

# Introduction to JavaScript

<http://learnxinyminutes.com/docs/javascript/>

# The History

- Javascript was created by Netscape's Brendan Eich in 1995. It was originally intended as a simpler scripting language for websites, complimenting the use of Java for more complex web applications, but its tight integration with Web pages and built-in support in browsers has caused it to become far more common than Java in web frontends.
- JavaScript isn't just limited to web browsers, though: Node.js, a project that provides a standalone runtime for Google Chrome's V8 JavaScript engine, is becoming more and more popular.

# The Basics

```
// Javascript has one number type (which is a 64-bit IEEE 754 double).
```

```
3 // = 3
```

```
1.5 // = 1.5
```

```
// All the basic arithmetic works as you'd expect.
```

```
1 + 1 // = 2
```

```
8 - 1 // = 7
```

```
10 * 2 // = 20
```

```
35 / 5 // = 7
```

```
// Including uneven division.
```

```
5 / 2 // = 2.5
```

# The Basics

```
// Strings are concatenated with +  
"Hello " + "world!" // = "Hello world!"
```

```
// and are compared with < and >  
"a" < "b" // = true
```

```
// Type coercion is performed for comparisons...  
"5" == 5 // = true
```

```
// ...unless you use ===  
"5" === 5 // = false
```

```
// You can access characters in a string with charAt  
"This is a string".charAt(0)
```

# Variables

```
// Variables are declared with the var keyword. Javascript is dynamically typed,  
// so you don't need to specify type. Assignment uses a single = character.  
var someVar = 5  
  
// if you leave the var keyword off, you won't get an error...  
someOtherVar = 10  
  
// ...but your variable will be created in the global scope, not in the scope  
// you defined it in.  
  
// Variables declared without being assigned to are set to undefined.  
var someThirdVar // = undefined  
  
// There's shorthand for performing math operations on variables:  
someVar += 5 // equivalent to someVar = someVar + 5; someVar is 10 now  
someVar *= 10 // now someVar is 100  
  
// and an even-shorter-hand for adding or subtracting 1  
someVar++ // now someVar is 101  
someVar-- // back to 100
```

# Arrays

```
// Arrays are ordered lists of values, of any type.
```

```
var myArray = ["Hello", 45, true]
```

```
// Their members can be accessed using the square-brackets subscript syntax.
```

```
// Array indices start at zero.
```

```
myArray[1] // = 45
```

# Control Flow

```
// The if structure works as you'd expect.
var count = 1
if (count == 3){
    // evaluated if count is 3
} else if (count == 4) {
    // evaluated if count is 4
} else {
    // evaluated if it's not either 3 or 4
,
// && is logical and, || is logical or
if (house.size == "big" && house.colour == "blue"){
    house.contains = "bear"
}
if (colour == "red" || colour == "blue"){
    // colour is either red or blue
}
```

# Loops

```
// As does while.  
while (true) {  
    // An infinite loop!  
}  
  
// the for loop is the same as C and Java:  
// initialisation; continue condition; iteration.  
for (var i = 0; i < 5; i++){  
    // will run 5 times  
}
```



# Iterating over Collections

```
var data = [{"lang": "en"}, {"lang": "en"},  
{"lang": "en"}, {"lang": "en"}, {"lang": "en"}];
```

```
// Iterates over array indexes  
for(var idx in data) {  
    console.log(idx);  
    console.log(data[idx]);  
}
```

```
var object = {"key1": "val1", "user" :  
"Martin"};
```

```
// Iterates over keys of the object  
for(var key in object) {  
    console.log(key);  
    console.log(object[key]);  
}
```

<http://jsbin.com/ogitel/2/edit>

# Matching and Comparing

- Most of what we do today is matching and comparing
- Rules of thumb:
  - Ignore stop words (and / or / the / into / ...)
    - [https://en.wikipedia.org/wiki/Stop\\_words](https://en.wikipedia.org/wiki/Stop_words)
  - Ignore word case (upper / lower)

# Matching and Comparing

```
// Split the data into words
var words = tweet.text.split(" ");
console.log(words.length);

var cleaned = [];

// Iterate over all words
for(var word in words) {
    // Convert to lower case
    var lc = words[word].toLowerCase();
    var found = false;
    // Check for the word in all stopwords
    for( var idx in STOP_WORDS_EN) {
        if (STOP_WORDS_EN[idx] == lc) {
            break;
        }
    }
    // If the word was not found in the stopwords
    // list, add it to the cleaned list
    if (!found)
        cleaned.push(lc);
}

console.log(cleaned);
```

**Stop word list is  
available as global  
variable in your code:  
STOP\_WORDS\_EN**

# Matching and Comparing

```
// Input data, check for tweet like words
var data = ["first", "tweet", "tweets", "berlin"];

for (var i in data) {
    var w = data[i];
    // find all words containing tweet and something like
    // tweets, tweeted, tweeteridu...
    if (w.match(/tweet(.*)\b/)){
        console.log(w);
    }
}
```

<http://regexpal.com/> - for testing regular expressions directly in the browser

[http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression)

# Further Readings

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/A\\_re-introduction\\_to\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript)
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript> overview of reference documentation
- <http://jsbin.com> to prototype some code
- <http://regexpal.com> to test regular expressions