

JavaScript Data Types

Our Goals

- Identify and explain primitive data types in JavaScript
- Define what a variable is and explain how and why we would use them
- Be able to discuss conventions and rules when creating variables
- Distinguish between loose and static typing
- Be able to use JavaScript comments
- Identify statements and expressions in JS code

What do I do with it?

- Data types are the types of information provided by a programming language
- **Primitive** data types are the basic building blocks of a language
 - They are immutable (cannot be changed)
- There are also **composite** types. These are types of data that can be constructed by combining primitive and other composite data types
 - They are not immutable

Primitive Data Types

- string
- number
- boolean
- null
- undefined
- (symbol)

There are more!

But they are all composite types...

- Objects
- Arrays
- Functions
- etc.

Strings

```
"Hello World";
```

```
"Jane's Bag";
```

```
'It\'s gibberish!';
```

Numbers

```
42;
```

```
1;
```

```
0.2;
```

```
-1242.151251;
```

Booleans

```
true;
```

```
false;
```


Undefined

```
undefined;
```

```
// No value yet
```

Null

```
null;
```

```
// This is explicitly empty
```

Random Harvest, 1942

I don't want to forget!

- Enter variables.
- Variables are ways to store information in memory.
 - You assign a name to a certain piece of data or collection of code. Think of them as named containers.
- The name of a variable is occasionally called the JavaScript identifier.

How do they work?

```
var name = "James Joyce";
```

```
// OR...
```

```
var name;
```

```
name = "James Joyce";
```

How do they work?

```
var x = 10;  
console.log( x );
```

```
x = 20;  
console.log( x );
```

Expressions vs. Statements

An expression **evaluates** - it produces a value

A statement **performs an action**

Variables can store results

```
var x = 2 + 2;
```

```
var y = 2 * 3;
```

```
var name = "James " + "Joyce";  
// Concatenation!
```


They can change types

```
var x = 2;
```

```
x = null;
```

```
x = "Hello World";
```

Static and Dynamic

- **Static typing** is where you have to say that a particular variable is going to be a string or number for example
- **Dynamic (loose) typing** is where it can change as it goes

What's JavaScript's type system?

How to name them

- Begin with letters, \$ or _
- Only contain letters, numbers, \$ or _
- Case sensitive
- Avoid reserved words
- Choose clarity and meaning
- Pick a naming convention
 - Use camelCase for multipleWords

Comments

Remember that there are two audiences for every program that you write! Humans and computers. Code for the humans, which means writing comments that explain what is going on is very good practice.

```
// A single line comment
```

```
/*  
    A multiline comment  
    (I'd avoid these)  
*/
```

Our Goals

- Identify and explain primitive data types in JavaScript
- Define what a variable is and explain how and why we would use them
- Be able to discuss conventions and rules when creating variables
- Distinguish between loose and static typing
- Be able to use JavaScript comments
- Identify statements and expressions in JS code

Have a go at **these**
exercises