



# LOOPS & CONDITIONALS

## LOOPS:

---

<b>for</b>	Runs while a condition is true, and includes syntax to declare and customize the iterator at the start	<pre>for (variable; condition; iteration) {   // do something }</pre>
<b>while</b>	Runs while a condition is true	<pre>while (condition) {   // do something }</pre>
<b>do while</b>	Runs while a condition is true, and ensures that the code block is executed at least once	<pre>do {   // do something } while (condition)</pre>

---

## ARRAY ITERATOR METHODS:

---

<code>forEach()</code>	Executes a provided function once per array element	<code>filter()</code>	Creates a new array with all elements that pass the test implemented by the provided function
<code>every()</code>	Tests whether all elements in the array pass the test implemented by the provided function	<code>map()</code>	Creates a new array with the results of calling a provided function on every element in this array
<code>some()</code>	Tests whether some element in the array passes the test implemented by the provided function		

---

### Usage:

```
arrayName.method(function() {  
  // do something  
})
```

### Example:

```
var friends = ['Curly', 'Larry', 'Moe'];  
  
friends.forEach(function() {  
  // do something  
})
```

## CONDITIONALS

---

```
if (condition) {  
    // do something  
}
```

```
if (condition) {  
    // do something  
} else {  
    // do something else  
}
```

```
if (condition) {  
    // do something  
} else if (condition){  
    // do something else  
} else {  
    // do something else  
}
```

## TERNARY OPERATOR

---

`(expression) ? trueCode : falseCode;`

## COMPARISON OPERATORS

---

<code>==</code>	<i>Equal to — DO NOT USE</i>
<code>===</code>	<i>Strict equal to</i>
<code>!=</code>	<i>Not equal to — DO NOT USE</i>
<code>!==</code>	<i>Strict not equal to</i>

<code>&gt;</code>	<i>Greater than</i>
<code>&lt;</code>	<i>Less than</i>
<code>&gt;=</code>	<i>Greater than or equal to</i>
<code>&lt;=</code>	<i>Less than or equal to</i>

## LOGICAL OPERATORS

---

<code>&amp;&amp;</code>	<i>and</i>
<code>  </code>	<i>or</i>
<code>!</code>	<i>not</i>

## FALSY VALUES

---

`false`

`0`

`""`

`NaN`

`null`

`undefined`