Functions 1

Javascript

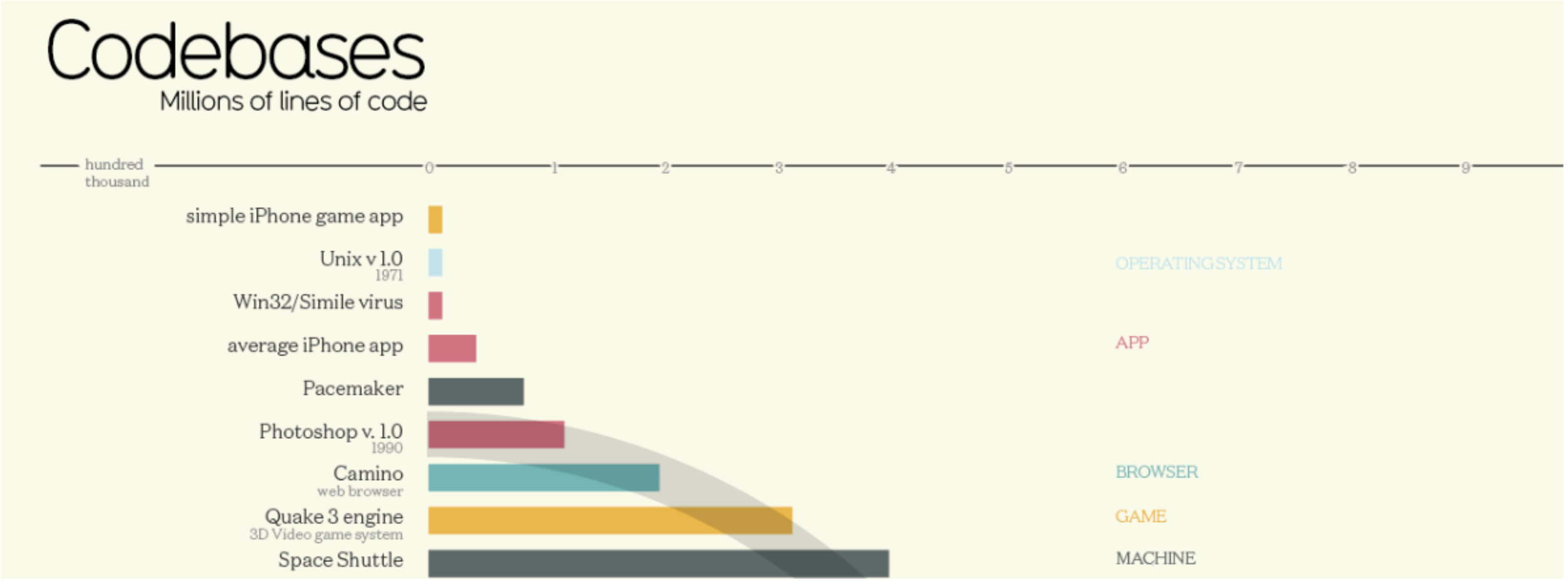
NIFR 2023

https://github.com/nicklasdean/ita-23-1-sem-code

Agenda

Javascript Functions

- Why Functions?
- Using functions
  - Utility Functions: Arrays
- Defining custom functions
  - Scope/Body
  - Argument/Parameters
  - Return values



https://www.visualcapitalist.com/millions-lines-of-code/

Applications are code

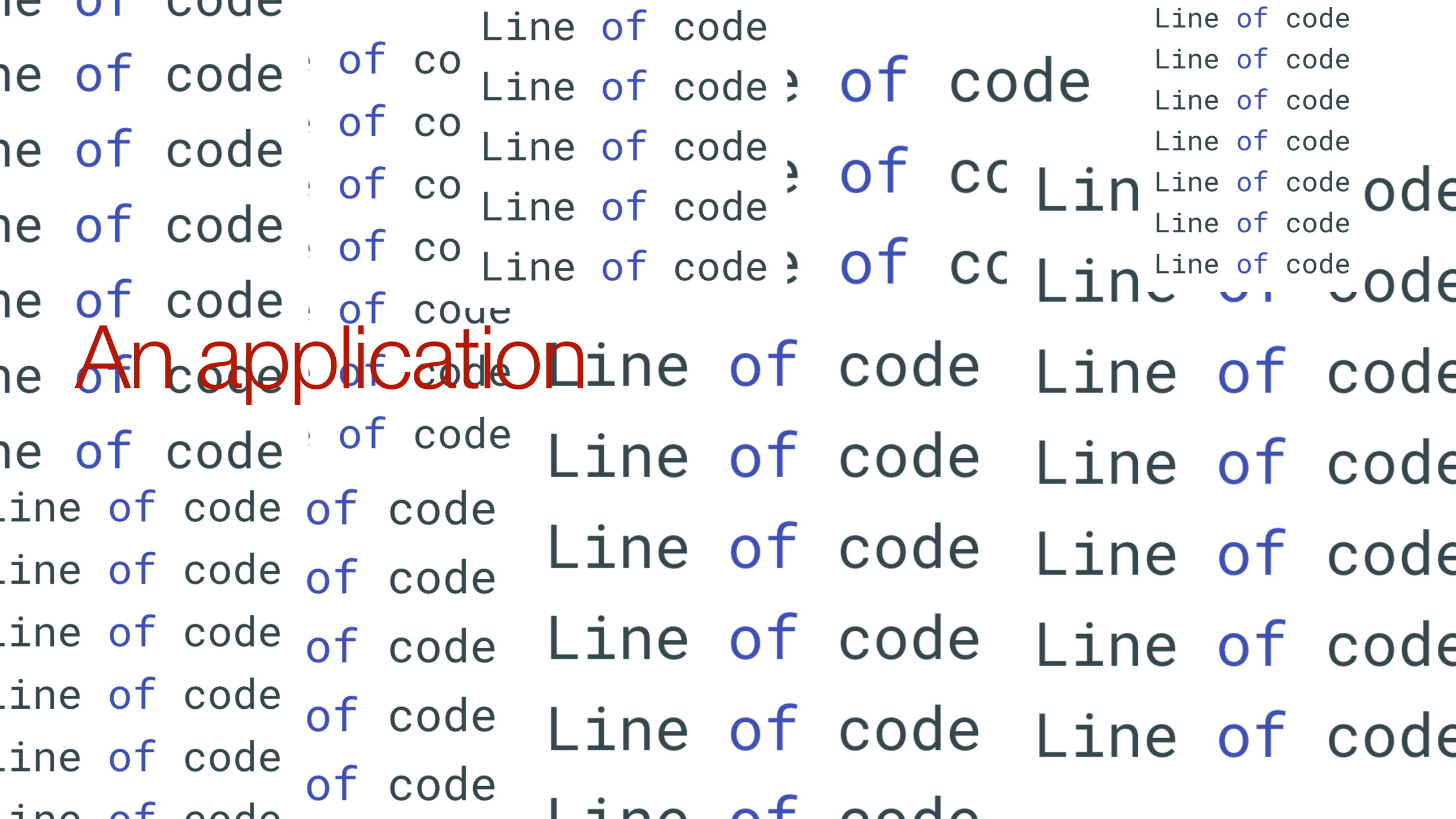
Code needs structure

Functions

Provide structure

```
function(){}
```

An application



```
function doSomething(){
    Line of code
    Line of code
    Line of code
    Line of code
    Line of code
    Line of code
    Line of code
}
```

```
function doSomethingElse(){
    Line of code
    Line of code
    Line of code
}
```

```
function doSomethingCompletelyElse(){
    Line of code
    Line of code
    Line of code
}
```

An application

function(){}

function(){}

function(){}

function(){}

function(){}

function(){}

function(){}

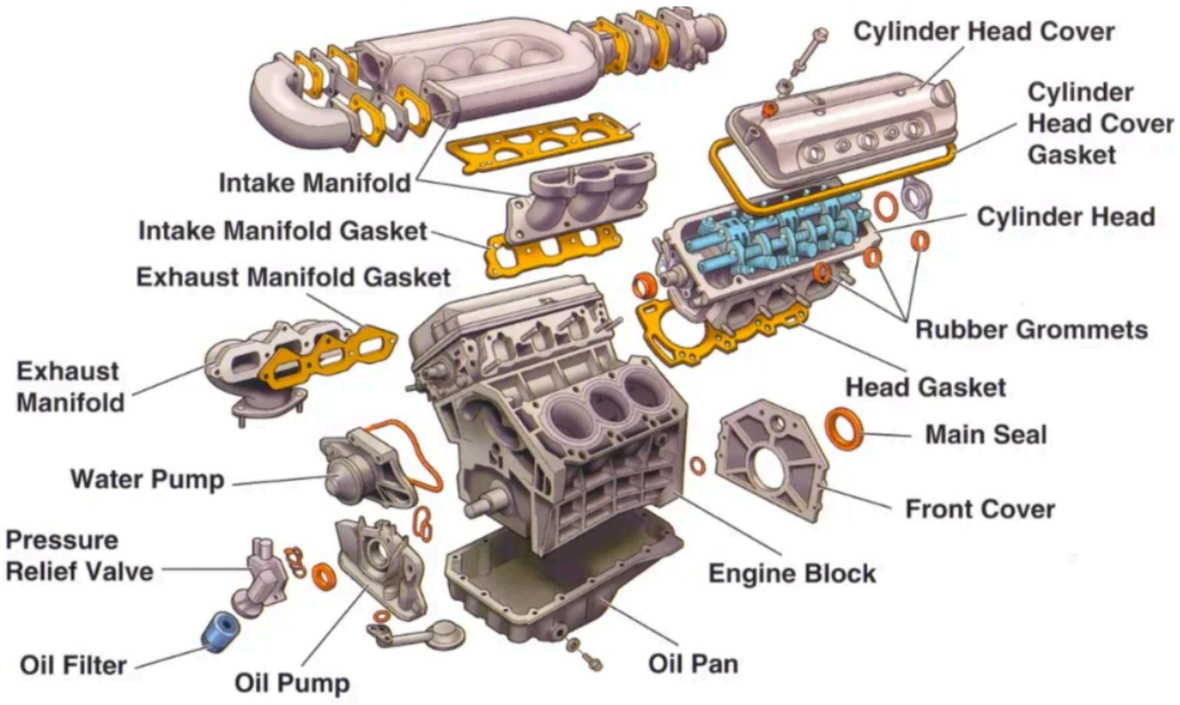
function(){}

function(){}

Functions

Why?

- Engineer codebases of reusable & modular parts
- Organise code into manageable & reasonable sizes
- Better readability
- Provide abstraction
- Reduce duplication of code
- Organise work & tasks



Today’s focus: Building small parts

Functions

Agenda

Javascript Functions

- ~~Why Functions?~~
- Using functions
  - Utility Functions: Arrays
  - Arguments
- Defining custom functions
  - Scope/Body
  - Parameters
  - Return values

How do we call a function?

Provide it with arguments (or not?)

```
console.log("Hello World");
```

Function

Argument

How do we call a function?

Provide it with arguments (or not?)

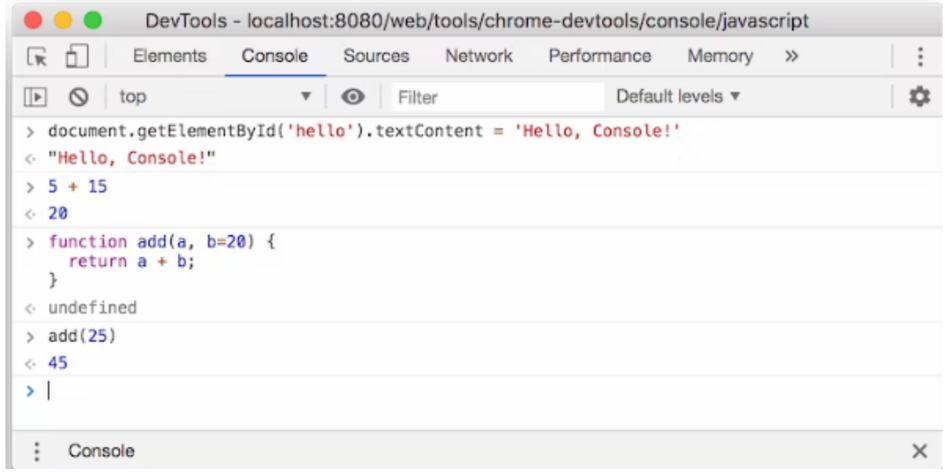
```
array = [5,2,4,3,4,4,123,4,5,6,7,2,3,4];
array.sort();
```

No argument

Using functions

Built-in functions

- Certain objects have built-in functions
- Common operations such as:
  - String manipulation
  - Datatype parsing (from string to number)
  - Array functionality
  - Regular expressions
  - Date handling



https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects

Why does this function (charAt) need an argument?

Provide it with arguments (or not?)

JS

```
"cat".charAt(1); // gives value "a"
```

Why does this function (toUpperCase) not need an argument?

Provide it with arguments (or not?)

JavaScript Demo: String.toUpperCase()

```
1 const sentence = 'The quick brown fox jumps over the lazy dog.';
2
3 console.log(sentence.toUpperCase());
4 // Expected output: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG."
5
```

split & concat example

Exercises set A

Agenda

Javascript Functions

- Why Functions?
- Using functions
  - Utility Functions: Arrays
- Defining custom functions
  - Scope/Body
  - Argument/Parameters
  - Return values

Anatomy of a

```
function(){}
```

```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

Name (action)

Parameters

```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

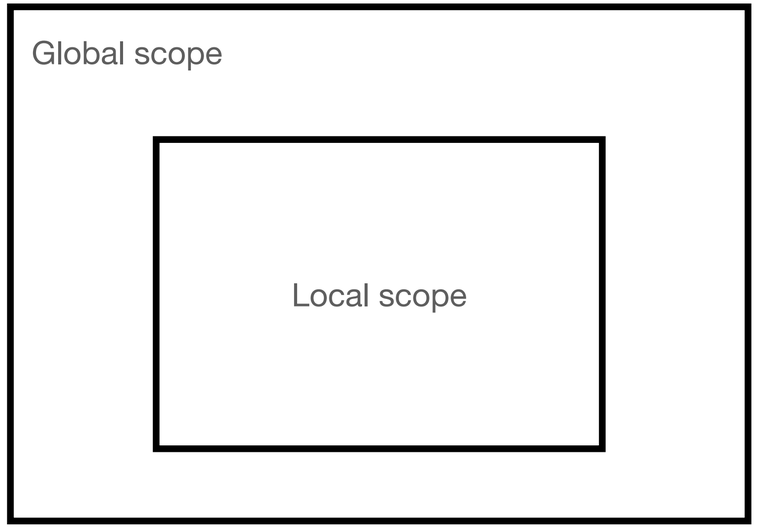
```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

Function body

Scope

Function scope

- The scope is like a house of mirror walls
- We can look outside
- The outside cannot look inside



Scope

Function scope

- Not all variables or constants are created equal
- If a variable or function is created within a function, they cannot be reached outside of the function
- That is why the example on the right will generate an error

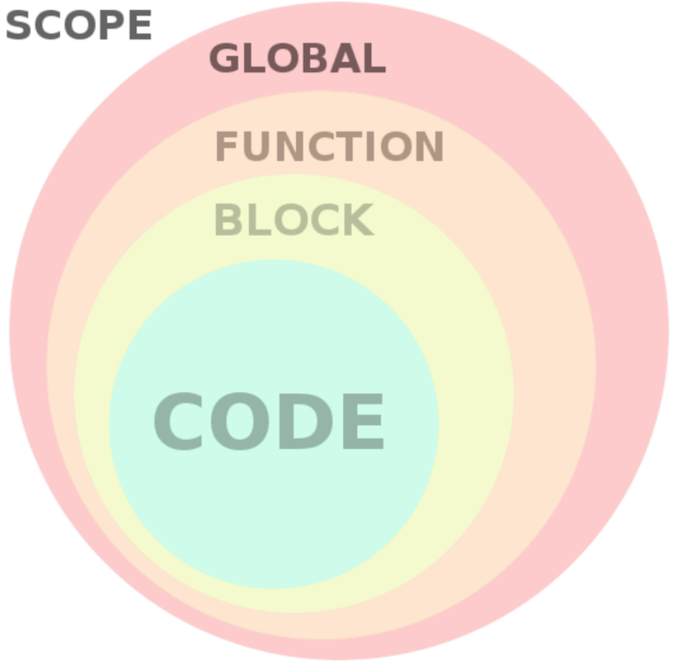
```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}

console.log(result);
```

Scope

Function scope

- Global variables can be reached within local functions
- Function scope cannot be reached within the global scope
- It will make sense with time




```
const applicationName = "calculator";
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(applicationName + " just printed " + result);
}
```

```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}

console.log(result);
```

Error: Due to local scope

```
const applicationName = "calculator";
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(applicationName + " just printed " + result);
}
```

Fine: applicationName is global

Return keyword

In functions we have two basic options:

- Get something
- Do something

Functions

Return or not

- Return: Get something from the fridge

- Void: Rearrange the fridge

Functions that get something will have a return value

```
function sumTwoNumbers(a,b){
    let result = a + b;
    return result;
}
```

Functions that does not return are called ‘void’ functions

```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

Examples

Printing the length of a string

Returning the length of a string

Exercise set B

Learning objectives

Javascript Functions

- Why Functions?
- Using functions
  - Utility Functions: Arrays
- Defining custom functions
  - Scope/Body
  - Argument/Parameters
  - Return values