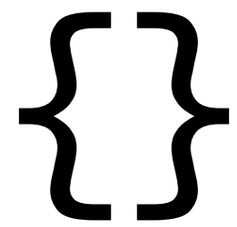
Syntax, Floating Point Numbers and Primitive Types

Syntax

There is no “Correct” JS style

Less Errors

Braces



```
if( true ){
  //do stuff
}
```

```
if( true ){ //do stuff }
```

```
if( true ) //do stuff
```

```
if( true ){
  //do stuff
}
```

Semi-colons

```
if( true ){
  var jeff = document.querySelectorAll( “.jeff” );
  jeff.className = “jeff isClassy”;
}
```

```
if( true ){
  var jeff = document.querySelectorAll( “.jeff” )
  jeff.className = “jeff isClassy”
}
```

Automatic Semi-colon Insertion

There for help. Not a crutch.

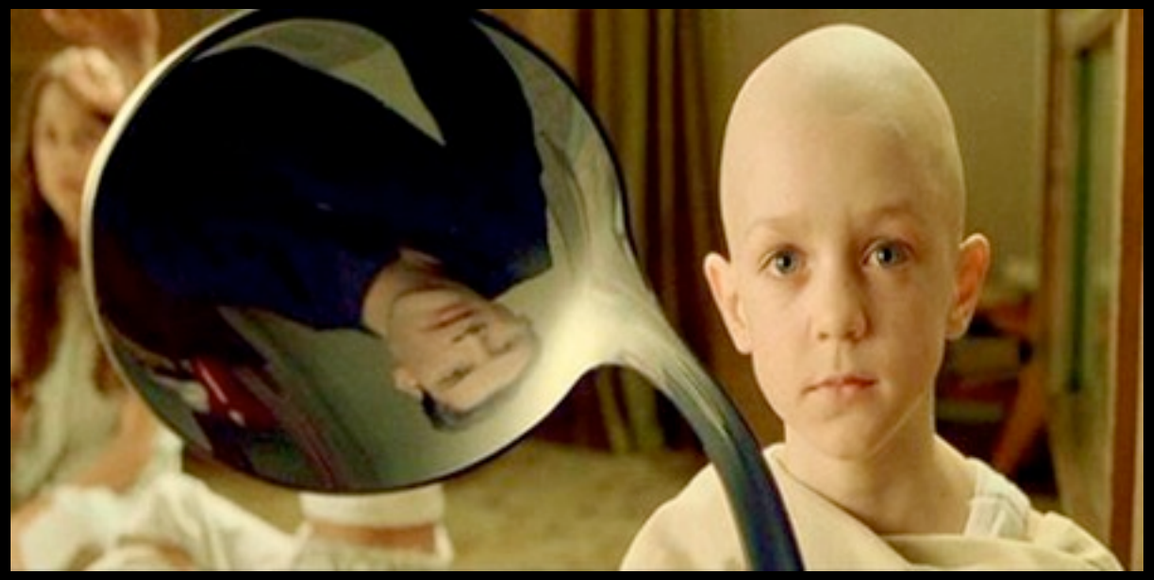
If, else if, else

```
var jeff = document.querySelectorAll( “.jeff” ),
    className;

if( bar === “foo” ){
  className = “jeff isClassy”
} else if( bar === “qua” ) {
  className = “jeff”;
} else {
  className = “jeff booooo”;
}

jeff.className = className;
```

There is no else if.



```
var jeff = document.querySelectorAll( “.jeff” ),
    className;

if( bar === “foo” ){
  className = “jeff isClassy”
} else if( bar === “qua” ) {
  className = “jeff”;
} else {
  className = “jeff booooo”;
}

jeff.className = className;
```

```
var jeff = document.querySelectorAll( “.jeff” ),
    className;

if( bar === “foo” ){
  className = “jeff isClassy”
} else {
  if( bar === “qua” ) {
    className = “jeff”;
  } else {
    className = “jeff booooo”;
  }
}

jeff.className = className;
```

Braces Optional.

while

```
while( condition ) {
    // Do stuff
}
```

for loops

```
var arr = [1,2,3,4,5];
for( var i = 0, l = arr.length; i < l; i++ ) {
    // Do stuff
}
```

Constructor

```
var f = new Foo();
```

Parentheses

```
var f = new Foo();

f.method();

method();

if(condition){
}

var fn = function(param){
};

function fn(param){
}

(function(){
}());
```

Primitives

Objects

```
var obj = Object.create( Object.prototype );

var obj = {};

var obj = {
    key: ‘value’,
    key2: ‘value’,
    key3: [values]
};
```

```
var obj = {};

typeof obj; //=> "object"
```

Functions

```
function fn(param){
}

var fn = function( param ){
};

var a = “a”;
fn(a);
```

```
function fn(){}
var fn1 = function(){};
```

```
typeof fn; //=> "function"
typeof fn1; //=> "function"
```

Arrays

```
var arr = [];

var arr = [1,2,3,4];

var arr = new Array(1,2,3,4);

var arr = new Array(1);
```

```
var arr = [];

typeof arr; //=> “object”
```

```
var arr = [];

Array.isArray( arr ); //=> true
```

```
var arr = [];
arr.length; //=> 0
arr[0]; //=> undefined

var arr = [1,2,3,4];
arr.length; //=> 4
arr[0]; //=> 1
```

```
var arr = new Array(1,2,3,4);
arr.length; //=> 4
arr[0]; //=> 1

var arr = new Array(1);
arr.length; //=>1
arr[0]; //=> undefined
```

Wait, what?

```
var arr = [];
arr.length; //=> 0
arr[0]; //=> undefined

var arr = [1,2,3,4];
arr.length; //=> 4
arr[0]; //=> 1
```

```
var arr = new Array(1,2,3,4);
arr.length; //=> 4
arr[0]; //=> 1

var arr = new Array(1);
arr.length; //=>1
arr[0]; //=> undefined
```

Always use [] for Array

Strings

```
var str = “hi”;

typeof str; //=> “string”
```

Boolean

```
var bool = true;

typeof bool; //=> “boolean”
```

Numbers

```
2.0.0p247 :001 > 1.0.class
 => Float
2.0.0p247 :002 > 1.class
 => Fixnum
```

Try that in IRB

Floating Point Numbers?

0.1 + 0.2 ===

0.30000000000000004

WTF

IEEE 754

Floating Point vs. Fixed Numbers

Floating Point is base 2

Squeezes infinite numbers into tiny space

Activity Time

- JSHint
- Sublimelinter
- https://github.com/rquinlivan/jshint-gem
- Rick Waldron's Idiomatic JS - https://github.com/rwaldron/idiomatic.js/
- Read about FP numbers - http://docs.oracle.com/cd/E19957-01/806-3568/