**Thinkful Prep Course Study Guide**

**<!DOCTYPE>**

* Instruction to web browser about what version of HTML the page is written in.
* <!DOCTYPE html>

**<head> element**

* container for metadata (data about data)
* HTML metatdata is data about the HTML document (not displayed)

e.g. <title>, <style>, <meta>, <links>, <script>, <base>, <footer>

**Semantic HTML and why is it important?**

* clearly describes its meaning to both the browser and the developer
* <form>, <header>, <div>, <p>, <table>, <section>

**Link to External File**

* <link rel="stylesheet" type="text/css" href="./main.css">

**CSS ID vs CSS Class**

* Id is unique in nature while class can be used for multiple elements.
* You can only have one id per element but you can use multiple classes per element
* Id gets a higher priority than class

e.g. <p id= “main” class = “class1 class2 class 3”> Blah </p>

e.g. #main {

}

vs.

.class1 {

}

**If else and else if**

* Use if to specify a block of code to be executed, if a specified condition is true
* Use else to specify a block of code to be executed, if the same condition is false
* Use else if to specify a new condition to test, if the first condition is false

**Object.keys()**

* Returns an array of a given object’s own property names
* Sample: object.keys(obj)

**Primitive data type**

* Boolean
* Null
* Undefined
* Number
* BigInt
* String
* Symbol
* Object

**CSS box-sizing property**

* Element padding and border are included in the width and height
* If you set a certain width or height it will be matched vs in the past

**Past:**

width + padding + border = actual visible/rendered width of an element's box

height + padding + border = actual visible/rendered height of an element's box

**Block-level vs Inline Elements**

* Block-level always starts on a new line and takes up the full width available.
* Inline does not start on a new line and only takes up as much width as necessary.

**<form>**

* Document section – interactive controls for submitting information to a web server.
* HTML form for user input.

**Sample:**

* <input>
  + Input info.
  + <input name= “firstname” type= “text” >
* <select>
  + Defines a drop-down list
  + Use **selected** for pre-selected option
  + Use **multiple**  to allow user to select more than one option
  + Use **size** for number of visible value
    - <select name= “foods” size= “3” >
    - Three initial options will be displayed
  + **<select name= “foods”>**

**<option value= “sushi”>Sushi</option>**

**<option value= “kbbq”> kbbq </option>**

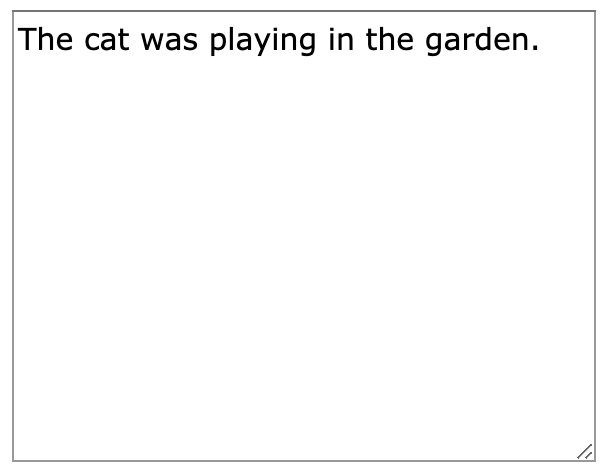
**<option value= “pasta” >Pasta</option>**

**</select>**

* <option>
  + Defines an option that can be **selected**
  + Use selected attribute to have a pre-selected option
  + <option value= “sushi” selected> Sushi </option>
* <textarea>
  + Defines a multi-line input field
  + <textarea name= “message” rows= “10” cols= “30”>

The cat was playing in the garden.

</textarea>



* <button>
  + Defines a clickable area
  + <button type = “button” onclick= “alert(‘Hello World!’)”>Click Me!</button>
* <datalist>
  + Specifies a list of pre-defined options for input controls
* <output>
  + Defines the result of a calculation

**Media queries**

* Apply different styles for different media types/devices
* @media rule to include a block of css properties if a certain condition is true
* sample:

**@media only screen and (max-width: 600px){**

**body{**

**background-color: lightblue;**

**}**

**}**

**Role**

* Adds to semantic