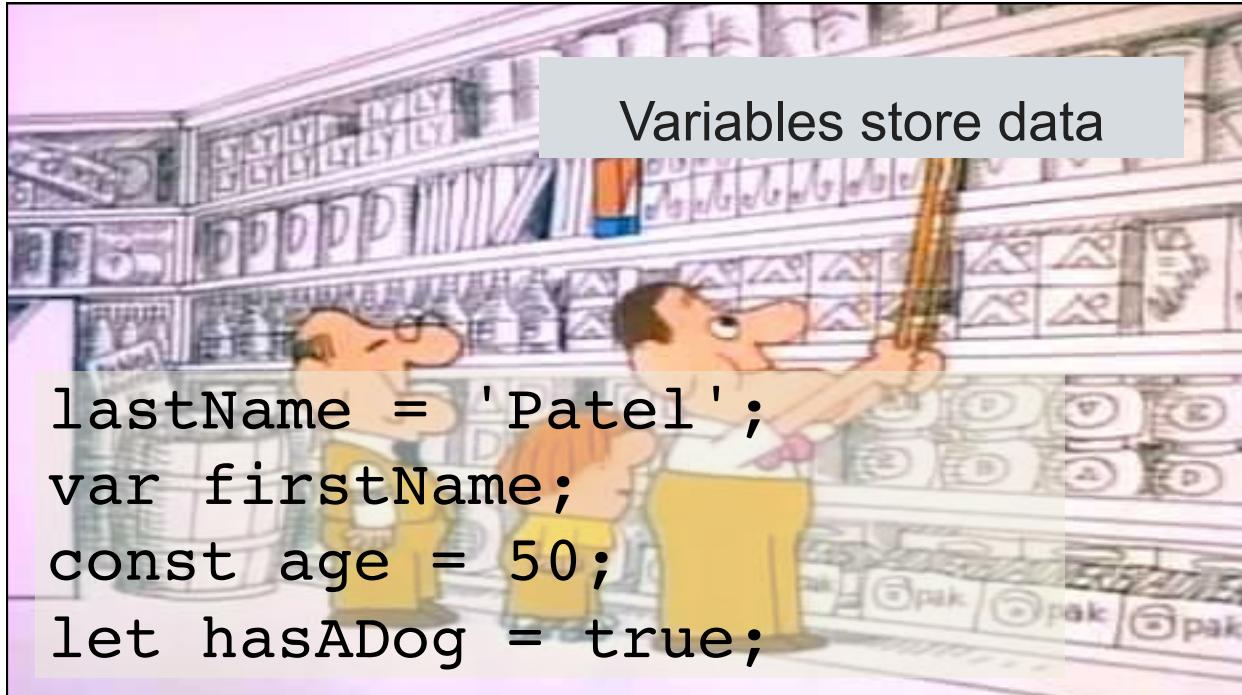


# Variables

tl;dr

- It hoists vars, giving them function scope but leaves lets and consts, giving them block scope
- It has dynamically-typed values
  - undefined
  - number
  - string
  - boolean
  - function
  - object



Let's start with this

To use a non-existent variable is a fatal error.

```
let x = 5;

let z = x + y; //ReferenceError: y is not defined
```

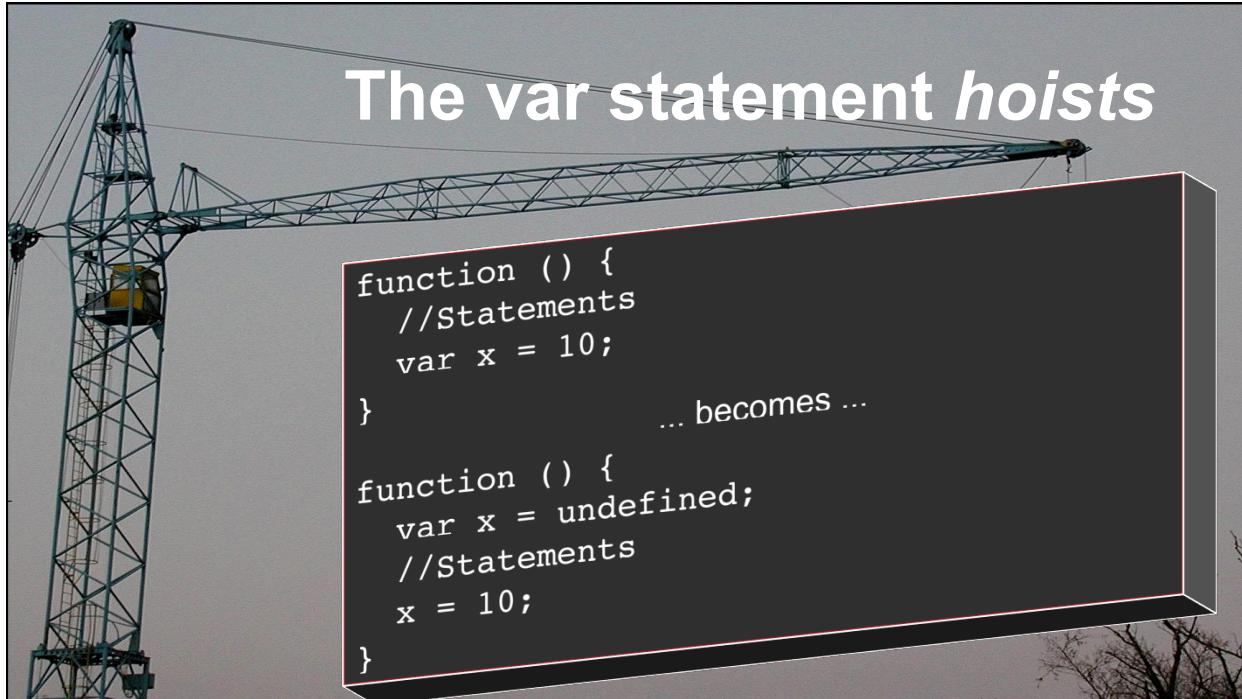
# Hoisting

What happens to x?

```
1 var city = document.getElementById('city').value;  
2 if (city === 'Albuquerque') {  
3   var x = 10;  
4 }  
5 console.log("x is ", x);
```

What happens on line 5?

- A) x is always 10
- B) x may be 10 or it may be undefined
- C) A ReferenceError is thrown



## Variable scoping

var, let, and const

Scoping refers to where a variable can be seen

Global scope  
Function scope  
Block scope

```
x = "foo";  
y = 5;
```



Variables are global by default

To give them function scope ...

1. Put them anywhere in a function
2. Use var

To give them block scope ...

Use let or const:

```
let foo;  
let bar = 5;  
const baz = 10;
```



## let isn't hoisted

- This doesn't work

```
console.log(bar);
// referenceError; bar isn't defined
let bar="value";
```

- This is okay, though

```
function readThere () {
    return there
}
let there = 'foo'
console.log(readThere())
```



## const behaves just like let

Not hoisted  
Block-scoped



except that const is constant ...

- Values must be assigned on declaration:

```
const x = 10;
```

```
x='foo'; // throws -- TypeError
```

```
const x = 'foo'; // throws -- redeclaration
```



... except when it's not

```
const simpsons = ['homer', 'marge', 'bart', 'lisa'];
simpsons.push('maggie'); // totally works.
const neighbor = {
  first: "Ned"
}
neighbor.kids = ["Rod", "Todd"]; // Also works.
```

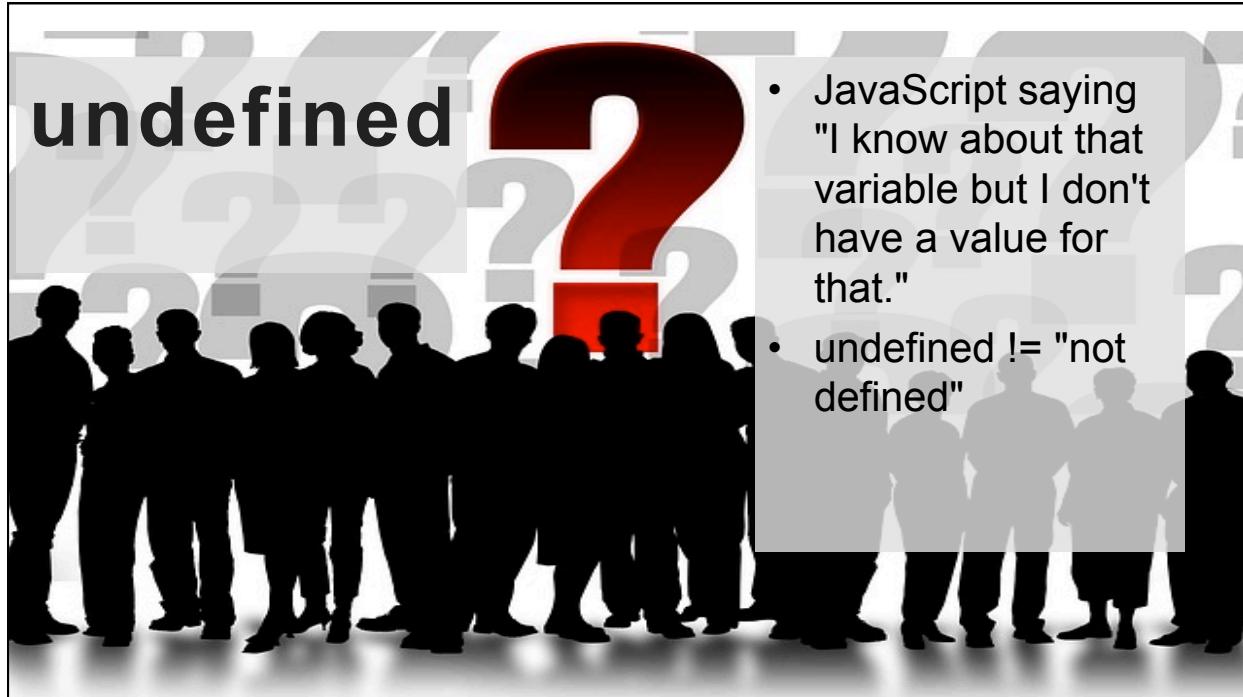
**const is not immutable**



# Data types

JavaScript is weakly typed

- undefined
- number
- string
- boolean
- function
- object



- JavaScript saying "I know about that variable but I don't have a value for that."
- undefined != "not defined"

## How do I know what I'm working with?

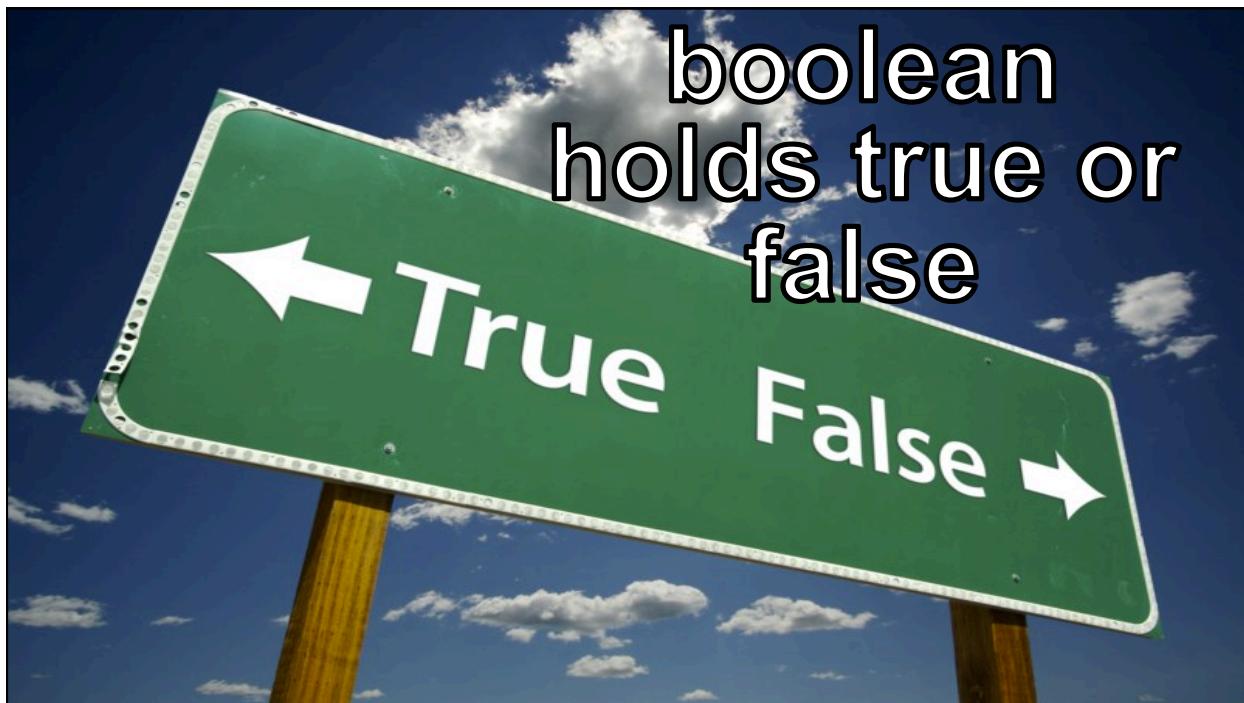
- The typeof operator

```
function foo(x) {  
    if (typeof x != "object")  
        throw "I need an object";  
    else  
        // Do stuff with that object  
}
```

## Numbers

- Numbers are IEEE754 double-precision floating point numbers
  - Max size is 1.8 E 308 or 9 quintillion

```
alert([Number.MAX_VALUE, Number.MIN_VALUE]);
```



## JavaScript Objects are simple hashes

- Merely a collection/hash/dictionary of key-value pairs. Very simple.

firstName	"Rufus"
middleName	"Xavier"
lastName	"Sarsparilla"
age	10
sister	rafaela
pet	rhinoceros

- Not based on classes at all.

## "Truthiness" = coercing to boolean

```
var x = _____;
if (x) console.log('true'); else console.log('false');
```

If x is ...	console says ...
true	true
false	false
"any string"	true
""	false
"false"	true
100	true
0	false
100/0	true
null	false
undefined	false
{}	true



## tl;dr

- It hoists vars, giving them function scope but leaves lets and consts, giving them block scope
- It has dynamically-typed values
  - undefined
  - number
  - string
  - boolean
  - function
  - object