



# CSS2



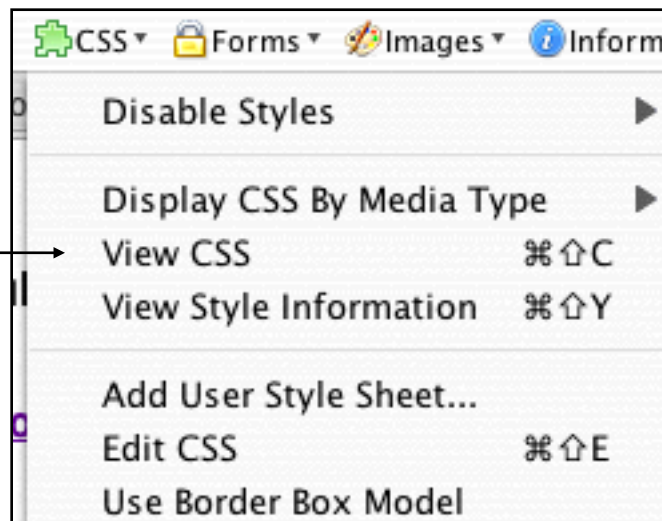
# Today's topics

- Box Model
- Flex-Box-Model
- List styles with links

# Finished Sample Files

- Remember: If you use Firefox it will be easy to view the .css files for each item listed.
- I have also tried to add description comments for all of these samples.

View CSS





# Let's set up today's folder



# Define site

- Take a moment to define the folder for today as your site folder.
  - Site > Manage Sites
  - Site > New Site
- Point to the folder of today as your root folder



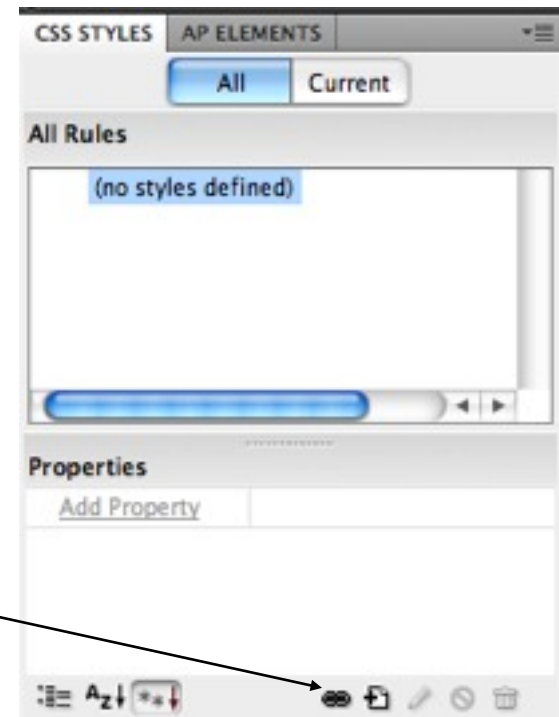
# XHTML - Box Model

Open up the [practice3.html](#)



# Let's create a .css

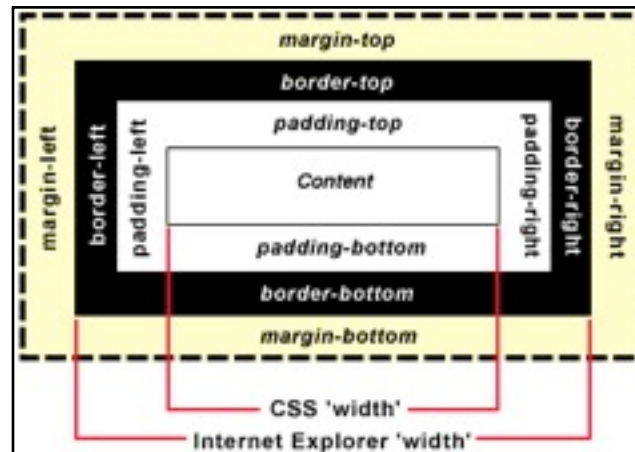
- Create a new .css and name it **practice3.css**
- Then we should attach it to the practice3.html.
  - **VERY IMPORTANT:** You must save this every time we make a change or you will not see anything occur in your html page.



Attach style sheet

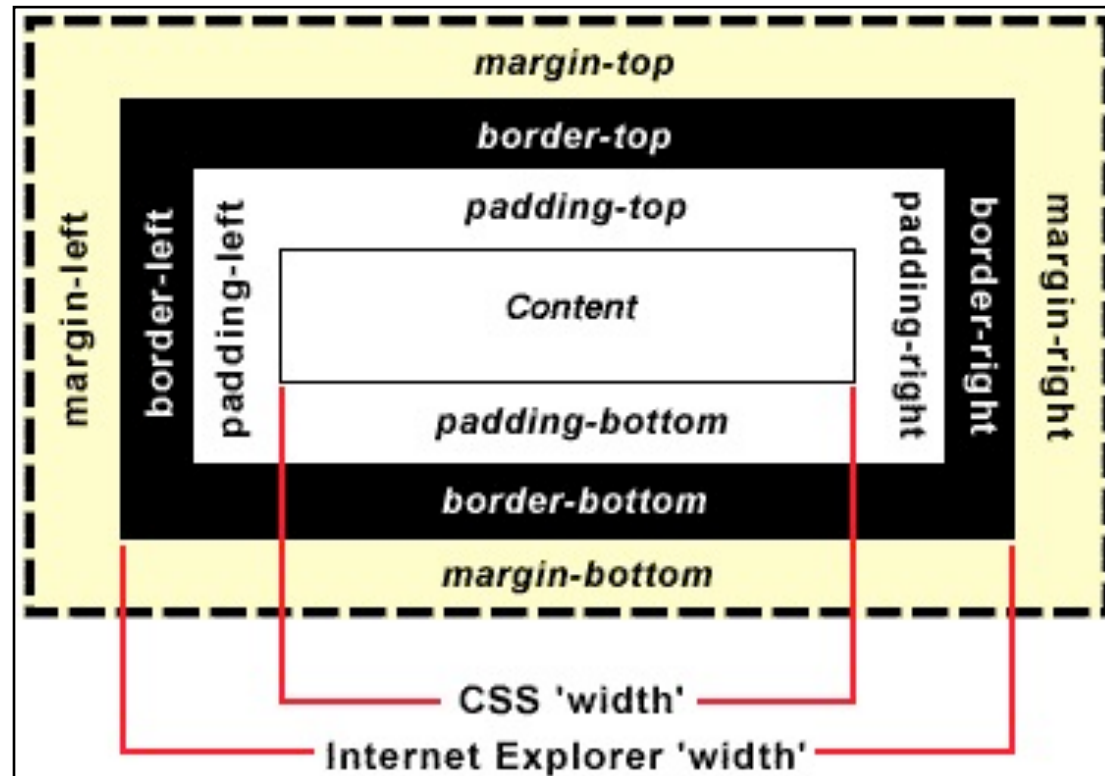
# Box Model

- What is the box model?
  - The box model is a way of adding spaces around content.
  - Many types of content (text, images, headings, links) can have these spaces around them with styles applied.





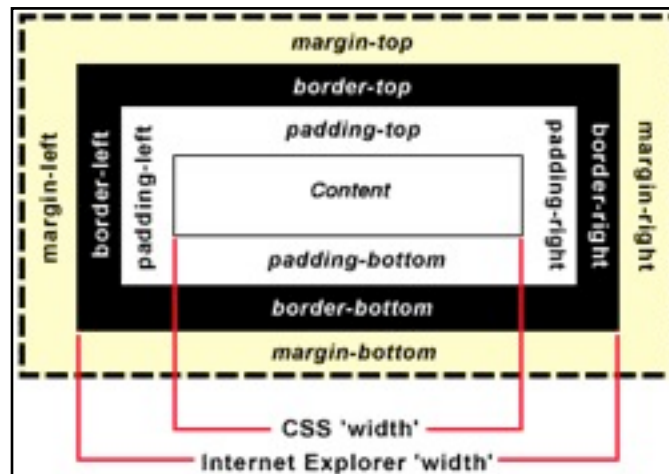
# XHTML - The W3C Box Model



- You can apply this box model to almost anything to add borders, padding, margins etc...

# A few terms to know

- Block elements
  - Margin and padding widths declared on **all four sides**.
- Inline elements
  - Margin and padding widths declared on the **left and right** hand sides.



# XHTML - Block Elements

- Block Elements are followed by a line break
  - h1, h2 etc
  - p
  - ul
  - ol
  - blockquote
  - div
- Block elements can have:
  - Margin and padding widths declared on all four sides.

# XHTML - Inline Elements

- Inline Elements are not followed by a line break. They follow each other like the words in this sentence
  - strong
  - em
  - img
  - a
  - span
- Inline elements can only have:
  - Margin and padding widths declared on the left and right hand sides.

# Margin

h1

{

margin: 20px;

}

- Sets all four sides

h1

{

margin-top: 30px;

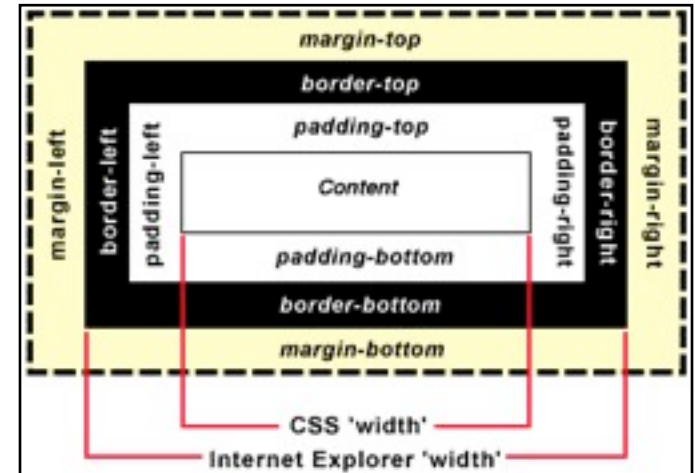
margin-right: 40px;

margin-bottom: 20px;

margin-left: 10px;

}

- Sets the top, right, bottom, left
- H1 is a block element so we can apply a top, right, bottom and left.
- It is best to do it in this order.



# Margins

- Shorthand for margin:

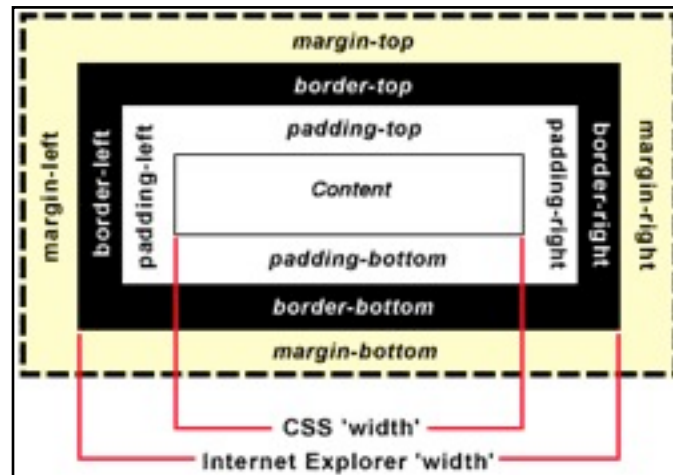
h1

{

margin: 10px 20px 15px 5px ;

}

- Top, right, bottom, left (clockwise)- **TRouBLE**
- It is considered good form to use these shortcuts. I will look for them in your code when I grade.



# Padding

h1

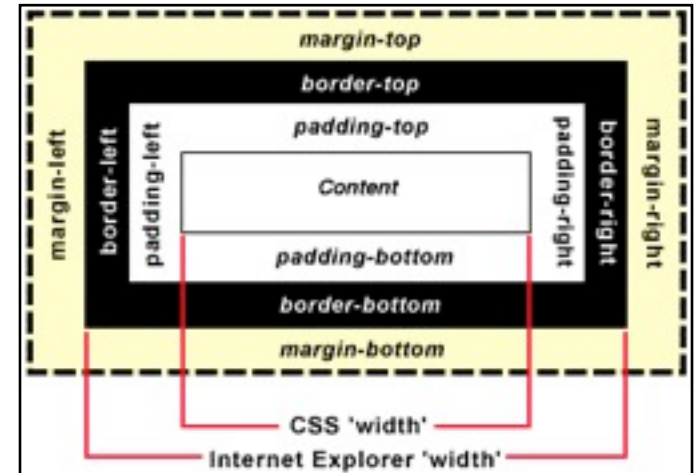
```
{
padding: 20px;
}
```

- Sets all four sides

h1

```
{
padding-top: 30px;
padding-right: 40px;
padding-bottom: 20px;
padding-left: 10px;
}
```

- Sets the top, right, bottom, left
- H1 is a block element so we can apply a top, right, bottom and left.



# Padding

- Shorthand for padding:

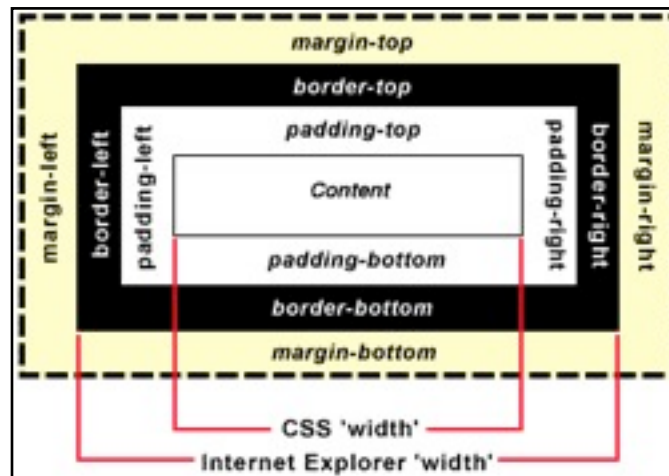
h1

{

padding : 10px 20px 15px 5px;

}

- Top, right, bottom, left (clockwise)- **TRouBLE**
- It is considered good form to use these shortcuts. I will look for them in your code when I grade.





# Box borders

- Let's say we want to add a border to our H1. This is the border shortcut.

```
h1
{
padding:20px;
border: 5px dotted #330066;
}
```

- This sets up the whole border!
- This must be previewed to view the change.
  - A few of the styles do not actually work as you would think so play is the best way to learn!.

Styles:

- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset

# Border: top, bottom, left, right

```
h1
{
padding:20px;
border: 5px dotted #330066;
border-top: 2px dashed #669966;
}
```

- This will overwrite the previous one on the top side because it is after it and it is more specific.

# Create a width

```
h1
{
padding:20px;
border: 5px dotted #330066;
border-top: 2px dashed #669966;
width:100px;
}
```

To see the correct padding, add a **width!**

# Background bug!

```
h1
{
border: 5px dotted #330066;
border-top: 2px dashed #669966;
background-color: #CC3366;
}
```

- Background-colors or images will fill up to the border's outer edge.
- This is a bit annoying when you have dashed or dotted edges.

# Box Model Rules Part One

- Top and bottom margins “collapse”.
  - I.e. If there are two boxes and they have top and bottom margins:
    - You don't add the two together to get the collective margin.
    - The largest one wins!
  - For example:
    - top box has a bottom margin of 50px
    - bottom box has a top margin of 40px
    - total margin between the two boxes is 50px.
- NB: this does not apply to side margins. they are the sum of the two margins as you would expect.

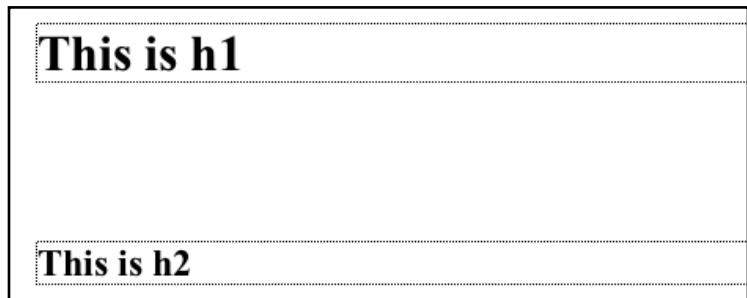
# Example

- Here is a good test to try this out.
- The 100px margin will win because it is the largest.

```

11 h1
12 {
13   border: 1px dotted;
14   margin-bottom:100px;
15 }
16
17 h2
18 {
19   border: 1px dotted;
20   margin-top:20px;
21 }

```





# Block vs Inline Elements



# Converting block and inline elements

- Occasionally you might like to convert inline elements to block elements and vice versa.
- Why convert an inline element to a block element?
  - To obtain the features of the opposite element.
    - Inline elements do not have a line break.
      - If you convert them to block you can add a line break to the elements.
      - You will also have the ability to set margins and paddings on all sides.



# Converting block and inline elements

- Why convert a block element to an inline element?
  - To obtain the features of the opposite element.
    - Block elements always have a break after them.
      - If you want to remove it, convert to an inline element.
      - Converting to inline means you can only set the margins and paddings on the left and right.
  - With the display property, you could change either.

# Converting to a block element

- To convert an inline element into a block element, use
  - `display: block;`
  - Let's try this on our anchors which are normally inline elements.
  - Why?
    - This will add a line break
    - This will allow us to put margins and padding on all four sides (with inline elements, you can only do left and right side!)

```
a
{
display:block;
padding:10px;
}
```

# Converting to an inline element

- To convert a block element into an inline element, use
  - `display: inline;`

```
li
{
display:inline;
padding:10px;
}
```

- Notice how the padding will only be added on the left and right because this is an inline element!



# \*Lists for linking

List style 1



# Pseudo classes/CSS selector

- These pseudo classes allow you to edit the anchor or link feature in four different ways.

a:link

a:visited

a:hover

a:active

It is recommended that  
you do them in this order,  
think LoVe HAtE



- More pseudo classes in CSS3 we will see them later on the module

# Changing the anchors

- a:link
- a:visited
- a:hover
- a:active

```
a
{
    color: maroon;
    font-family: century gothic;
    font-size: 15pt;
    font-style: oblique;
}

a:link
{
    color: green;
}

a:visited
{
    color: green;
}

a:hover
{
    color: yellow;
}

a:active
{
    color: pink;
}
```

# Why are we using lists?

- We will need to style text that will be used for links.
- The lists provide us already existing tags to work with.
  - This makes the text easy to select.
- The `<li>` also creates a line break which we would not get with the `<a>` tag.

```
<ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Rooms</a></li>
    <li><a href="#">Rates</a></li>
    <li><a href="#">FAQ</a></li>
    <li><a href="#">Contact</a></li>
</ul>
```



# Mixing links with lists





# List 1

- Open up **navigation1.html**
  - **Have a look at the simple code.**
- We will turn this simple unordered list into a navigation.
- We use lists for navigation because it can give you more control over the style of the anchors.
- Let's look at a sample.

# Create a stylesheet

- Let's create a stylesheet to go along with the `nav1.html`.
- Let's call this `nav1.css`
- Now using the `css style palette` we can link the `nav1.css` into the `navigation1.html`

# Adding it to our list

- We only want to affect one list on our page, after all, we may have other lists later.
- We need to give the list an individual name. We could just create a class, but that can be used on many objects, not simply one.
- We want something that is only for our list.
- I have placed the list in a <div>. Let's give this div an id.

```
<div id="navigation">
```

# The id...

- The id identifies this specific list.
- This prevents us from confusing it with any other list.

```
<div id="navigation">
```

# Adding our id to our css

- In the css the id is represented by the #name.
- We only want to affect li items in an unordered list with the id navigation.

```
#navigation ul li
{

}
```

# Note: Inheritance: Descendant selectors

- You can put two or selectors (tags) together when they are nested inside of each other.
- Place them in the order that they appear in HTML.
- Then just separate them by a space.

```
<div id="navigation">
  <ul>
    <li><a href="#">Home</a></li>
```



```
#navigation ul li
{
}

```

Remember: This selects the list item, not the <a> tag. If we wanted to select the <a> tag we would need to state that.

# Adding styles

- First we will:
  - Take away the bullet points
  - Add a margin to our list to put some spacing in between.

```
#navigation ul li  
{  
  list-style-type:none;  
  margin:20px;  
}
```

# Important note

- You will notice that you need to choose the selector which causes the problem that you want to solve.
  - This will apply to all CSS styles!
- Example:
  - We want to remove the bullet points
  - These are caused by the `<li>` tag
  - As a result, we need to select this tag and apply the style to it



# Reducing redundancy

- You will also notice that we have only added one id="navigation".
  - We could have added this to each `<li>`, but instead we have added it to the `<div>`.
  - This helps to reduce redundancy.

# Changing the links directly

- This helps you select the `<a>` links that are inside of the list item in the unordered list.

```
#navigation ul li a
{
}
}
```

Here we are stating specifically that we want to select the `<a>` tag

# Adding colour and decoration

- Next we want:
  - To change the colour of our links
  - Remove the underline
- This gives us quite a bit of control. Instead of just selecting all “anchor” tags, we are only selecting the ones with this id.

```
#navigation ul li a
{
color:#000000;
text-decoration:none;
}
```

# Selecting the mouse over

- We want to create a little trick with the border when you mouse over the link.
  - This is how you would select the mouse over.

```
#navigation ul li a:hover
{

}
```

# Creating a rollover

- We are going to create a trick.
- When you rollover, we want to show the left border of the link and make it a blue square.
- We will:
  - Then we will create the border-right and give it a colour.

This is the square rollover →

```
#navigation ul li a:hover
{
border-left: 1em solid #4866ad;
}
```

# a:hover

- Here we will add some padding to separate the rollover from the element.
- Color: will change the color of the text when you rollover.

```
#navigation ul li a:hover
{
border-left: 1em solid #4866ad;
padding-left: 10px;
color: #3300CC;
}
```

# Problem to fix!!!!

- The borders are not aligned so it looks a bit messy.
- For the <a> tag we must:
  - Make the border the same as the rollover, but it will be white. If the border is white you will not be able to see it. It will act as a place holder.
  - Make the padding the same as the a:hover, to ensure that it matches in alignment.

```
#navigation ul li a
{
color:#000;
text-decoration:none;
border-left: 1em solid #ffffff;
padding-left:10px;
}
```

Add this to the "a"

Notice that this border is white which will match the background colour of the page

# Mouse down on current page

- We also want to create a style that will show you what page you are currently on.
- We will assume that this page is the home page and we want to show that as the current link.
- Let's give it an id.

```
<li><a href="#" id="currentpage">Home</a></li>
```



# Styling the current page

- To style the current page we can set up our style this way.
  - Why have we connected the a with the #currentpage this? Let's have a look....

```
#navigation ul li a#currentpage
{

}
```

# Styling the current page

- To style the current page we can set up our style this way.
  - If you have an item (attribute such as id or class) within another tag you connect them

```
<div id="navigation">
  <ul>
    <li><a href="#" id="currentpage">Home</a></li>
```

↓

```
#navigation ul li a#currentpage
{
}
}
```

# Our style sheet will look like this

- It will be the same as the mouse over, but it will always be shown on the current page!

```
#navigation ul li a#currentpage
{
border-left: 1em solid #4866ad;
padding-left: 10px;
color: #3300CC;
}
```

Same colour and  
placement as  
a:hover

# Note

- You may write this code as #current page instead of the whole line.
- Important: **This doesn't always work.**
  - Sometimes the browser has trouble matching the css with the HTML.
  - When in doubt, refer to the the previous slide as a solution.

**Another option**  
**Less reliable**

```
#currentpage
{
border-left: 1em solid #4866ad;
padding-left: 10px;
color: 3366FF;
}
```

# Finished

- We are finished with this list.
- Please close the style and the html page.



# List styling 2



# Open up `navigation 2.html`

- This is the same list that we created before but this time we will style it in a different way.
- First let's create a style sheet.
- Let's call it `nav2.css` and connect it to `navigation2.html`

- I have already placed the id's in the document for you.

```
<div id="navigation">
```

```
<a href="#" id="currentpage">
```



# List items

- Lets select the list items to style them.

```
#navigation ul li
{

}
```

# Changing our list

- First we will:
  - Change the font to verdana
  - Put some space in between each item using a margin.

```
#navigation ul li
{
font-family:verdana;
margin:20px 0px 0px 0px;
}
```



Or margin-top:20px

# Changing the bullets

- Then we want to change the bullets to be an arrow. I have an arrow .gif in the folder for today.
- To replace the bullets with the arrow:

```
#navigation ul li
{
font-family:verdana;
margin:20px 0px 0px 0px;
list-style-image:url(arrowleft.gif);
}
```

# Style the anchor

- To create styles to affect the anchor...

```
#navigation ul li a
{
}
}
```

# Text colour and decoration

- Next we will:
  - Set the text colour to be black instead of blue.
  - Remove that nasty underline which is caused by the anchor/link.

```
#navigation ul li a
{
color:#000000;
text-decoration:none;
}
```

# a:hover

- For the mouse over we want to change the text colour to red.

```
#navigation ul li a:hover  
{  
  color:#CC0000;  
}
```

# Current page link

- This will select the currentpage id.

```
#navigation ul li a#currentpage  
{  
  
}
```

# Current link

- For the current link we want to decide on a colour, border, and padding.

This will prevent it from changing to red when you roll over

```
#navigation ul li a#currentpage
{
color:#000000;
border: 1px solid #CC0000;
padding:2px;
}
```

This does not work in Safari!





# Adding comments to CSS



# It is important to comment

- One of the requirements for your CSS is that it has comments to explain all of your CSS code.
- This helps you understand what is going on. It also helps other members of your team understand the code.
- To comment:
  - `/* this is your text */`

# Adding a comment

`/* This style changes the colour of all links to pink and makes the background light blue*/`

```
a
{
color: #FF3333;
background: #006699;
}
```

- You should comment on every style section within your page.
  - I will be looking for this in your projects.
- We will see another use for commenting in a moment.



# Drop Down List



# Drop down list

- This type of drop down list involves a simple list with a little Javascript.
- It only seems to work well on a vertical, not a horizontal plane.
- Instead of typing all of this out, I have created the file for you.
- I will explain the code section by section.

# Drop Down List

- Open up [dropdown\\_nav.html](#)
- This drop down list starts with some really simple html code to make an unordered list.
- There are four of these already set up for you.
- Notice how there is an unordered list inside each list item.
  - This creates the submenus

```
<li><a>About</a>
    <ul>
        <li><a href="#">Our mission</a></li>
        <li><a href="#">History</a></li>
        <li><a href="#">Location</a></li>
    </ul>
</li>
```

# Drop down list

- First we must set the ids and classes within the html code.
- The entire code is surrounded by a div tag which allows us to style the whole area.
- Here we have given the main div with an id="menu".

```
<div id="menu">
</div>
```

# Drop down list

- Each section has also been given an id. The example below has an id="aboutSub"
- This name will be different in each section to give each of them an unique name.

```
<li><a >About</a>
    <ul id="aboutSub">
        <li><a href="#">Our mission</a></li>
        <li><a href="#">History</a></li>
        <li><a href="#">Location</a></li>
    </ul>
</li>
```



# Drop down list

- Open up the `dropdown_nav.css`
- I have attached the `dropdown_nav.css` to the `dropdown_nav.html` page.
- Let's have a look at the `.css`. Open up `dropdown_nav.css`.

# Drop down list

- The first item styles the menu with a background colour, text colour, font and width.
- The second item styles the entire unordered list. It will remove the bullet and also style the margin and padding for the unordered list.
- The fourth style removes that nasty underline from the `<a>` tags.

```
#menu
{
    background-color:#FFFF99;
    border:1px solid #3300CC;
    width:300px;
    font-family:arial;
}

#menu ul
{
    list-style-type:none;
    margin-left:5px;
    padding:2px;
}

#menu li a
{
    text-decoration:none;
}
```

# Drop down list

- The last style will ensure that the sub menus do not show when you first arrive on the page using the `display:none`.
- The font weight, colour and font size are also set here.

```
#aboutSub,#roomsSub,#activitiesSub,#facilitiesSub
{
    display:none;
    font-weight:normal;
    color:#fff;
    font-size:12px;
}
```

# Drop down list

- Each section has an href with a Javascript function.
- Notice how this function mentions the id that is listed in the unordered list.
- This basically says when you click on this anchor use the javascript to target this id.

```
<li><a href="javascript:showSub('aboutSub')">About</a>
    <ul id="aboutSub" >
        <li><a href="#">Our mission</a></li>
        <li><a href="#">History</a></li>
        <li><a href="#">Location</a></li>
    </ul>
</li>
```

# Drop down list

- The code that runs the javascript is on the top of the document.
- Basically this means that when you click on the anchors open and close the items that are listed by id.
  - This is done by **display="block" which overwrites the display="none"**.

```
function showSub(id) {
    if (document.getElementById(id).style.display == "none" || document.getElementById(id).style.display == "") {
        document.getElementById(id).style.display = "block";
    } else {
        document.getElementById(id).style.display = "none" ;
    }
}
```

# Drop down list disclaimer

- Although this is a useful feature, you should consider avoiding Javascript for main content.
  - The navigation is very important and should be accessible.
- Reason: This might make it difficult for accessibility devices to read the material.



# Development tips



# Development Tips Part One

- Always use the most standards compliant browser to code with, eg
  - Firefox or other Mozilla Browser
  - Opera V7.5 (now in beta for mac)
  - IE V 5+
- Never use Dreamweaver's Design view for previewing.
  - Always look at the browser!



# Development Tips Part Two

- Apply background colours and borders to elements whilst developing.
  - This will reveal the underlying structure and can be helpful when troubleshooting problems.
- Use Shortcuts whenever you can. It speeds things up and can make the code more readable, eg
  - #fff and #123 instead of #ffffff and #112233
  - Margin: 20px if all 4 margins are the same
  - For different margins (and paddings) you can use a similar format, eg;
    - Margin: 10px 20px 5px 15px.
    - The order is Top, Right, Bottom, Left (TROUBLE)
    - Border: 1px solid #000

# Links for lists

- This is a really good site for list/links tutorials:
  - Look under Listamatic and Listamatic 2
  - <http://css.maxdesign.com.au>

# Next lesson

- CSS
  - Creating a scroll box
  - Changing your cursor
  - Normal and disjointed rollover
  - Accessibility