

**If you have not already done so,  
please download Aptana:**

**<http://aptana.com>**

# **HTML/CSS Class I: HTML Basics**

Slides:

**<http://bit.ly/GDIhtml1>**

**Alexis Goldstein**

@alexisgoldstein

[alexis@autfaciam.com](mailto:alexis@autfaciam.com)

# Introductions

- Before we begin, I'd like to learn a little bit more about everyone here!
- Can you please introduce yourself, and tell us why you're here and what you hope to learn.

# Goals for Today

We hope that by the end of class today:

- You will have mastered some of the basic terms & jargon.
- You will know the most common HTML tags.
- You will get some practice using the **Aptana** software
- You will have built a very simple, HTML-only webpage.
- **Time permitting:** You will understand the basics of web servers, and what they are used for.

# Ask me questions!

We are going to cover **LOTS** of content today

- If you missed something I just said, let me know and I'd be happy to repeat it.
- If anything isn't clear, tell me, and I will do my best to clarify.

# Background Concepts

# What is HTML?

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not actually a programming language! It's a markup language.

# What is a Markup Language?

- A system for annotating text.
- Comes from publishing industry: you *mark up* a manuscript prior to publishing.
- The revisions (mark up) editors make for the designers, so they know how to lay it out.
- They were traditionally done in blue pencil on author's manuscripts.
- Other markup languages you may have heard of: LaTeX, XML

# What is HTML?

- HTML is a markup language that describes webpages.
- It tells our browsers how to layout the page.
- It describes webpages using markup tags.
- We usually just refer to HTML's markup tags as “HTML tags”
- I like to think of HTML as one of the main languages of websites.



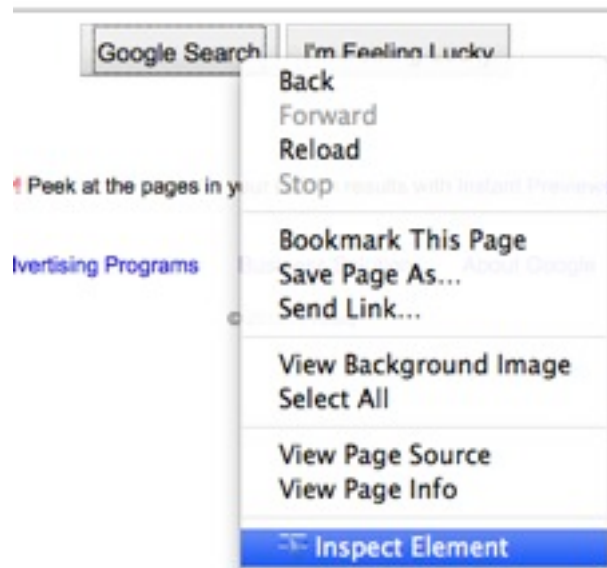
# What does HTML look like?

- Right-click on ANY website, and choose “View Source”
- You can see the HTML and CSS of every single website on the web!
- HTML/CSS are open platforms.

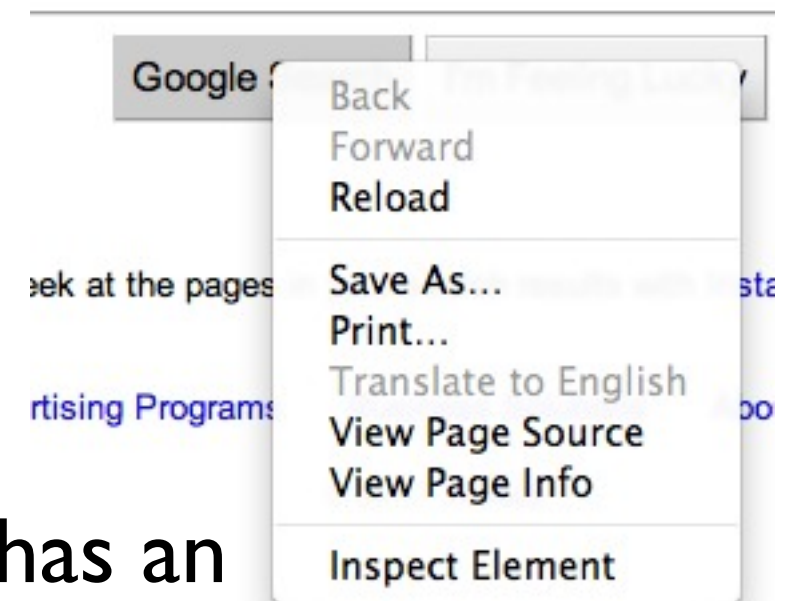
# Other Useful Tools

- Another great way to learn HTML is to inspect HTML elements on webpages you visit.
- There are two tools that can help you do this:

- The Chrome browser has a right-click (control-click on a mac!) and “Inspect Element” tool built in



- The Firefox browser has an extension called Firebug that also allows you to “Inspect Element”



# What does HTML look like?

```
<html>
```

```
  <body>
```

```
    <h1>My First Heading</h1>
```

```
    <p>My first paragraph.</p>
```

```
  </body>
```

```
</html>
```

Try this yourself! Go to: [http://w3schools.com/html/tryit.asp?filename=tryhtml\\_intro](http://w3schools.com/html/tryit.asp?filename=tryhtml_intro)

# HTML: A Brief History

- 1989: Tim Berners-Lee invents the Web with HTML as its publishing language
- Berners-Lee was working at CERN in Switzerland, a particle physics lab.
- Since particle physics often requires international collaboration, Berners-Lee wanted to create a way for researchers all over the world to share information easily.

# HTML: A Brief History

- The HTML that Berners-Lee created was based on SGML (Standard Generalized Mark-up Language)
- SGML was used to mark up text into structural units such as paragraphs, headings, and list items.
- HTML added something new: the hypertext link--what we've come to know of today as just "links"

# HTML: A Brief History

- 1994:
  - HTML 2 specification is released.
  - Netscape is formed. Begins adding to HTML without consulting international community.
  - World Wide Web Consortium (aka the w3 consortium) is formed to “fulfill the potential of the Web through the development of open standards.”

# HTML: A Brief History

- 1995:
  - HTML is extended with lots of new tags, including ones for formatting like BGCOLOR and FONT FACE.
    - “You’re not supposed to do that with HTML!”
  - HTML 3 is released.
  - Internet Explorer browser comes out.
  - Netscape submits a proposal for FRAMES.
- 1998: HTML 4
- 2010: HTML 5 in development, draft submitted.

# HTML: A Brief History

- To learn more:
  - <http://www.w3.org/People/Raggett/book4/ch02.html>
  - Steven Johnson's "Where Good Ideas Come From"



# HTML vs. CSS

- CSS stands for Cascading Style Sheets.
- We will cover CSS in detail in class 2.
- How does HTML fit in with CSS?
- CSS was created to allow the separation of **document content** from **document presentation**.

# HTML vs CSS

- HTML defines the content of a document:

**This is a HEADING**

- this is a new bullet!
- CSS defines the *formatting* and style of the content your website.
  - I am some blue text!
  - I am Monaco font!

# HTML/CSS and Browsers

- You can think of HTML and CSS as languages that web browsers speak
- Your Internet Explorer, Firefox, Chrome or Safari Browser reads the HTML and CSS on a webpage, and creates what you see.

# HTML Vocabulary

# HTML Vocabulary

- HTML Terms:
  - Tag
    - Opening Tag
    - Closing Tag
  - Element
  - Attribute

# HTML term: Tag

- HTML surrounds your text with what's called a “tag”
- Tags describe what the content is (is it a paragraph of text? A heading? A table? A list of items?)
- Tags are surrounded by angle brackets `<>`
  - The name of the tag goes in between the angle brackets: `<tag>`

# HTML term: Tag

- Tags usually come in pairs:
  - Starting tag: `<html>`
  - Ending tag: `</html>`
- Here's how you could create a paragraph of text in HTML, using the paragraph (`<p>`) tag:
  - `<p>Hello, world! This is my first paragraph of text</p>`

# Learning HTML tags

- You learn HTML one tag at a time.
- We are going to start by learning the following tags:
  - html, head, title
  - body
  - p, h1-h6, strong
- We will then move on to:
  - a, img
  - table



# HTML Vocabulary

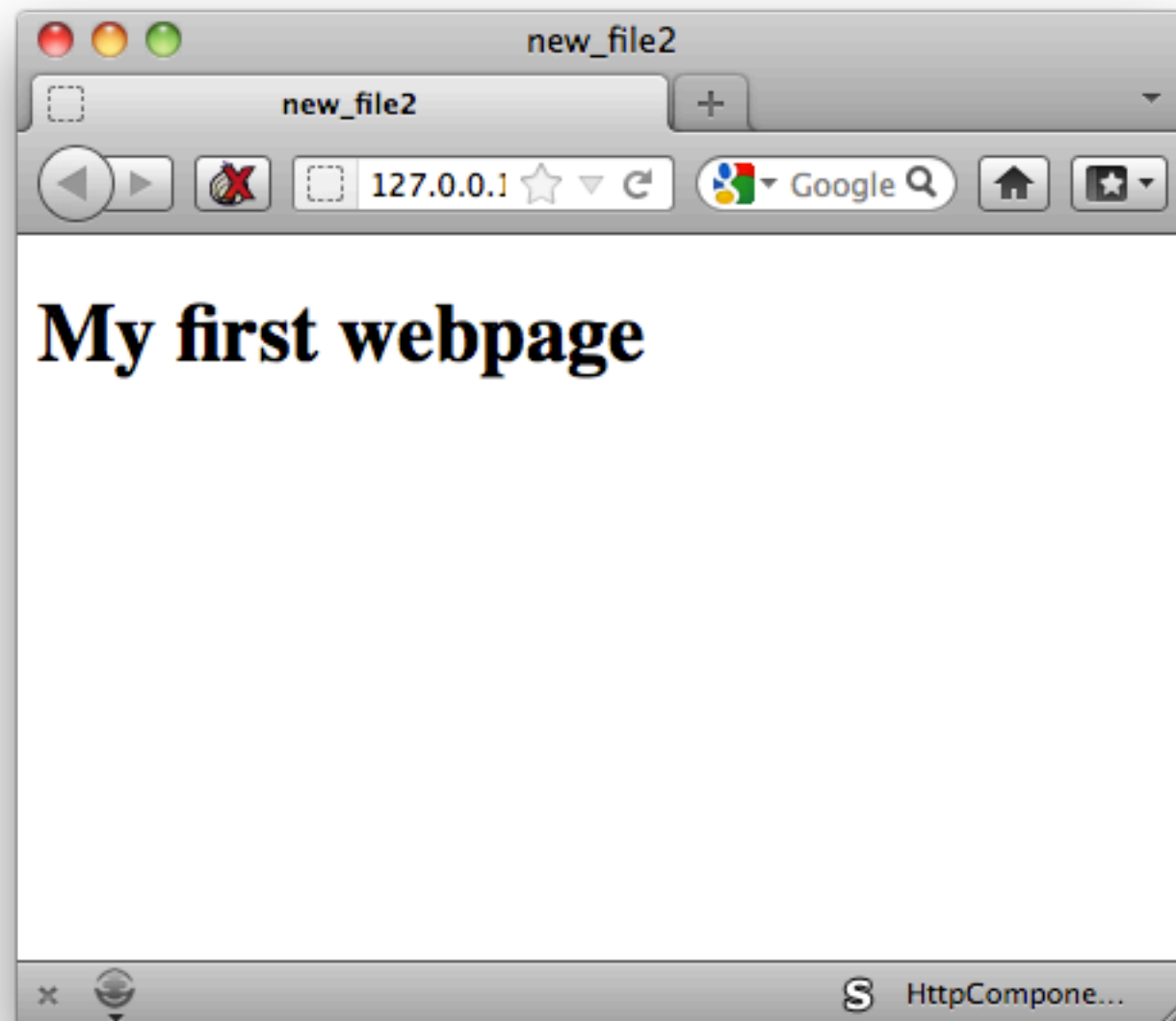
- HTML Terms:
  - Tag
    - Opening Tag
    - Closing Tag
  - Element
  - Attribute

# HTML term: Element

- A starting tag + some text/content + an ending tag is called an HTML Element.
- Examples of elements:
  - `<p>this is my great paragraph. I really hope you like it, I put a lot of thought into it. No, really, I did.</p>`
  - `<strong>this is some bold text!!</strong>`
- Element = `<tag>` + text + `</tag>`

# A Basic HTML Document

# A basic HTML page



```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html lang="en">  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# First things first: Doctype

- The very first thing on an HTML page is the doctype.
- Doctype tells us what kind of standards the page is using.
- It is an instruction to the web browser about what version of the markup language the page is written in.
- To read more: [http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp)

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html lang="en">  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# <html>

- Next comes the <html> **opening tag**.
- All of your HTML code will go in between the <html> and the </html>
- The very last line of every webpage you create should always be </html>

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# <html>

- Since most tags come in pairs, there will be an **opening tag** and a **closing tag**.
- The **closing tag** will have a backslash / before the tag name.
- <html> is the **opening tag**.
- </html> is the **closing tag**.
- Everything in between them are other HTML tags.

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# The <head> element

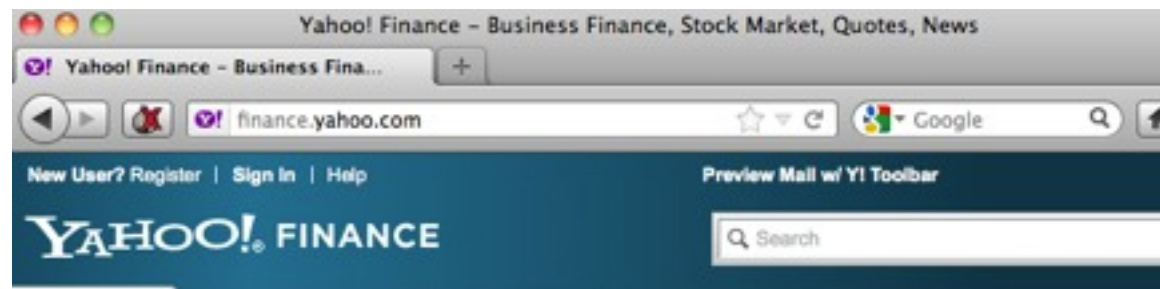
- There are two main sections in every HTML page:
  - the <head>
  - the <body>
- <head> allows you to define metadata for search engines, as well as things like the page's title.
- Just like with <html>, <head> has an opening and closing tag.

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```



# <title>

- The **title** element allows us to set the text displayed for our page in the browser's tab, or the top of the browser window:



```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# <title>

- **title** always lives within the **head** element

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# Nesting of HTML tags

- Some tags can be nested within other tags.
- **title**, for example, is nested inside the head tag:

```
<head>
```

```
    <title>My Amazing Webpage</title>
```

```
</head>
```

# The **body** element

- The body element typically contains the main content of your page.
- All the visible content of your page will go inside the <body> opening and </body> closing tags.

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# The **h1** element

- `<h1>` will create a new heading for your website
- `<h1>` stands for the 1st Heading
- There are six different levels of headings:
  - h1, h2, h3, h4, h5, and h6
- `<h1>` is typically used to set the title of your website

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
  </body>  
</html>
```

# The h2 element

- `<h2>` will create a new heading for your website
- `<h2>` stands for the 2nd Heading
- There are six different levels of headings:
  - h1, h2, h3, h4, h5, and h6
- `<h2>` is typically used to set the subtitle of your website

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-  
Type" content="text/html;  
charset=utf-8">  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
    <h2>Where I learn HTML</h2>  
  </body>  
</html>
```

# The **p** element

- `<p>` will create a new paragraph when you surround a section of text with `<p>` and `</p>`
- It will create space above and below any text you wrap inside it.

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
    <h2>Where I learn HTML</h2>  
    <p>  
      This is my first paragraph  
of text!  
    </p>  
  </body>  
</html>
```

# Spacing your content

- There are two ways to add whitespace around your content:
  - `<p>...</p>`
  - `<br>`
- `p` stands for paragraph.
  - It will create space above and below any text you wrap inside it.
- `br` stands for break.
  - It will create a new line break (a carriage return) anywhere you place it.



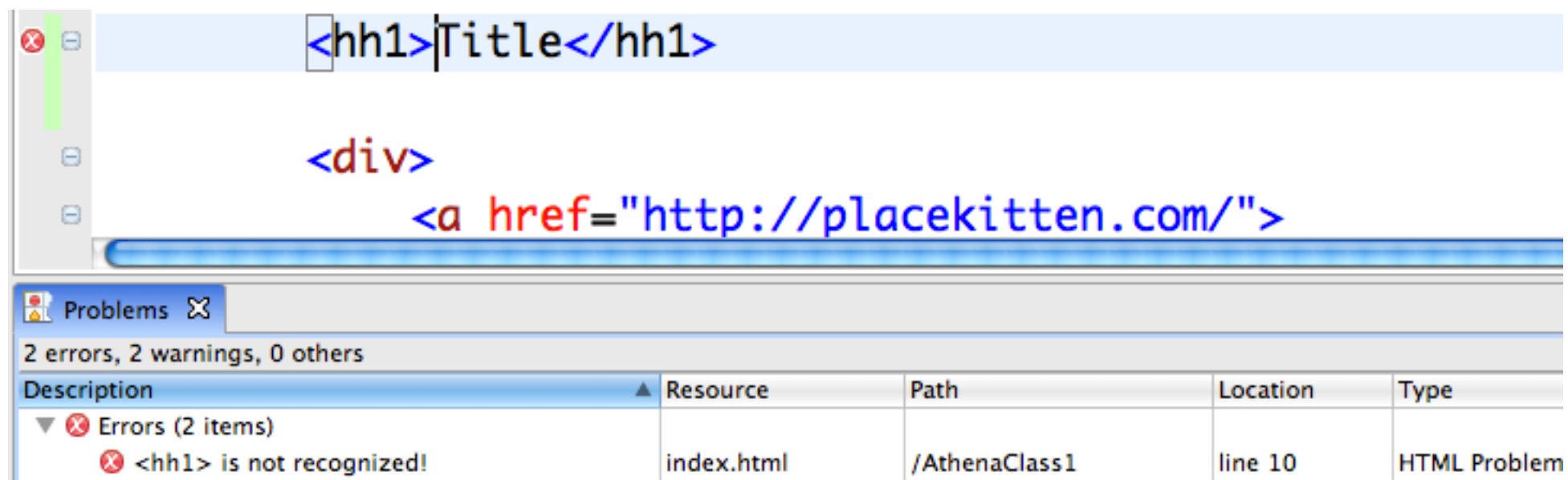
# HTML Editors

# Free HTML/CSS editors

- You don't need anything more sophisticated than Notepad (on a PC) or TextEdit (on a Mac) to build an HTML page.
- However, it is often easier to use an HTML editor.
- Here are some free ones:
  - Aptana (Windows, Mac OS 10.5+, Linux):  
<http://www.aptana.com/products/studio2/download>
  - Komodo (Windows, Mac OS 10.4+, Linux):  
<http://www.activestate.com/komodo-edit> **note, free trial only**

# Why use an Editor?

- Editors like Aptana will highlight mistakes in your HTML or CSS code



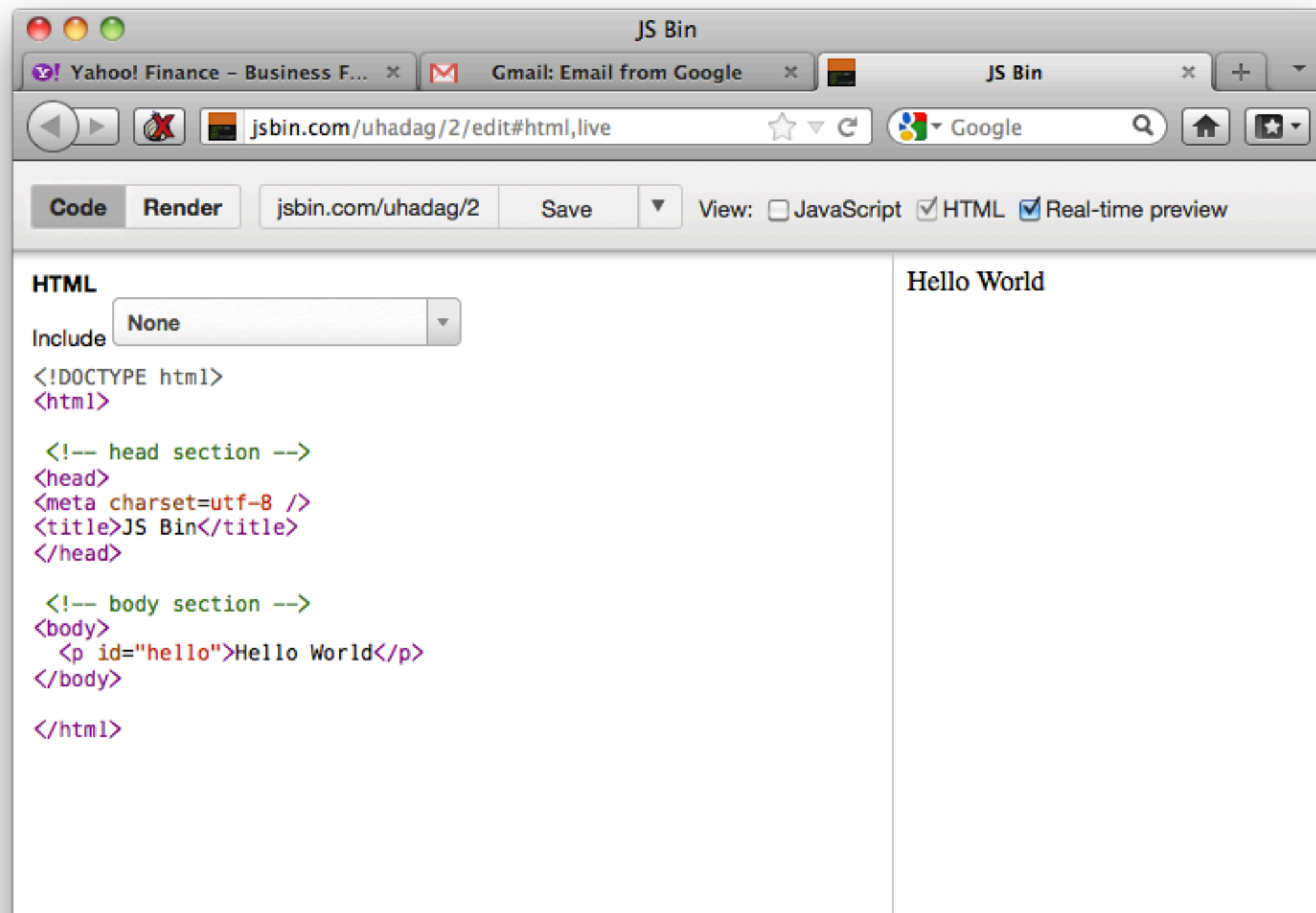
# Why use an Editor?

- They will highlight different parts of your code in different colors, making it easier to read
- This is called “syntax highlighting” in programming

```
<!-- comments are one color -->  
<body>  
    <!-- tags are yet another color -->  
    <h1>  
        <!-- the text inside tages are yet another color -->  
Athena HTML/CSS Class 1
```

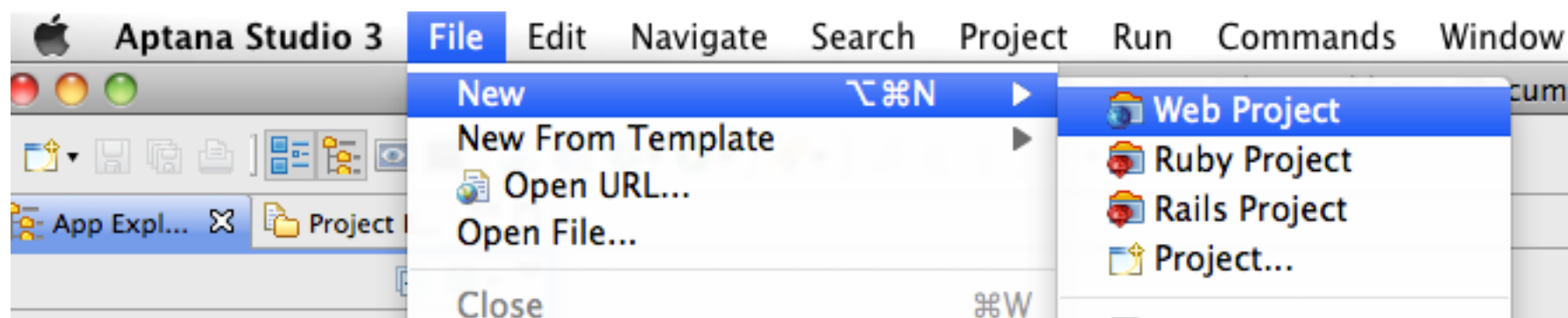
# If you don't have an editor today...

- If you don't have an editor today, I'd suggest going to <http://jsbin.com/uhadag/2/edit>



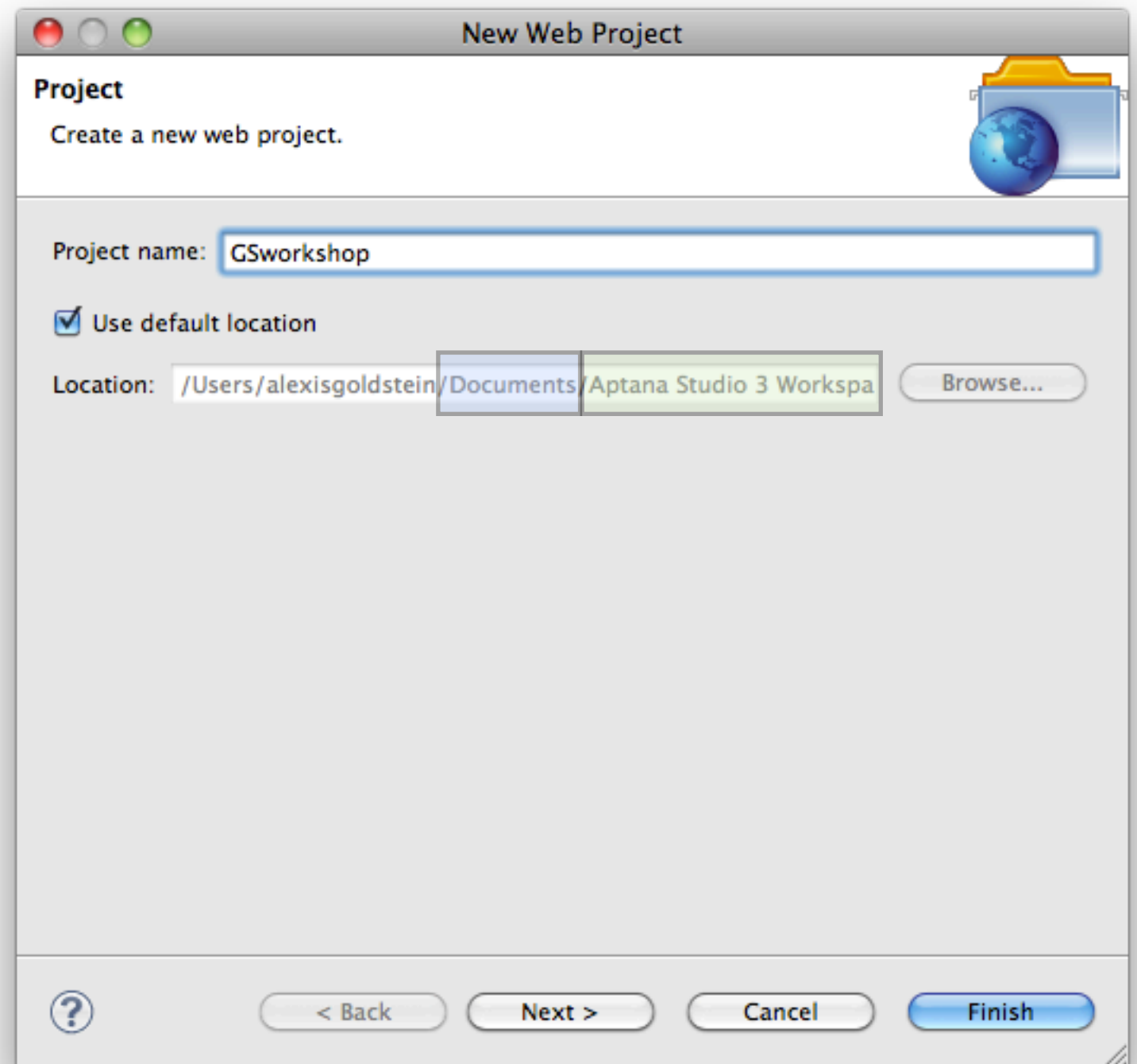
# Getting started with Aptana v3

- Before we can start exploring Aptana, we need to create a new project.
- The first step is to go to **File > New > New Web Project**



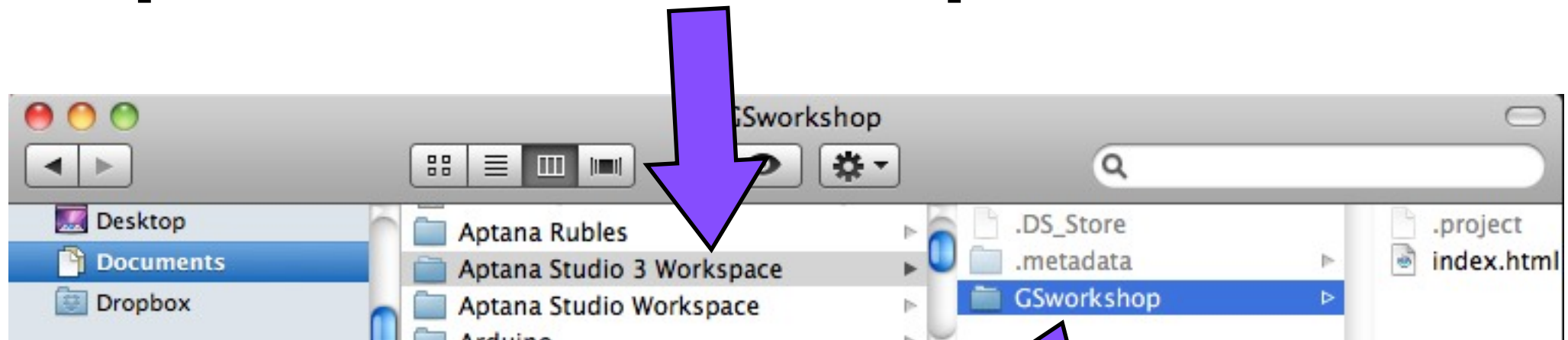
# Where Projects are saved

- The second step is to give you Project a name
  - By default, the new Project will be saved to your Documents folder, under a folder named “Aptana Studio 3 Workspace”
- Don't click **Finish** just yet!



# Where Projects are saved

- By default, the new Project will be saved to your Documents folder, inside a folder named **Aptana Studio 3 Workspace**

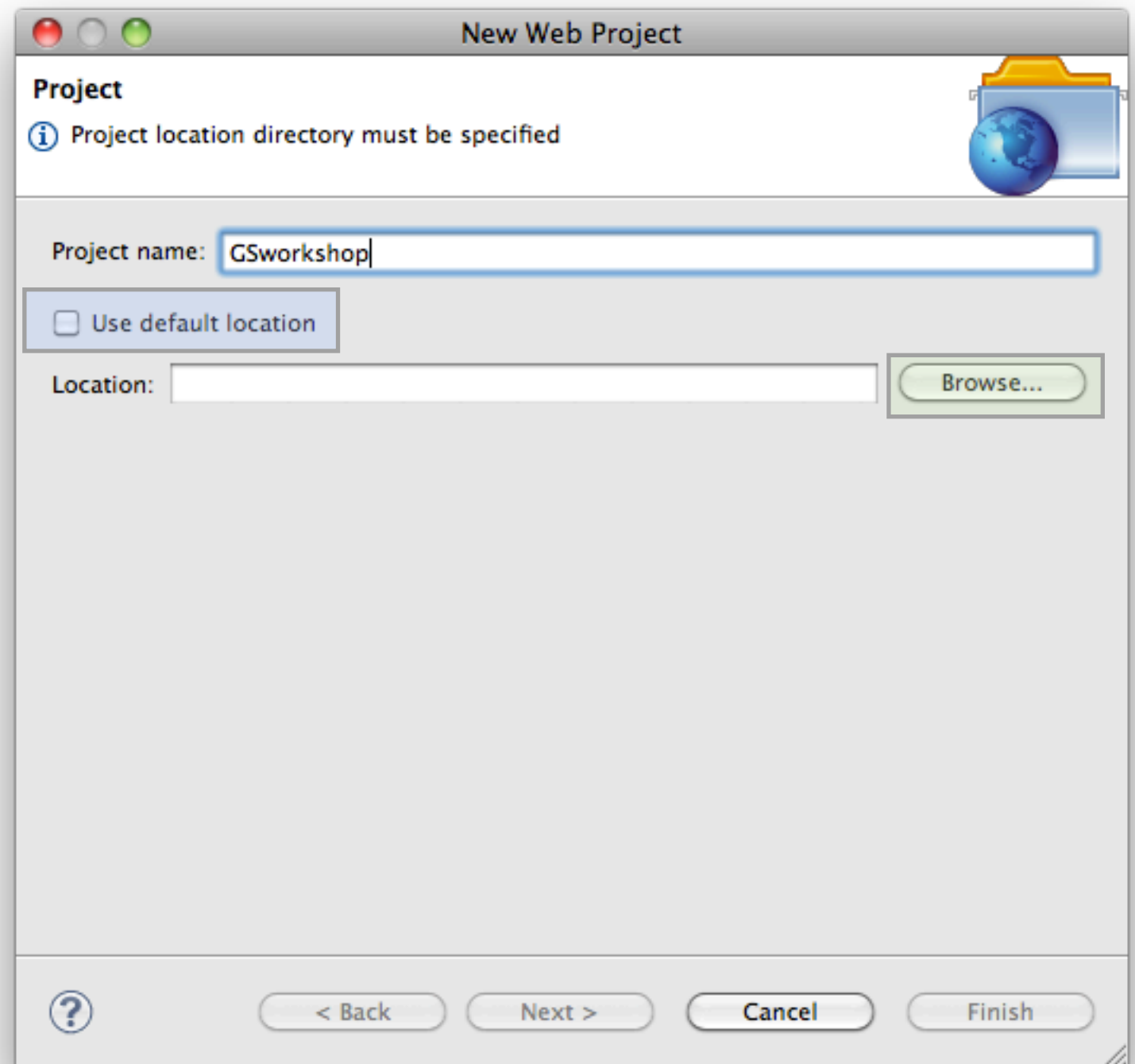


- I named my project GSworkshop. This folder will hold my webpage files.



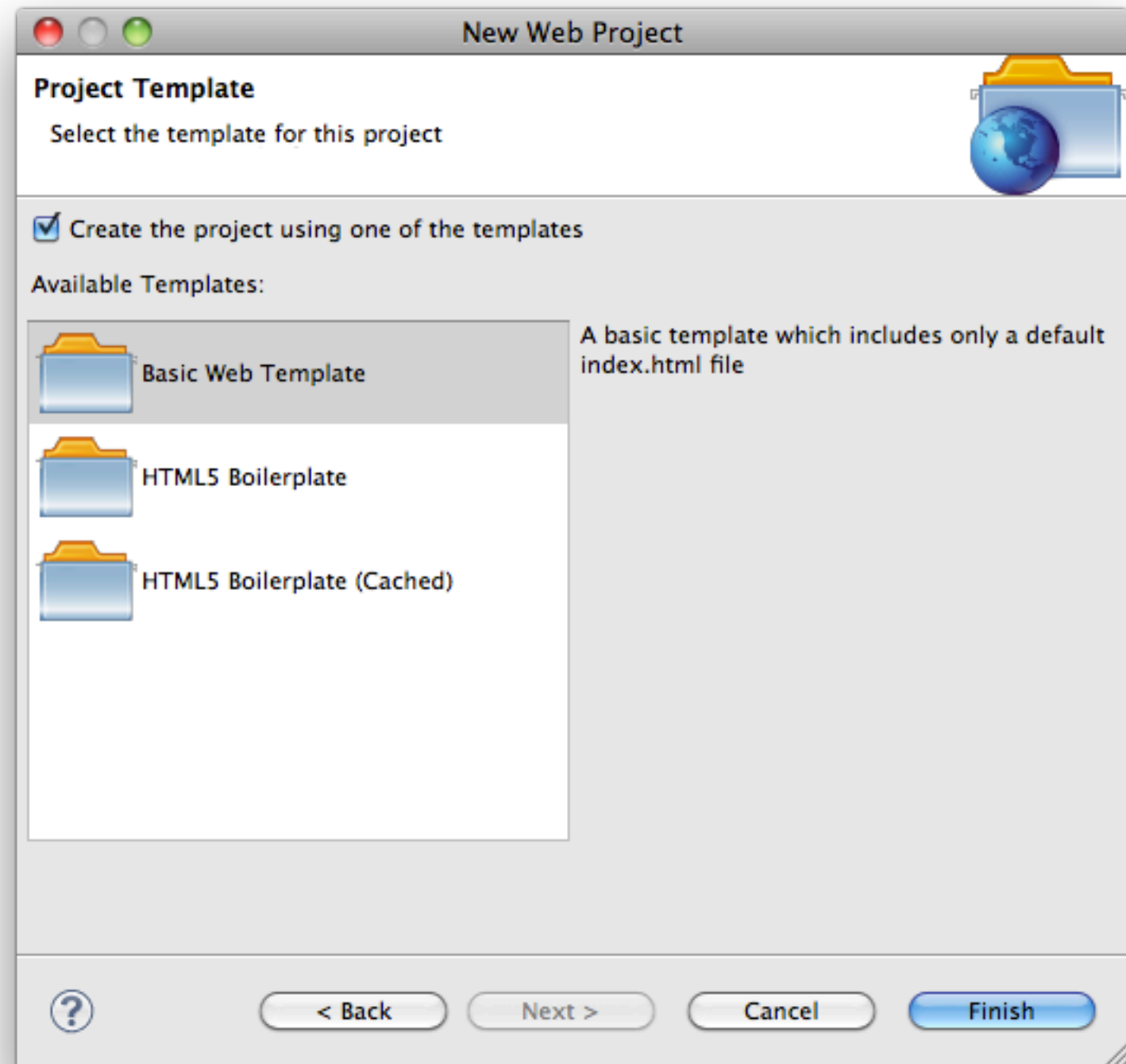
# Where Projects are saved

- If you would rather save your Project to a different location:
  - Uncheck “use default location” checkbox
  - Click the **browse** button and choose a different folder to save to.

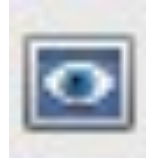


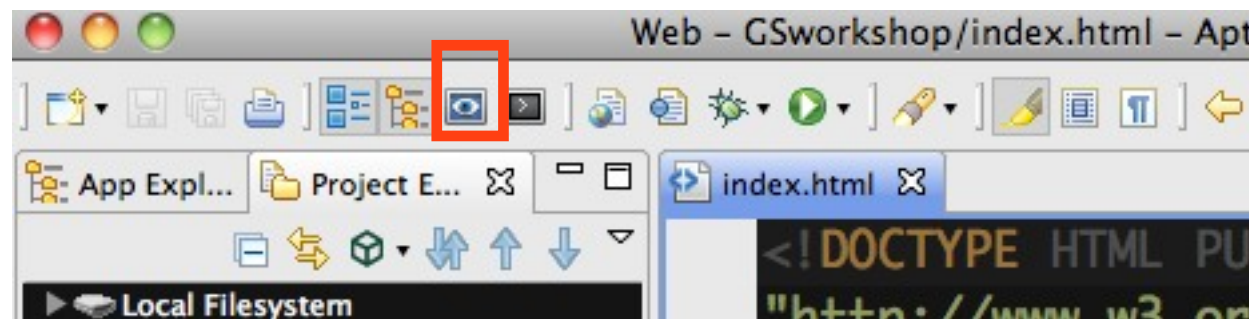
# Make the Project use the Basic Web Template

- Click “Create the project using one of the templates”
- Choose “Basic Web Template”
- Click **Finish**



# Setting up our Preview window

- To view a preview of your page in Aptana version 3, click the icon that looks like an eye. 
- It's at the top of Aptana in the toolbar



# Setting up our Preview window

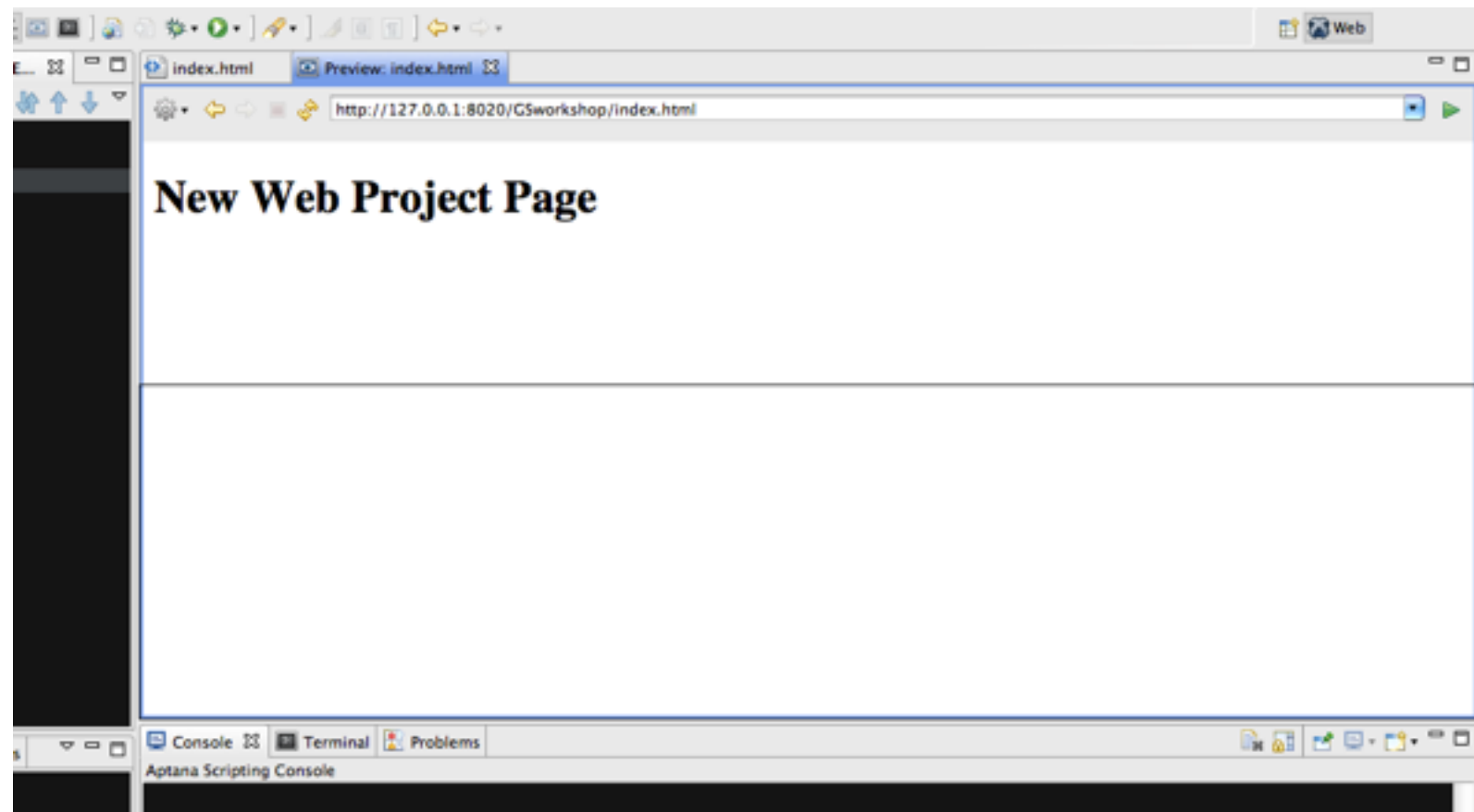
- Clicking this eye icon will create a new tab called “Preview”



- This will allow us to view the changes we are making to our HTML file.

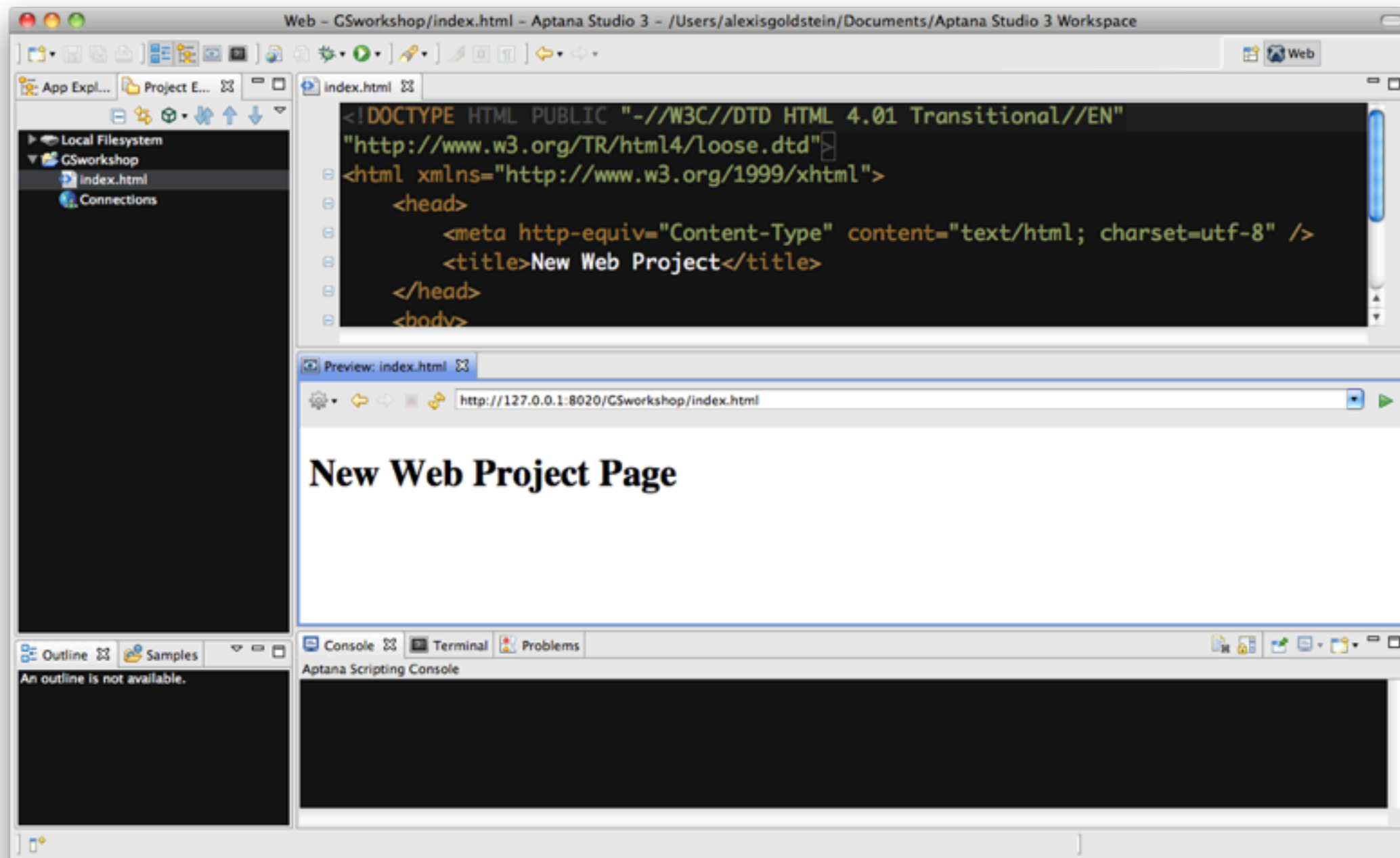
# Setting up our Preview window

- Instead of having the preview in a separate tab, I prefer to see the preview underneath my HTML file.
- To achieve this, click and drag the preview tab down towards the bottom of the screen, then let go when you see a black rectangle:



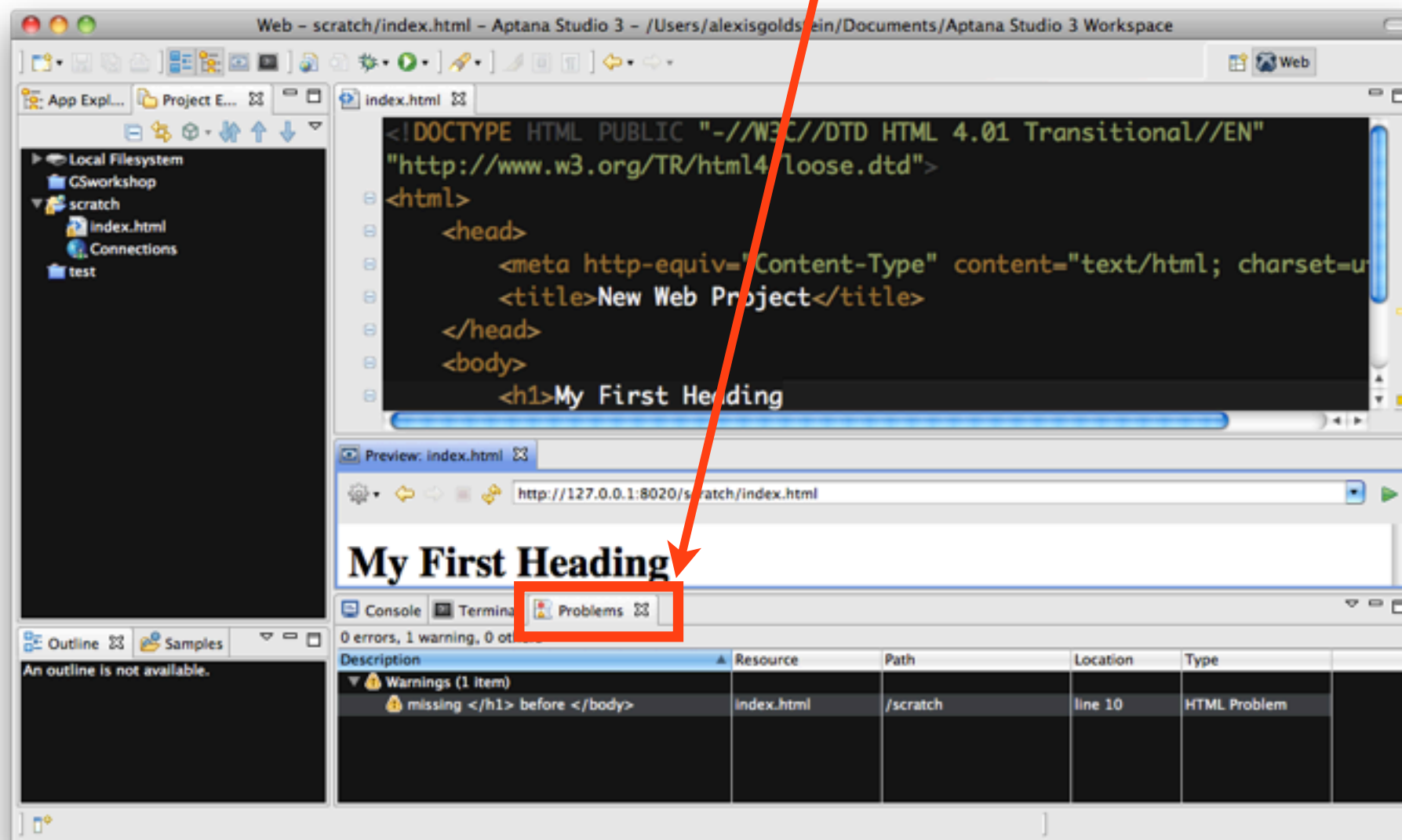
# Setting up our Preview window

- The preview tab should anchor itself below your code, so you get a nice split screen.



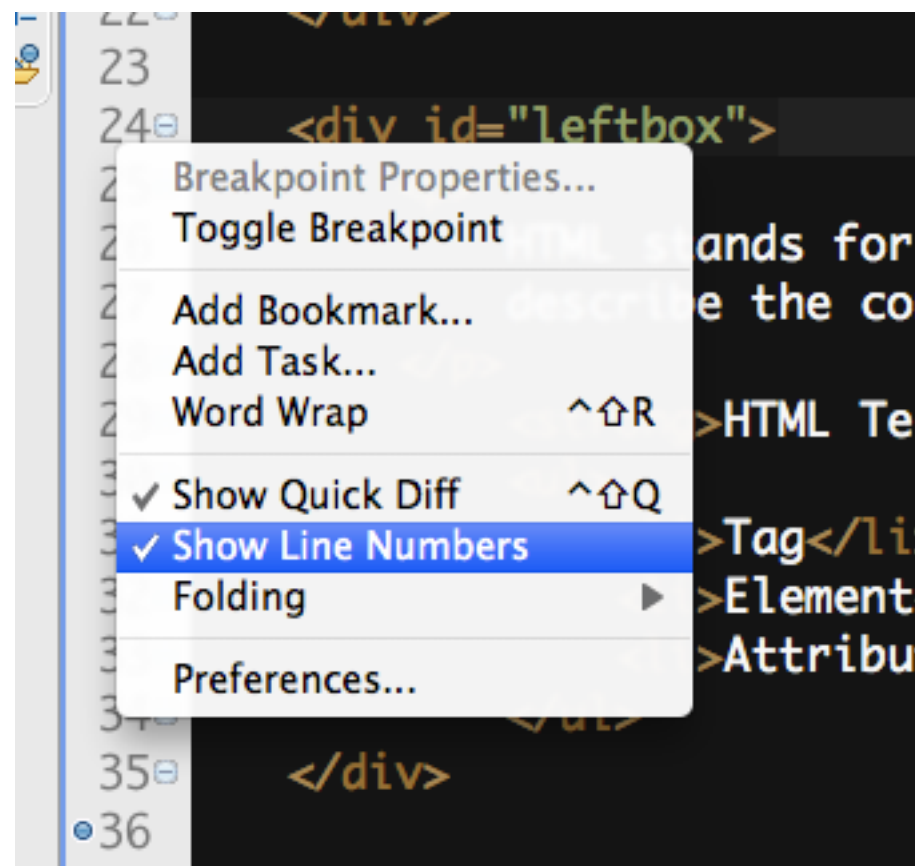
# When things go wrong

- If you have any errors in your HTML, they will show up in the **Problems** tab:



# Show line numbers

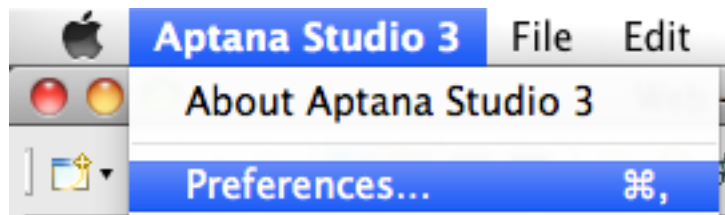
- Right-click in the gray, left-side margin (or control + click on a Mac) and make sure you choose “Show Line Numbers”





# One more thing: font size

- To change the default font size, go to:
  - On a Mac: Aptana Studio 3 > Preferences



- On a PC: Window > Preferences

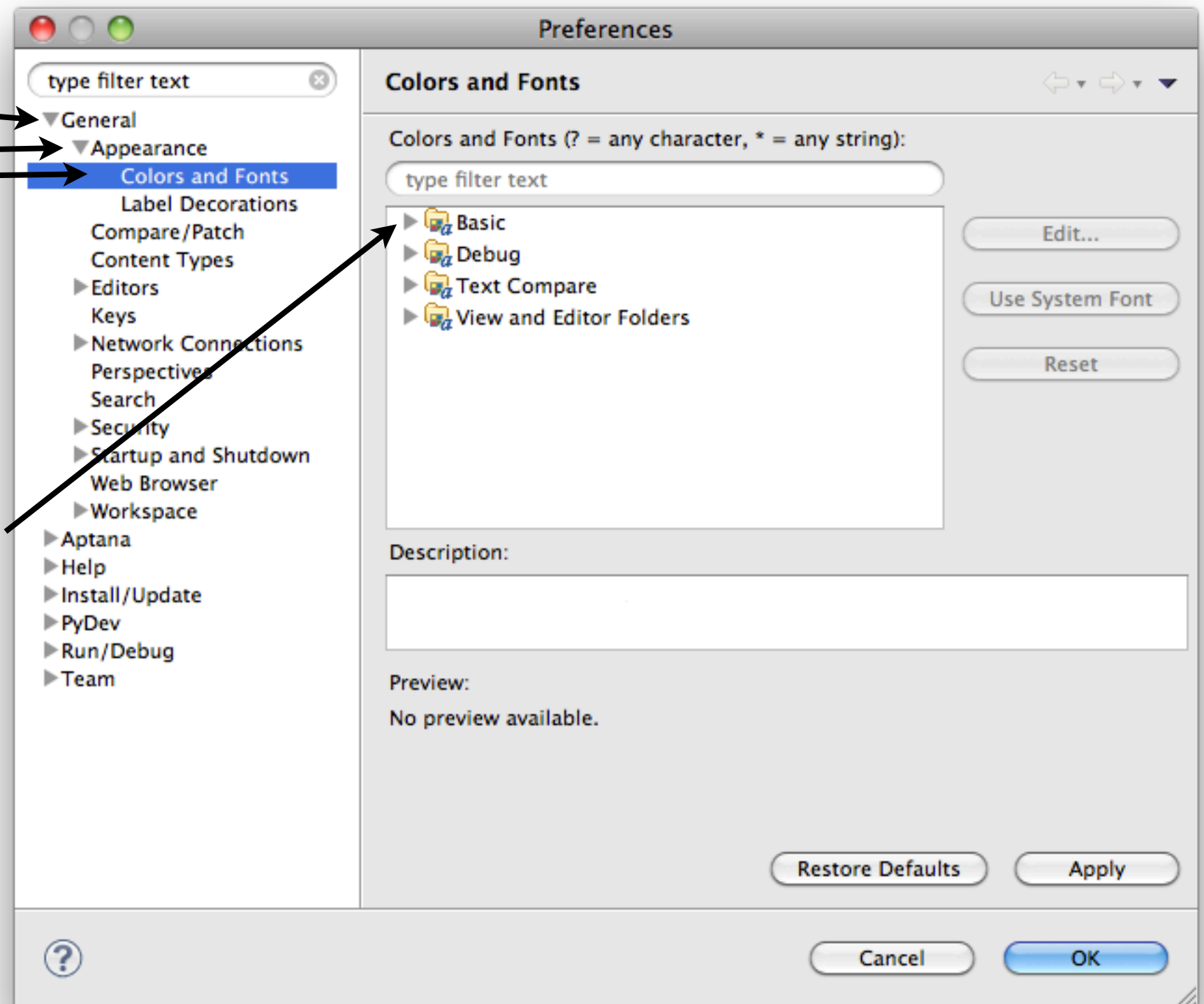
# If you'd like to change the font size

Expand **General**

Expand **Appearance**

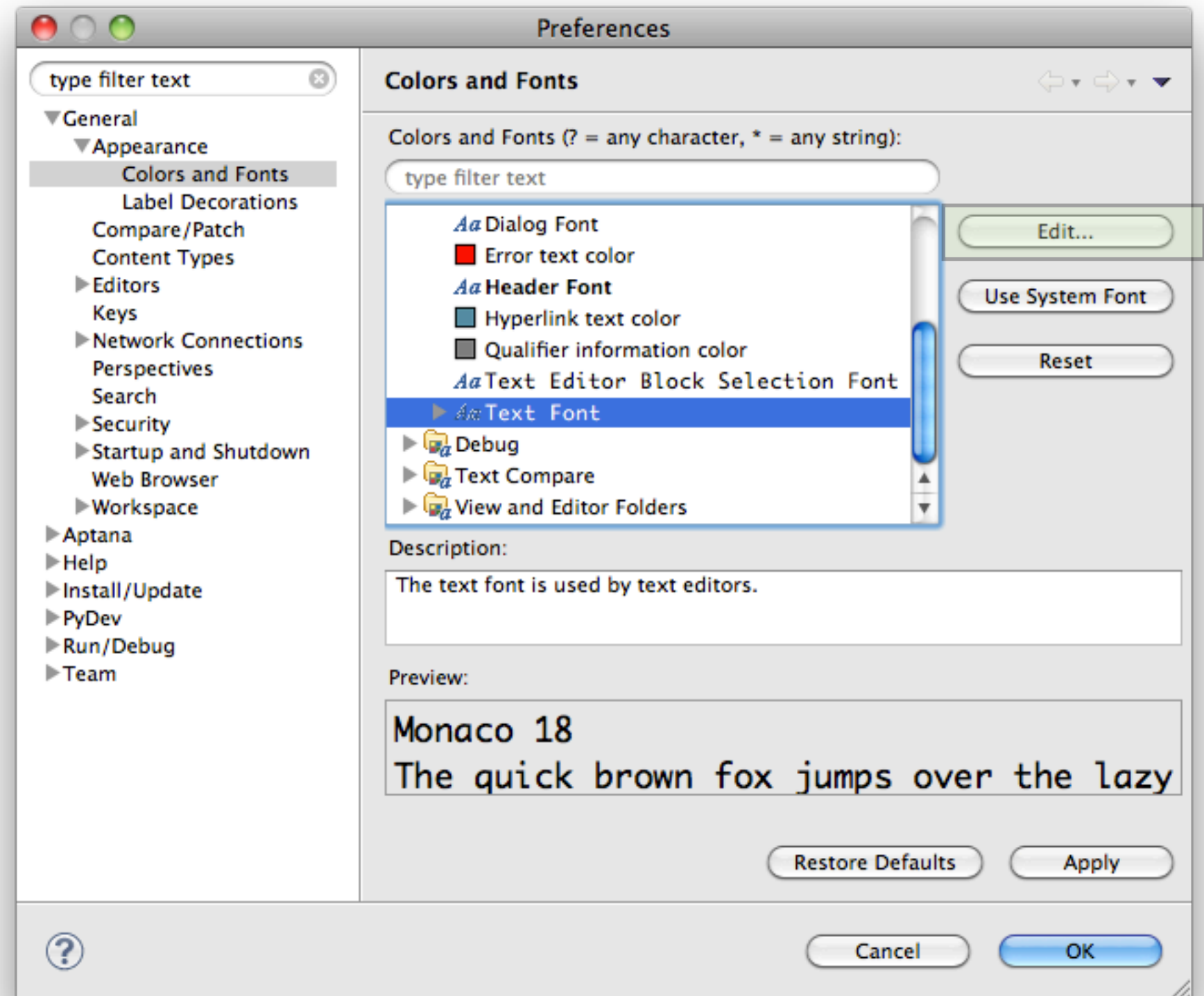
Click on **Colors and Fonts**

Expand the **Basic** folder



# If you'd like to change the font size

- Inside the **Basic** folder is a choice called **Text Font**. Click it.
- Now click the **Edit** button, and you can choose a new size



# Let's practice

- Aptana should have given us a good place to start.
- Let's add some tags inside the <body> section.
- Let's add a <h1>, <h2> and <p>

```
<!DOCTYPE html PUBLIC "-//W3C//  
DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/  
loose.dtd">  
<html>  
  <head>  
    <title>new_file2</title>  
  </head>  
  <body>  
    <h1>My first webpage</h1>  
    <h2>Where I learn HTML</h2>  
    <p>  
      This is my first paragraph  
of text!  
    </p>  
  </body>  
</html>
```

# Exercise

1. Add some basic text to your page. Just a few sentences.
2. Try wrapping one sentence in `h1` tags, and see what happens
3. Try wrapping another sentence in `<h2>` tags
4. Try wrapping a long block of text in `<p>` tags

```
<body>  
  <h1>Test again</h1>  
  <h2>Test</h2>
```

```
<p>
```

This is the HTML/CSS course for the Athena Leadership Lab.  
This is the HTML/CSS course for the Athena Leadership Lab.

```
</p>  
</body>
```

# &nbsp; and Character Codes

- &nbsp; is what is called a Character Code
- There are character codes for many different characters in many different languages
- Here is a solid list: <http://rabbit.eng.miami.edu/info/htmlchars.html>

# Other Character Codes

- Delta  $\delta$ 
  - `&delta;`
- Copyright symbol ©
  - `&copy;`
- Grave `
  - `&grave;`
- An “a” with the grave à
  - `&agrave;`

# Exercise

1. Add a few `<br>` tags to your page, in between text
2. Try adding a few character codes, like `&copy;` or `&delta;`;
3. Try adding several `&nbsp;` codes to create several spaces

```
<body>
Hello
<br><br><br><br>
Goodbye

<br>
Test &nbsp;&nbsp;&nbsp;&nbsp;&nbsp; Test2
<br>
&copy; 2011
<!-- This is the character code for the delta symbol-->
&Delta;
</body>
```



# Self-Closing Tags

- The `<br>` tag is our first example of a self-closing tag.
- You can write is `<br/>` or `<br>`, both will work.
- `<br>` is preferred in HTML5

# Self-Closing Tags

- Tags usually come in pairs...but not always
  - Most tags have a starting and ending tag.
  - However, some tags are “self-closing” tags.
    - They don’t require a closing tag, because you don’t have anything to sandwich between them.
    - All the information you need is in the first tag!
  - Example self-closing tags: br, img

# More HTML Tags: a

- The tag a is probably one of the most frequently tags in all of HTML.
- It links you to an external page, or a sub-section of your own page.
- `<a href="http://www.twitter.com">Go to Twitter!</a>`
- Try it yourself: [http://w3schools.com/tags/tag\\_a.asp](http://w3schools.com/tags/tag_a.asp)

# HTML Vocabulary

- HTML Terms:
  - Tag
    - Opening Tag
    - Closing Tag
  - Element
  - Attribute

# Our first look at HTML attributes

- We're going to specify where the link should take the user to.
- We use an HTML attribute to specify where we want the **a** tag to link to.

# HTML Term: Attribute

- Attributes provide additional information about HTML elements.
- Attributes are formatted like this:  
`attr="value"`
- Attributes are always specified in the opening tag.
- Attribute values should always be in quotes.
  - An example: in `<a href="http://www.google.com">`, href is the attribute.

# Exercise

- Let's try adding some **a** tags with **href** attributes to our page.

```
<a href="http://google.com">
```

```
This is a link to google!
```

```
</a>
```

```
<a href="http://twitter.com">
```

```
This is a link twitter!
```

```
</a>
```

# More HTML Tags: img

- Another very common HTML tag is the `img` tag, which creates images.
- The `img` tag has lots of attributes you can specify.
  - The most important one is the **src** attribute.
  - Without a **src**, you won't display an image!



# More HTML Tags: img

- The img tag has lots of attributes you can specify.
- **alt**: Specifies an alternate text for an image. Used by search engines, and by screen readers.
- **height**
- **width**

```

```

# Exercise: <img>

- Let's try adding a few images to our page.
- Search on google to find a few images.
- Right-click the image and choose “copy image location”
- Paste the URL into your src attribute:
  - ``

# Exercise: <img>

- Make sure you include both the **src** attribute and the **alt** attribute.

```

```

# More HTML Tags: ol, ul

- Let's make a list!
  - HTMLs allow you to specify several items in either bulleted or numbered lists
  - `<ol>` creates an **ordered** list with numbers
  - `<ul>` creates an **unordered** list with bullets

# More HTML Tags: ol, ul

- `<ol>` creates an **ordered** list with numbers
- `<ul>` creates an **unordered** list with bullets
- All the items in our list will live inside `<li>` elements
  - We must combine two elements together to make a list

# More HTML Tags: ol, ul

- `<ol>` creates an **ordered** list with numbers
- `<ul>` creates an **unordered** list with bullets
- All the items in our list will live inside `<li>` elements

```
<ol>  
  <li>Do my laundry</li>  
  <li>Pay my bills</li>  
  <li>Go the bank</li>  
  <li>Turn off the projector!</li>  
</ol>
```

1. Do my laundry
2. Pay my bills
3. Go the bank
4. Turn off the projector!

```
<ul>  
  <li>Do my laundry</li>  
  <li>Pay my bills</li>  
  <li>Go the bank</li>  
  <li>Turn off the projector!</li>  
</ul>
```

- Do my laundry
- Pay my bills
- Go the bank
- Turn off the projector!

# Adding a table to our page

- Tables are composed of rows and columns.
  - Rows in HTML are denoted by the tag **tr**.
  - Columns are the tag **td**.
- Let's try this at the w3schools: [http://w3schools.com/tags/tag\\_table.asp](http://w3schools.com/tags/tag_table.asp)

# Adding a table to our page

- Copy/paste this into your page.

- ```
<table>
  <tr>
    <td>
      <!-- left column section -->
    </td>

    <td>
      <!-- right column section -->
    </td>
  </tr>
  <tr>
    <td colspan=2>

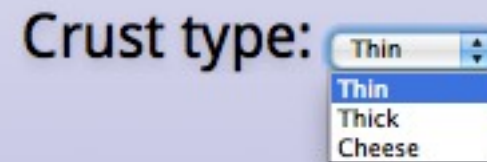
  </td>
</tr>
</table>
```



# HTML Forms

- You can also use HTML to define forms, including things like:

- drop-down boxes



- text fields

```
<input type="text" name="animal" placeholder="e.g. fox">
```

e.g. fox

- radio buttons

Pizza Size: ☐ Small ☐ Medium ☐ Large

- buttons

Submit order

# HTML Forms

- What I suggest for creating forms is to use Google Forms:
- <http://docs.google.com/support/bin/answer.py?answer=87809>

# HTML Forms

- Let's add a really simple form to our page:

```
<form action="mailto:admin@example.com" enctype="text/plain"
method="post">

    <input name="info" type="text">
    <button type="submit">Send Mail</button>
</form>
```

# HTML Forms

- A form is usually received by and interpreted by a program on a server, written in a programming language like PHP or Ruby.
- If you don't have a server, you can also have all the form contents sent to the email address of your choice, using extra attributes in the form element.
- That is what we have done in our example

```
<form action="mailto:admin@example.com" enctype="text/plain"
method="post">
```

```
    <input name="info" type="text" />
```

```
    <button type="submit">Send Mail</button>
```

```
</form>
```

# HTML Forms

- To learn more about forms, two great resources are:
  - <http://www.teaching-materials.org/htmlcss/lesson3/slides.html>
  - [http://w3schools.com/html/html\\_forms.asp](http://w3schools.com/html/html_forms.asp)

# HTML Validation

- A good way to make sure your HTML is valid, and doesn't have any errors, is to validate it.
- Aptana can do this for you.
- You can also find many sites on the web that will allow you to copy/paste your code in, and it will point out any errors.

# Using Firebug to learn more

- Demo!

# Your Homework

- As a homework assignment, you should:
- 1) Review the following links:
  - The a tag: [http://www.w3schools.com/tags/tag\\_a.asp](http://www.w3schools.com/tags/tag_a.asp)
  - The img tag: [http://www.w3schools.com/tags/tag\\_img.asp](http://www.w3schools.com/tags/tag_img.asp)
- 2) Go to the htmldog HTML tag list: <http://htmldog.com/reference/htmltags/>
  - Pick three tags we did NOT cover today and read through what they do
  - Experiment with these tags in a HTML page



# Extra Topics

- Time permitting!

# How do I create a website?

1. Register a Domain
2. Get Hosting for that Domain
3. Upload your HTML files, images and CSS files to your web server

# Jargon Alert!

## DOMAIN



Domain

<http://www.something.com>



Suffix

## DOMAIN REGISTRATION

The process of claiming / reserving a domain name.

Lasts for one year, then you must renew the domain if you want to keep it. Should not cost you more than \$10/year.

# Step One: Registering a Domain

- Lots of choices:
  - [godaddy.com](http://godaddy.com)
  - [register.com](http://register.com)
  - [dreamhost.com](http://dreamhost.com)
  - [networksolutions.com](http://networksolutions.com)
- I recommend dreamhost

## Step Two: Getting Hosting for the

- Once you've secured the domain, you need a way to make it available for others to see!
- You need someone to host your domain in order to use a domain you've registered.
- A hosting service provides you with a **server**.
- The server is what actually makes your website viewable from any browser, on any network, anywhere.

# Jargon Alert!



## SERVER

A server is just a computer!

A dedicated computer that does nothing but sit around and wait for you to call.

When someone types in your domain (blahblah.com), the server receives the request, and **serves** you back the website.



# How Do Servers Receive your Request?

- When someone types in your domain (blahblah.com), how does your request get to the server?
  - Through DNS: Domain Name System
  - DNS is like a phone book.
  - It takes a domain (blahblah.com) and looks up the IP address for that domain. This is the IP address of the server that hosts your domain.

# Jargon Alert!

## IP ADDRESS

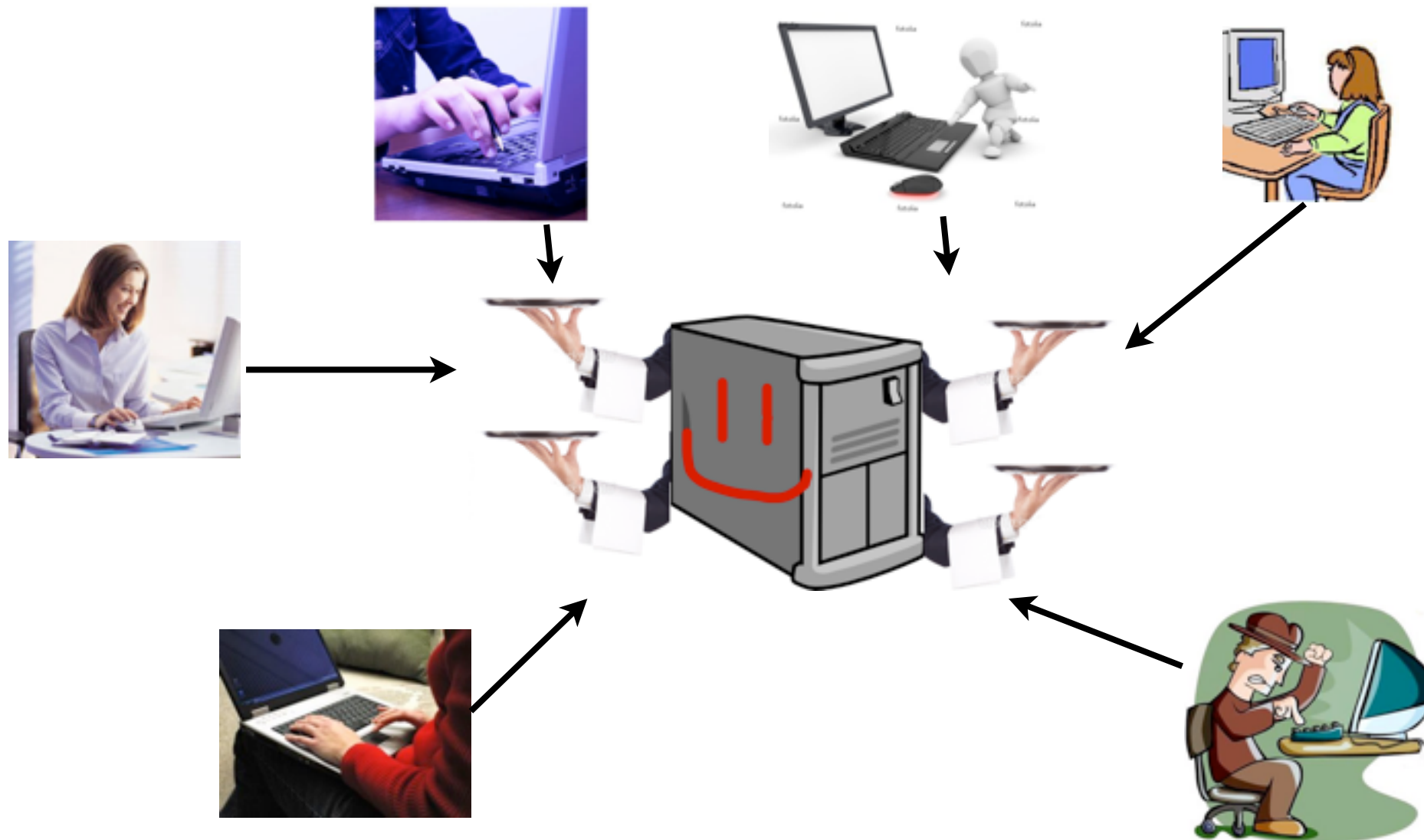
The “address” of your computer. It tells the Internet how to connect to a given machine.

Every single computer that is connected to the internet has a unique IP address.



# Hosting your Domain, Continued

- A dedicated server will allow multiple people to connect to your site at the same time.



# Hosting your Domain, Continued

- A good hosting company will allow many thousands of people to simultaneously view your site, with no crashes.
- Even a good hosting service may not be able to prepare you for “The Oprah Effect”
  - If you are on Oprah, and a million people all try and access your site at once, your server may crash!
- If you have plans to go on Oprah, hire a network specialist to guard your site against crashing!

# Hosting your Domain, Continued

- Who should I use for hosting?
- Again, you have lots of choices. Some common ones:
  - godaddy.com
  - bluehost.com
  - rackspacecloud.com
  - [dreamhost.com](http://dreamhost.com) -- what I use and recommend
- Comprehensive list: <http://ietherpad.com/3GA5A4CG9F>