Intro to JavaScript

Cassidy Williams

@cassidoo

What is JavaScript?

It can do almost ANYTHING.

Any application that can be written in JavaScript, will eventually be written in JavaScript.

- Atwood's Law

- Go to repl.it
- Click JavaScript

Type the following in the console and hit "Run":

```
console.log('Hello, World!');
```

Variables

```
var x = 2;
console.log(x);
```

Variables let you **save values** that you'll use later.

Use `var` to make a new variable, name it (ours is named `x`), and then you can use it later.

What can variables be?

Short answer: Anything.

Long answer: A number, a string, a boolean, `null`, `undefined`, and mooooore!

What the heck are all of those things?

Numbers

Numbers can be whole numbers or decimals.

```
var myAge = 23;
var pi = 3.141592653;
```

Strings

Strings are wrapped in quotes.

```
var greeting = 'Hello, World!';
var beauty = 'Cassidy';
```

Booleans

```
var dogsSuck = false;
var hackathonsRule = true;
```

Notice how booleans look a lot like strings. They're **not**. Notice the lack of quotes.
Plus, there's only **two** possible values for booleans.

Null and Undefined

```
var thisIsNotDefinedYet;
```

`undefined` variables are ones that are not assigned anything. They don't have a value!

On the other hand, this is a `null` variable:

```
var goodPickupLines = null;
```

`null` variables are explicitly empty, and are defined to be empty!

Other notes about variables

- They start with a letter, a $, or a _
- They can only have letters, numbers, $, and _ in their names
- They are case-sensitive
- It's best to use camelCase for multiple word variable names

Variables can define expressions

```
var name = 'Cassidy';
var greeting = 'Hello ' + name;
var rank = 3 - 2;
var title = 'the #' + rank + 'developer in the world!';
var formalGreeting = greeting + ', ' + title;
```

This would print:

```
Hello Cassidy, the #1 developer in the world!
```

You can change variables

Let's just say you define a variable `fish` to be the string `"salmon"`, but you want it to be `"tuna"`.

```
var fish = 'salmon';
// uh oh that's not right
fish = 'tuna';
```

Boo yah. Also, notice, I left a **comment** in the code. What.

Comments

If you want to make notes for yourself, or you want to remove some code but still save it, you can add a comment!

```
// I'm pretty sure this variable isn't right
var potatosSuck = false;

/*
But how can we be sure?
If potatoes suck, do I suck?
What does it all MEAN?
*/
```

Notice that a single line comment starts with //,
and a multi-line comment has /* and */ bookending the statement.

Functions

Functions are re-usable collections of statements.

```
function crapAlternative() {
  console.log('turd');
}
```

And then, you can call your function as many times as you want by typing out:

```
crapAlternative();
```

Functions can also accept parameters.

```
function sayMyName(name) {
  console.log('Hi, ' + name);
}

sayMyName('Beyonce');
```

They can accept multiple parameters, too:

```
function add(n1, n2) {
  var result = n1 + n2;
  console.log(result);
}

adds(3, 21); // This will print 24!
```

They can also accept variables!

```
var ten = 10;
adds(ten, 11); // This will print 21!
```

The `return` statement

The return keyword returns a value to whoever calls the function, and also exits the function.

```
function add(n1, n2) {
  var result = n1 + n2;
  return result; // Stuff after this line won't be executed
}

var sum = add(15, 2);
```

You can call functions any way you want, as much as you want.

```
var bigSum = add(2, 50) + add(35, 2);
var massiveSum = add(add(52, 2), add(3, 7));
```

Conditionals

Conditionals are statements that are, well, conditions!

```
if (/* some boolean */) {
  /* execute this only if the boolean above is true */
} else {
  /* Do this instead */
}
```

You don't have to include an `else` statement.

Conditional Example

```
function tooBig(num) {
  if ( num > 100 ) {
    return 'That number is too big!';
  } else if ( num === 50 ) {
    return 'That number is nice.'
  } else {
    return 'That number is too small.';
  }
}
```

Using these bits of information, you can conquer anything. With JavaScript.

Additional resources

- developer.mozilla.org
- w3schools.com
- GitHub
- Stack Overflow

Cassidy Williams

cassidoo.co
github.com/cassidoo