JS Review

Fundamentals

OBJECTIVES

- Review and discuss DOM selection techniques
- Review and discuss Array and Object usage
- Review and discuss iterating using for loops and conditions

DOM SELECTION

- DOM selection fetches an Element from the page using the document API
  - `document.getElementById("some-id")`

DOM SELECTION

- DOM selection fetches an Element from the page using the document API
  - `document.getElementById("some-id")`
  - `document.querySelector("#some-id")` or `document.querySelector(".some-class")`
  - `document.querySelectorAll("#some-id")` or `document.querySelectorAll(".some-class")`

NEW MATERIAL

DOM SELECTION

- DOM selection fetches an Element from the page using the document API
  - `document.getElementById("some-id")`
  - `document.querySelector("#some-id")` or `document.querySelector(".some-class")`
  - `document.querySelectorAll("#some-id")` or `document.querySelectorAll(".some-class")`

- querySelector finds the first css selector match
- querySelectorAll finds all matches

NEW MATERIAL

DOM SELECTION

- Create a CodePen with the following
  - HTML
    - `<div id="foo">Hello</div> <div id="foo">World</div>`
- In the console try the following
  - Use getElementById to select “foo”
  - Use querySelector and querSelectorAll to select id of “foo”
  - How do the different selection methods differ?

DOM SELECTION

- In the console try the following:
  - Change the backgroundColor of one of the div’s with id foo to be “blue”
  - Bonus: Use document.querySelectorAll to select both of the elements with id “foo”. Iterate through the elements and change their font color to be white.

Arrays

- When accessing arrays we use the index to set and read values

```
var friends = ["Ruby", "Sam", "Taylor", "Alex"];


friends[0]
// => "Ruby"
friends[2]
// => "Taylor"

friends[3] = "Morgan";

friends
// => ["Ruby", "Sam", "Taylor",  "Morgan"]
```

Arrays

- Utilize push, pop, shift, and unshift to update an array

```
var friends = ["Ruby", "Sam", "Taylor", "Alex"];


friends.push("Morgan")

friends
// => ["Ruby", "Sam", "Taylor", "Alex",  "Morgan"]
```

LOOPS

- When looking to loop over a list of items in a sequential fashion we learned to utilize a `for` loop
  - The following iterates over all friends and prints their name to the console.

```
var friends = ["Ruby", "Sam", "Taylor", "Alex"];


for (var index = 0; index < friends.length; index += 1) {
  console.log(friends[index])
}
```

Array Access

- Given the following array of numbers
  - Google: Sort the numbers in the array
- Find the median value

```
var numbers = [18, 1, 2, 22, 32, 3];
```

LOOPING

Utilize looping to print out only the even indexed names and third name

- NOTE: 0 is even.

```
var friends = ["Ruby", "Sam", "Taylor", "Alex"];

// Add your code
```

LOOPING

Utilize looping to print out only the first half of the names.

```
var friends = ["Ruby", "Sam", "Taylor" , "Morgan", "Alex"];

// Add your code to print half of your friends.


var bestFriends = ["Rey", "Jamie", "Adrian", "Devin"];

// Copy your loop above
//  Use it to print half of your bestFriends.
```

Looping

- Utilize Math.random to create an array of 100 integers ranging from 1 to 1000.

```
var numbers = [];

// Your code here.
```

LOOPING

Utilize looping to print out the items in reverse order

```
var friends = ["Ruby", "Sam", "Taylor", "Morgan", "Alex"];

// Add your code
```

Selected Answer: Looping

Utilize looping to print out only the first half of the names.

```
var friends = ["Ruby", "Sam", "Taylor", "Morgan", "Alex"];

// Add your code to print half of your friends.
for (var index = 0; index < (friends.length)/2; index += 1) {
  console.log(friends[index])
}
```

Selected Answer: Looping

Utilize looping to print the items in reverse order.

```
var friends = ["Ruby", "Sam", "Taylor", "Morgan", "Alex"];

// Add your code to print half of your friends.
for (var index = friends.length - 1; index >= 0; index -= 1) {
  console.log(friends[index])
}

// or

for (var index = friends.length; index-- > 0;) {
  console.log(friends[index])
}
```

Conditions

- Utilize conditions to handle different cases

```
var ages = [36, 23, 20, 21, 56, 19, 28, 47];

// Code to only print ages over 21

for (var index = 0; index < ages.length; index += 1) {
  if (ages[index] > 21) {
    console.log("You are over 21");
  } else {
    console.log("You are not over 21");
  }
}
```

Conditions

Utilize a loop and condition to iterate through the values 1 through 100

- Print each number that
- When the number is a multiple of three print “Fizz” instead of the number
- When the number is a multiple of five print “Buzz” instead of the number
- When the number is a multiple of five and three print “FizzBuzz”.

Objects

- Utilize Objects to hold properties associated to their values

```
var myName = {};

// set each value individually
myName.first = "John";
myName.middle = "Joe";
myName.last = "Doe";

myName
// => { first: "John", middle: "Joe", last: "Doe" }

myName.first
// => "John"
```

Objects

- Utilize Objects to hold properties associated to their values

```
// set them all during initialization
var myName = {
    first: "John",
    middle: "Joe",
    last: "Doe"
}

myName
// => { first: "John", middle: "Joe", last: "Doe" }

myName.first
// => "John"
```

Objects

Use nested objects to structure data and organize data.

```
var myName = { first: "John", middle: "Joe", last: "Doe" };
var friend = {};

friend.name = myName;
friend.age = 28;

friend
// => {
    age: 28,
    name: { first: "John", middle: "Joe", last: "Doe"}
};

friend.name.first;
// => "John"
```

REVIEW

- Selecting and manipulating element in the document
- Array accessing and methods
- Using for loops and conditionals
- Using nested objects to structure data and organize data.

NEXT...

- You should review your DOM events and control flow assessment
- Be prepared to review the assessment and ask questions
- We will be using these concepts to create our own functions later.