JavaScript: Variable Scope

Why learn about variable scope?

- In order to program JavaScript well, you'll need to understand how scope affects the value of variables
- Without this understanding, you won't be able to predict the outcome of your JavaScript

Meet the Scopes

Global Scope

A variable that can be accessed globally, within almost any part of your script

```
var z = 1;

function printIt(){
  console.log(z); //Will return 1
}
```

Local Scope

A variable that only pertains to the function you are currently in

```
var z = 3;

function someF(z){
  console.log(z);
}
someF(10)
>>10

console.log(z);
>>3
```

Local Scope

Another example

```
var z = 3;

function someF(){
  var z = 20;
  console.log(z);
}
someF()
>>20

console.log(z);
>>3
```

Exercise: Global vs. Local

Create a script that exemplifies global scope and local scope by logging a variable to the console.

Block Scope

- Within any "block" of code, for instance an `if` statement, variables will mantain separate values than globally scoped versions of the variable
- This concept does not exist in JavaScript, **there is no block scope in JavaScript**

Block Scope: Example

```
var c = 10;

function hallo(){
  if(true){
    var c = 2;
  }
  console.log(c);
  // returns 2, not the global value 10
  // implication: variables don't have a
  // separate scope within a block
}
```

Function Scope

An argument is only accessible within the function it gets declared in

```
function haha(argument_uno){
  console.log(argument_uno);
}

haha("hello");
>>"hello"
console.log(argument_uno);
>>ReferenceError: argument_uno is not defined
```

Exercise

Write a "cheat sheet" of runnable JavaScript code and comments that explain the concepts in this lecture so you have a reference

Resources

TeamTreeHouse

JavaScript Basics - Creating Reusable Code with Functions