Pro Dev Session

Pro Dev Session

You may want to write this down.

For this professional Development Session, we'll be going through how to link JavaScript documents to our HTML documents.

If your first guess is to use a **link** tag like we do when linking our CSS stylesheets to HTML documents, you’re pretty close!

In HTML, all JavaScript is contained inside of a **script** tag.

Pro Dev Session

You may want to write this down.

The **script** tag can go either in the **head** or **body** tags, depending on where you want the JavaScript to load.

It’s generally a good idea to leave the **script** tag in the body as we’ll eventually want scripts to run within our page layout.

Linking JS to HTML

Where does the logic go?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>


</body>
</html>
```

Linking JS to HTML

Where does the logic go?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>

                <script></script>
</body>
</html>
```

Pro Dev Session

You may want to write this down.

Now all we have to do is tell our **script** tag where our JavaScript file exists using the **src** attribute.

Linking JS to HTML

Where does the logic go?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>

                <script src=”js/index.js”></script>
</body>
</html>
```

Practice

- Start by creating a brand new HTML file.
- Then, create 3 new files adjacent to your newly created HTML file (same folder level) called:

1. **logic1.js**
2. **logic2.js**
3. **logic3.js**

- Inside of each **logic#.js** file, add the following code: **alert('NEW MESSAGE HERE');**
- Be sure to make the message different for each file!
- Back inside of your HTML file, add the necessary starter code for a basic website.

Pro-Tip: Using Visual Code you can type "!" and hit enter to get HTML start code quickly!

- Create 3 script tags and link all three **logic#.js** files to your HTML file.

- Using HTML commenting (<!-- Example -->) and comment all but one script tag out and open your HTML document in a browser.

- Repeat the process of uncommenting a script tag while commenting out the other two script tags and see the result in the browser!

Stand Up!

JavaScript Crash Course

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Notebooks Ready? It’s time for a mini lecture.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

JavaScript is the only programming language with major browser support. All websites use JavaScript.

JavaScript is syntactically similar to Java. We're going to learn some JavaScript fundamentals today.

Some of this may feel like review, but pay careful attention to the differences between Java and JavaScript.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

As a reminder:

Java is used on the server to build an API and send data from the database to the client.

JavaScript is used on the client (along with HTML and CSS) to send data to and receive data from the server and display it on the screen for users.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
int num = 4;
double num2 = 4.2;
String name = "Daisy";
boolean isCool = true;
int[] numArr = {1, 2, 3, 4};
```

JavaScript

```
let num = 4
let num2 = 4.2
let name = "Daisy"
let isCool = true
let numArr = [1, 2, 3, 4]
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
int num = 4;
num = true;
```

✗

JavaScript

```
let num = 4
num = true
```

✓

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

var, let, and const can all be used to declare any type of variable in JavaScript.

var and let are similar with two key exceptions:

1. var is function scoped and let is blocked
2. var is hoisted. let is not.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

let and const are similar except that once you assign a const a value, it cannot be reassigned.

```
let num = 4
num = 5
```

✓

```
const num2 = 4
num2 = 5
```

✗

JavaScript also has a value that is assigned to all uninitialized variables, `undefined`.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
int num;
System.out.println(num);
```

✗

JavaScript

```
let num
console.log(num)
```

✓

Check-in Time

- What are the differences between a strongly typed and loosely typed language?
- What are some benefits of each?
- When might const be preferable to let or var?

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
for (int i = 0; i < 5; i++){
        System.out.println(i);
}
```

JavaScript

```
for (let i = 0; i < 5; i++){
        console.log(i)
}
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
int[] numArr = {1,2,3,4,5,6};

for (int i = 0; i < numArr.length; i++){
        System.out.println(numArr[i]);
}
```

JavaScript

```
let numArr = [1,2,3,4,5,6]

for (let i = 0; i < numArr.length; i++){
        console.log(numArr[i])
}
```

Loops

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create an array of your 5 favorite colors.
- Loop through the array and print all the colors.
- Loop through the array in reverse order and print all the colors
- Create an array of your 5 favorite numbers.
- Loop through the array and multiply each by 10 then print the array.

Challenge:

- Find out what other types of loops are available in JavaScript and determine which loop would be best for this activity

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
int x = 4;

if(x > 5){
        System.out.println("High");
}else{
        System.out.println("Low");
}
```

JavaScript

```
let x = 4

if(x > 5){
        console.log("High")
}else{
        console.log("Low")
}
```

Syntactically JavaScript is very similar to Java. But it also likes to play it fast and loose with rules.

Watch this!

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

```
let x = 4

if(x){
        console.log("High")
}else{
        console.log("Low")
}
```

Welcome to the world of truthy and falsy!

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

JavaScript will happily make a boolean of any value in a conditional.

All values are truthy except:

undefined
null
false
0
empty String
NaN

Conditionals

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Write a conditional, as concisely as possible, that assigns the value of “Hello World” to a string if the string doesn’t have a value
- How would you change the program to only assign the value of “Hello World” to a string if the string has less than 5 characters?

10 minutes!

Stay Seated &Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We'll start back in a few minutes.

Crash Course Continued

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
public int add(int num1, int num2){
        return num1 + num2;
}
```

JavaScript

```
// Function Declaration
function add(num1, num2){
        return num1 + num2
}

// Function Expression
const add = function(num1, num2){
        return num1 + num2
}

// Arrow Function
const add = (num1, num2) => num1 + num2
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Things get a little trickier with objects.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Note: There are no access modifiers in JavaScript. Everything is public.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void greet(String hi) {
        System.out.println(hi +
            this.name);
    }
}
```

JavaScript

```
// Class
class Person {
    constructor (name, age) {
        this.name = name;
        this.age = age;
    }

    greet(hi) {
        console.log(hi + this.name);
    }
}
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void greet(String hi){
        System.out.println(hi +
            this.name);
    }
}
```

JavaScript

```
// Function Constructor
function Person(name, age) {
    this.name = name;
    this.age = age;
    this.greet = function(hi) {
        console.log(hi + this.name);
    }
}
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
Person page = new Person("Page", 12);

String pName = page.getName();
```

JavaScript

```
const page = new Person("Page", 12)

const pName = page.name
```

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

We don’t have to use a function constructor or class in JavaScript.

We can create objects directly in JavaScript.

```
const page = {
    name: "Page",
    age: 12,
    greet = function(hi) {
        console.log(hi + this.name)
    }
}
```

Objects

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- In 2 ways, create the equivalent of a Java motorcycle class with the properties VIN, make, model, year, maxSpeed, currentSpeed.
- Include methods accelerate and break which increment and decrement the currentSpeed by 5 with limits of 0 and maxSpeed.

20 minutes!

lunch.

Inheritance

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

JavaScript lacks abstract classes and interfaces, but it does have a mechanism for inheritance called Prototypal Inheritance.

Although internally JavaScript inheritance differs from Java inheritance a bit, on the surface the two are very similar.

JS Crash Course

A WHOLE LOT OF INFO IN A LITTLE BIT OF TIME

Java

```
class Person extends Animal {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void greet(String hi) {
        System.out.println(hi +
            this.name);
    }
}
```

JavaScript

```
class Person extends Animal {
    constructor (name, age) {
        this.name = name
        this.age = age
    }

    greet(hi){
        console.log(hi + this.name)
    }
}
```

Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Looking at the code provided, try to guess where the following properties and methods exist:
  - numOfFeet, sound, color, name, makeNoise, sayName, rollOver
- Run the code found in index.js. Use console.log() to print the rover object to the console. Note which functions and properties are on the object.
- Navigate through the object and find the reference to its prototype. Expand the prototype. Note which functions exist there

10 minutes!

Review Activity

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
const rufus = {
    name: 'rufus',
    color: 'brown'
}

console.log(rufus);
```

Review Activity

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
class Dog {
    constructor(color, name) {
        this.name = name;
        this.color = color;
    }

    sayName() {
        console.log(this.name);
    }
}
const rufus = new Dog ('brown', 'rufus');
console.log(rufus);
```

Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Working with your partner(s), think of an object that could be created from a class that extends another class.
- Write the code to create those classes and instantiate an object from them. A good tip to keep in mind as you are working on this activity is how objects can be related to each other and in what ways certain objects are different.
- Hint: Think about vehicles! Within the category of vehicles, you might have cars, trains, and airplanes. Within cars, you can make coupes, sedans, or SUVs. Within sedans, you could have a Ford Taurus, Honda Accord, or a Toyota Camry.

Lab Time

Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Using the provided starter files, link each JavaScript file to **index.html** and complete the following activities.

Remainder of Class

Functions

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create 5 functions:
    - subtract: take in two variables and return the difference
    - subtractOrZero: take in two variables, if the difference is less than zero, return 0, otherwise return the difference
    - max: take in three variables and return the largest number
    - min: take in three variables and return the smallest number
    - isLarger: take in two variables and return true if the first variable is larger than the second, otherwise return false
- Run all of these functions and return the values to the console

Remainder of Class

Arrays

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create 4 functions
  - total: take in an array and return the sum of all values in the array
  - totalOdd: take in an array and return the sum of only the odd values in the array
  - totalEven: take in an array and return the sum of only the even values in the array
  - secondLargestNumber: take in an array and return the second largest number in the array.
- Run all of these functions and return the values to the console

Remainder of Class

Classes

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Using the UML diagram provided, create the JavaScript classes as shown in the image

Remainder of Class

Inheritance

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create a JavaScript Car class
  - This should have a make, a model, and milesDriven
  - This should also have a drive method as well as a displayMilesDriven method
- Create a JavaScript Sedan class that extends Car
  - A sedan should have 4 doors and an option to have poweredWindows
- Create a JavaScript Accord class that extends Sedan
  - Give the Accord a year, engine, and transmission

Remainder of Class