Scope

original
mint
SCOPE
MOUTHWASH
Original Mint
1 L (33.8 FL OZ)

Variable scope

Globals, BOOOOOO

Global variables crowd up the window

They could mess with libraries

They’re difficult to track

Let’s avoid them

Use var

```
var x = “hi”; //local scope
x = “hi”; //global scope
```

Functional scoping

```
var fn = function(){
  var a = "a"";

  var fn2 = function(){
    var b = "b";

    console.log( a ); //=> "a"
    console.log( b ); //=> "b"
  };

  console.log( b ); ==> undefined
};
```

Hoisting

```
var fn = function(){
  var foo = "foo";
  console.log( foo );
  console.log( bar );
};

fn();
// => ReferenceError: bar is not
defined
// => "foo"
```

```
var fn = function(){
  var foo = "foo";
  console.log( foo );
  console.log( bar );
  var bar = "bar";
};

fn();
// => "foo"
// => undefined
```

• • •

All explained by the act of Hoisting

```
var fn = function(){
  var foo = "foo";
  console.log( foo );
  console.log( bar );
  var bar = "bar";
};

fn();
// => "foo"
// => undefined
```

is the same as

```
var fn = function(){
  var foo = "foo";
  var bar;
  console.log( foo );
  console.log( bar );
  bar = "bar";
};

fn();
// => "foo"
// => undefined
```

Review

var declarations are at the top of their function

```
var fn = function(){
  var foo = "foo";
  var bar;
  console.log( foo );
  console.log( bar );
  bar = "bar";
};

fn();
// => "foo"
// => undefined
```

IIFEs

Immediately-invoked Function Expressions

```
(function( window ){
  var x = “hi”;
  console.log( y );
  //=> Reference Error
}(this));

(function( window ){
  var y = “hey”;
  console.log( x );
  //=> Reference Error
}(this));
```

Use to wrap your JS files

tyra
darren
wants a drama-free relationship

So does your JavaScript. Pay attention to scope

Activity!

```
function bar() {
    return foo;
    foo = 10;
    function foo() {}
    var foo = 11;
}
alert(typeof bar());
```

```
function bar() {
    return foo;
    foo = 10;
    function foo() {}
    var foo = 11;
}
alert(typeof bar());
```

function

```
x = 1;
function bar() {
    this.x = 2;
    return x;
}
var foo = new bar();
alert(foo.x);
```

```
x = 1;
function bar() {
    this.x = 2;
    return x;
}
var foo = new bar();
alert(foo.x);
```

2