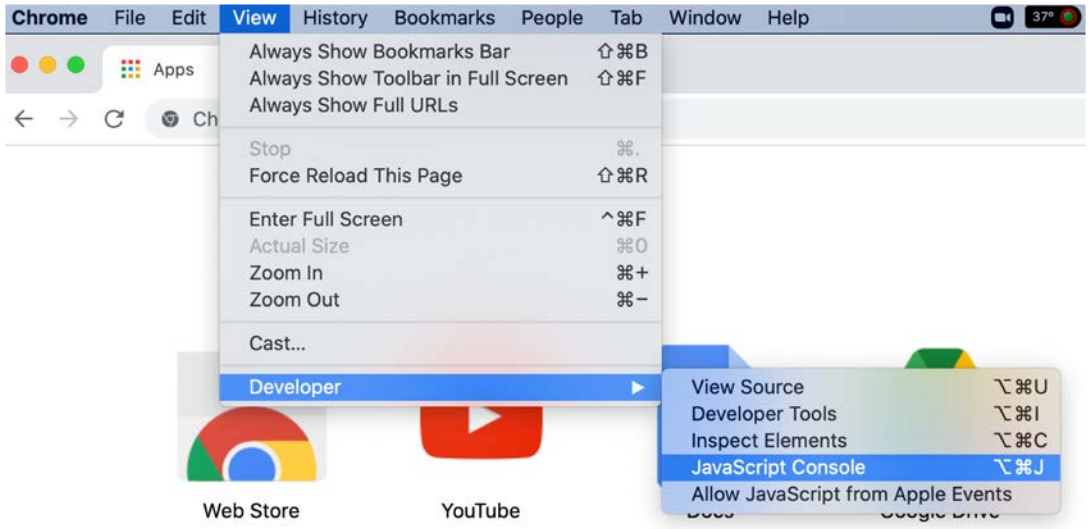
In-Class Exercise/Demo

In-Class Exercise / Demo

- For our demo exercises, we will use a different tool to test out JavaScript codes.
- We will use Chrome browser's JavaScript Console. If you have not heard or use this before, today you are going to learn how to use this quick and easy tool.
- You might ask: is this console also available on other browsers, like Firefox and Safari? The answer is yes.
- But we chose to use the more popular Chrome for our Bootcamp.
- So why use browser console? Why can't we test JavaScript codes using VS Code? Of course you can. The console provides you with quick testing and the ability to write, manage, and monitor JavaScript on demand without having to create separate HTML support codes. It is designed for and as part of the development process.

In-Class Exercise / Demo

- Let's walk through quickly how to get to this console feature:
  - Open a new Chrome browser tab.
  - Select View → Developer → JavaScript Console



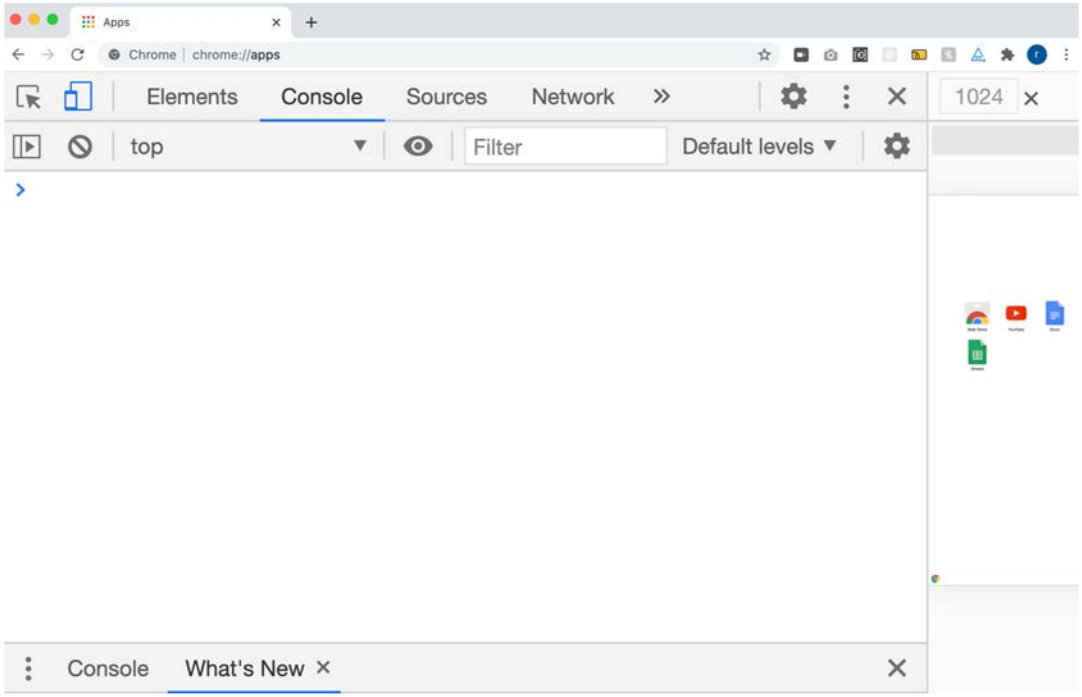
Or you can use keyboard shortcut:

cmd + option + j (mac)

ctrl + shift + j (windows)

In-Class Exercise / Demo

- The console view (may vary slightly between mac & window):



In-Class Exercise / Demo

- Because this week's demo exercises are all done on the browser console, you will be tasked to take a screenshot of your console for every exercise you finished. Below are instructions on how to take screenshots on your computer:

PC Windows:

Win + Alt + Print Screen – *Captures only the active window*. This command saves an image to *C:Users<user name>Videos>Captures* by default.

MAC:

Command + Shift + 4 – Marquee the area of the screen you want to screenshot. Saves the screenshot as a PNG file on your desktop.

In-Class Exercise / Demo

- Alright, now that you have learned how to open the JavaScript Console in Chrome, we can proceed to begin our code testing.
- In our first demo, we will write a simple JavaScript function and see how it works.
- On a new console, enter the following:

```
> function newStudent() {
    console.log("New student is added to the system.");
  }
```

- Press enter or return on the keyboard. Undefined is returned because this piece of code just creates a function and nothing else.

Note:
What is console.log? It is a window method to print to the console window whatever is written within the parenthesis.

Don't forget to take a screenshot!

In-Class Exercise / Demo

- If you recall in the lecture, functions are not executed or run until it is called.
- To call the function, enter the following and then hit enter or return key:

```
> function newStudent() {
    console.log("New student is added to the system.");
  }
<· undefined
> newStudent();
```

- The function runs and execute the codes contain within the function and print to the console the following:

```
> newStudent();
  New student is added to the system.
<· undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- Next, let's pass an argument to the function. First, clear the console window – click the browser refresh button, enter the following codes and then press enter or return key:

```
var name = "John";
function newStudent(x) {
  console.log("New student " + x + " is added to the system.");
}
undefined
```

- Here we create a variable with an assigned value "John". In the function, we create a variable x (name whatever you like) as the parameter to be received into the function for use. The function when run, will print to the console a message that consist of strings and variable. We use the plus + symbol to concatenate them together.

In-Class Exercise / Demo

- We will now call the function by entering the function name along with an argument (the variable) to be passed to it and then hit enter or return key. The concatenate message will be printed to the console:

```
var name = "John";
function newStudent(x) {
  console.log("New student " + x + " is added to the system.");
}
undefined
newStudent(name);
New student John is added to the system.                    VM1
undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- In our next demo, we will look at conditional statements. The if-else or if-elseif-else statements are examples of this decision making process. We will create a variable that contains a value or data to be used as part of the decision making.
- On a refreshed console window, enter the following codes. When finished hit the enter or return key. If it's true, it will print to the console the first message. Otherwise, it will print to the console the second message (else) .

```
var name = "Hanna";

if (name == "John") {
    console.log("Student name is " + name);
} else {
    console.log("It's a different student. His/her name is " + name + ".");
}
```

```
It's a different student. His/her name is Hanna.    VM92:6
undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- Before you attempt the next demo, first return to the console window and click refresh (and clear button if necessary) to erase the existing codes.
- In the next demo, you will be writing the same codes but with a small change. So instead of writing the codes again, let's do a shortcut. The console actually record all recent codes you have written. Press the up arrow key on your keyboard to cycle through the codes. Stop when you see the following:

```
var name = "Hanna";

if (name == "John") {
    console.log("Student name is " + name);
} else {
    console.log("It's a different student. His/her name is " + name + ".");
}
```

In-Class Exercise / Demo

- Change the value of the name variable from Hanna to John. Then hit the enter or return key. This time the statement evaluates the condition to true. As a result, the first message will be printed to the console.

```
var name = "John";

if (name == "John") {
    console.log("Student name is " + name);
} else {
    console.log("It's a different student. His/her name is " + name + ".");
}
```

```
Student name is John                                          VM1857:4
undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- Next, let's practice an if-elseif-else statement. Return to your console window and clear the codes. In this demo, we are constructing a multiple elseif statement to see how it works. Enter the following codes.

```
var name = "Andrea";

//if else statement
if (name == "John") {
    console.log("Student name is " + name);
} else if (name == "Hanna") {
    console.log("It's a different student. His/her name is " + name);
} else if (name == "Michael") {
    console.log("It's a new student. His/her name is " + name);
} else if (name == "Julie") {
    console.log("It's a returning student. His/her name is " + name);
} else {
    console.log("Sorry, no student by the name of " + name + " exist.");
}
Sorry, no student by the name of Andrea exist.                    VM125
undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- In this demo we'll practice an if-else statement that test more than one condition at the same time. The first example uses the and (&&) to evaluate to a true statement. Enter the following on a cleared console:

```
var doorlocked = true;
var alarmon = false;

//Example 1: Using and (&&)
if ((doorlocked == true ) && (alarmon == true)) {
    console.log("House secured");
} else {
    console.log("House not secured");
}
```

```
House not secured
undefined
```

Don't forget to take a screenshot!

Note: both conditions must be evaluated to true in order to satisfy the statement's test. In this case, it didn't because only one satisfied not the other.

In-Class Exercise / Demo

- The second example uses the or (||) to evaluate to a true statement. Enter the following on a cleared console:

```
var doorlocked = true;
var alarmon = false;

//Example 2: Using or (||)
if ((doorlocked == true ) || (alarmon == true)) {
    console.log("House secured");
} else {
    console.log("House not secured");
}
```

```
House secured
undefined
```

Don't forget to take a screenshot!

Note: in this example, as long as one of the conditions is evaluated to true even though the other didn't, it will satisfy the statement's test. In this case, it met the criteria.

In-Class Exercise / Demo

- What if the values of both variables are reversed?

```
var doorlocked = false;
var alarmon = true;

//Example 2: Using or (||)
if ((doorlocked == true ) || (alarmon == true)) {
    console.log("House secured");
} else {
    console.log("House not secured");
}
```

```
House secured
```

```
undefined
```

Don't forget to take a screenshot!

Note: even if they are reversed, things doesn't change. As long as one of the conditions is evaluated to true, it satisfies the statement's test.

In-Class Exercise / Demo

- The final demo is the switch statement. Switch works similarly to the if-elseif-else statement. Return to your console, clear all codes and enter the following:

```
var name = "Brent";

//switch statement
switch(name) {
  case 'John':  // if (name == 'John')
    console.log("Student name is " + name);
    break;

  case 'Hanna':  // if (name == 'Hanna')
    console.log("It's a different student. His/her name is " + name);
    break;

  default:
    console.log("No such student name exist");
}
```

```
No such student name exist                                    VM45
undefined
```

Don't forget to take a screenshot!

In-Class Exercise / Demo

- **Submission**:
  - Grab all your screenshots and save into a folder.
  - Name this folder: **week3-day1**.
  - Zip this folder.
  - Submit **week3-day1.zip** in GAP – Week 3 Day 1 dropbox at the end of this class session.