Anything **orange** is code that always stays the same. Wherever you see black text with a gray background you can choose what to type.

Syntax Reference

|  |
| --- |
| **What is syntax?** |
| In coding, **syntax** is the set of rules that describe the combination and sequence of symbols (including letters and numbers) that form a correctly structured program for a specific language. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Name** | **Example 1** | **Example 2** |
| / | Forward Slash | <body><**/**body> | <img src="https:**//**brit.co**/**1.jpg"> |
| - | Hyphen | font**-**size: 20px; | $(".two").css("background**-**color","10px"); |
| " " | Quotes | <img src = **"**awesome.jpg**"**> | $(**"**.div1**"**).hide(); |
| < > | Angle Brackets | **<**head**> <**/head**>** | **<**!DOCTYPE html**>** |
| { } | Curly Brackets | p **{**  color: blue;  **}** | if(password === "1234") **{**  $(".result").show();  **}** |
| [ ] | Square Brackets | let favColor = colors**[**1**]**; | let colors = **[**"red", "blue", "yellow"**]**; |
| ( ) | Parentheses | $**(**"h1"**)**.hide**()**; | for**(**let song of playlist**)**{  $("ol").append(`<li> ${song} </li>`);  } |
| ; | Semicolon | let word = "hello"**;** | colors.push("purple")**;** |
| : | Colon | .two {  font-size**:** 20px;  } | #two {  width**:** 300px;  } |
| . | Dot | $("**.**yourclass")**.**text("hi"); | **.**yourClass {  color: red;  } |
| ` | Backtick | $(".message").append(**`**Hi**`**); | $("h2").text(**`**Hi, ${name}. Welcome**`**); |

|  |  |
| --- | --- |
| **Comments** | |
| Comments allow you to include information for other coders and are ignored by the computer. | |
| **<!--** These are comments in the code. **-->** | Add a comment in HTML |
| **//** One line of comments. | Add one line comment in JavaScript |
| **/\*** Type a long section in the comments **\*/** | Add a section of comments in JavaScript and CSS |

Environment Set Up

|  |  |
| --- | --- |
| **Resource Linking** | |
| **<link rel="stylesheet" href="../css/style.css">** | A style sheet is used to define the style for many HTML pages. To use an external style sheet, add a link to it in the <head> section of the HTML page. |
| **<script src="../script/script.js"> </script>** | A script file is used to add Javascript for many HTML pages. To use an external script, add a <script> tag at the end of the <body> tag. |

|  |  |
| --- | --- |
| **Command Line** | |
| **pwd** | The **pwd** command tells you in which directory you are currently located. **pwd** stands for print working directory. |
| **ls** | The ls command is used for viewing files and directories. The **ls** command, shows all of the major directories filed under a given file system. **ls** stands for list. |
| **cd** <directory name>  **cd** ..  **cd** ~  y - | The **cd** command will allow the user to change between directories. .. represents the parent directory and **~** represents the root directory. Use **-** to go back to the last directory. **cd** stands for change directory. |
| **mv** | The **mv** command - move - allows a user to move a file to another directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents".  **mv** stands for move. |
| **mkdir** <directory name> | The **mkdir** command allows the user to make a new directory. **mkdir** stands for make directory. |
| **touch** <file name> | The **touch** command - a.k.a. the make file command - allows users to make files. Just as the mkdir command makes directories, the **touch** command makes files. |
| **rm**  **rmdir** | The **rm** command like the **rmdir** command is meant to remove files. The **rmdir** command will remove directories and files within them. The **rm** command will delete any created files. **rm**  stands for remove and **rmdir** stands for remove directory. |
| **clear** | The **clear** command clears the screen and wipes the board clean. |
| **python3 -m http.server 3000** | This command is used in Gitpod to open the browser view of your code. |

|  |  |
| --- | --- |
| **Git and Github** | |
| Screen Shot 2017-08-31 at 11.32.37 AM.png | **Forking** a repository creates a new copy of the repository on your GitHub profile. |
| git clone <your repo link here> | **Cloning** a repository to your local environment makes a local copy of your repository. |
| git status  git add .  git commit -m "<your message here>"  git push | To put your local changes on GitHub, first check the **status** of what changes you have made, next stage all of your changes by using the **add** command, then **commit** all of your changes and finally **push** your changes to the repo. |

HTML

|  |  |
| --- | --- |
| **Basic Structure of an HTML document (or webpage)** | |
| **<!DOCTYPE html>**  **<html>**  **<head>**  **<title>**My Page**</title>**  **</head>**  **<body>**  <p>My first paragraph</p>  **</body>**  **</html>** |  |

|  |  |  |
| --- | --- | --- |
| **HTML Element** | | |
| **an individual component of a webpage** | | |
| Opening Tag | Content | Closing Tag |
| **↓** | **↓** | **↓** |
| **<**p**>** | This is a paragraph | **</**p**>** |

|  |  |  |  |
| --- | --- | --- | --- |
| **HTML Elements** | | **Code Example** | **Output** |
| [**paragraph**](https://popcode.org/?snapshot=e2361707-c368-4121-8897-8750105f30fb) | **<p></p>** | **<p>**This is a paragraph.**</p>** |  |
| [**heading**](https://popcode.org/?snapshot=29f038a4-9ffc-461b-950a-e686f10af498) | **<h1></h1>**  **<h3></h3>**  **<h6></h6>** | **<h1>**Heading level 1**</h1>**  ...  **<h6>**Heading level 6**</h6>** |  |
| [**ordered list**](https://popcode.org/?snapshot=15d42867-010b-47a0-a659-379cd293ec6e)  **(with numbers)** | **<ol>**  **<li></li>**  **</ol>** | **<ol>**  **<li>**George Washington**</li>**  **<li>**John Adams**</li>**  **</ol>** |  |
| [**unordered list**](https://popcode.org/?snapshot=581f408c-b53f-4e54-9c7d-90f1ad329717)  **(with bullets)** | **<ul>**  **<li></li>**  **</ul>** | **<ul>**  **<li>**George Washington**</li>**  **<li>**John Adams**</li>**  **</ul>** |  |
| [**button**](https://popcode.org/?snapshot=ee054575-ee6e-4dba-9b4e-1038f52ec0e4) | **<button></button>** | **<button>**Click Me**</button>** |  |
| [**div**](https://popcode.org/?snapshot=0670ec94-6a39-4465-91d6-eeb9d58e8ded) | **<div></div>** | **<div>**This is a div**</div>** |  |
| [**input**](https://popcode.org/?snapshot=68147a4c-6225-425a-a9f5-593f06daa687)**\*\*** | **<input>** | **<input>** | Screen Shot 2017-08-08 at 2.55.31 PM.png |

**\*\*Self-closing: Does not have a closing tag.**

|  |  |  |
| --- | --- | --- |
| **Nesting and Indentation in HTML** | | |
| In coding, nesting is when you put one tag completely inside another tag's content.  Indentation helps you organize your code and makes it more readable. Remember to indent (press the tab key) when you're nesting an element inside another. | **<div>**  **<h1>**Weekday**</h1>**  **<p>**Monday**</p>**  **</div>** | On the left, the <h1> and <p> tags are nested within the <div> tags. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **HTML Attribute Syntax** | | | | | |
| An attribute adds extra information to an HTML element. In HTML syntax, attributes are part of an HTML opening tag. | Opening  tag attribute Closing tag  **↓ ↓ ↓**  **<a href="**www.google.com**">** Google it!**</a>** | | | | |
|
|

|  |  |  |  |
| --- | --- | --- | --- |
| **HTML elements w/ attributes** | | **Code Example** | **Output** |
| [**Image**](https://popcode.org/?snapshot=8631f268-a673-4af5-8ca9-84f260d89ae1) **\*\*** | **<img src=" ">** | **<img src="**https://imgur/cats.png**">** |  |
| [**Link**](https://popcode.org/?snapshot=d356d014-fcf7-4866-a18d-930797b12021)  **(anchor tag)** | **<a href=" "></a>** | **<a href="**<https://www.google.com>"> This is a link to Google**</a>** |  |
| [**Adding classes**](https://popcode.org/?snapshot=df62458f-5571-4a66-9247-a6a6d0d277b1)**\*** | **class=" "** | **<**h1 **class="**thisClass**">**text**</**h1**>** |  |
| [**Input w/ placeholder**](https://popcode.org/?snapshot=418f1c7a-cef7-46cd-b8ce-8bb749f3fc1b)**\*\*** | **<input placeholder=" ">** | **<input placeholder="**type here**">** | Screen Shot 2017-08-08 at 3.12.43 PM.png |

**\*You can add a class to any HTML element (<img>, <a>, <li>, <ul>, etc.)**

**\*\*Self-closing: Does not have a closing tag.**

|  |  |  |  |
| --- | --- | --- | --- |
| **id vs. class** | | | |
| **ids** and **classes** are HTML attributes that you can add to HTML elements. | | | |
| assign [**class**](https://popcode.org/?snapshot=f426364c-da4c-4205-9e26-42aefb53acc6) in HTML | **class=" "** | **<**div **class="**myClass**">** | * The symbol that you use to select a class is a **.** (dot). * You can use the same class on multiple HTML elements. * You can use more than one class on the same HTML element * Classes are case-sensitive. |
| select [**class**](https://popcode.org/?snapshot=f426364c-da4c-4205-9e26-42aefb53acc6) in CSS | **.** | **.**myClass {  text-align: right;  } |
| assign [**id**](https://popcode.org/?snapshot=bc2f8c85-7303-435f-8eca-118ce4cacb84) in HTML | **id=" "** | **<**div **id="**myID**">** | * The symbol that you use to select an id is a **#** (hashtag). * Each HTML element can only have one id. * Each page can only have one HTML element with that id. * ids are case sensitive. |
| select [**id**](https://popcode.org/?snapshot=bc2f8c85-7303-435f-8eca-118ce4cacb84) in CSS | **#** | **#**myID {  color: blue;  } |

CSS

|  |  |
| --- | --- |
| **CSS Syntax** | |
|  | 1. **Selector:** Identifies the parts of your page that will be affected by this CSS rule. You can select using the tag name, id, or class. 2. **Property:** The thing you want to change for the element(s) you’ve selected. Each property should be followed by a : (colon) . 3. **Value:** What you want to set this property to. Each value should be followed by a ; (semicolon) . |
|
|

|  |  |  |  |
| --- | --- | --- | --- |
| **CSS Properties and Values** | | | |
| **Change ...** | **Code Examples** | **Output** | **What it does** |
| [**text**](https://popcode.org/?snapshot=3130cd9f-4ee2-4f22-b461-a9646b12afa3) | **font-family:** "Comic Sans";  **font-size:** 12px;  **text-align:** center;  **color:** blue; |  | Changes the **font** to Comic Sans.  Changes **font size** to 12 pixels.  **Aligns the text** to the center.  Changes the **font color** to blue. |
| [**color**](https://popcode.org/?snapshot=1ea2ec7e-7997-40e0-82bc-bd50bdf96cac) | **background-color:** #000000;  **color:** yellow; |  | Changes the **background color** to the hex code #000000, which is black.  Changes the **font color** to a specific shade of yellow. |
| [**background**](https://popcode.org/?snapshot=44989e68-b79d-470f-bd44-6793144e3ab0) | **background-color:** pink;  **background: url("**ex.png"**);** |  | Changes the **background color** to pink.  Changes the **background to an image** w/ URL "www.ex.png" |
| [**size**](https://popcode.org/?snapshot=5834c89a-974e-400c-b949-85043aa39c82) | **width:** 50px;  **width:** 50%;  **font-size:** 20px; |  | Changes the **width** to 50 pixels.  Changes the **width** to 50% of the screen, whatever the size.  Changes the **font-size** to 20 pixels. |
| [**border-radius**](https://popcode.org/?snapshot=f6c0dbbd-e428-4624-8fd6-67c8f4c83ebd) | **border-radius:** 500px; |  | Makes the corners of a div slightly rounded |
| [**opacity**](https://popcode.org/?snapshot=20929e2d-d93f-4ea1-98a0-6a5b5e2818c0) | **opacity:** 0.5; |  | Make the whole div and all its content semi-transparent. Values can be between 0 and 1. |

CSS Layout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CSS Box Model** | | | | |
| box-model-alt-small.png | | | | All HTML elements are shaped like boxes.  Each box has a content area (text, image, link, etc.) and optional surrounding padding, border, and margin areas. |
|
|
|
| **Change...** | **Code Examples** | **What it does** | | |
| [**content**](https://popcode.org/?snapshot=ab5bfb24-0ac4-46b8-ac67-fdbddd680ae4) | <p>hey</p>  <img src="cat.jpg"> | Any HTML element (paragraph, image, link, etc.). *This is not a property.* | | |
| [**padding**](https://popcode.org/?snapshot=f7f247b6-94ad-46af-9638-82ea6bb4fbe1) | **padding:** 20**px;** | Spacing between the content and border. | | |
| [**border**](https://popcode.org/?snapshot=0d3f9278-8570-44d6-9a45-6b38acf1ff7e) | **border:** 2**px** solid red;  **border:** 10**px** dotted yellow;  **border:** 50**px** groove red; | Surrounds the padding. Think of it like an outline around a picture.  Border takes 3 values that define how thick the border is, the style, and the color. | | |
| [**margin**](https://popcode.org/?snapshot=f8a23964-c16a-4b33-af52-daca78d8690f) | **margin:** 150**px;** | Spacing between the border of this element and the start of another element. | | |
| **If we define only one value, it will be applied to all 4 sides of the content.** | | | | |
| **padding:** 10px**;** | | 10px padding applied to all sides | | |
| **We can define a different value for all 4 sides (top, right, bottom, left).** | | | | |
| **margin:** 10px 20px 30px 40px**;** | | 10px margin to **top** of content, 20px margin to r**ight** of content, 30px margin to **bottom** of content, 40px margin to **left** of content | | |
| **You can define a value for a specific side of the property.** | | | | |
| **padding-left:** 100px;  **margin-top:** 25px; | | 100px padding to the left only  25px margin to the top only | | |
| **Similarly, you define a border for a specific side of the box.** | | | | |
| **border-right:** 10px solid black;  **border-bottom:** 20px dotted green; | | 10px solid black border to the right only  20px dotted green border to the bottom only | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CSS Flexbox** | | | | | | |
| When using flexbox, turn on **flexbox** for the parent element, using the property display and value flex. | | | | | | .container {  display: **flex**;  } |
| **Arranged in a row** | | | | | | |
| Use the **justify-content** property to align the child elements to a specific side. | | | | | | |
| **Change...** | **Code Examples** | | | **What it does** | | |
| [**flex-start**](https://popcode.org/?snapshot=eeac3d31-7361-4422-8fb2-df477d25121f) | .container {  display: flex;  **justify-content: flex-start;**  } | | |  | | |
| [**center**](https://popcode.org/?snapshot=49d4c27c-47bc-48f0-8f46-bce661e71cad) | .container {  display: flex;  **justify-content: center;**  } | | |  | | |
| [**flex-end**](https://popcode.org/?snapshot=94a421bb-2582-4c8c-9934-0bf9d4fb0184) | .container {  display: flex;  **justify-content: flex-end;**  } | | |  | | |
| [**space-between**](https://popcode.org/?snapshot=05c42872-8c22-487e-8e6d-d32981d901e8) | .container {  display: flex;  **justify-content: space-between;**  } | | |  | | |
| [**space-around**](https://popcode.org/?snapshot=b85448de-d292-42a1-acd5-d77641b68f88) | .container {  display: flex;  **justify-content: space-around;**  } | | |  | | |
| **Arranged by columns** | | | | | | |
| **Step 1:** Turn on flexbox for the parent element (see above).  **Step 2:** Define the width for the child elements. | | | | | | .section **{**  **display: flex;**  **}**  .left **{**  **width:** 25%**;**  **}**  .right **{**  **width:** 75%**;**  **}** |

jQuery

|  |  |  |
| --- | --- | --- |
| **jQuery Syntax** | | |
| **<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>** | | |
|  | | |
| 1. The **$** symbol lets the computer know that you are using jQuery, the JavaScript library. 2. The **selector** is exactly like a CSS selector. It selects or identifies the element on the page. You can use the name of an **HTML element** (<p>, <h1>, <body>) or **class** (.results, .div1). | 3.  4. | The jQuery **action()** to be performed on the element. See more options below.  The **argument** tells more information about how to change the element. Sometimes, there is no argument, i.e. .show(), and sometimes, there are several arguments, i.e. .css(). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Click Handler** | | | |
| 1  2  3 | **$("**.yourClass**").click(function(){**  $("img").hide**;**  **});** | 1  2  3 | When the user **clicks** the HTML element with a class yourClass  The HTML element img **hides**.  Closes the **click handler**. |

|  |  |  |
| --- | --- | --- |
| **Action** | **Code Example** | **What it does** |
| [**Show**](https://popcode.org/?snapshot=34e5f8bc-26dd-4734-88f9-bfd8ea4b5075) **an element.**  [**Hide**](https://popcode.org/?snapshot=34e5f8bc-26dd-4734-88f9-bfd8ea4b5075) **an element.** | **$("**.yourClass**").show();**  **$("**.myClass**").hide();** | **Show** all HTML elements with the class yourClass.  **Hide** all HTML elements with the class myClass. |
| **Replaces the content of an** [**HTML**](https://popcode.org/?snapshot=34e66089-f114-4940-a9bf-38915cf6816a) **element.** | **$("**body**").html(**`<p>Hi!</p>`**);** | In the **HTML**, replace the content inside the <body> with <p>Hi!</p>. |
| **Add/change the** [**CSS**](https://popcode.org/?snapshot=5f003676-69c8-40de-9a6a-2e1f88247c70)**, or style, of an element.** (Change the property and/or value) | **$("**.container**").css("**color**", "**red**");** | Add/change the **CSS** property color to red for all HTML elements with a class of container. |
| **Add/change the** [**text**](https://popcode.org/?snapshot=1a8187e7-c76b-4152-a447-cccd5d7fbd2b) **in an element.** | **$("**.results**").text("**You won!**");** | Add/change the **text** to "You won!" for the HTML element with the class results. |
| **Add/change an** [**HTML attribute**](https://popcode.org/?snapshot=603f4793-2a10-4f68-8570-020f5d31bbd7)**.** (See page 4 for info about attributes.) | **$("**img**").attr("src", "**[http://pics.com/blah.jpg](http://coolpics.com/rhinoceros.jpg%27)**");** | Add/change the **HTML attribute** src, or source, to that URL for all <img> tags. |
| **Action** | **Code Example** | **What it does** |
| [**Append**](https://popcode.org/?snapshot=8b20b95f-758a-42d2-ba7c-003d78b6fd25) **(add) content to an element.** | **$("**div**").append**(**"**Bye!**");** | **Append**, or add, the text "Bye!" to the **end** of the all the <div> tags. |
| **Retrieve a** [**value**](https://popcode.org/?snapshot=183498e9-5024-4e84-9a1c-59873eff4fa2) **from an <input>** | **let** firstName **= $("**input**").val();** | Retrieve a **value** from the input tag and store it in a variable named firstName. |

JavaScript

|  |  |
| --- | --- |
| **Variable Syntax** | |
| [**Variables**](https://popcode.org/?snapshot=ef5b4ded-a8b9-44c1-a4c7-0dcf7b0e0bd2) are containers for storing data values. | |
|  | **Parts:**   1. The keyword let declares a variable, or creates a new variable. 2. The variable name winner 3. The equal **=** sign assigns a value.   **Line 1: Declares a variable** and gives it the name, winner.  **Line 2: Assigns a value** to the variable winner.  **Line 3:** **Re-assigns** a different value to the variable winner. The value of winner is no longer "Taylor Swift". It is now "Beyonce".  **Line 4:** **Declares a variable** named loser and **assigns it a value** "Kanye" all in one line of code! |

|  |  |  |
| --- | --- | --- |
| **Data Types** | | |
| [**Number**](https://popcode.org/?snapshot=458fb853-9701-47a2-9d94-046bdfdb9cda) | Duh… you know what a number is… No quotation marks, may start with a + or -, may include a decimal. | **let** temperature **=** -1**;**  **let** price **=** 5.99**;** |
| [**String**](https://popcode.org/?snapshot=6aa1289c-a0b2-4f2b-87d4-b9c8ea1b9ce1) | Always inside single (**''**) or double (**""**) quotes. Can be an empty string (" "). Can include letters, spaces, symbols, numbers… as long as it's in quotes. | **let** greeting **= "**Joliz is here!**";**  **let** space **= ' ';**  **let** price **= "**5.99**";** |
| [**Boolean**](https://popcode.org/?snapshot=0327ecc7-5306-4473-b241-8f1148073f69) | **true** or **false** has no quotation marks | **let** codeNationIsAmazing **= true;**  **let** brunoMarsOverrated **= false;** |
| [**Array**](https://popcode.org/?snapshot=d313da7e-2415-46d0-9ac6-5cff245b6f7f) | A list of multiple values separated by commas inside square brackets **[ ]** | **let** oddNumbers **= [**1**,** 3**,** 5**,** 7**,** 9**];**  **let** airport = ["JFK", "LGA", "SFO"]; |
| [**Object**](https://popcode.org/?snapshot=0b590376-3528-4f8f-acf0-3f8e040c495b) | A collection of properties separated by commas inside curly brackets **{ }**. A property is an association between a name (or key) and a value separated by a colon : | **let student = {**  **name : "Erica",**  **school : "Columbia HS",**  **};** |

|  |  |  |
| --- | --- | --- |
| **Input and .val** | | |
| **JavaScript** | **Code Example** | **What it does** |
| [**input field**](https://popcode.org/?snapshot=257b97a8-f607-4d40-8d27-79fe959c13ff) | 1 **<input id="**myID**">** 2 **<button id="**yourID**">** Go! **</button>** | Creates an **input** field in HTML with an id, myID.  Creates a button that says Go! with an id, yourID. |
| [**input.val**](https://popcode.org/?snapshot=7a344013-594c-4462-a9be-c9b1660126ee) | 1 **$("#**yourID**").click(function(){**  2 **let** message **= $("**#myID**").val();**  3 **});** | When the user clicks the HTML element with an id yourID (which is the button),  retrieve the value from the **input** field. |

|  |  |  |
| --- | --- | --- |
| **Math Library** | | |
| **JavaScript** | **Code Example** | **What it does** |
| [**Math.random();**](https://www.w3schools.com/js/js_random.asp) | 1 **let num = Math.random();**  2 **console.log(num);** | Returns a random number between 0 (inclusive), and 1 (exclusive) |
| [**Math.floor();**](https://www.w3schools.com/jsref/jsref_floor.asp) | 1 **let num = Math.random();**  2 num = Math.floor(num);  3 **console.log(num);** | Round a number downward to the nearest integer. Example: Math.floor(1.4) = 1 |
| [**Math.ceil();**](https://www.w3schools.com/jsref/jsref_ceil.asp) | 1 **let num = Math.random();**  2 num = Math.ceil(num);  3 **console.log(num);** | Round a number upward to the nearest integer. Example: Math.ceil(1.4) = 2 |

|  |  |  |
| --- | --- | --- |
| **Conditional Syntax** | | |
| **Conditional statements** are used to perform different actions based on conditions. | | |
| 1) [**if statement**](https://popcode.org/?snapshot=13fbc7ee-51d3-4191-90e8-8ccdfa30d99a) |  | **Conditional Statements** can be created using a combination of the three statements on the left.   1. The keyword if indicates that this is an **if statement** 2. The **condition** goes between the **( )**; the result should be true or false. If you need multiple conditions, you will need an else-if statement. 3. **Curly brackets** indicate the body of the condition statement. 4. **Body** - This is the code that executes if the condition is true. If the condition is false, then the code will NOT execute. 5. The keyword else if indicates an **else-if statement**. 6. The keyword else indicates an **else statement**. No condition with an **else statement**   An **if statement** is required to create a conditional statement, however an **else-if statements** and **else statements** are as needed. You can also use more than one **else-if statement**. |
| 2)[**else-if statement**](https://popcode.org/?snapshot=b204fa3b-ea20-4137-b51a-a178e6020c8a) |  |
| 3) [**else statement**](https://popcode.org/?snapshot=35d87d7f-640b-4810-98fe-705dac4b5268) |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Basic Conditional Statement Example** | | | |
| 1  2  3  4  5  6 | **let** number **=** 3**;**  **if (**number < 5**) {**  **$("**.btn**")**.hide**();**  **} else {**  **$("**.btn**")**.show**();**  **}** | 1  2  3  4  5  6 | Declare variable named number and assign it a value of 3.  **If** the variable number is less than 5...  Hide the HTML element with the class btn.  Or **else**…  Show the HTML element with the class btn  End of **conditional statement**. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Conditional Statement with Multiple Conditions Example** | | | |
| 1  2  3  4  5  6  7  8 | let num = 11;  if (num < 5) {  console.log("Less than 5");  } else if (num < 10) {  console.log("Less than 10");  } else {  console.log("Greater than 10");  } | 1  2  3  4  5  6  7  8 | Declare variable named num and assign it a value of 11.  **If** the variable value is less than 5...  Print "Less than 5" to the console  **Else if** the number is less than 10...  Print "Less than 10" to the console  **Else**  Print "Greater than 10 to the console  End of **conditional statement**. |

|  |  |  |  |
| --- | --- | --- | --- |
| [**Compound Conditional**](https://popcode.org/?snapshot=35007ccb-914b-48d7-835b-e8fded3aefc6) **Statement Example** | | | |
| 1  2  3  4  5 | if (age > 16 && passedTest===true) {  console.log("you can drive.");  } else {  console.log("no driving yet.");  } | 1  2  3  4  5 | **If** the value of age is greater than 16 AND(**&&**) passedTest is true  Log "you can drive." to the console.  **Else**  Log "you can't drive." to the console  End of **conditional statement**. |

|  |  |  |
| --- | --- | --- |
| **Array Syntax** | | |
| An [**array**](https://popcode.org/?snapshot=5149544d-94b6-4054-8982-781bb085713a) is a way to store more than one value at a time. Think of it like a list. | | |
| 1. **Declare a variable** called classNames. 2. An **array** is a list of values — they can be numbers, strings, or a combination of different value types. Square brackets start and end an **array**. 3. Each **array element**, or individual item (i.e. "History") in the array, is separated by a comma. | | |
|  | | D. Arrays have properties that you can use, including **length**. Use the name of the array, in this case, classNames + .length to represent the **length**. The **length** of this array is 3, because there are 3 total elements in this array. The value of arrayLength is 3. |
|  | | E. To use a specific array element, use the **array index**. It (see above) represents the location of an array element and always begins with 0. The **array index** uses the name of the array + **[**the **index** surrounded by square brackets**]**. The value of favElement is "English". |

|  |  |  |
| --- | --- | --- |
| **Object Syntax** | | |
| An [**object**](https://popcode.org/?snapshot=1382f950-eaf8-4702-8e52-2cf0d2425e94) is a way to store data as properties with keys and values. | | |
| 1. **Declare a variable** called classroom. 2. An **object** is a collection of **key/value pairs** separated by commas. 3. Each **key/value pair**, in the **object** has a unique name or **key** used to identify it. 4. Each key has a corresponding **value** separated by a colon **:** . | | |
|  | | 1. To access a specific value from an object in **dot notation**, use the corresponding **key**. The value of myTeacher is "Ms. C". |
|  | | 1. To access a specific value from an object in **bracket notation**, you can also use the corresponding **key.** The value of myClassSubject is "English". |
|  | | To change the value of a property you can use either **dot notation** or **bracket notation** and assign the **property** a new **value**. |

|  |
| --- |
| **forEach Loop Syntax** |
| **Loops** repeat an action. A [**forEach loop**](https://popcode.org/?snapshot=68f3bc0b-00ed-4256-821b-be9b5fd3e6e8) repeats until all elements in an array have been selected |
| **anArray.forEach(function(arrayElement) {**  **//loop body goes here**  **});** |
| 1. The **forEach loop** is used to **iterate** over an **array**. It can be any array with any number of values or **array elements** 2. **Iterating** over an array means looping over the **elements** of the **array** and selecting each **element** one at a time. This **variable** represents the **array element** that is currently selected. You can name this variable anything. 3. The **forEach body** goes between the curly brackets. This block of code executes every time an **element** is selected from the array. Usually the code is doing something to the **array element** that is currently selected. |

|  |  |  |  |
| --- | --- | --- | --- |
| **For Each Loop Example** | | | |
| 1  2  3  4  5 | **let courses = ["history", "math", "science"];**  **courses.forEach(function(course) {**  **$("#schedule").append("<p>" + course + "</p>");**  **});** | 1  2  3 | Creates an **array** to iterate over.  Uses a **forEach** loop to iterate over the array.  The variable course represents the array element that is currently selected. The first time the loop runs course is equal to "history", the second time it is "math", and the third time it is "science". |

|  |  |  |
| --- | --- | --- |
| **String Method and Properties** | | |
| **Action** | **Code Example** | **What it does** |
| **. length property returns the length of a string** | let txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  let sln = txt.length; | Returns the length of the array. sln will evaluate to 26. |
| **.slice() extracts a part of a string and returns the extracted part in a new string.** | let str = "Apple, Banana, Kiwi";  let res = str.**slice**(7, 13); | The method takes 2 parameters: the starting index (position), and the ending index (position). This example slices out a portion of a string from position 7 to position 13. The result of res is “Banana”. |
| **A string is converted to uppercase with .toUpperCase() or to lower case with .toLowerCase():** | let text1 = "Hello World!";  let text2 = text1.**toUpperCase()**;  let text3 = text1.**toLowerCase()**; | The result of text1 is "Hello World!". The value of text2 is "HELLO WORLD!". The value of text3 is "hello world!" |
| **A string can be converted to an array with the .split() method:** | let txt = "a b c d e";  txt.split(" "); | Converts txt from a string into an array splitting on each space. The result of txt is the array ["a","b","c","d","e"]. |

|  |  |  |
| --- | --- | --- |
| **Array Methods and Properties** | | |
| **Action** | **Code Example** | **What it does** |
| **.length tells us how many items there are in the array** | let fruits = ["Banana", "Orange", "Apple", "Mango"];  let x = fruits.**length**; | Returns the number of elements in the array. x will evaluate to 4. |
| **The .pop() method removes the last element from an array:** | let fruits = ["Banana", "Orange", "Apple", "Mango"];  let x = fruits.**pop()**; | Removes the last element ("Mango") from fruits. The value of x is ["Banana", "Orange", "Apple"] |
| **The .push() method adds a new element to an array (at the end):** | let fruits = ["Banana", "Orange", "Apple", "Mango"];  fruits.**push**("Kiwi"); | Adds a new element ("Kiwi") to fruits. The result of fruits is ["Banana", "Orange", "Apple", “Kiwi”] |
| **You can re-assign an array value. Array elements are accessed using their index number:** | let fruits = ["Banana", "Orange", "Apple", "Mango"];  fruits**[0]** = "Kiwi"; | Changes the first element of fruits to "Kiwi". The result of fruits is ["Kiwi", "Orange", "Apple", “Kiwi”] |
| **The .join() method also joins all array elements into a string.** | let fruits = ["Banana", "Orange","Apple", "Mango"];  let x = fruits.**join**(" \* "); | Joins all elements into a string separated by " \* ". The result of x is  "Banana \* Orange \* Apple \* Mango". |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Syntax** | | | |
| A [**function**](https://popcode.org/?snapshot=ffa5ed69-34f7-471c-b4aa-8edc0a0dd22c) is a set of instructions-- the basic building block of a program.  A **function declaration** creates the set of instructions. | | | |
|  | | | 1. The keyword function is *always* used to start a **function declaration**. 2. The **name** of this function is checkAnswer. 3. Some functions use **parameters.** The name of this parameter is input. You may also accept *multiple* parameters, separated by commas. 4. **Curly brackets** **{ }** surround the body of the function. 5. The **body** of the function is the list of instructions, enclosed in the curly brackets. |
| To use the list of instructions, you must make a **function call.** | | | |
|  | | 1.  2. | To **call the function**, use the function name checkAnswer.  In a function call, you should pass an **argument** for every parameter in the function declaration. The parentheses **( )** are *always* included, even if there isn't an **argument**. (see above). |

|  |  |  |  |
| --- | --- | --- | --- |
| [**Function Example with Return**](https://popcode.org/?snapshot=b8bac04d-8cda-4eb6-acbd-d748ef4863da) **Statement** | | | |
| 1  2  3  4  5  6 | function compoundWord(a,b) {  return a + b;  }  let word1 = compoundWord("can","not");  let word2 = compoundWord("fire","work"); | 1  2  3  4  6 | **Declare** **function** compoundWord that takes 2 parameters.  **Body**: Return parameter a + parameter b.  **End** of function compoundWord.  **Call function** compoundWord, w/ arguments "can" & "not". Assign it to the variable word1. The value is "cannot".  **Call function** compoundWord, with arguments "fire" and "work". The value of variable word2 is "firework. |

APIs

|  |
| --- |
| APIs allow websites to access the features or data of another service. This data is returned as an object that can be accessed using **JSON** (Javascript Object Notation). |
| **API Request URL** |
| **API or Application Programming Interface Request URL** |
|  |
| 1. **Base Url** is the consistent part of your url. This will not change. 2. **End Point** refers to some object or set of objects that are exposed at an API endpoint. 3. **Query String** comes after the endpoint. This starts after the ? and includes the **query parameters** and their associated **values** separated by & signs.    1. The name of the **Query Parameter**    2. **Value** is the data that is associated with a query parameter. |

|  |
| --- |
| **fetch() Request** |
| [**fetch()**](https://popcode.org/?snapshot=86bf7b40-9de3-43f9-a31f-31e698c74d29)is used to retrieve data from an **API** |
|  |
| 1. Begin the request with fetch(). The **API URL** goes between the parentheses. **Three basic properties of a fetch() request are (there are others not listed):** 2. **url**: Indicates where you are making the request to. 3. First **.then**: makes the API call by running a function that takes **response** as a parameter and returns all the data from the API call in JSON. **This never changes.** 4. Second **.then**: runs upon a successful return from the first .then. This function takes **data** as a parameter that becomes the object which contains all the data found in **response**. **Data** can be traversed to return whichever piece of the API call that you want to access. |

|  |
| --- |
| **API Methods** |
| Methods tell the server what the user wants to do with the requested data |
| **GET**: asks the server to retrieve data.  **POST**: asks the server to create a new resource and add data provided by the user  **PUT**: asks the server to edit/update existing data, provided by the user  **DELETE**: asks the server to delete data |

|  |  |  |
| --- | --- | --- |
| **React** | | |
| **React** is a JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called **components**. | | |
| **components** | import **React** from 'react';  Import Colors from '/.Colors.js';  export default function **App()** {  return (  <div>  <h1> hi </h1>;  <Colors name={"green"} />;  </div>  );  } | **Components** are each written in their own separate .js file  The first line calls the React library into your component file  The name of this component is App, it is created by writing a function with a return |
| **props** | import React from 'react';  export default function Colors(**props**) {  return <h1> {**props.name**} </h1>;  } | Add **props** to customize your component so it can be reused  Props must be called as the argument of the component  Return props as an object defined in your component |
| **onClick function** | function speak(){  alert("meow");  }  <div **onClick**={speak}></div> | Add interactivity with an **onClick** function  Call the onClick function in curly brackets in your component |
| **state** | import React, {**useState**} from 'react';  let [numSnacks, setNumSnacks] = useState(0);  setNumSnacks(numSnacks + 1); | Use **state** to update your component  Set the state of a variable using useState() |

|  |  |  |
| --- | --- | --- |
| **Mathematical Operators\*\*** | | |
| **Symbol** | **Definition** | [**Code Example**](https://popcode.org/?snapshot=9e5b1b2d-a557-4d7e-9cbc-c9f642973bdb) |
| **+** | Addition\*\*\*\* | a **+** b; |
| **-** | Subtraction | a **-** b; |
| **\*** | Multiplication | a **\*** b; |
| **/** | Division | a **/** b; |
| **%** | Modulo | a % b; |

\*\* Follow the order of operations rule **PEMDAS**: 1) Parentheses, 2) Exponents, 3) Multiply/Divide, 4) Add/Subtract

\*\*\*\*Can *ALSO* be used to concatenate, or combine, strings, not just add numbers.

|  |  |  |
| --- | --- | --- |
| **Comparison Operators** | | |
| **Symbol** | **Definition** | [**Code Example**](https://popcode.org/?snapshot=ef52d340-cbc8-441b-b849-8f6b9cc6efa4) |
| **<** | Less than | if (number **<** 10) |
| **>** | Greater than | else if (grade **>** 70) |
| **<=** | Less than or equal to | if (points **<=** 100) |
| **>=** | Greater than or equal to | else if (age **>=** 16) |
| **===** | Equal to | if (username **===** "scripted1") |
| **!==** | NOT equal to | else if (password **!==** "p@$sw0rd") |

|  |  |  |
| --- | --- | --- |
| **Logical Operators** | | |
| **Symbol** | **Definition** | [**Code Example**](https://popcode.org/?snapshot=11f2551e-1c71-4326-955d-ff08cfe28c04) |
| **&&** | And | if (resume > true **&&** interview < true) |
| **||** | Or | if (grade > 65 **||** passedRegents) |
| **!** | Not | if (**!**(number < 10)) |