



# Dev Mountain - Javascript Cheat Sheet

Color coding is only mean to visually distinguish different or related items in a single example.

## VARIABLES

Set amount of information you want the computer to store for later use, that is associated with a user-defined keyword (called an identifier).

```
var name = data;
```

## DATA TYPES

As a coder, you signify to the Javascript interpreter what type of data you will be using by the syntax you use.

**Number**    var myNum = 5;

**String**    var myString = "Words";

**Boolean**   var trueOrFalse = true;

**Object**    var newObject = { }; (curly)

**Array**     var newArray = [ ]; (square)

## OPERATORS

### Mathematical Operators

+    add

-    subtract

\*    multiply

/    divide

%    remainder

### Assignment Operators

=    equals (assigns value)

+=   add by (adds number to value)

-=   subtract by

++   increment by 1

--   decrement by 1

### Mathematical Operators

<    greater than

>    less than

>=   greater than or equal to

<=   less than or equal to

!=    is not equivalent to

==    equivalent to

&&    and

||    or

## OBJECTS

Javascript objects are a collection of related data stored in a single variable and indexed with key value pairs.

```
var objName = {key: value, };
```

The syntax for calling an item out of an object can be written one of two ways:

```
objName . key
value
```

or

```
objName [key]
value
```

## ARRAYS

Arrays are collections of data items that do not have a key value pair, but instead are number indexed automatically in the order items are added to the array.

```
var arrayName = [value1, value2, ,];
```

The syntax for calling an item from an array is written like this:

```
arrayName [key]
value
```

Use the push function to add a data item to an array.

```
arrayName . push (value)
```

## FUNCTIONS

Functions allow us to define a set of actions to be performed when certain keywords are called.

```
function name (inputs) {actions}
```

Calling a function is as simple as giving the name and the arguments:

```
name (arguments)
```

## METHODS

A Method is a function that has been assigned as one items within a Javascript object.

```
var objName = {
  funcName : function () {code},
};
```

To call a method, access it from the object and pass in the arguments.

```
objName : methName (arguments);
```

## CONDITIONALS

Conditionals are used to define logic and evaluation within code, and check to see if a statement is true before performing a specified action. Conditionals may evaluate multiple statements, but as soon as one is found to be true, they will only execute one action.

```
if (condition1) {action1}
else if (condition2) {action2}
else {action3};
```

## LOOPS

Loops are how computers perform actions multiple times. The loop will check to see that a statement is true, and as long as it is, the loop will continue to run.

```
for (initial; condition; increment) {
  action
};
```

Looping through an array takes special syntax:

```
for (i=0; i<array.length; i++){action};
```

To loop through an object, use a for...in loop:

```
for (i in objName) {action};
```