|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Box Model** |  |

|  |
| --- |
| **Your Tasks** |
| * Apply the width, height, padding, border, and margin properties to create a box * Apply the width and height properties to style block elements * Have Ms. Pluska check off the above tasks * Apply the border property to style the block elements * Differientate between the use of class and id attributes * Have Ms. Pluska check off the above tasks * Interpret the specificity of css rules * Write code to *chain* selectors * Write code to select nested elements * Receive credit for the group portion of this lab |

* **Apply the width, height, padding, border, and margin properties to create a box**

The box model comprises the set of properties which define parts of an element that take up space on a web page. The model includes the content area’s size (width and height) and the element’s padding, border, and margin. The properties include:

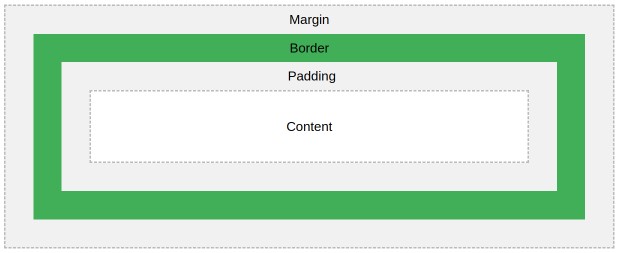
**Width and height** — specifies the width and height of the content area.

**Padding** — specifies the amount of space between the content area and the border.

**Border** — specifies the thickness and style of the border surrounding the content area and padding.

**Margin** — specifies the amount of space between the border and the outside edge of the element.

The image below is a visual representation of the box model.



A common tag used to organize content into boxes is the <div></div> tag. And, just as the tag implies, the <div> tag is used to *divide* content.

|  |  |
| --- | --- |
| **Index.html** | **Styles.css** |
| <!DOCTYPE html>  <html>  <head>  <title>My Box</title>  <link href="style.css" type="text/css" rel="stylesheet">  </head>  <body>  <div>  This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.  </div>  </body>  </html> | div {  background-color: lightgrey;  width: 300px;  border: 15px solid green;  padding: 50px;  margin: 20px;  } |
| **Output** | |
|  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Consider the code snippet below. Write a style rule for the salad-picclass to create the following output (do your best ;-))  Each property and their usage are summarized below,   |  |  |  | | --- | --- | --- | | **Property** | **Values** | **Description** | | height | 1em, 2em, 2.5em, etc | Specifies the height of the box | | width | 5%, 10%, 15%, etc | Specifies the width of the box | | padding | 1em, 1.5em, 2em, etc | Specifies the amount of space between the content area and the border | | margin | 1em, 1.5em, 2em, etc | Specifies the amount of space between the border and the outside edge of the element. | | border | 2em solid green,  1em dashed blue,  .5em dotted grey,  2.5em double purple,  etc | Specifies the thickness and style of the border surrounding the content area and padding. Notice there are three attributes associated with this property. | | |
| **Index.html** | **Styles.css** |
| <!DOCTYPE html>  <html>  <head>  <title>My Recipes</title>  <link href="style.css" type="text/css" rel="stylesheet">  </head>  <body>  <div class = “salad-pic”>  This is some salad content  </div>  </body>  </html> |  |
| **Ouput** | |
|  | |

* **Apply the width and height properties to style block elements**

Any element that takes up space on a web page is considered a box element. Consider the following elements: <p></p>, <h1></h1>, <h2></h2>, <img>. Because all these elements take up space, they can be styled using the same box properties as applied above.

If the width property is not specified for an element, the element will take up 100% of the screen. If the height property is not specified, the height will adjust to size of the content. This is illustrated below,

|  |  |
| --- | --- |
| **Index.html** | **Styles.css** |
| <!DOCTYPE html>  <html>  <head>  <title>Vacation World</title>  <link href="style.css" type="text/css" rel="stylesheet">  </head>  <body>  <h1>Taco recipe</h1>  <p class = “recipe-info”>  How to make tacos  </p>  <img src=”tacos.jpg”>  </body>  </html> | h1{  background-color:lightgray;  } |
|  | |

The below css rule for the <h1> element illustrates how to set the height and width of an element.

|  |
| --- |
| h1{  width:50%;  height: 2em;  background-color:lightgray;  } |

Notice in the above rule, the width is specified as a percentage and the height is specified in *em* units which is relative to the current size. This is important especially for scaling your website.

Setting the *height* and *width* of an image can be tricky, because setting both can cause the image to be distorted if they are not scaled properly. To avoid distortion, simply set one parameter (height or width) and the other will scale automatically,

|  |  |  |
| --- | --- | --- |
| **CSS rule** | **Original** | **Resized** |
| img{  width:10em;  } |  |  |

|  |  |
| --- | --- |
| In the Styles.css page,   1. write a rule for the class attribute *recipe-info* to confine the contents of the block element to only 25% of the width of the screen. Give it a background-color of lightgray to make the effect more visible. 2. Write a rule for the image so that it scales to a width of 5em. | |
| **Index.html** | **Styles.css** |
| <!DOCTYPE html>  <html>  <head>  <title>Vacation World</title>  <link href="style.css" type="text/css" rel="stylesheet">  </head>  <body>  <div id=”recipe-info”>This is my paragraph. This is my paragraph. This is my paragraph.</p>  <img src = “cat.jpg”>  </body>  </html> |  |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Apply the border property to style block elements**

A border is a line that surrounds an element, like a frame around a painting. Borders can be set with a specific width, style, and color.

***width*** — The thickness of the border. A border’s thickness can be set in pixels or with one of the following keywords: thin, medium, or thick.

**style** — The design of the border. Web browsers can render any of [10 different styles](https://developer.mozilla.org/en-US/docs/Web/CSS/border-style" \l "Values). Some of these styles include: none, dotted, and solid.

**color** — The color of the border.

|  |
| --- |
| p {  border: thick solid coral;  } |

In the example above, the border has a *width* of thick, a *style* of solid and a *color* of coral. All three properties are set in one line of code.

The default border is medium none color, where color is the current color of the element. If width, style, or color are not set in the CSS file, the web browser assigns the default value for that property.

|  |
| --- |
| p {  border: solid coral;  } |

In the above example, the border style is set to solid and the color is set to coral. The width is not set, so it defaults to medium.

|  |  |
| --- | --- |
| In Styles.css,   1. Add a dotted 1 pixel red border to the div element with id = banner 2. Add a thin solid coral border around each <li> element | |
| **Index.html** | **Styles.css** |
| <!DOCTYPE html>  <html>  <head>  <title>The Terminal</title>  <link href="Styles.css" rel="stylesheet">  </head>  <body>  <ul class="navigation">  <li>LOCAL</li>  <li>NATIONAL</li>  <li class="logo">THE TERMINAL</li>  <li>GLOBAL</li>  <li>OPED</li>  <li class="donate">DONATE</li>  </ul>  <div id="banner">  <h1>Conservation Efforts at Lake Tahoe Being Praised by Nation's Leaders</h1>  </div>  </body>  </html> |  |

* **Apply the padding and margin property to style block elements**

The space between the contents of a box and the borders of a box is known as padding. Padding is like the space between a picture and the frame surrounding it. In CSS, you can modify this space with the padding property.

|  |  |
| --- | --- |
| Consider the output below. Write the code in the Index.html for the section marked with a rectangle. Then write appropriate rule-sets that could be used to produce the output. Be mindful of when to use class attributes and id attributes. | |
| **Output** | |
| This text is purple and bold  This text is purple | |
| **Index.html** | **MyStyles.css** |
| <!DOCTYPE html>  <html>  <head>  <link rel = "stylesheet" type = "text/css" href = "MyStyles.css" />  </head>  <body>  </body>  </html> |  |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Interpret the specificity of css rules**

Specificity is the order by which the browser decides which CSS styles will be displayed. A best practice in CSS is to style elements while using the lowest degree of specificity, so that if an element needs a new style, it is easy to override.

ids are the most specific selector in CSS, followed by classes, and finally, tags. For example, consider the following HTML and CSS:

|  |
| --- |
| <h1 class="headline">Breaking News</h1> |

*Breaking News* would appear as *firebrick,* because classes override selectors. In otherwords they have a higher specificity.

|  |
| --- |
| h1 {  color: red;  }  .headline {  color: firebrick;  } |

In the example code above, the color of the heading would be set to firebrick, as the class selector is more specific than the tag selector. If an id attribute (and selector) were added to the code above, the styles within the id selector’s body would override all other styles for the heading. The only way to override an id is to add another id with additional styling.

Over time, as files grow with code, many elements may have IDs, which can make CSS difficult to edit, since a new, more specific style must be created to change the style of an element.

To make styles easy to edit, it’s best to style with a tag selector, if possible. If not, add a class selector. If that is not specific enough, then consider using an id selector.

|  |  |
| --- | --- |
| Predict the output of the code indicated with a rectangle. For colored text indicate the color in parentheses after the text. | |
| **Index.html** | **Styles.css** |
| <h1 id="article-title" class = "green all-caps”>Vacation Spots</h1>  <h2 class="green">1. Florence, Italy</h2>  <h5 class="all-caps">Top Attractions</h5>  <ul>  <li>Museums</li>  <li>Bike Tours</li>  <li>Historical Monuments</li>  </ul>  <h2 class="green">2. Beijing, China</h2>  <h5 class="all-caps">Top Attractions</h5>  <ul>  <li>Biking</li>  <li>Historical Sites</li>  <li>Restaurants and Dining</li>  </ul> | #article-title{  color: blue;  text-decoration: underline overline;  }  h1 {  color: red;  }  .green{  color: green;  }  .blue{  color: blue;  }  .all-caps{  text-transform: uppercase;  } |
| **Output** | |
|  | |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Write code to chain selectors**

When writing CSS rules, it’s possible to require an HTML element to have two or more CSS selectors at the same time.

This is done by combining multiple selectors, which we will refer to as chaining. For instance, if there was a *.special* class for <h1> elements, the CSS would look like:

|  |
| --- |
| h1.special {  } |

The code above would select only the <h1> elements that have a class of *special*. If a <p> element also had a class of *special*, the rule in the example would not style the paragraph.

|  |  |
| --- | --- |
| Predict the output of the code indicated with a rectangle. For colored text indicate the color in parentheses after the text. Do the same for bolded text. | |
| **Index.html** | **Styles.css** |
| <h1 id="article-title" class = "green special”>Vacation Spots</h1>  <h2 class="green special">1. Florence, Italy</h2>  <h5 class="first-letter-caps attractions">Top Attractions</h5>  <ul>  <li class=“attractions”>Museums</li>  <li class=“attractions”>Bike Tours</li>  <li class=“attractions”>Historical Monuments</li>  </ul>  <h2 class="green special">2. Beijing, China</h2>  <h5 class="first-letter-caps attractions">Top Attractions</h5>  <ul>  <li class=“attractions”>Biking</li>  <li class=“attractions”>Historical Sites</li>  <li class=“attractions”>Restaurants and Dining</li>  </ul> | #article-title{  text-decoration: underline overline;  color: red;  }  h1.special {  font-style: italic;  text-transform: uppercase;  }  .green{  color: green;  }  .blue{  color: blue;  }  .first-letter-caps{  text-transform: capitalize;  }  li.attractions{  font-weight: bold;  font-style: italic;  } |
| **Output** | |
|  | |

* **Write code to select nested elements**

In addition to chaining selectors to select elements, CSS also supports selecting elements that are nested within other HTML elements. For instance, consider the following HTML:

|  |
| --- |
| <ul class='main-list'>  <li> ... </li>  <li> ... </li>  <li> ... </li>  </ul> |

The nested <li> lements are selected with the following CSS:

|  |
| --- |
| .main-list li {  } |

In the example above, *.main-list* selects the *.main-list* element (the unordered list element). The nested <li> are selected by adding *li* to the selector, separated by a space, resulting in *.main-list li* as the final selector (note the space in the selector).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| For the list below, write a css rule that could be used to style the list as shown. In addition to the css properties you learned previously, you can also use the properties below,   |  |  |  | | --- | --- | --- | | **Property** | **Values** | **Description** | | float | left, right, none | Places elements side by side | | list-style-type | none, disc, circle, square, decimal, etc | Removes or changes the default bullet in a list | | padding | .5 em, 1em, 1.5em, etc | Adds space around the elements content | | |
| **Index.html** | **Output** |
| <ul class = "nav">  <li><a class="active" href="#main">Home</a></li>  <li><a href="#news">News</a></li>  <li><a href="#contact">Contact</a></li>  <li><a href="#about">About</a></li>  </ul> |  |
| **Styles.css** | |
|  | |

* **Receive Credit for the group portion of this lab**



* Indicate the names of all group members.
* Have Ms. Pluska check your Intro to CSS tasks
* Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
* Do not submit your lab until you have Ms. Pluska’s (or her designated TA’s) signature

\_\_\_\_\_\_\_