Loops & Collections

GENERAL ASSEMBLY

Learning Objectives

- **Explain** looping structures and their uses in JavaScript
- **Experiment** with JavaScript loops
- **Identify** the need for JavaScript Arrays and Objects
- **Practice** using them
- **Experiment** with iteration

Agenda

- Loops
- JavaScript Arrays
- JavaScript Objects

Review

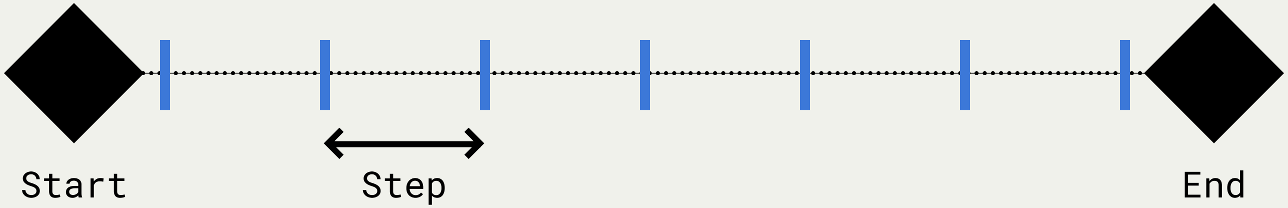
Loops

GENERAL ASSEMBLY

What are they?

- A piece of code that can execute over and over again
- There are three main parts of any loop:
  - A **starting point**
  - An **increment**, or **step**
  - An **ending point**
  - You always need these things!
- You get access to the current value
- We can use other JS functionality in a loop (e.g. conditionals, call functions)

Start, Step, End



Types of Loops

- The **for** loop
- The **while** loop

The while loop

```
while ( CONDITION ) {
    CODE TO EXECUTE
}
```

```
var num = 0;

while ( num < 5 ) {
    console.log( num );
    num = num + 1;
}
```

The for loop

```
for ( start; end; increment ) {
    CODE TO EXECUTE
}
```

```
for ( var i = 0; i < 10; i += 1 ) {
    console.log( i );
}

for ( var current = 5; current > 0; current -= 1 ) {
    console.log( current );
}
```

Exercise!

For Loop Exercises

Collections

GENERAL ASSEMBLY

What are they?

A collection is a way to group similar, or related, data in JavaScript

Two main types:

- Arrays (ordered and unnamed)
- Objects (unordered and named)

What are they like?

- An array is like a todo list
- An object is like a dictionary

Arrays

GENERAL ASSEMBLY

What are they?

- An array is an ordered list
- It can be filled with any data type
- Each item is accessible through an **index** (a number)
    - The first item is accessed through index 0 (often called zero-based)
- They are able to be **iterated** through (meaning looped through)

Creating Arrays

```
var emptyArray = [];

var randomNumbers = [ 12, 42, 1, 3, 92 ];

var rainbowColours = ['red', 'orange', 'yellow', 'green'];

var weirdInstruments = [
    "Badgermin",
    "The Great Stalacpipe Organ",
    "Stylophone",
    "Ondes Martenot",
    "Sharpischord",
    "Hydraulophone",
    "Pyrophone"
];
```

Accessing Elements

```
var weirdInstruments = [
    "Badgermin",
    "The Great Stalacpipe Organ",
    "Stylophone",
    "Ondes Martenot",
    "Sharpischord",
    "Hydraulophone",
    "Pyrophone"
];

weirdInstruments[0];
weirdInstruments[5];
weirdInstruments[ weirdInstruments.length - 1 ];
```

Reassigning Elements

```
var weirdInstruments = [
    "Badgermin",
    "The Great Stalacpipe Organ",
    "Stylophone",
    "Ondes Martenot",
    "Sharpischord",
    "Hydraulophone",
    "Pyrophone"
];

weirdInstruments[0] = "Roli Seaboard";
weirdInstruments[5] = "Makey Makey Banana Piano";
weirdInstruments[ weirdInstruments.length - 1 ] = "OP1";
```

Looping through Arrays

```
var ordinals = [
    "Zeroth",
    "First",
    "Second",
    "Third"
];

ordinals[0];
ordinals[1];
ordinals[2];
ordinals[3];

// Fair bit of consistency there!
```

Looping through Arrays

```
var ordinals = [
    "Zeroth",
    "First",
    "Second",
    "Third"
];

for ( var index = 0; index <= 3; index += 1 ) {
    var currentElement = ordinals[index];
    console.log( currentElement );
}
```

Looping through Arrays

```
var ordinals = [
    "Zeroth",
    "First",
    "Second",
    "Third"
];

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

Properties & Methods

```
var ordinals = [
    "First",
    "Second",
    "Third"
];

ordinals.length; // => 3

ordinals.pop(); // Remove the last element
ordinals.push( "Third" ); // Add "Third" to the end

ordinals.shift(); // Remove the first element
ordinals.unshift( "First" ); // Add "First" to the start

ordinals.indexOf( "Second" ); // Get the index of "Second" => 1
```

Exercise!

JavaScript Array Exercises

Resources

- CodeAcademy
- Array Documentation
- Speaking Javascript: Arrays
- Eloquent Javascript: Arrays
- Javascript.info's Description

Objects

GENERAL ASSEMBLY

What are they?

- Like a dictionary
- A collection of **key-value** pairs
- It's not index-based (like arrays), it is accessible through words
- They are unordered

Why use them?

- Encapsulation and modularity
    - Ways to group functionality
- Give names to values
- Work with large amounts of data effectively

Creating Objects

```
var emptyObject = {};

var movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

var bookSeries = {
    name: "In Search of Lost Time",
    author: "Marcel Proust",
    books: [
        "Swann's Way",
        "In the Shadow of Young Girls in Flower",
        "The Guermantes Way",
        "Sodom and Gomorrah",
        "The Prisoner",
        "The Fugitive",
        "Time Regained"
    ]
};
```

Accessing Values

```
var movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

var movieName = movie.name;
var movieDirector = movie.director;
var movieDuration = movie.duration;
```

Changing Values

```
var movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

movie.name = "Sátántangó";
movie.director = "Béla Tarr";
movie.duration = 534;
```

Adding new values

```
var movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

movie.language = "Hungarian";
movie.rating = 2141252322461698 2; // Out of 5
movie.parts = 12;
```

Nested Objects

```
var explorer = {
    firstName: "Jacques",
    lastName: "Cousteau",
    birth: {
        day: 11,
        month: 6,
        year: 1910
    }
};

var birthDay = explorer.birth.day;
var birthMonth = explorer.birth.month;
var birthYear = explorer.birth.year;
```

Complex Data Structures

```
var marxFamily = [
    { name: "Groucho", birthYear: 1890 },
    { name: "Harpo", birthYear: 1888 },
    { name: "Chico", birthYear: 1887 },
    { name: "Zeppo", birthYear: 1901 },
    { name: "Gummo", birthYear: 1893 }
];

for ( var i = 0; i < marxFamily.length; i++ ) {
    var brother = marxFamily[ i ];
    console.log( brother.name, brother.birthYear );
}
```

Exercise!

Loops, Arrays and Objects - All together now

Resources

- Sitepoint
- Speaking JavaScript
- Eloquent JavaScript
- Code Academy
- JavaScript.info

Homework

- Finish off the Loop Exercises
- Finish off the Array Exercises
- Finish off the Object Exercises
- Prepare for Milestone 1

Next Lesson

- More JavaScript
  - JavaScript Plugins
  - jQuery
  - JS and Webpages

Q & A

GENERAL ASSEMBLY

Feedback Time

Lesson 11: **Loops and Collections**

https://ga.co/fewd32syd

Thanks!

GENERAL ASSEMBLY