

Loops

Sometimes you want to go through a piece of code multiple times

Why?

- Showing a timer count down
- Displaying the results of a search
- Adding images to a slideshow

The while loop

The while loop tells JS to repeat statements while a condition is true:

```
while (expression) {  
  // statements to repeat  
}
```

The while loop

Try it out!

```
var x = 0;  
while (x < 5) {  
  console.log( x );  
  x++;  
}
```

Review: '++' means increment by 1!

Danger!!

What happens if we forget x++? ?

The loop will never end!!

The for loop

The for loop is a safer way of looping

```
for (initialize; condition; update) {  
  // statements to repeat  
}  
  
for (var i = 0; i < 5; i++) {  
  console.log(i);  
}
```

Less danger of an infinite loop. All conditions are at the top!

Array

An array is a data-type that holds an ordered list of values, of any type:

```
var arrayName = [element0, element1, ...];  
var colors = ['Red', 'Orange', 'Yellow'];  
var favoriteNumbers = [16, 27, 88];
```

Arrays

Arrays can hold variables

```
var fun = " roller coasters" ;  
var favoriteThings = ['raindrops', fun ];  
  
document.write( favoriteThings );
```

Array

The length property reports the size of the array:

```
document.write( favoriteThings.length );
```

Arrays -- returning values

You can access items with "bracket notation".
The number inside the brackets is called an "index"

```
var arrayItem = arrayName[indexNum];
```

Arrays in JavaScript are "zero-indexed", which means we start counting from zero.

```
var colors = ['Red', 'Orange', 'Yellow'];  
document.write( colors[0] ); // prints out 'Red'
```

Arrays -- returning values

Spaces between commas don't matter
Sometimes you'll see arrays spaced out like this:

```
var rainbowColors = [ 'Red',  
                     'Orange',  
                     'Yellow',  
                     'Green',  
                     'Blue',  
                     'Indigo',  
                     'Violet'  
                   ];  
var firstColor = rainbowColors[0]; // 'Red'  
var lastColor  = rainbowColors[6]; // 'Violet'
```

Arrays -- updating values

You can also use bracket notation to change the item in an array:

```
var awesomeAnimals = ['Corgis', 'Otters', 'Octopi'];  
awesomeAnimals[0] = 'Sloths';
```

This replaces the first item 'Corgis' with 'Sloths'

```
document.write( awesomeAnimals );  
// should print out 'Sloths', 'Otters', 'Octopi'
```

Arrays -- Adding values

Use bracket notation to ADD values to an array:

```
var awesomeAnimals = ['Corgis', 'Otters', 'Octopi'];  
awesomeAnimals[3] = 'Llamas';
```

This adds a value to the specified index.

If you skip past the last element, it fills in the gaps with undefined values

```
awesomeAnimals[10] = 'Sharks';
```

Arrays -- Push

You can add values with the push method:

```
awesomeAnimals.push('Ocelots');
```

push always adds the value to the end of the array

Before you use the push method, the array must be

declared

initialized using an empty array or one with values

```
var emptyArray = [];  
var valueArray = [ 1, 2, 3 ];
```

Loops and Arrays

Use a for loop to easily look at each item in an array:

```
var rainbowColors = ['Red',
                    'Orange',
                    'Yellow',
                    'Green',
                    'Blue',
                    'Indigo',
                    'Violet'
                    ];

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

Let's Develop It

Add a new link to the exercise from last week

Add an onclick to the link for a function called favoriteThings()

Create a new function called favoriteThings() in the javascript file

In the function, create an array and loop through the results

Post the results in an alert

"My favorite things are XX, YY, ZZ"

Bonus -- add an 'and' in the sentence before the last item

Objects

Objects are a data type that let us store a collection of properties and values.

```
var objectName = {
    propertyName: propertyValue,
    propertyName: propertyValue,
    ...
};

var charlie = {
    age: 8,
    name: "Charlie Brown",
    likes: ["baseball", "The little red-haired girl"],
    pet: "Snoopy"
};
```

Objects -- returning values

Access values of "properties" using "dot notation":

```
var charlie = {
    age: 8,
    name: "Charlie Brown",
    likes: ["baseball", "The little red-haired girl"],
    pet: "Snoopy"
};
```

```
var pet = charlie.pet;
var name = charlie.name;
```

Objects -- returning values

Or using "bracket notation" (like arrays):

```
var name = charlie['name'];
```

Non-existent properties will return undefined:

```
var gender = charlie.gender
```

Objects -- changing values

Use dot or bracket notation with the assignment operator to change objects.

Change existing properties:

```
charlie.name = "Chuck";
```

Or add new properties:

```
charlie.gender = "male";
```

You can also delete properties:

```
delete charlie.gender;
```

Arrays of Objects

Arrays can hold objects too!

```
var peanuts = [
  {name: "Charlie Brown",
   pet: "Snoopy"},
  {name: "Linus van Pelt",
   pet: "Blue Blanket"}
];
```

We can loop through the objects in the array

```
for (var i = 0; i < peanuts.length; i++)
{
  var character = peanuts[i];
  console.log( character );
}
```

Objects in functions

You can pass an object into a function as a parameter

```
var peanut = {
  name: "Charlie Brown",
  pet: "Snoopy"
};

function describeCharacter(character) {
  console.log(character.name + ' has a pet named ' + character.pet + '.');
}

describeCharacter(peanut);
```

Let's Develop It

Add another link that calls the function myFriends() onclick

Add a new function to the javascript myFriends

In the function, create an array of friends objects, with their names and hair colors

Use a for loop to go through each friend and describe them

Alert the results

Bonus -- make a separate functions that describe the friends

DOM

"Document Object Model"

A way to interact with the HTML elements on a webpage

Chrome and Firefox -- Right click --> Inspect Element

DOM Interaction

On every webpage, the document object gives us ways of accessing and changing the DOM.

Every DOM "node" has properties. They are connected like a family tree.

Parent (parentNode), children (childNodes, firstChild), siblings (prevSibling, nextSibling)

```
var bodyNode = document.body; // <body>
var htmlNode = document.body.parentNode; // <html>
for (var i = 0; i < document.body.childNodes.length; i++) {
  var childNode = document.body.childNodes[i];
  //could be <p>, <h1>, etc.
  //any html element
}
```

DOM Interaction: Easier

Finding every element on the page by siblings and children is time consuming!

The document object also provides methods for finding DOM nodes without going one by one

Find element by id

```

```

```
var img = document.getElementById('mainpicture');
```

DOM Interaction: Easier

Find element by tag name (p, li, div, etc.)

```
<li class="peanut">Charlie Brown</li>
<li class="peanut">Linus van Pelt</li>
```

```
var listItems = document.getElementsByTagName('li');
for (var i =0; i < listItems.length; i++) {
  var listItem = listItems[i];
}
```

Methods

Methods are functions that are associated with an object

They affect or return a value for a specific object

Used with dot notation

Previously seen example:

```
var img = document.getElementById('mainpicture');
```

DOM Nodes -- Attributes

We can use node methods to set and retrieve attributes

getAttribute/setAttribute

```
var img = document.getElementById('mainpicture');
img.getAttribute('src');
img.setAttribute('src', 'http://girldevelopit.com/assets/pink-logo.png');

var img = document.getElementById('mainpicture');
img.getAttribute('class');
img.setAttribute('class', 'picture-class');
```

DOM innerHTML

Each DOM node has an innerHTML property:

```
document.body.innerHTML;
```

You can set innerHTML yourself to change the contents of the node:

```
document.body.innerHTML = '<p>I changed the whole page!</p>';
```

You can also just add to the innerHTML instead of replace everything:

```
document.body.innerHTML += "...just adding this bit at the end of the page.";
```

DOM Modifying

The document object can create new nodes:

```
document.createElement(tagName);
document.createTextNode(text);
document.appendChild();

var newImg = document.createElement('img');
newImg.src = 'http://girldevelopit.com/assets/pink-logo.png';
document.body.appendChild(newImg);

var newParagraph = document.createElement('p');
var paragraphText = document.createTextNode('New Paragraph!');
newParagraph.appendChild(paragraphText);
document.body.appendChild(newParagraph);
```

Let's Develop It

Put it all together

Modify your existing three functions to add new elements to the screen instead of fire an alert

Keep in mind how to find an element, how to append an element, and how to change the inner html of an element

There are lots of possible solutions! Be creative!