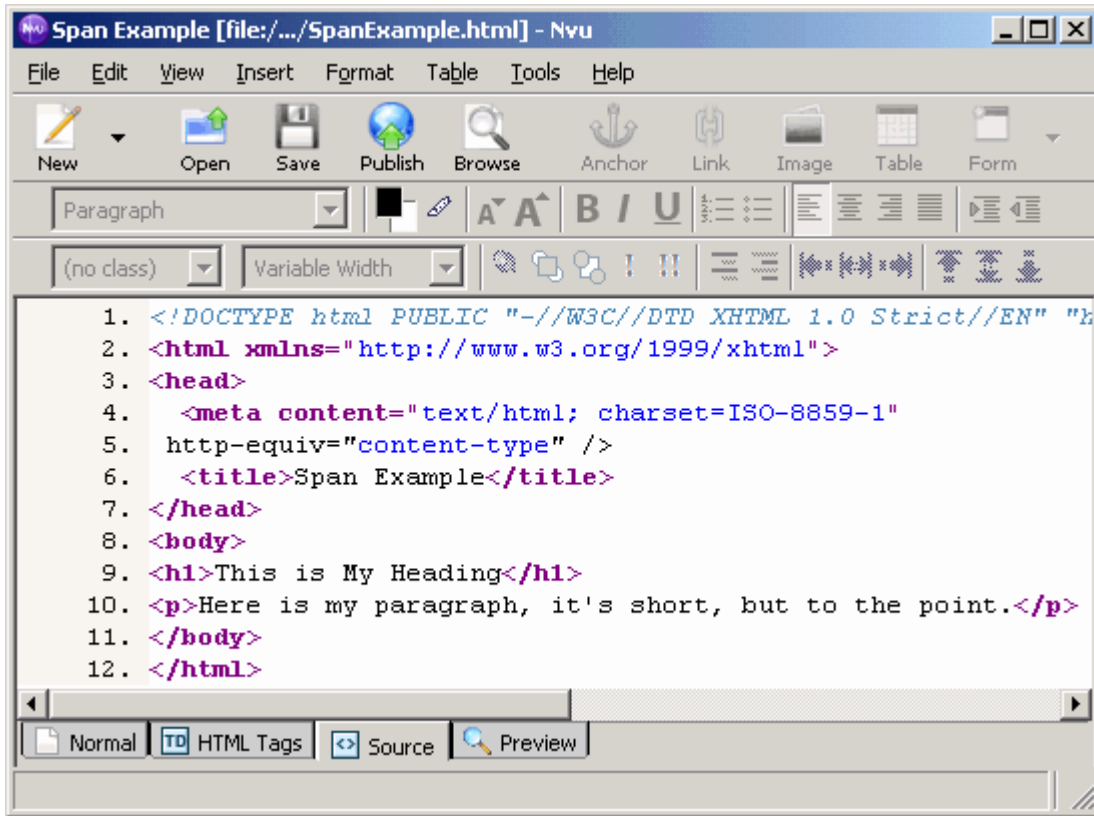
Notice that we only have one style attribute and all of the properties are listed together, separated by semi-colons. You can list as many properties as you want for an element.

### *The <span> Tag*

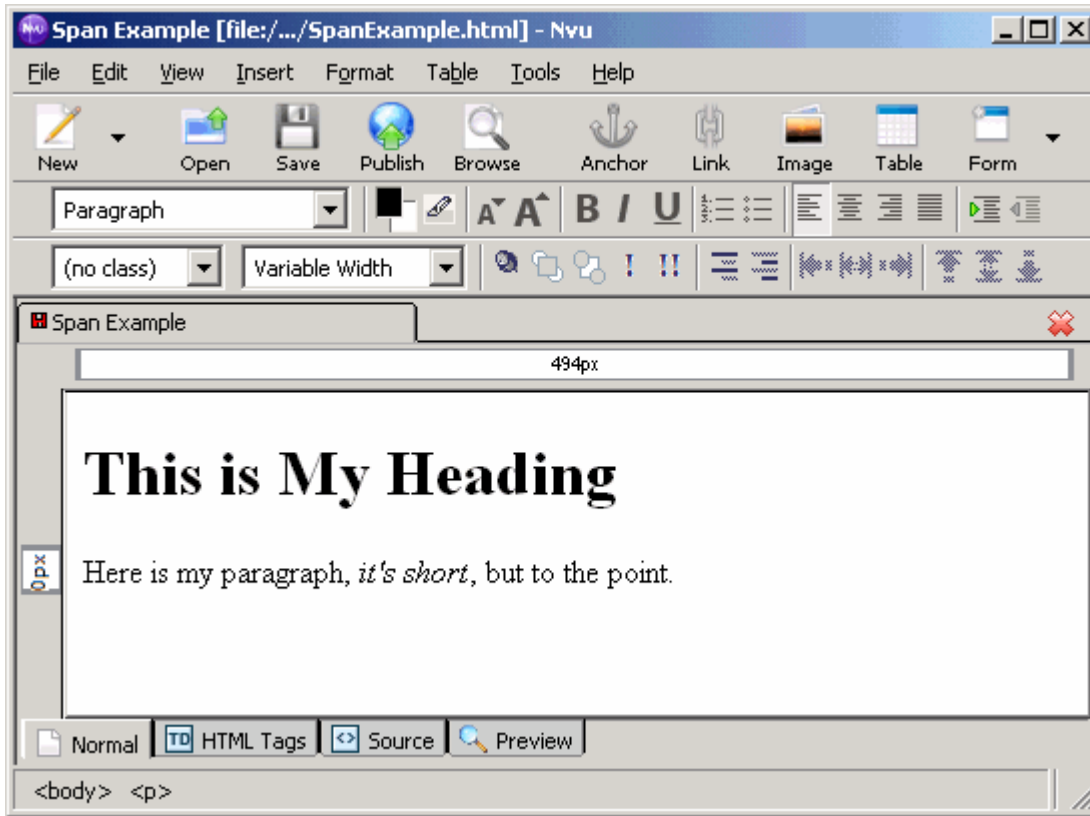
Sometimes you may choose to format something on your page that does not have a tag associated with it, for example, a few words in a paragraph. Since the words lie within paragraph tag, and you do not want to set the formatting for the entire paragraph, Nvu inserts the **<span>** tag when you apply formatting to bits of text. The span tag has no effect on the text, other than the inline formatting rules that you set. For example, let's look at the following page,



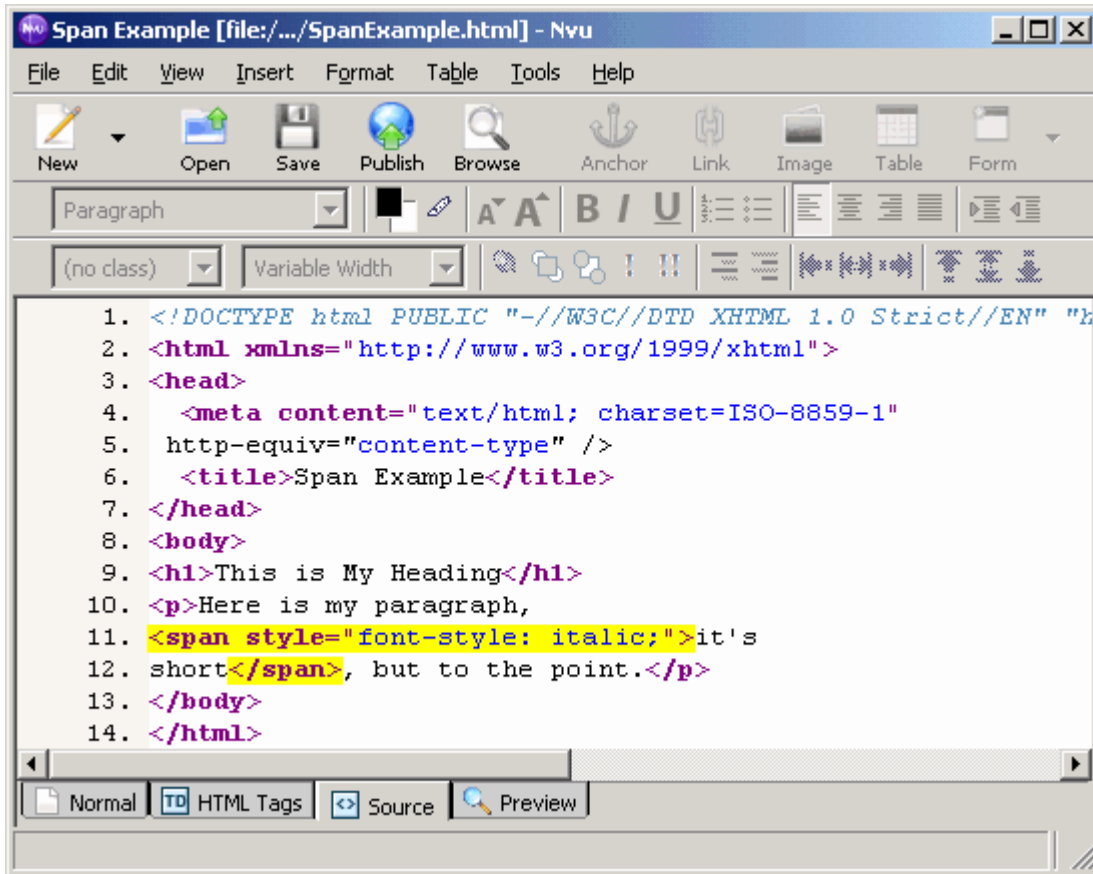
and it's source code,



We can see in the source code that there is no formatting applied yet. Once we add some formatting in the following example,



we can look and see how our source code changes to include the span tag:

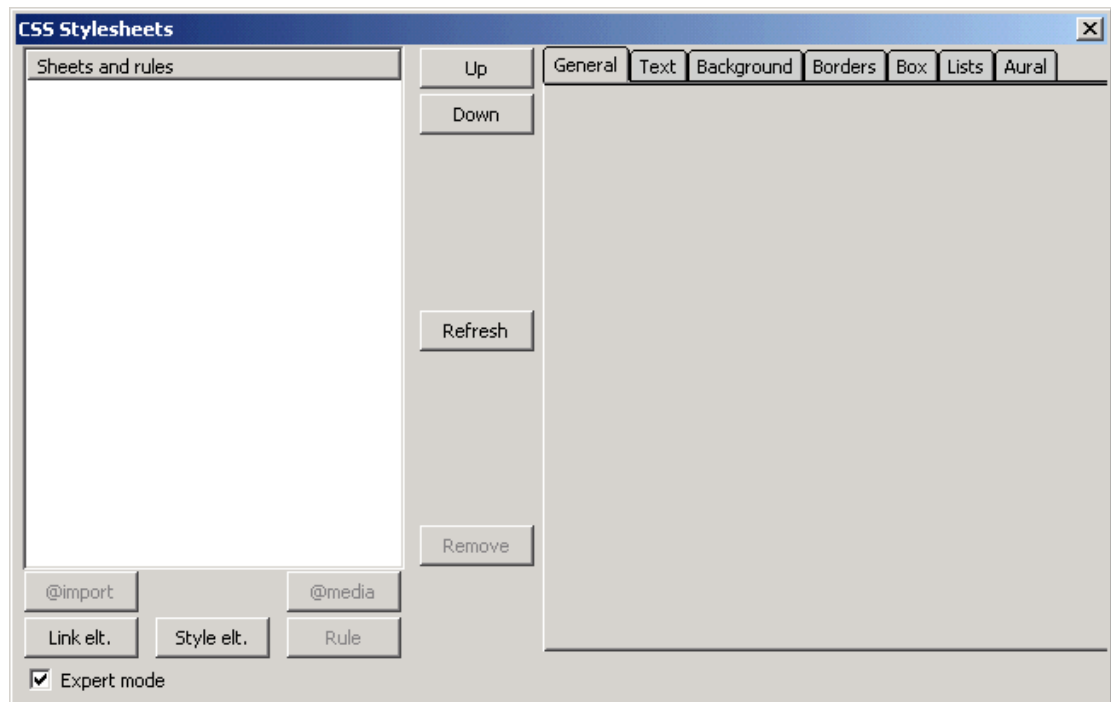


Notice how the span tag is within our paragraph, but only encloses the words that are italicized. Also notice how the **style** attribute is used to apply the formatting.

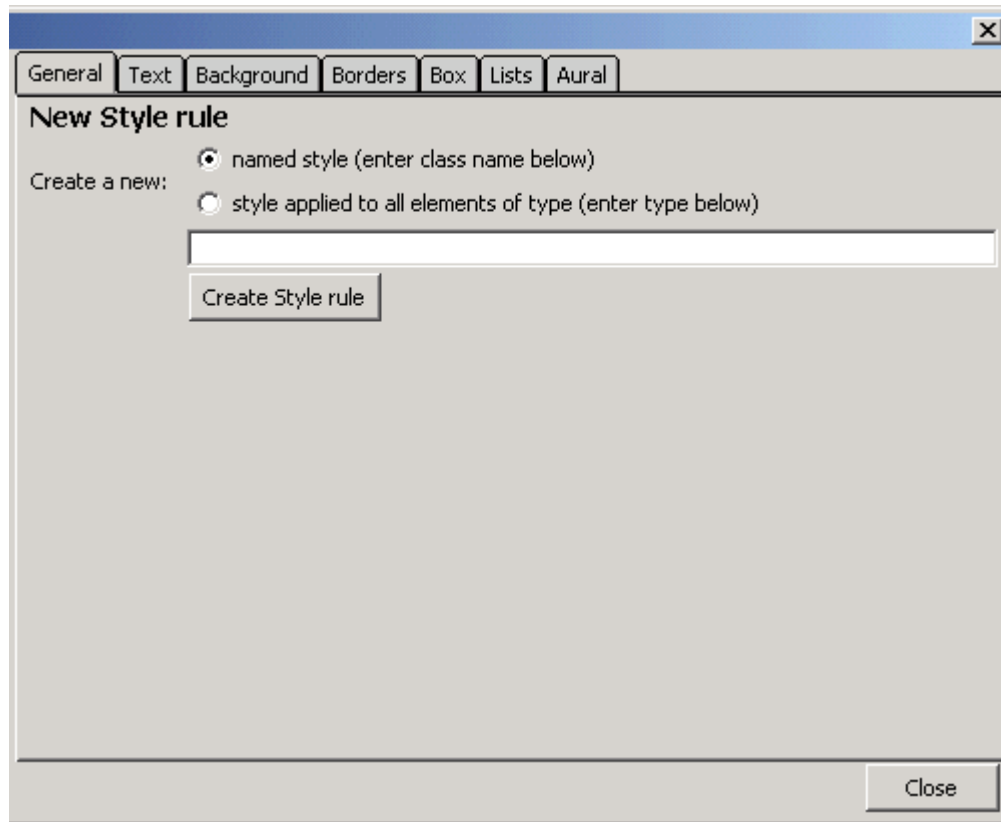
### Global Styles with the CSS Editor

Nvu has a CSS Editor that allows you to create custom CSS global style rules. You will notice the global style rules appear in the head of your source code in Nvu as you create them with the CSS Editor. Nvu's CSS Editor is under the **Tools -> CSS Editor** menu. This is where you develop rules that define the formatting of your webpage.

1. To begin creating your rules, uncheck the **Expert mode** checkbox in the lower left corner.



2. You will be given two options for your rule:
  - named style - creates a class with the name you supply
  - style applied to all elements of type - allows you to specify a tag (ie. h1) to which the style will be applied

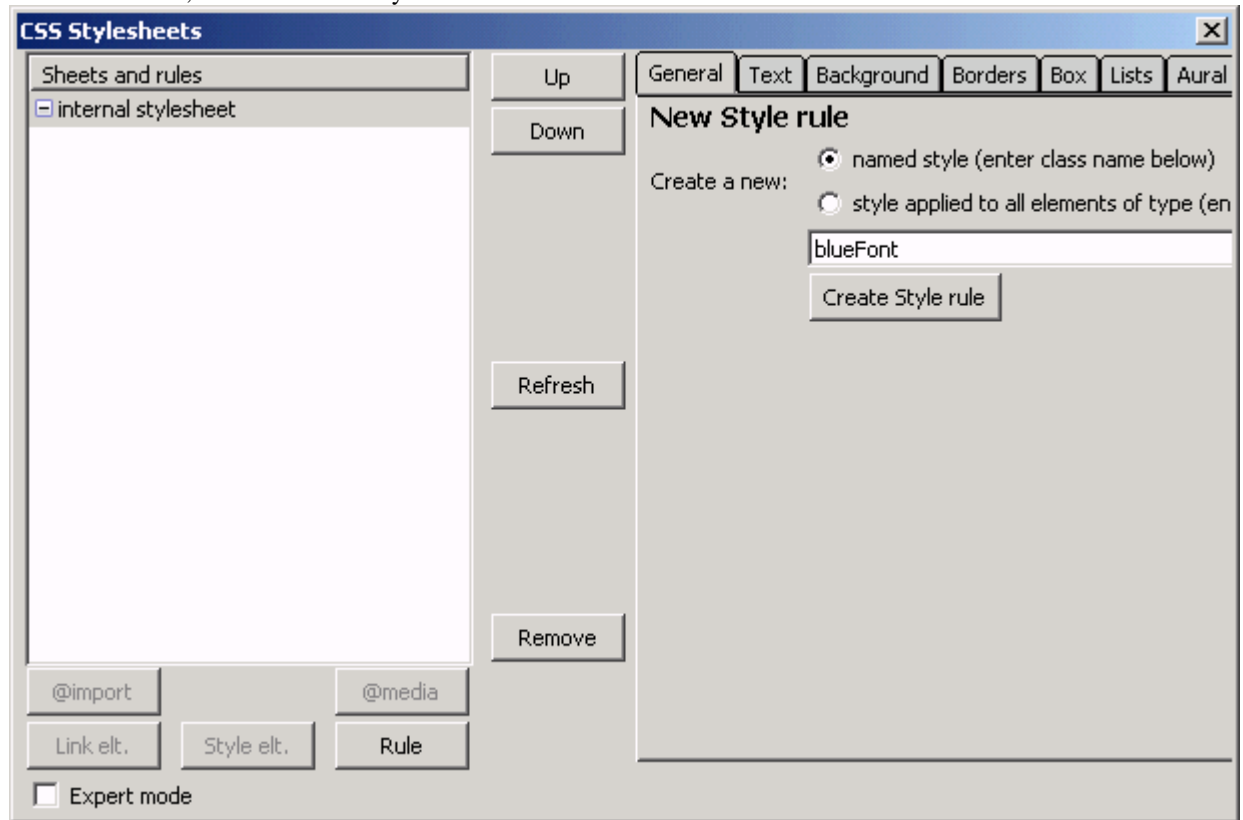


### *Named Style*

The **named style** option allows you to create a **class** that has style elements associated with it. You can then apply the class to whichever elements on your page that you desire. For example, if you wanted to have some headings and some paragraphs on your page have a green background, you can create a named style called greenBackground. In our case, we're going to create a named style with the class blueFont which displays the elements using blue text.

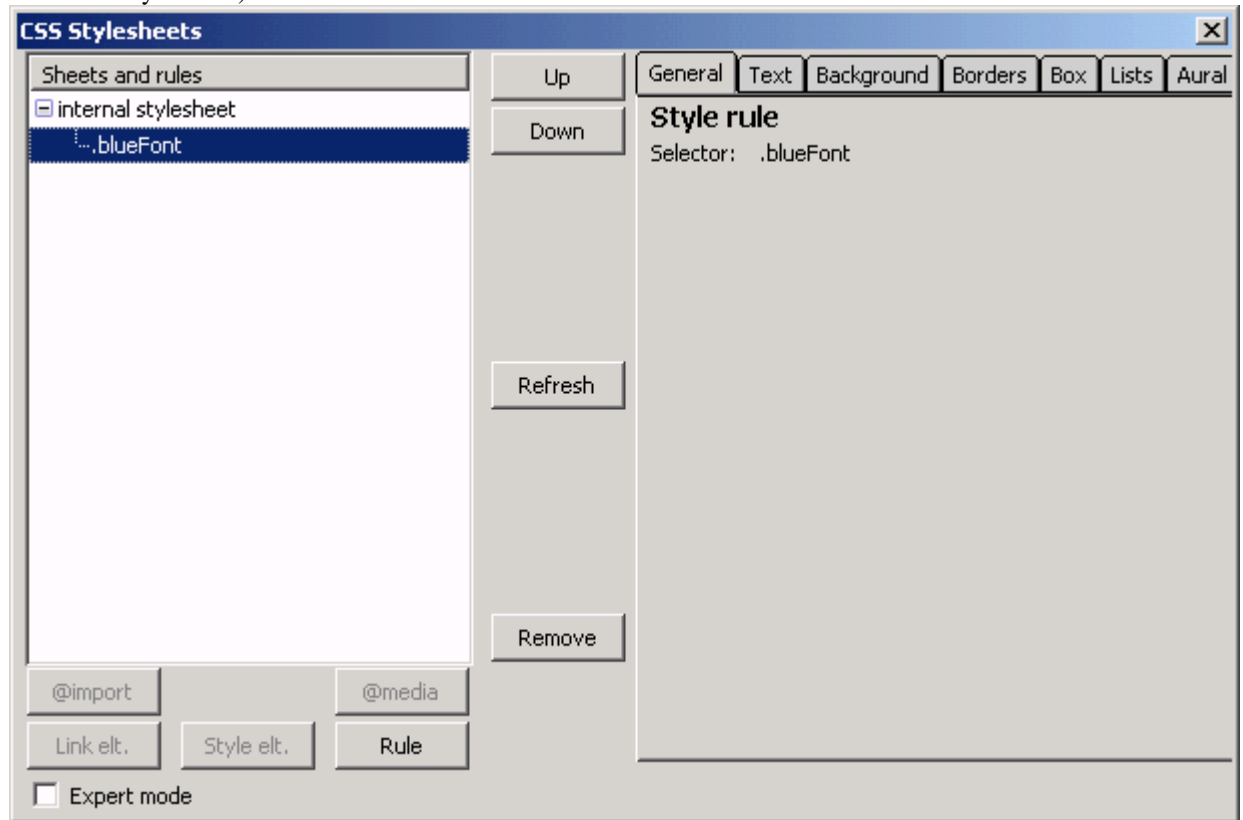


1. To create a class, enter the name of your class in the text box



2. Click on the **Create Rule** button and you will see your class appear on the left hand side in the **Sheets and Rules** area. Select your rule (you may have to click the + sign beside **internal style**

sheet to see your rule).

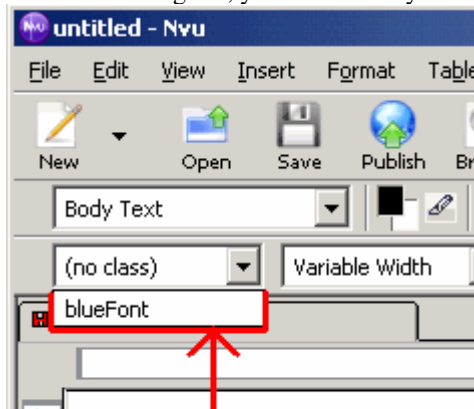


3. To apply style to a rule, choose one of the tabs on the right hand side:
  - Text - To format text font, style, color, alignment, etc.
  - Background - Allows you to set the background color or image for elements on your page
  - Borders - Allows you to put a border around elements on your page, lets you choose width and color
  - Box - Many features including setting margins and paddings (useful in tables), also allows you to set the width and height of elements
  - Lists - Allows you to specify the style for the bulleting of a list (ie. roman numerals, disc, image)
  - Aural - Rules for audio, not used in this course

4. In our example, we're going to select the **Text** tab. In the **Color** text box we will set our color to "blue".



5. Once you've set your formatting, click the **Close** button in the bottom right corner. You should now be back in Nvu's main window. If you click on the drop-down menu that says **(no class)** in Nvu's formatting bar, you should see your class there



**Class Selector**

6. Now that you have your class created, the next step is to apply the class to elements on your webpage. To do this, select the element you wish to apply the rule to and select the class from the

Selector list

The screenshot shows the Nvu web editor interface. The title bar reads "CSS Examples [file:///.../CSSExamples.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. Below the toolbar are two rows of formatting options: the first row includes a Paragraph dropdown, a blue font color selector, and buttons for Bold (B), Italic (I), and Underline (U); the second row includes a "blueFont" class selector, a "Variable Width" dropdown, and various alignment and list icons. The main workspace shows a preview of a web page with a width of 548px. It contains a heading 

# This is a Heading 1

 in blue, bold font, followed by a paragraph 

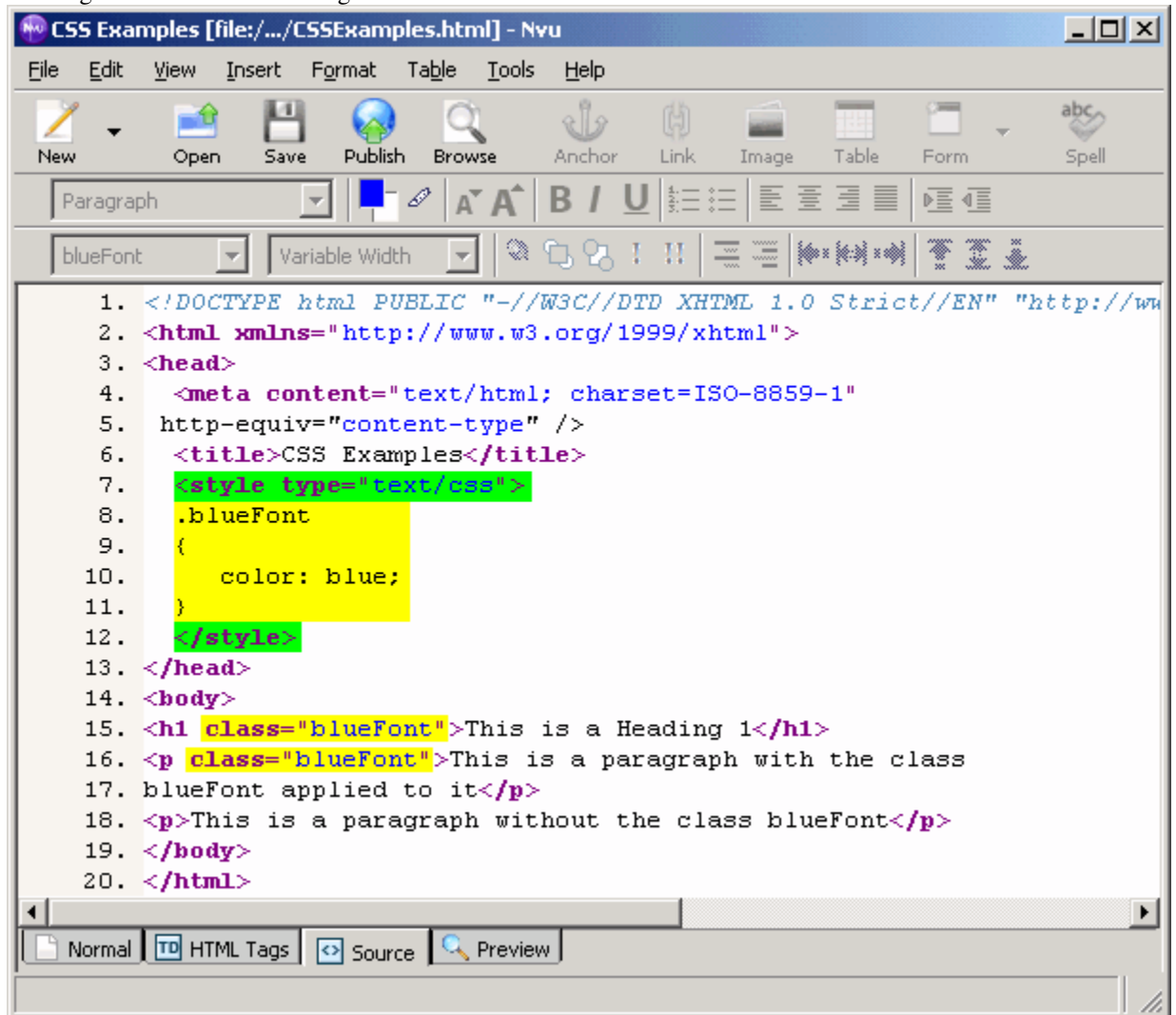
This is a paragraph with the class blueFont applied to it

 in blue font, and another paragraph 

This is a paragraph without the class blueFont

 in black font. At the bottom, there are tabs for "Normal", "HTML Tags", "Source", and "Preview". The "Source" tab is active, showing the code `<body> <p>`.

7. Looking at the source code Nvu generates



The screenshot shows the Nvu editor window titled "CSS Examples [file:///.../CSSExamples.html] - Nvu". The interface includes a menu bar (File, Edit, View, Insert, Format, Table, Tools, Help), a toolbar with icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell, and a formatting toolbar with options for Paragraph, blueFont, Variable Width, and various text and alignment tools. The main editing area displays the following HTML source code:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title>CSS Examples</title>
7.   <style type="text/css">
8.     .blueFont
9.     {
10.       color: blue;
11.     }
12.   </style>
13. </head>
14. <body>
15. <h1 class="blueFont">This is a Heading 1</h1>
16. <p class="blueFont">This is a paragraph with the class
17. blueFont applied to it</p>
18. <p>This is a paragraph without the class blueFont</p>
19. </body>
20. </html>
```

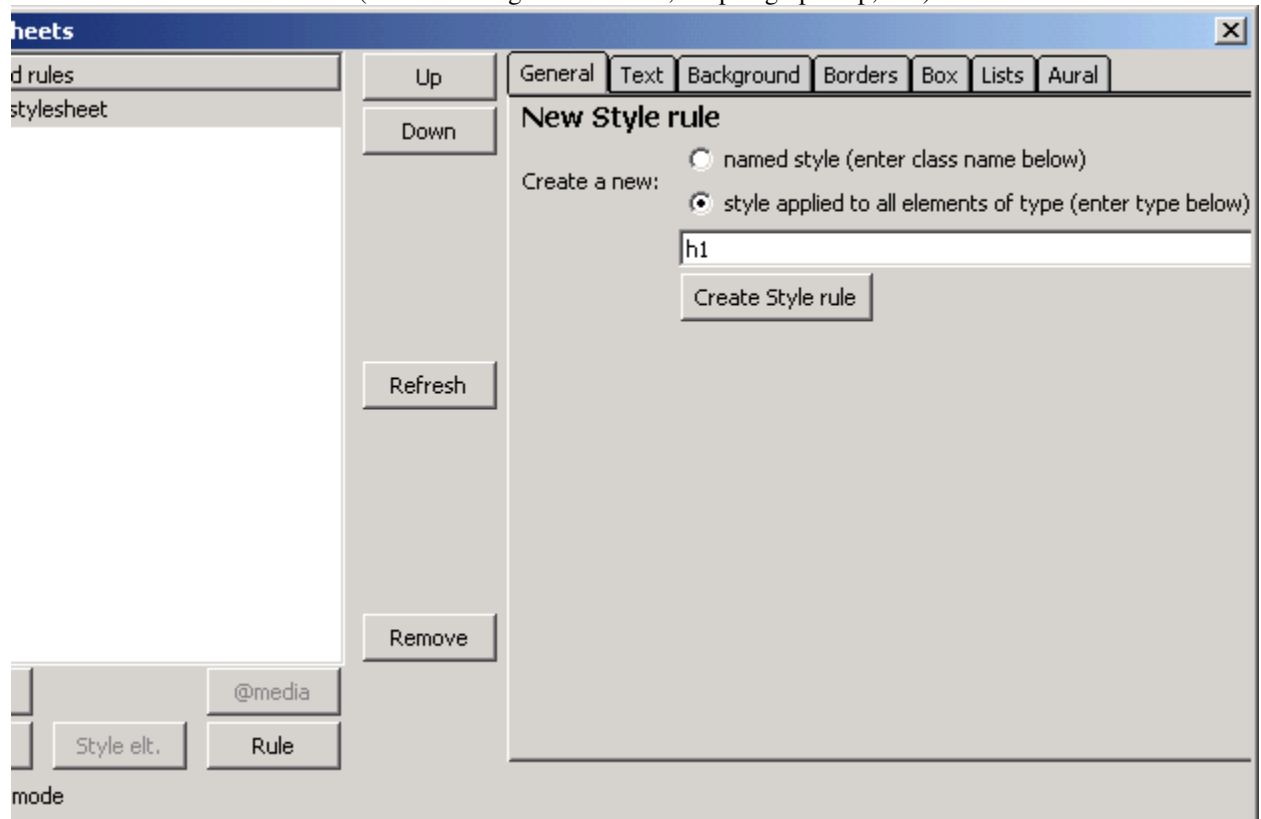
At the bottom of the editor, there are tabs for "Normal", "HTML Tags", "Source", and "Preview".

We see inside our head, Nvu creates `<style></style>` tags. All of our global CSS rules will be nested within these tags. Also notice that our class name appears as `".blueFont"`. The period before the name indicates that it is a class that can be applied to different elements. Finally, we see our class applied to elements by using the `class` attribute within our opening tags. It is important to remember that classes can be applied to different types of elements, and by applying it to one type of element, does not mean that it will be applied to all other elements of that type.

*Style applied to all elements of type*

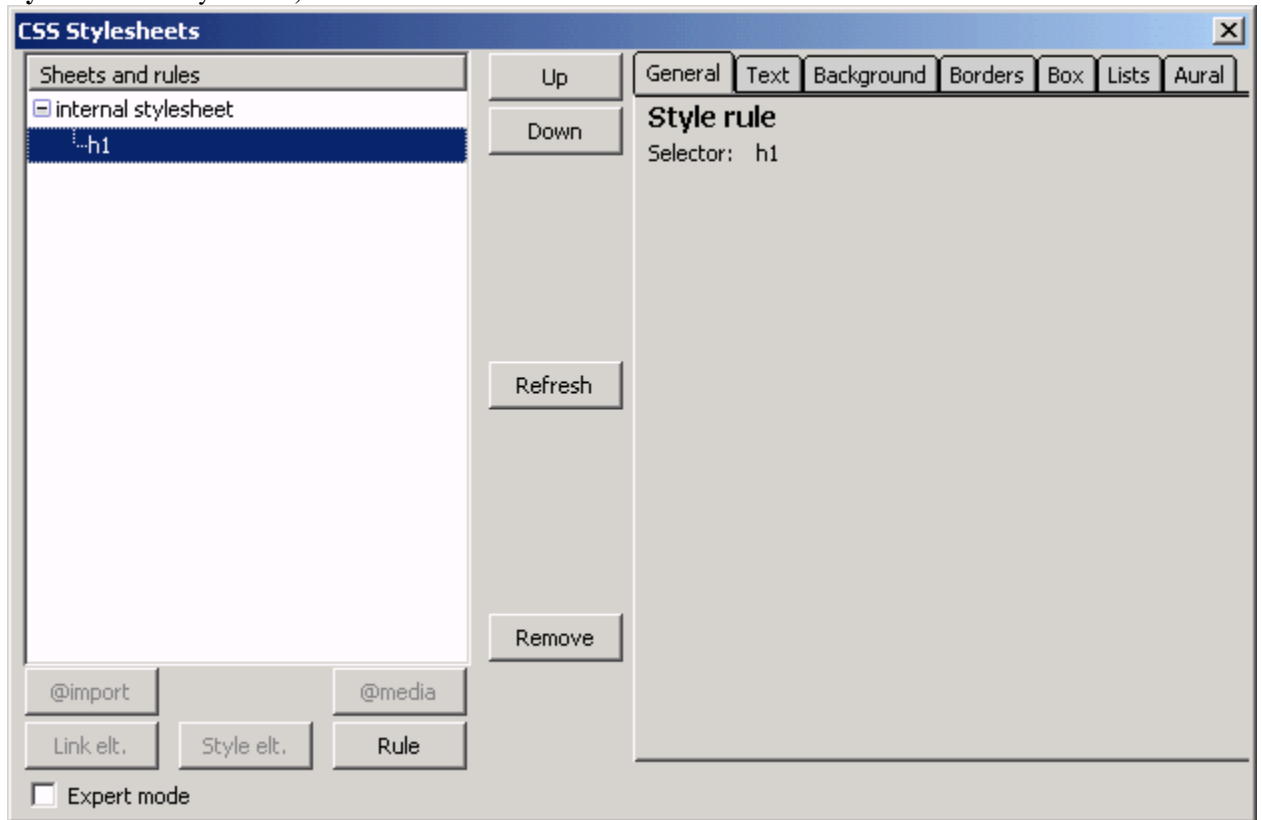
The **Style applied to all elements of type** option allows you to create rules that will be applied to all elements of a specific type. For example, if you wanted all headings of size 1 on your page to be purple and italicized, you would create a rule for all h1 elements.

1. First, select the style applied to all elements of type radio button. Enter the element you wish to create rules for into the text box (ie. for heading of size 1 - h1, for paragraphs - p, etc.)



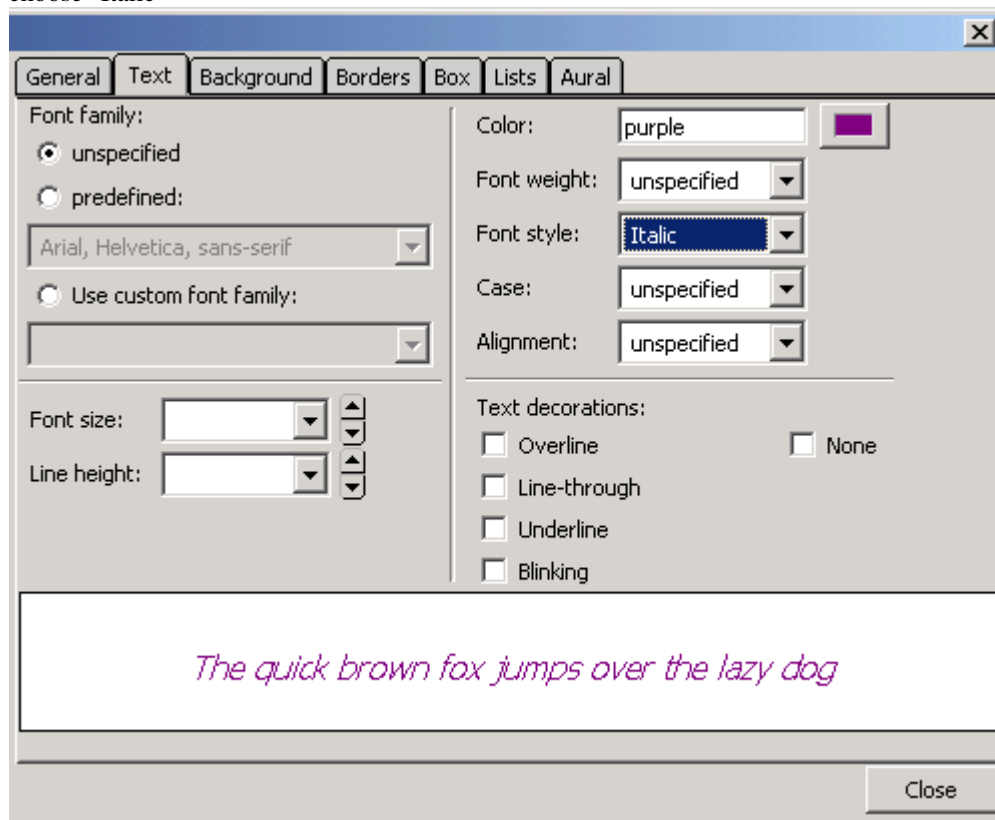
2. Click on the **Create Style Rule** button and you will see your rule appear on the left hand side in the **Sheets and Rules** area. Select your rule (you may have to click the + sign beside **internal**)

style sheet to see your rule).



3. Style can be applied by selecting one of the tabs as listed above in the "Named Style" section. For our example we are going to make all our headings of size 1 purple and italicized. To do this we first click on the **Text** tab. We will set the color to purple and in the **Font style** select list we will

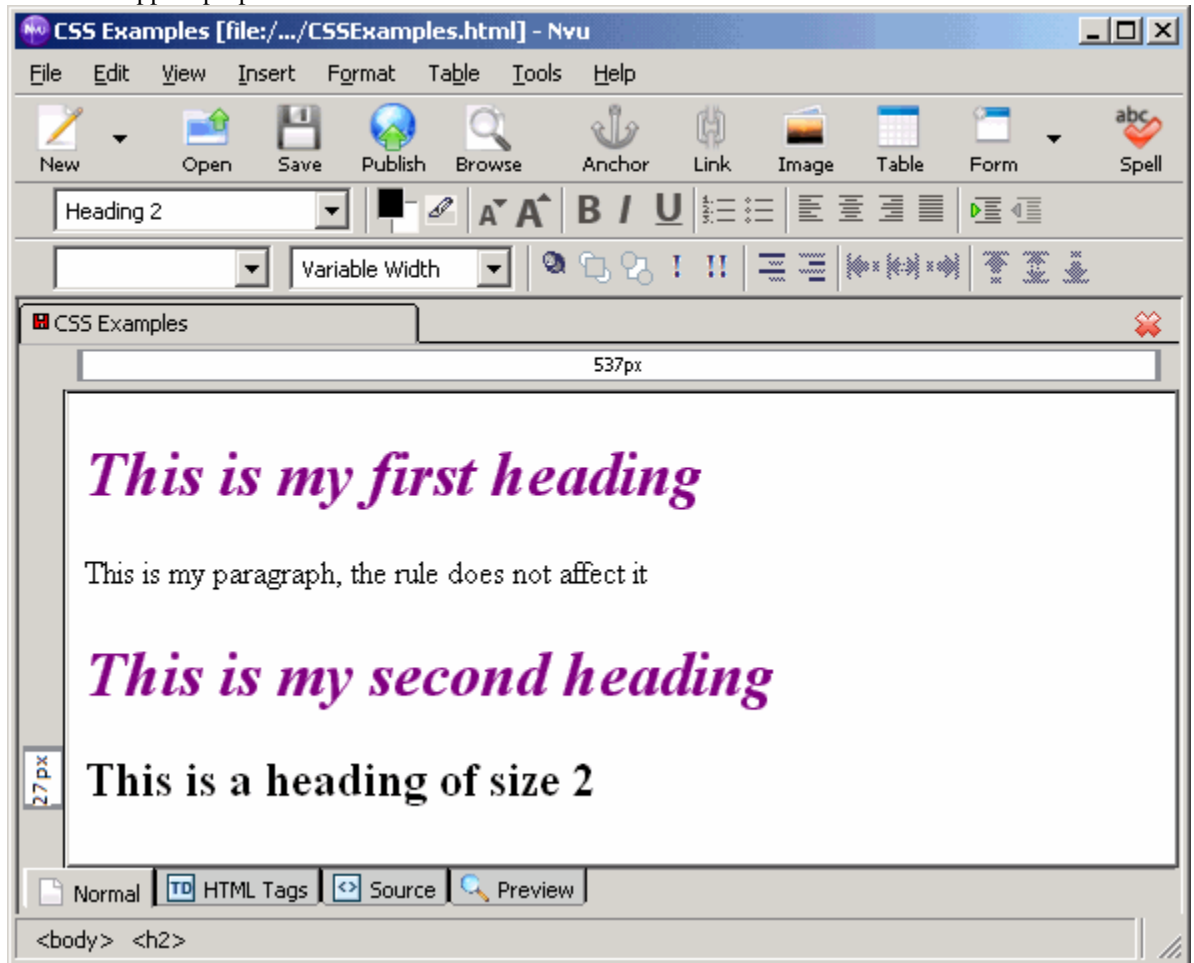
choose "Italic"



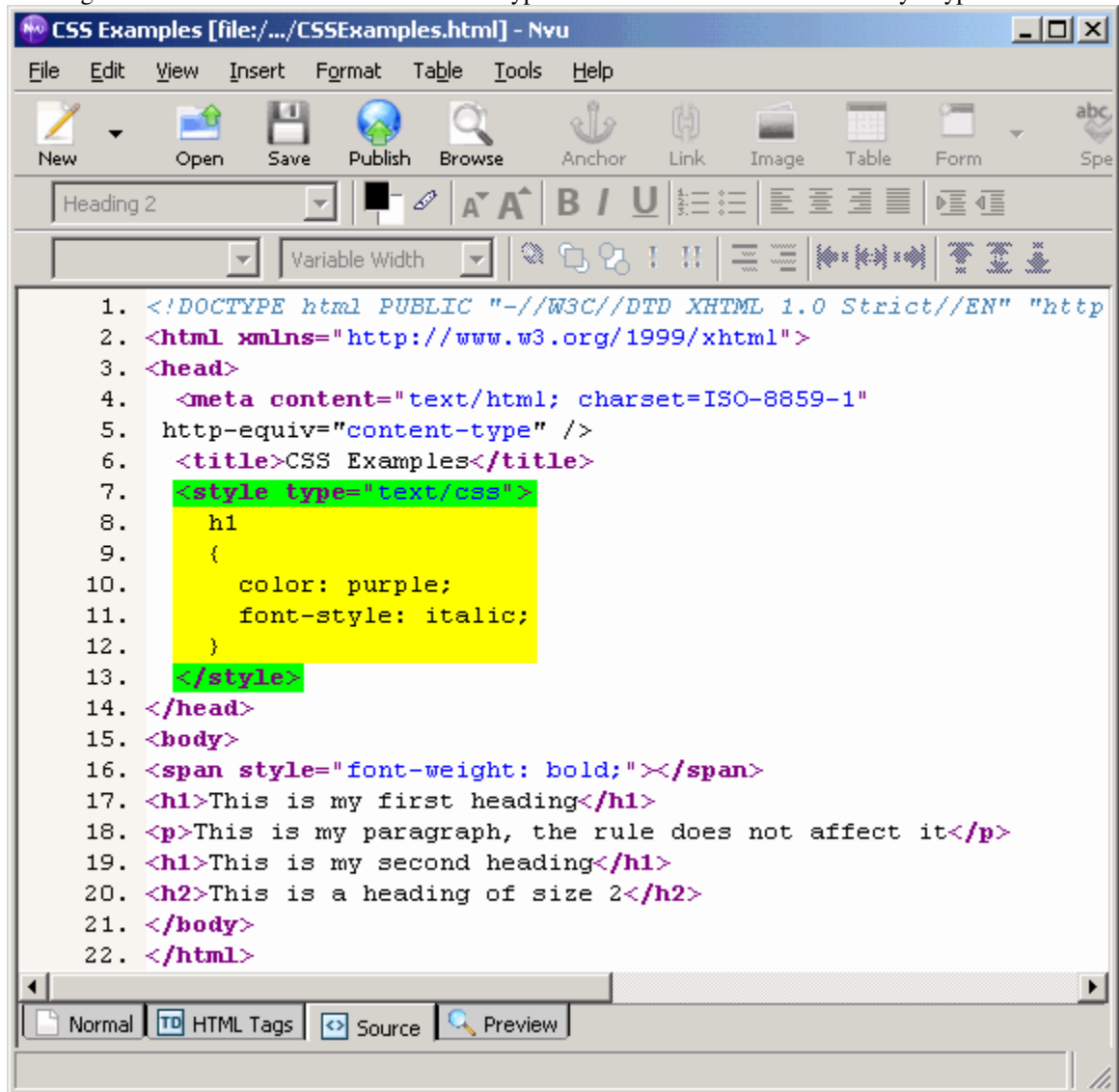
4. Hit the close button in the CSS Editor and you will be returned to Nvu's main window. You will see that any headings of size 1 already on the page, as well as any new headings of size 1 you



create will appear purple and italicized.



5. Looking at the source code we can see how this type of rule differs from the Named Style type



The screenshot shows the Nvu editor window titled "CSS Examples [file:///.../CSSExamples.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, and Spell. The main editing area shows the following HTML source code:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title>CSS Examples</title>
7.   <style type="text/css">
8.     h1
9.     {
10.      color: purple;
11.      font-style: italic;
12.     }
13.   </style>
14. </head>
15. <body>
16. <span style="font-weight: bold;"></span>
17. <h1>This is my first heading</h1>
18. <p>This is my paragraph, the rule does not affect it</p>
19. <h1>This is my second heading</h1>
20. <h2>This is a heading of size 2</h2>
21. </body>
22. </html>
```

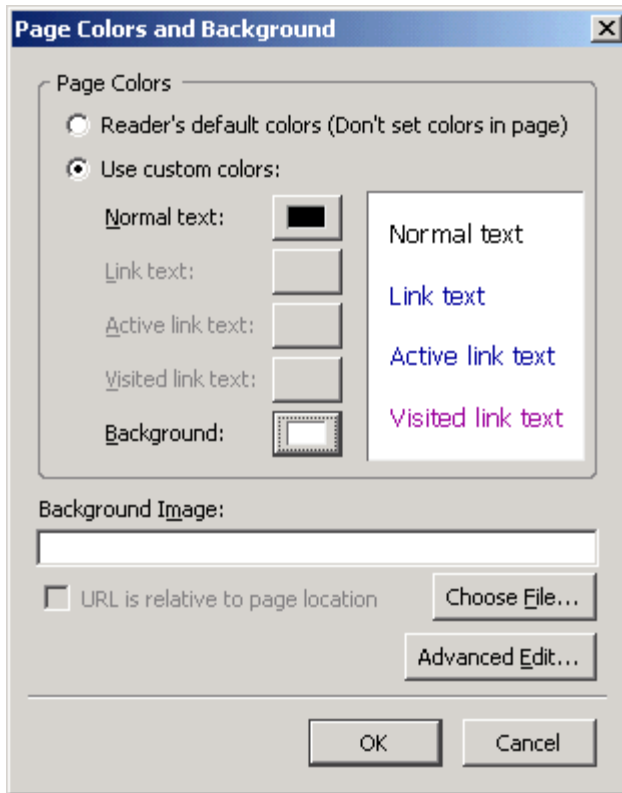
The code is displayed in a source view, with tabs for Normal, HTML Tags, Source, and Preview at the bottom.

Nvu still creates the `<style></style>` tags inside the head. We also see our rule created as `h1` does not have a period in front of it. This is because it is not a class. By not having the period, the rules for `h1` will be applied to all `<h1>` tags. Also notice that we do not need to use the class attribute, as these types of CSS rules are applied directly to all elements of the type you specify (in this case `h1`).

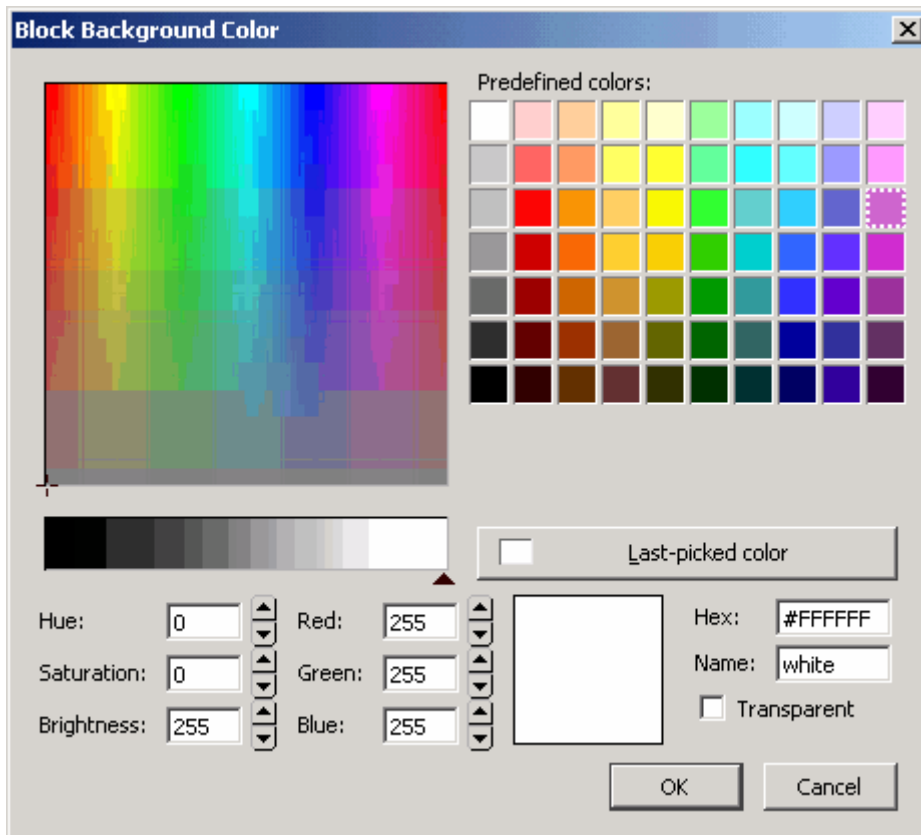
### Setting the Page Background

Nvu sets the formatting for specific elements on your page by using **inline style rules**. If you are setting the formatting for just one heading, a selection of text, one table cell, or even the background of the page, an inline style rule is used. Nvu places the **style** attribute inside the opening tag of that specific element and will then set that attribute equal to the style rule for the formatting you applied. There is no need to create or apply any class.

You can set the background color and default text color of your webpage. To do this click on **Format -> Page Colors and Background...**



The above window will pop up. By default, the **Reader's default colors** radio button will be selected. To set your own colors check the **Use custom colors** radio button. By clicking on the square beside **Normal text** you can set the default color for all text on your webpage. When you click the button a window like this will pop up:



You can chose your color numerous ways:

- Use the rainbow square
- Select one of the predefined color tiles
- Enter a value between 0-255 for each of Red, Green and Blue. The higher the value, the more of that color will appear in your custom color.
- Enter one of XHTMLs 16 predefined color names into the **Name** text box.

#### Predefined Colors

aqua	black	blue	fuscia
gray	green	lime	maroon
navy	olive	purple	red
silver	teal	white	yellow

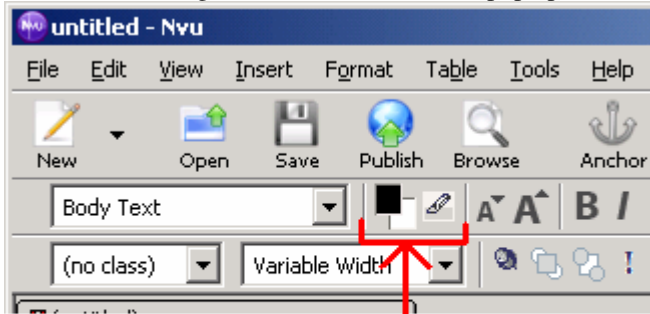
- Enter a Hexadecimal value for the color into the **Hex** text box
  - Hexadecimal colors consist of 6 digits: the first two represent the amount of red, the second two represent the amount of green, and the last two represent the amount of blue
  - To further complicate things, hexadecimal digits range from 0-F
  - So, for example, a hexadecimal value for red is #FF0000. This means full red, no green and no blue. The color #FF00FF would be a mix of full red, no green and full blue, resulting in purple
  - [Click here for a Hexadecimal Color Chart](#)

You can set the background color for your page by clicking on the square beside **Background**. If you wish to use an image as the background of your webpage, click on the **Choose File...** button and locate the image

file you wish to use. Note that selecting a background image will override any background color you have chosen.

You can also set the background color for specific elements on your page, whether it be a table cell, a paragraph, a list item, one word, etc.

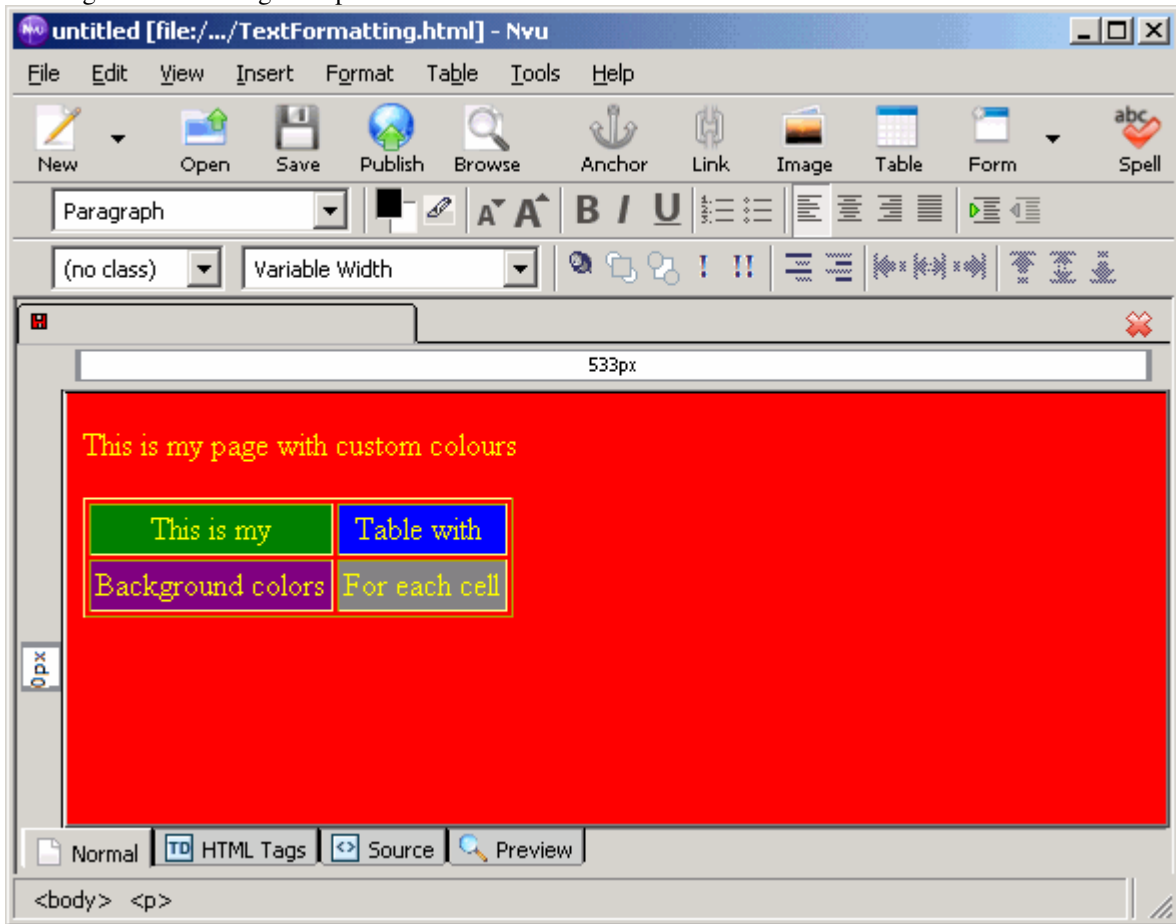
1. To do this, first select the element
2. Next, click on the bottom of the two color tiles in the formatting bar (in this picture it is the white one) and the Background Color window will pop up



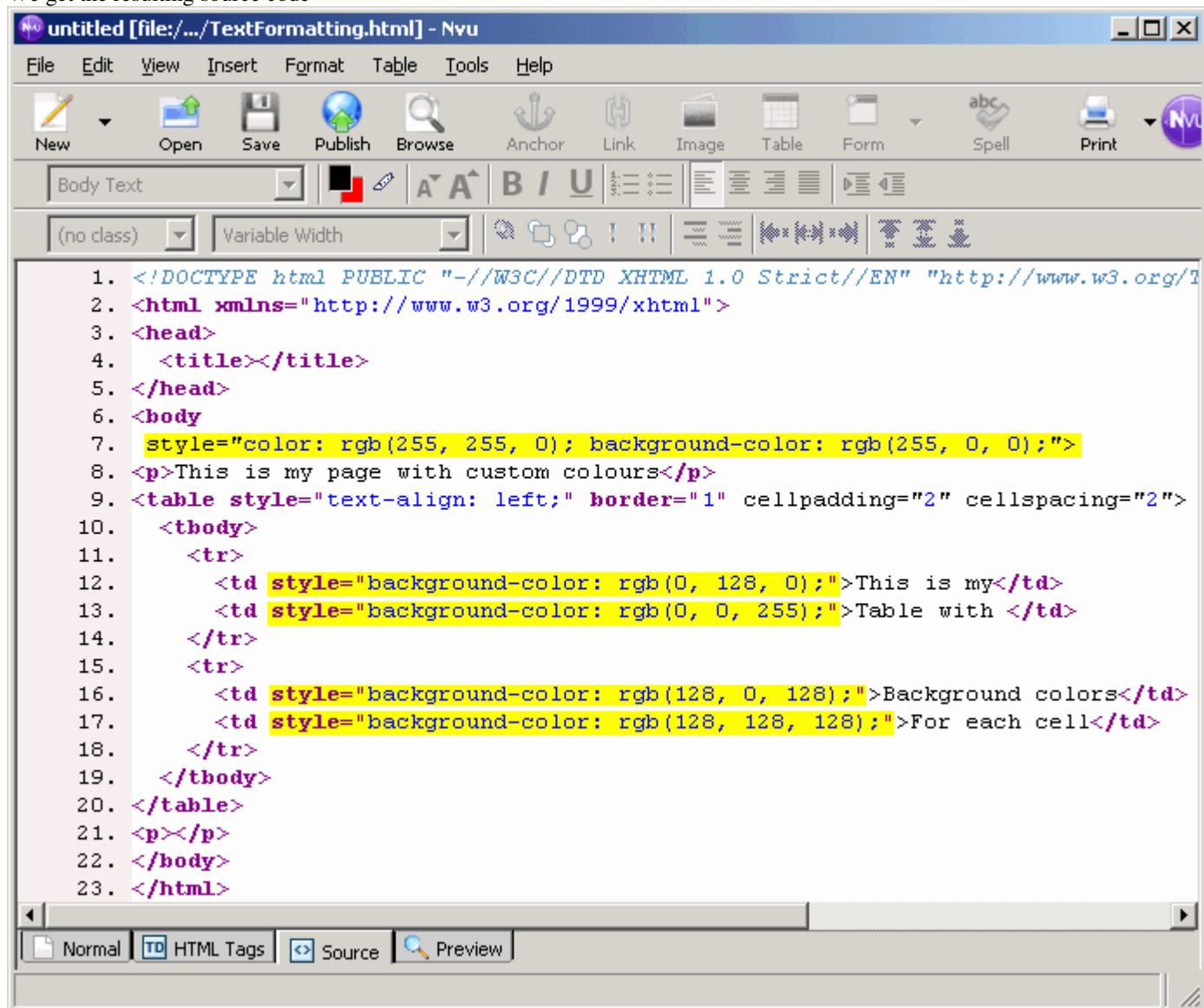
**Color Tiles**

3. Select the color you desire from the color window and click OK

Looking at the following example



We get the resulting source code



```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/1
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <title></title>
5. </head>
6. <body
7.   style="color: rgb(255, 255, 0); background-color: rgb(255, 0, 0);">
8. <p>This is my page with custom colours</p>
9. <table style="text-align: left;" border="1" cellpadding="2" cellspacing="2">
10.  <tbody>
11.   <tr>
12.     <td style="background-color: rgb(0, 128, 0);">This is my</td>
13.     <td style="background-color: rgb(0, 0, 255);">Table with</td>
14.   </tr>
15.   <tr>
16.     <td style="background-color: rgb(128, 0, 128);">Background colors</td>
17.     <td style="background-color: rgb(128, 128, 128);">For each cell</td>
18.   </tr>
19. </tbody>
20. </table>
21. <p></p>
22. </body>
23. </html>
```

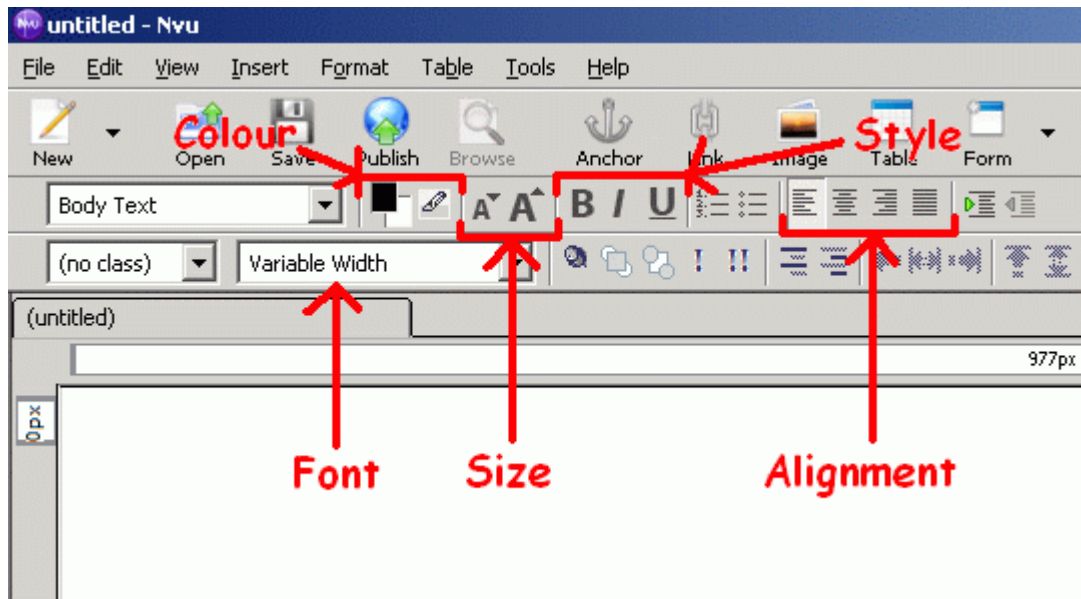
First, to set the colors for the entire page, Nvu places the **style** attribute in the opening body tag. This causes the style rules to be applied to everything on our webpage, since everything on our webpage is in our `<body></body>` tags. For the **property** Nvu sets the **background-color**, which sets the color for the page's background, and the **color**, which sets the defaults text color. The **rgb(#, #, #)** is the value of the color. "rgb" stands for "Red, Green, Blue". And the numbers represent the amount of those colors, respectively.

Next we see that in each opening `<td>` tag we have style attributes that define the background color for each individual cell. By placing the style attribute in the `td` element, the style gets applied to just that element.

### Changing Text Colour, Style, Size and Font

To change the formatting of your text, whether it be a heading or paragraph, you first need to select the text you want to format. Once you have selected the text you wish to edit, formatting is like it would be for a

word processing document. Nvu's Formatting Bar looks like this



You can choose your font using the drop down menu that by default says "Variable Width" in the formatting bar.

To edit the style you can select any of the formatting options from the formatting bar **B**, *I*, U (bold, italic, underline).

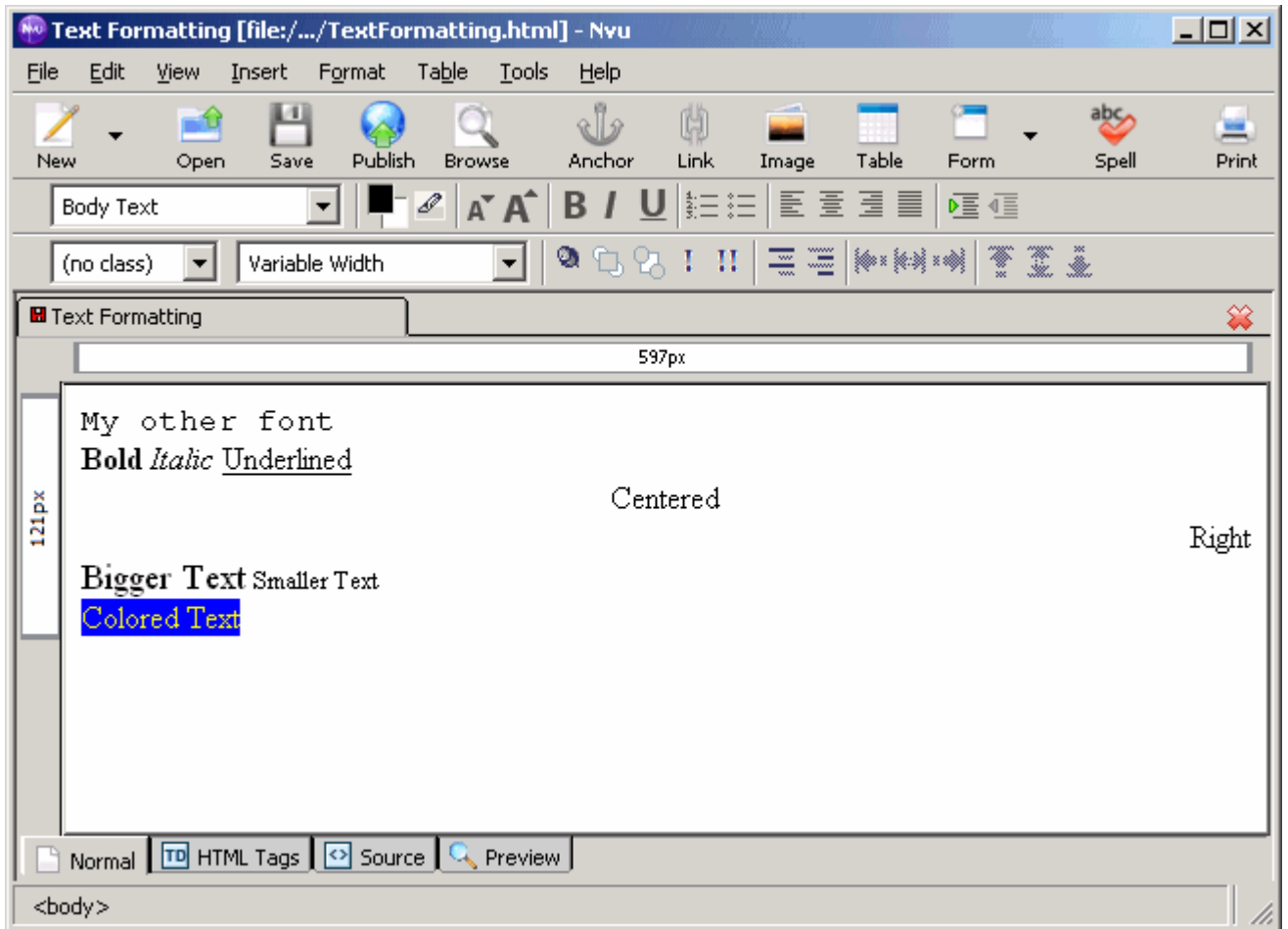
You can change the colour by clicking on the colour tiles. By clicking on the top tile you can choose the colour for the text. Clicking on the tile behind that, you can choose the colour for the background behind the text. The icon beside the two tiles lets you select a highlight colour for the text. Colours are chosen exactly like you do for the page background.

To increase or decrease the size of your text, click on the large or small A buttons in the formatting bar.

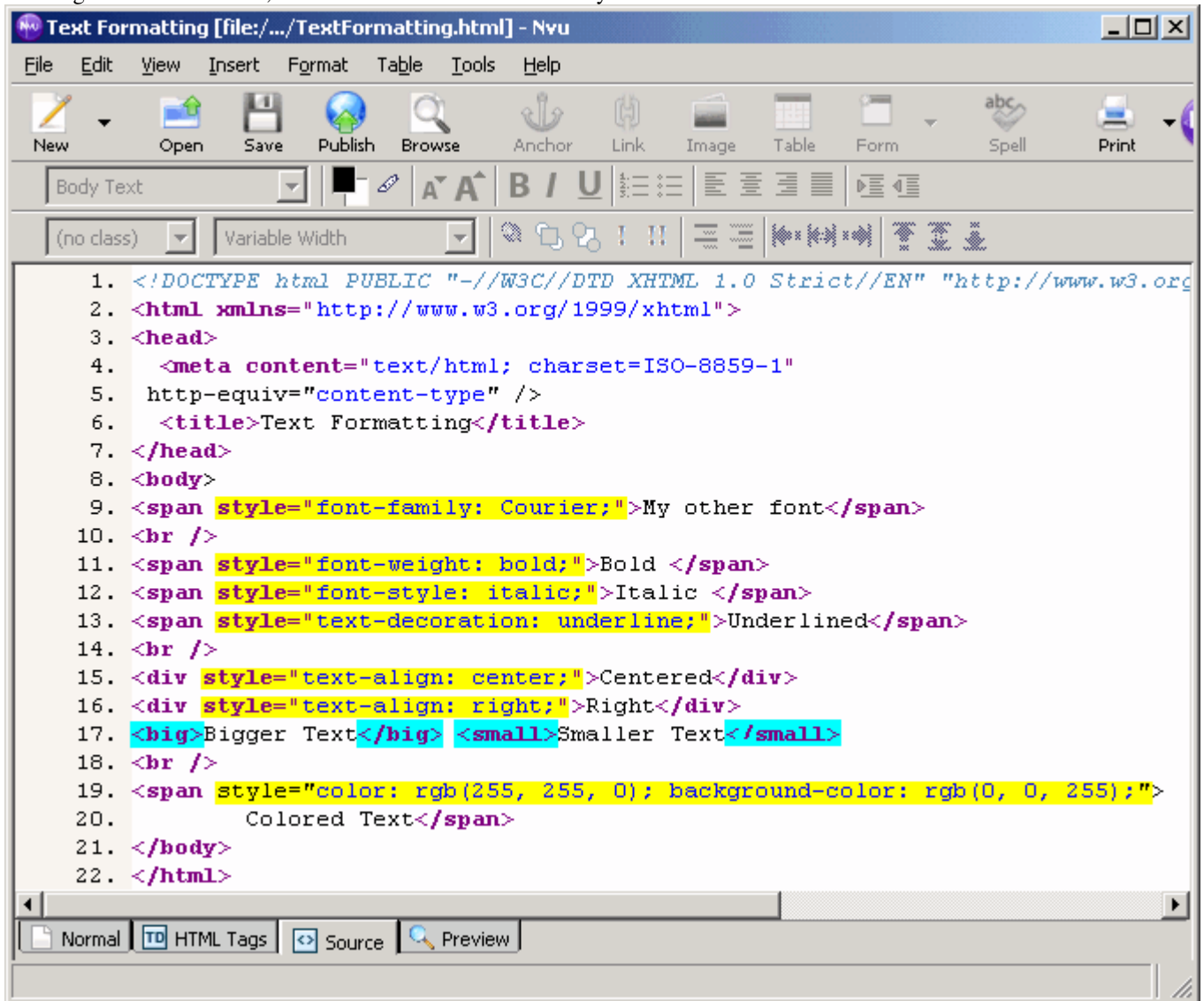
To align you text, pictures and even tables, select the content you wish to align and choose the alignment (left, center, right, justified) just like in MS Word.



Here is an example of common text formatting



Looking at this source code, we can see how Nvu inserts the style into our XHTML code



```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" />
6.   <title>Text Formatting</title>
7. </head>
8. <body>
9.   <span style="font-family: Courier;">My other font</span>
10. <br />
11.   <span style="font-weight: bold;">Bold </span>
12.   <span style="font-style: italic;">Italic </span>
13.   <span style="text-decoration: underline;">Underlined</span>
14. <br />
15.   <div style="text-align: center;">Centered</div>
16.   <div style="text-align: right;">Right</div>
17.   <big>Bigger Text</big> <small>Smaller Text</small>
18. <br />
19.   <span style="color: rgb(255, 255, 0); background-color: rgb(0, 0, 255);">
20.     Colored Text</span>
21. </body>
22. </html>
```

In almost all the text formatting examples `<span></span>` tags are used. The span tags are useful in selecting the text you wish to format and then applying the formatting by using the **style** attribute within them. As you can see the value for the style attribute is a CSS Rule. This rule is only applied to whatever is in between the span tags. We can clearly see this in the case of the colored text, where the background color is set for only the area where that text occurs.

It is also important to notice that all these rules are **Inline Style Rules** as they all use the style attribute within the opening tag of the element. Remember that inline style rules are only applied to the element they occur in.

In our source code we also see `<big></big>` and `<small></small>` tags. These tags are inserted when we click on the large or small A button in the button bar. The text in between these tags is made relatively bigger or smaller, depending on which tags surround it.

Under the **Format** -> option, you can access the same formatting options as the buttons. There are also more options for the text style, such as "superscript", "strikethrough", "code", just to name a few. There is

also the option to **Remove/Discontinue Text Styles**. This is helpful if you set the formatting for a section of text and decide you want to restore it to its default.

## CSS Reference

This reference combines into one place all of the style summaries in each of the lessons. You can create most of these styles by using Nvu's CSS editor. However, this page is included as a reference if you want to start editing the source code, or if you want to see how many styles are available to use in CSS.

## Quick navigation

[Heading, paragraph, html styles](#)

[Hyperlink styles](#)

[Horizontal rule styles](#)

[Image styles](#)

[List styles](#)

[Ordered list styles](#)

[Table styles](#)

[Input element styles](#)

[More CSS references](#)

## CSS Syntax

```
<link rel="stylesheet" href="sitestyles.css" type="text/css" />
```

Linked stylesheet

```
<style type="text/css">
  selector, selector
  {
    property: value;
  }
```

Global style

```
</style>
<style type="text/css">
  selector.className
  {
    property: value;
  }
```

Global style, with a class

```
</style>
...
<element class="className"></element>
<element style="property:value"></element>
```

Inline style

## Heading, paragraph, html styles

Property	Values	Explanation	In Nvu
		<b>Foreground and Background Colors</b>	
color	[red, blue, green...]	Sets the foreground (font) color.	Click on <b>Format -&gt; Page Colors and Background</b> to set colors for the entire page.
	#FF0000	Can be any named color value or any RGB hexadecimal value.	
background-color	[red, blue, green...]	Sets the background color.	For a specific element, select what you wish to change and use the color tiles in the formatting bar.
	#FF0000	Can be any named color value or any RGB hexadecimal value.	

## Font Properties

		Sets the type of font used.	
font-family	[arial, courier, helvetica,...]	Can be any named font (the font must be already installed on the viewing computer).	Highlight the text and select the font from the <b>Variable Width</b> select list in the formatting bar
	[serif, sans-serif, cursive, fantasy, monospace]	Can be a "category" of fonts like serif or cursive. The browser will pick the first installed font it finds on the viewing computer that falls into this particular category.	
font-style	[normal, italic]	Sets the style of font used.	Highlight the text and click the italic icon in the formatting bar
		Can be normal or italics. Sets the weight/thickness of the font used.	
font-weight	[normal, bold, bolder, light, lighter]	Can be set to predefined thicknesses like "normal" or "light".	Highlight the text and click the bold icon in the formatting bar
	[100 - 900]	Can be set to a specific thickness from 100 to 900, with 100 being the lightest and 900 being the thickest. Sets the size of the font.	
font-size	[x-small, small, medium, large, x-large]	Can be set to predefined sizes like "small" or "large".	Highlight the text and click the small A or large A icon in the formatting bar to make the text relatively smaller or bigger.
	8pt	Can be set to a specific point size. 12pt is a normal font size for written documents. Larger numbers mean a larger font. Must include pt after the number (no space between the number and the pt).	To set a specific font size, use the CSS Editor
<b>Margins, Borders, and Padding</b>			
Sets the width of the margins around an element. This sets all four margins together.			
margin	10px	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).	
	5%		
margin-top		Can be set as a percentage of the page (so the margins adjust if you make your window bigger/smaller). Sets only the width of only the specified margin (top, bottom, left, right) of an element.	
margin-bottom	10px		
margin-left	5%	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).	
margin-right		Can be set as a percentage of the page.	
border-style	[none, dotted, dashed, solid,	Creates a border around an element.	

	double, groove, ridge, inset, outset]	Can be set to predefined styles like "solid" or "dashed".
		Must set a border-style before you can apply a border-color or border-width.
border-top-style		Creates a border along the specified edge (top, bottom, left, right) of the element.
border-bottom-style	[none, dotted, dashed, solid, double, groove, ridge, inset, outset]	Can be set to predefined styles like "solid" or "dashed".
border-left-style		Must set a border-style before you can apply a border-color or border-width.
border-right-style		Sets the width of the border.
		Can be set to predefined thicknesses.
border-width	[thin, medium, thick]  10px	Can be set to a specified number of pixels. Must include px after the number (no space between the number and the px).
		Must set a border-style before you can apply a border-color or border-width.
border-top-width		Sets the width of the border for a specified edge (top, bottom, left, right).
border-bottom-width	[thin, medium, thick]	Can be set to predefined thicknesses.
border-left-width	10px	Can be set to a specified number of pixels. Must include px after the number (no space between the number and the px).
border-right-width		Must set a border-style before you can apply a border-color or border-width. Sets the color of the element border.
border-color	[red, blue, green...]  #FF0000	Can be any named color value or any RGB hexadecimal value.
		Must set a border-style before you can apply a border-color or border-width.
border-top-color		Sets the color of the element border for a specified edge (top, bottom, left, right).
border-bottom-color	[red, blue, green...]  #FF0000	Can be any named color value or any RGB hexadecimal value.
border-left-		Must set a border-style before you can apply a border-color or border-width.

color		
border-right-color		Sets the amount of whitespace between the border and the contained element.
padding	10px	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
padding-top		Sets the amount of whitespace between the border for a specified edge and the contained element.
padding-bottom	10px	
padding-left		Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
padding-right		

### Indenting and Alignment

		Sets the size of the indentation for the first line of text in an element.
text-indent	10px 5%	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
		Can be set as a percentage of the page. Sets the alignment for an element.
text-align	[left, right, center, justify]	Can be set to predefined alignments like "left" or "center".

### Hyperlink styles

Property	Values	Explanation
<b>Foreground and Background Colors</b>		
color	[red, blue, green...]	Sets the foreground (font) color.
background-color	[red, blue, green...]	Sets the background color.
	#FF0000	Can be any named color value or any RGB hexadecimal value.
	#FF0000	Can be any named color value or any RGB hexadecimal value.
<b>Font Properties</b>		
		Sets the type of font used.
font-family	[arial, courier, helvetica,...]	Can be any named font (the font must be already installed on the viewing computer).
	[serif, sans-serif, cursive, fantasy, monospace]	Can be a "category" of fonts like serif or cursive. The browser will pick the first corresponding font it finds installed on the viewing computer.
font-style	[normal, italic]	Sets the style of font used.

		Can be normal or italics. Sets the weight/thickness of the font used.
font-weight	[normal, bold, bolder, light, lighter]	Can be set to predefined thicknesses like "normal" or "light".
	[100 - 900]	Can be set to a specific thickness from 100 to 900, with 100 being the lightest and 900 being the thickest. Sets the size of the font.
font-size	[x-small, small, medium, large, x-large]	Can be set to predefined sizes like "small" or "large".
	8pt	Can be set to a specific point size. 12pt is a normal font size on written documents. Larger numbers mean a larger font. Must include pt after the number (no space between the number and the pt).

### Horizontal Rule Styles

Property	Values	Explanation
background-color	[red, blue, green...] #FF0000	Sets the background color. Can be any named color value or any RGB hexadecimal value.
width	10px 5%	Sets how long the line is horizontally. Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
height	10px	Can be set as a percentage of the page. Sets how thick the line is vertically. Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).

### Image styles

Dimensions		
Property	Values	Explanation
height	20px	Sets the height of an element in pixels.
	10%	Sets the height of an element by % of space available.
width	20px	Sets the width of an element in pixels.
	10%	Sets the width of an element by % of space available.
Margins, Borders, and Padding.		
margin	10px	Sets the width of the margins around an element. This sets all four margins together.
	5%	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).

		Can be set as a percentage of the page (so the margins adjust if you make your window bigger/smaller).
margin-top margin-bottom margin-left margin-right	10px 5%	Sets only the width of only the specified margin (top, bottom, left, right) of an element.  Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).  Can be set as a percentage of the page.
border-style	[none, dotted, dashed, solid, double, groove, ridge, inset, outset]	Creates a border around an element.  Can be set to predefined styles like "solid" or "dashed".  Must set a border-style before you can apply a border-color or border-width.
border-top-style border-bottom-style border-left-style border-right-style	[none, dotted, dashed, solid, double, groove, ridge, inset, outset]	Creates a border along the specified edge (top, bottom, left, right) of the element.  Can be set to predefined styles like "solid" or "dashed".  Must set a border-style before you can apply a border-color or border-width.
border-width	[thin, medium, thick]	Sets the width of the border.  Can be set to predefined thicknesses.  Must set a border-style before you can apply a border-color or border-width.
border-top-width border-bottom-width border-left-width border-right-width	[thin, medium, thick] 10px	Sets the width of the border for a specified edge (top, bottom, left, right).  Can be set to predefined thicknesses.  Can be set to a specified number of pixels. Must include px after the number (no space between the number and the px).  Must set a border-style before you can apply a border-color or border-width.
border-color	[red, blue, green...] #FF0000	Sets the color of the element border.  Can be any named color value or any RGB hexadecimal value.  Must set a border-style before you can apply a border-color or



		border-width.
border-top-color border-bottom-color border-left-color border-right-color	[red, blue, green...]  #FF0000	Sets the color of the element border for a specified edge (top, bottom, left, right).  Can be any named color value or any RGB hexadecimal value.  Must set a border-style before you can apply a border-color or border-width.
padding	10px	Sets the amount of whitespace between the border and the contained element.  Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
padding-top padding-bottom padding-left padding-right	10px	Sets the amount of whitespace between the border for a specified edge and the contained element.  Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
<b>Alignment</b>		
*text-align	[left, right, center, justify]	Sets the alignment for an element.  *Applies to a paragraph surrounding the image, not to the image directly.  Can be set to predefined alignments like "left" or "center".
float	[left, right, none]	Sets the position of an image. Text will wrap around images using the float property.
* clear	[left, right, both, none]	*Applies to an element after a floating image (like a paragraph). It will force the area next to the image to remain clear and move the current element below the image.

### List Styles

Property	Values	Explanation
list-style	[disc, square, circle]	Sets the type of bullet used.
list-style-image	url("image.gif")	Sets the type of bullet to the image indicated by "image.gif". Note that this can be any image file, but it must be located in the same folder as your XHTML file.

## Ordered Lists

The type of enumeration for an ordered list can be set using the property described here:

Property	Values	Explanation
list-style	[decimal, decimal-leading-zero, upper-roman, lower-roman, upper-alpha, lower-alpha, upper-latin, lower-latin, hebrew]	Sets the type of enumeration used.

## Table styles

Property	Values	Applies to	Explanation
			<b>Table Width</b>
width	100%	<table>	Alters the horizontal size of the table.
			<b>Foreground and Background Colors</b>
			Sets the foreground (font) color. Will affect fonts and table borders.
color	[red, blue, green...] #FF0000	<table>, <tr>, <td>,<th>	Can be any named color value or any RGB hexadecimal value.
			Sets the background color.
background-color	[red, blue, green...] #FF0000	<table>, <tr>, <td>,<th>	Can be any named color value or any RGB hexadecimal value.
			<b>Font Properties</b>
			Sets the type of font used.
font-family	[arial, courier, helvetica,...] [serif, sans-serif, cursive, fantasy, monospace]	<table>, <tr>, <td>,<th>	Can be any named font (the font must be already installed on the viewing computer).
			Can be a "category" of fonts like serif or cursive. The browser will pick the first installed font it finds on the viewing computer that falls into this particular category.
font-style	[normal, italic]	<table>, <tr>, <td>,<th>	Sets the style of font used.
			Can be normal or italics.
font-weight	[normal, bold, bolder, light, lighter]	<table>, <tr>, <td>,<th>	Sets the weight/thickness of the font used.
			Can be set to predefined thicknesses like "normal" or "light".
	[100 - 900]		Can be set to a specific thickness from 100 to 900, with 100 being the lightest and 900 being the thickest.
			Sets the size of the font.
font-size	[x-small, small, medium, large, x-large] 8pt	<table>, <tr>, <td>,<th>	Can be set to predefined sizes like "small" or "large".
			Can be set to a specific point size. 12pt is a normal font size for written documents. Larger numbers mean a larger font. Must include pt after the number (no space between the number and the pt).
			<b>Margins and Padding</b>
margin	10px	<table>	Sets the width of the margins around a table. This sets all four margins together.

	5%		Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
	auto		Can be set as a percentage of the page (so the margins adjust if you make your window bigger/smaller).
margin-top			Can be set to "auto", the table will be centered horizontally by the browser. Sets only the width of only the specified margin (top, bottom, left, right) of a table.
margin-bottom	10px		Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
margin-left	5%	<table>	Can be set as a percentage of the page.
margin-right	auto		Can be set to "auto", the browser will automatically assign a value, usually attempting to center the element. Sets the amount of whitespace within table cells.
padding	10px	<table>, <td>, <th>	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
padding-top			
padding-bottom			Sets the amount of whitespace for a specified edge of the table cell.
padding-left	10px	<table>, <td>, <th>	Can be set to a certain number of pixels. Must include px after the number (no space between the number and the px).
padding-right			
vertical-align	[top, bottom, middle]	<table>, <tr>, <td>, <th>	<b>Alignment</b> Sets the vertical alignment for the contents of table cells. Can be set to predefined values such as "top".
text-align	[left, right, center, justify]	<table>, <tr>, <td>, <th>	Sets the alignment for the contents of the table cells. Can be set to predefined alignments like "left" or "center".

### Input element styles

Property	Values	Applies to	Explanation
<b>Foreground and Background Colors</b>			
color	[red, blue, green...]	<select>, <textarea>, <option>	Sets the foreground (font) color. Will affect fonts and table borders.
	#FF0000		Can be any named color value or any RGB hexadecimal value.
background-color	[red, blue, green...]	<select>, <textarea>, <option>	Sets the background color.
	#FF0000		Can be any named color value or any RGB hexadecimal value.
background-	url(name.gif)	<select>, <td>, <th>	Sets the background image for an element.

image	url(name.jpg)	<textarea>, <option>	Must be the name of an image file in the same directory.
<b>Font Properties</b>			
	[arial, courier, helvetica,...]		Sets the type of font used.
font-family	[serif, sans-serif, cursive, fantasy, monospace]	<select>, <textarea>, <option>	Can be any named font (the font must be already installed on the viewing computer).
			Can be a "category" of fonts like serif or cursive. The browser will pick the first installed font it finds on the viewing computer that falls into this particular category.
font-style	[normal, italic]	<select>, <textarea>, <option>	Sets the style of font used.
			Can be normal or italics.
			Sets the weight/thickness of the font used.
font-weight	[normal, bold, bolder, light, lighter]	<select>, <textarea>, <option>	Can be set to predefined thicknesses like "normal" or "light".
	[100 - 900]		Can be set to a specific thickness from 100 to 900, with 100 being the lightest and 900 being the thickest.
			Sets the size of the font.
	[x-small, small, medium, large, x-large]	<select>, <textarea>, <option>	Can be set to predefined sizes like "small" or "large".
font-size	8pt		Can be set to a specific point size. 12pt is a normal font size for written documents. Larger numbers mean a larger font. Must include pt after the number (no space between the number and the pt).

### Additional CSS References

- [W3 Schools CSS Tutorial](#)
  - This tutorial covers styles and formatting using CSS. Some knowledge of XHTML is assumed.
- [CSS Property Index](#)
  - An alphabetical listing of CSS properties (even more than we cover in this class). Helpful reference. Even shows cross-browser compatibility.

### Conclusion

In this lesson we learned how to use Nvu's formatting bar to perform such actions as align our text, change its color, size and font. We also saw how to change the background to a color or image to give a webpage a unique look and feel. Finally we saw how to use Nvu's CSS Editor to create advanced formatting techniques not available in Nvu's formatting bar.

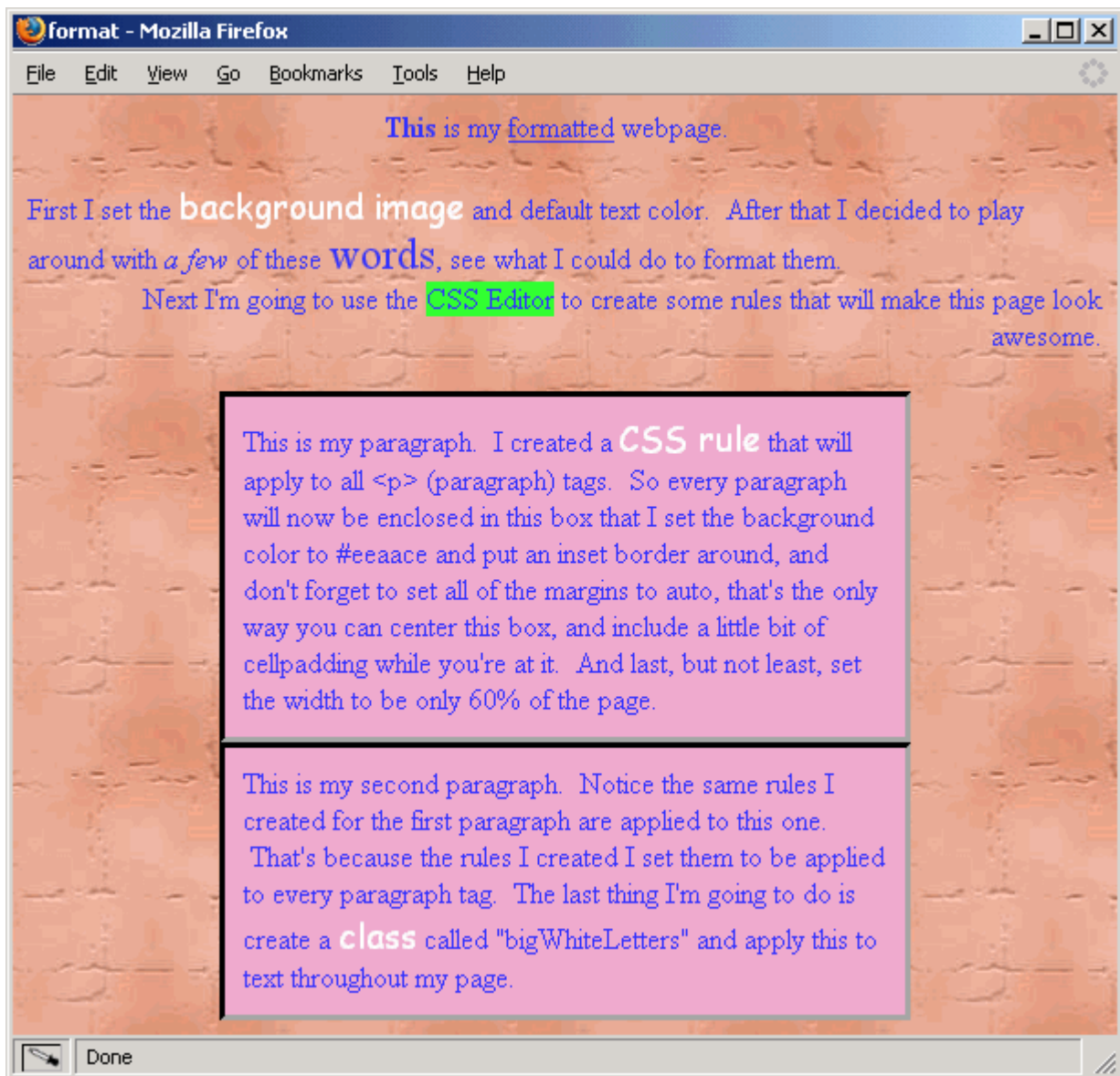
### Exercises

1. Define all the keywords in this lesson
2. Create the page below

- o Use the following image as the background



(This image is freely available at <http://www.nutrocker.co.uk/freetiles.html>)



## Basic XHTML: My Scuba Adventure- A simple web page

This tutorial involves the construction of a simple website using basic XHTML and CSS. This page will be about your new favorite hobby - scuba diving!

### Video Clips of the Tutorial

Note that it might take a few minutes for the video clips to load.

- [Clip 1](#)
- [Clip 2](#)
- [Clip 3](#)
- [Clip 4](#)

### Text Version of the Tutorial

#### *Basic Page Layout*

1. Save ONE of the following three images to your computer. You may want to create a folder especially for this tutorial to keep all of the related files together. You can save the image by right-clicking the image and choosing "save image as". Name the image "**scuba.jpg**" (you may need to change the default name that shows up).





These images are freely available at: <http://gimp-savvy.com/PHOTO-ARCHIVE/> and credit for the photos goes to the US National Oceanic and Atmospheric Administration

2. Open up Nvu and create a new page by clicking **File -> New**
  - Make sure you make it XHTML Strict
  - Click the save icon
  - Title the page "Scuba Diving"
  - Be sure to save your page to the same folder as you saved the picture above and name your file **scuba.html**
3. Add a heading to your web page. Headings appear within the content of your web page.
  - You should save your changes and look at your page through a browser to see if it looks as expected. You should get into the habit of doing this frequently. It is much easier to find and correct errors this way since you're only adding a few lines of code at a time.
4. Before moving too far into this page, we are going to save a copy of it to use as the second page of our website. Saving a copy at this point will save us from having to create a new page again.
  - In Nvu, click on File -> Save As
  - Save your file as "page2.html"
  - Change your heading to say "My Scuba Adventure - Page 2"
  - Save the changes.

- Put "page2.html" aside for now by closing the page2.html tab in Nvu. We will return to it later in the tutorial.
  - Re-open your "scuba.html" file in Nvu.
  - Continue working on scuba.xhtml
5. In "scuba.html", add a paragraph about how you started scuba diving.
  6. Insert the image of the scuba diver. The image file is called "scuba.jpg".
    - The images and your xhtml files must be stored in the same folder, otherwise your image will not appear
    - The filename must match exactly, including uppercase/lowercase combinations
    - You must always include Alternate Text for your images, for those using text-only versions of the webpage.
  7. Create a sub-heading of size h2 for the next section of your page - the details of your first scuba diving trip. Add a paragraph describing your adventure.
  8. Add a few blank lines and a horizontal rule to your page.
  9. Create a third section to your page - a set of interesting links.
    - Add a heading of size h2 -- Links
    - Add a paragraph introducing the links
    - Create a relative link to your second page - page2.html
      - Remember, your second page must be saved in the same folder as your current page
    - Create an absolute link to the PADI website (www.padi.com) - for those who want to learn to scuba
      - Always include http:// at the start of the URL for a website, this tells the browser that you are looking for a website, not just a page on your computer.
    - Create an internal link back to the top of your current page
      - You must name insert an anchor at the top of your page. This gives you something to reference when you create the hyperlink.
      - Create your hyperlink, including a # in front of the location name to signify to the browser that this location is on the current page.
  10. Make your scuba.jpg image a hyperlink to your "page2.xhtml" as well.

### ***Adding Styles and Formatting***

1. Set the background color for the entire webpage. We'll use a light blue color, that has the hexadecimal code #99cccc.
  - You can find color codes by searching on the web for RGB color charts
2. Center the top half of your page, up until the horizontal rule.
  - Select everything you want to be centered and click the center icon in the formatting bar.
3. Format the headings from the top section. Both the h1 and the h2 heading will get the same formatting instructions, so we'll define a class that can be assigned to both. Format both the font and the color of the headings.
  - Use Nvu's CSS Editor to create a class .heading that will set the font to arial, italicize the font and change the font color to #336666
  - Apply this class to the h1 and the h2 headings ("My Scuba Adventure" and "My First Scuba Trip")
4. Format the "My First Scuba Trip" paragraph and the "Links" heading so that they appear to be contained in white boxes. Since several elements will be using the same formatting details, we'll use the CSS Editor to create a class and the CSS Rule.
  - Create an inset border around the box and set the background color to white.
  - Add some padding inside the box so that the words aren't stuck to the edges.
  - Set the margins of the box so that it is indented from the edges of the page.
  - Apply this class to the "My First Scuba Trip" paragraph and to the "Links" heading.
5. Once again using the CSS Editor, create a rule for when the user moves their mouse over one of the hyperlinks. Set the background of the link to black and the font to white.



- Notice the formatting of the selector is different (includes a colon), a:hover, because you are specifying formatting for a particular state of the hyperlink.
  - We do not need to apply the rule to anything on the page. It will automatically be applied to every hyperlink on the page.
6. Format some words in your second paragraph so that they appear larger and in bold.
  7. Validate your webpage to make sure it contains correct code by clicking on **Tools -> Validate HTML**. Make sure you have saved your changes first.

*By the end of this tutorial, you should be familiar with:*

- Creating and saving XHTML files
- HTML terminology - tags, opening/closing tags, elements, head, body, title vs. headings, CSS, styles
- Using comment tags
- Creating headings
- Creating paragraphs
- Creating horizontal rules and breaks
- Inserting images
- Creating hyperlinks (relative, absolute, internal)
- Creating CSS Rules
- Using classes to define CSS rules

## **Nvu - Creating Webpages**

In this tutorial we will work with headings, tables, and lists to create a webpage of attractive vacation destinations.

### **Video Clips of the Tutorial**

Note that it might take a few minutes for the video clips to load.

- [Clip 1](#)
- [Clip 2](#)
- [Clip 3](#)
- [Clip 4](#)
- [Clip 5](#)

### ***Text Version of the Tutorial***

1. This is the webpage we will be creating:



2. Save the following 4 images to your computer. You may want to create a folder especially for this tutorial to keep all of the related files together. You can save the image by right-clicking the image and choosing "save image as". These images are freely available in the Public Domain at <http://pdphoto.org>





3. Open up Nvu and create a new page
  - Make sure you make it XHTML Strict
  - Click the save icon
  - Give the page an appropriate title when you are prompted
  - Be sure to save your page to the same folder as you saved the pictures above and name your file **vacation.html**
4. Add a heading of size 1 to your page - "Wonderful Vacations"
  - Align it to be centered
  - Open up the CSS Editor (under Tools) to create a style rule
  - Make sure to uncheck the "Expert Mode" checkbox
  - Add a rule by clicking on the **Rule**
  - Name this rule "yellowheadings"
  - Select your .yellowheadings rule and make the colour for this rule yellow
  - Apply the class .yellowheadings to your heading
5. In the Format -> Page Colors and Background, set the page background to the **bground1.gif** image we saved and set the default text colour to white
6. Create a table that is 5 rows X 3 columns and apply the following formatting to the table:
  - The width should be 95% of the page
  - The border should be size 5 and the cellpadding 0
  - Set the horizontal and vertical alignment within the cells to center and middle, respectively
  - Center the table
  - Set the background color of the table to be #5f9ea0
7. **Row 1:** In the first cell, create a heading of size 2 - "Vacation Packages"
  - Apply the class ".yellowheadings" to your heading
  - Set the cell to merge with the other 2 cells in Row 1 so that it spans across the entire table
8. **Row 2:** Contains Headings and images for each of the three vacation packages
  - Cell 1: add a heading of size 2 ("Arctic Adventure") with the font set to arial and color to navy. Add the image glacier.jpg
  - Cell 2: add a heading of size 2 ("Awesome Autumn") with the font set to courier and the color to orange. Add the image forest.jpg
  - Cell 3: add a heading of size 2 ("Space Spectacular") with the font set to times and the color to black
  - To make sure that each cell takes up 1/3 of the table, create a rule in the CSS Editor called "onethird". In this rule, set the width to 33%. Apply this class to the three cells in row 2.
9. **Row 3:** Contains a description of each of the three packages
  - Cell 1: Contains a nested table of 2 rows and 2 columns

- create the outline for the table
  - set the border of the table to 0
  - set the width of the table to 75%
  - set the background colour to #4682b4
  - center the table within the cell
  - first row, cell 1: "Features"
  - first row, cell 2: "cold water", "wonderful view", "peaceful" with line breaks in between
  - second row, cell 1: spans 2 columns and larger font, "New Low Price = \$1000"
  - second row, cell 2: deleted due to colspan
  - Cell 2: Contains an unordered list of features and a price for the vacation package
    - Create an unordered list with two items
    - Add the words "Only \$900" and make it a larger font
  - Cell 3: Contains an invisible table and an ordered list of features and price for the vacation package
    - Create a nested table with one row, one column. This is to center the list as a whole rather than individual list items
    - Center the table within the cell
    - Create an ordered list with three features and the price of the trip (in larger font)
10. **Row 4:** Contains availability dates for the vacation packages
- Cell 1: The Winter Dates apply for both the Arctic Adventure and Awesome Autumn packages, so this cell will span two columns
    - Type in "Winter Dates" and make it a larger font
    - Enter two dates for the packages
  - Cell 2: is deleted due to the colspan
  - Cell 3: This cell spans two rows. It contains the availability details for the Space Spectacular package, which are the same year-round.
11. **Row 5:** Contains availability dates for the vacation packages
- Cell 1: The Summer Dates apply for both the Arctic Adventure and Awesome Autumn packages, so this cell will span two columns
    - Type in "Summer Dates" and make it a larger font
    - Enter two dates for the packages
  - Cell 2: is deleted due to the colspan
  - Cell 3: Is deleted due to rowspan from row 4

*By the end of this tutorial, you should be familiar with:*

- Creating and saving webpages in Nvu
- Creating headings
- Inserting Images
- Creating tables
- Creating lists
- Creating nested tables
- Merging cells in table
- Formatting in Nvu