LEARN WEB

ale66

github.com/ale66/learn-web

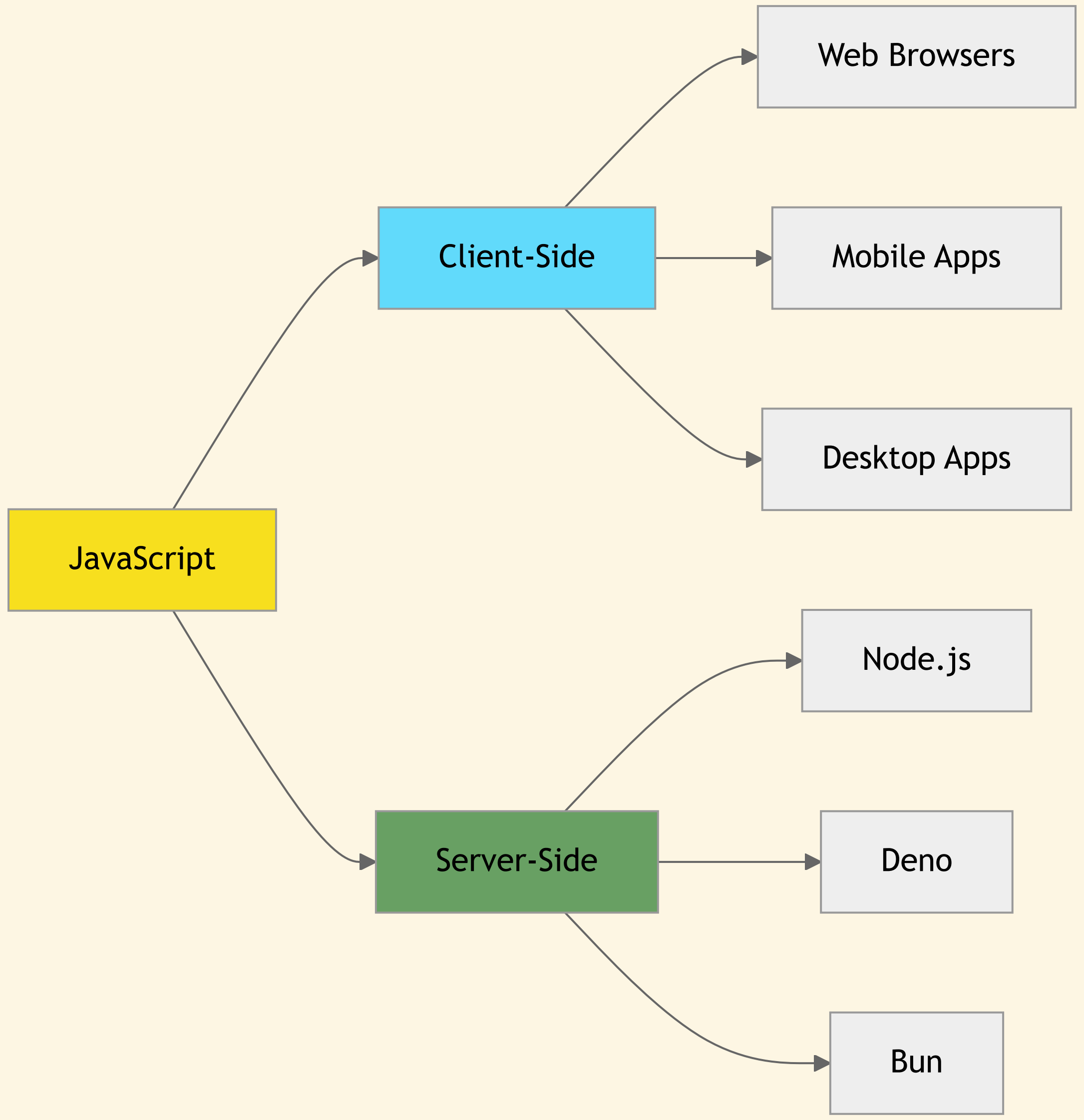
JAVASCRIPT

JS

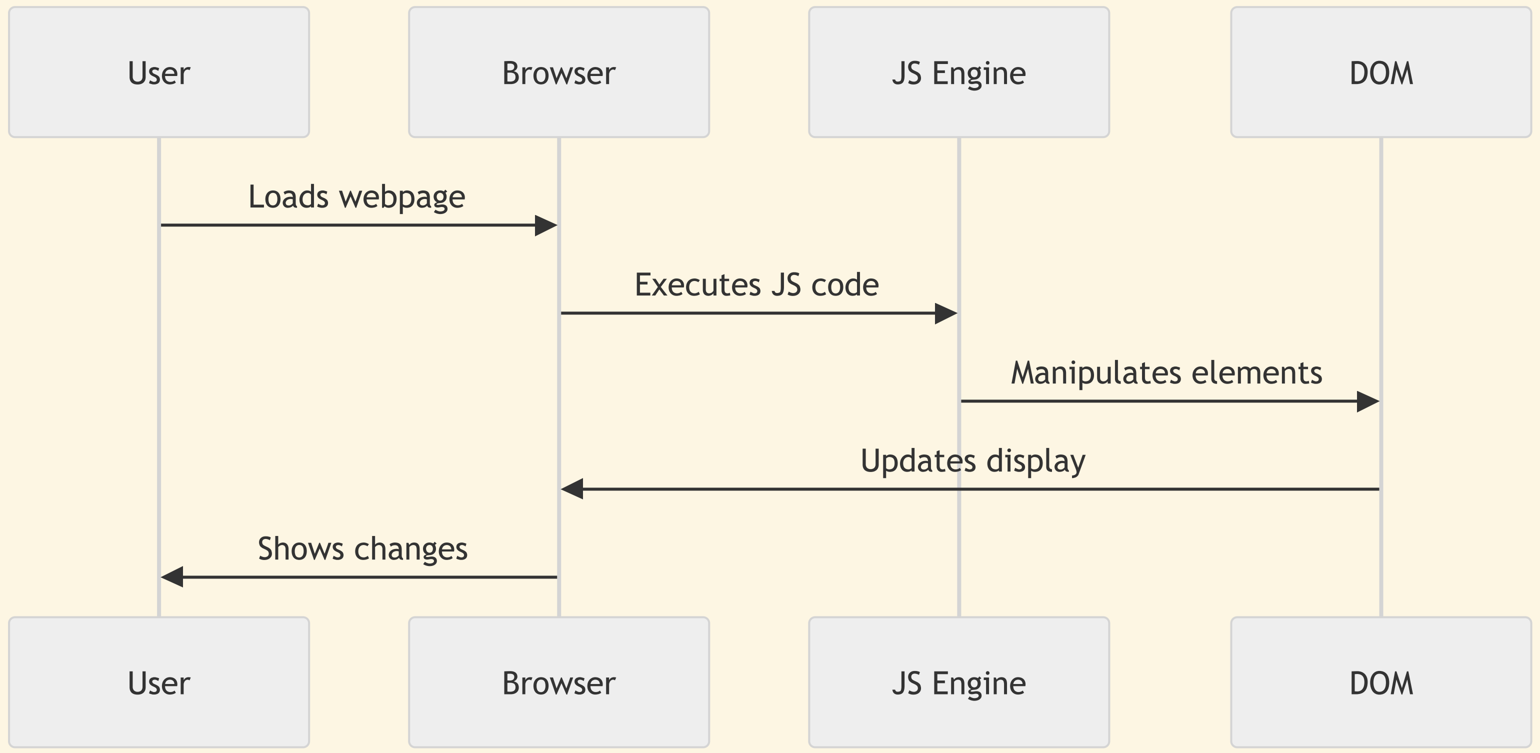
- animates web pages
- taylors pages to users and let them *act*
- the only programming language that *runs in the browser* (now also on the server)

- extremely hard and unforgiving for learners
- a core technology of the web is a mistake, essentially

THE JS *ECOSYSTEM* TODAY



JS BETWEEN USERS AND THEIR PAGES



JS EXAMPLES

```javascript
// Variables
let name = "Alice";
const age = 25;

// Functions
function greet(person) {
  return "Hello, " + person + "!";
}

// Calling a function
console.log(greet(name));
```

notice `;` as line terminator

HELLO WORLD! IN JS

```html
<html>
<head></head>
<body>
  <h1 id="greeting">Welcome</h1>

  <button onclick="changeGreeting()">Click Me</button>

  <script>
    function changeGreeting() {
      document.getElementById('greeting').textContent =
        'Hello, JavaScript!';
    }
  </script>
</body>
</html>
```

Result: Button click changes the heading text

EXAMPLE 2: INTERACTIVE COUNTER

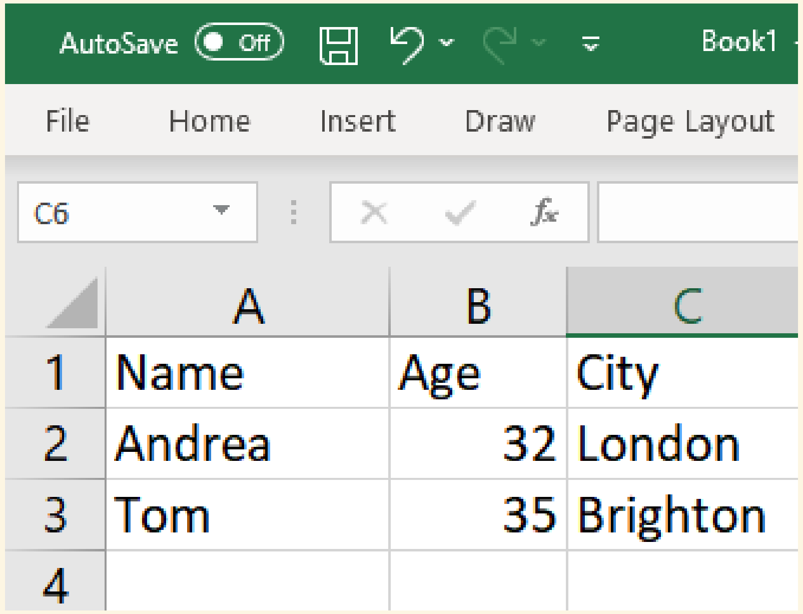
```html
<html>
<head></head>
<body>
  <h1>Count: <span id="count">0</span></h1>
  <button onclick="increment()">Increment</button>
  <button onclick="decrement()">Decrement</button>
  <button onclick="reset()">Reset</button>

  <script>
    let count = 0;

    function increment() {
      count++;
      updateDisplay();
    }

    function decrement() {
      count--;
```

A STEP BACK: THE BASICS

VARIABLES

A JS variable is a symbolic name for some content, the *value*, that is kept in memory

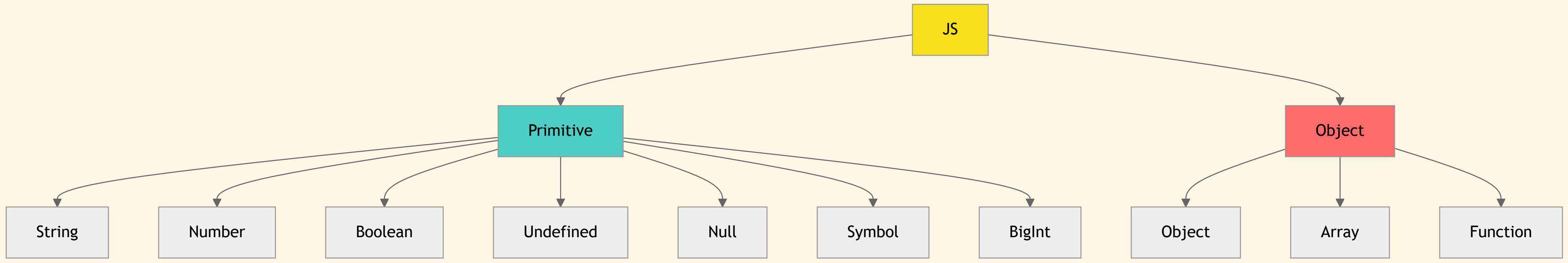
In spreadsheets, cells are variables



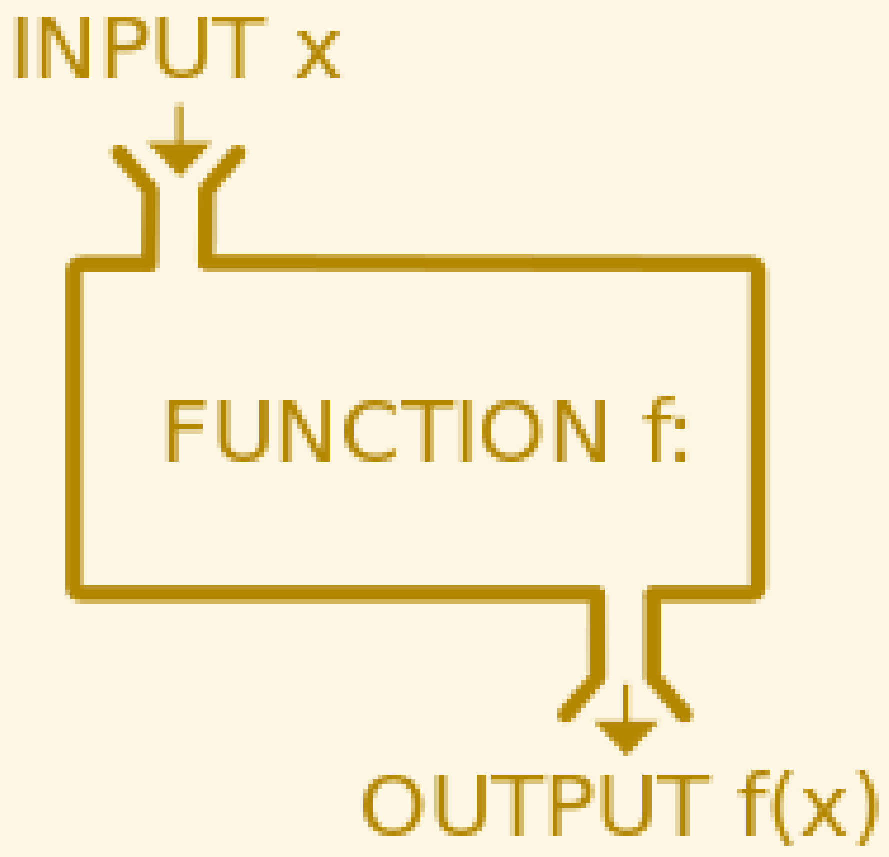
	A	B	C
1	Name	Age	City
2	Andrea	32	London
3	Tom	35	Brighton
4			

`B2 = 35` is a variable with name `B2`, content `35` and type `int`

DATA TYPES


JS
Primitive
Object
String
Number
Boolean
Undefined
Null
Symbol
BigInt
Object
Array
Function

JS FUNCTIONS



Functions are a key abstraction to model nature and processes

a regular input/output or cause/effect behaviour is identified and *given a name*

```
The higher the temperature the quicker pizza cooks.

Cooking time is a function of the temperature in the oven.
```

FUNCTIONS IN CODING

A function is a block of code (instructions) that

- has a clear input/output definition and
- executes in a separated environment

Spreadsheets: `B4 = (B2 + B3)/2` is a function

```
/* Convert Italian exam marks into percentages */
function marks2pc(marks) {

  let converted = (marks / 30) * 100

  // Math.round() is a 'foreign' function that rounds up 50.65 --> 51 etc.
  let pc = Math.round(converted);

  return pc;
}
```

`marks` is a *parameter* of the f.

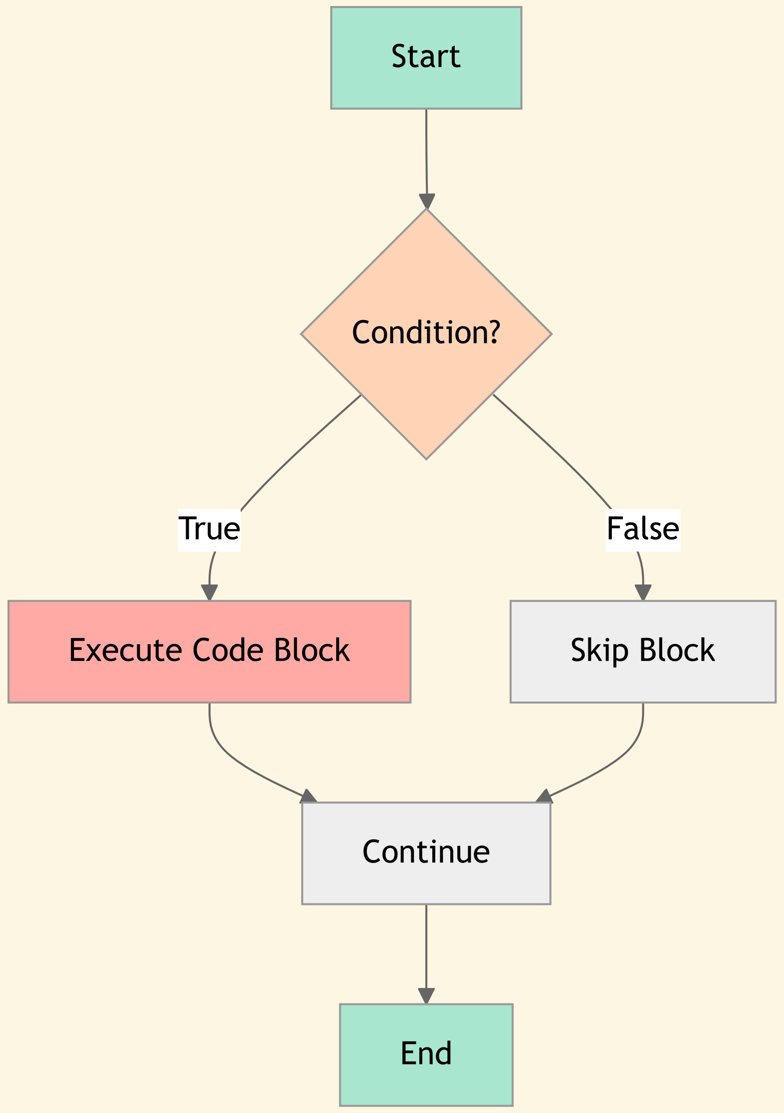
`pc` is the *return value* of the f.

OBSERVATIONS

Functions only run when they are called ('invoked') within a code in execution

```
/* convert and show on the page the UK version of the Webcomm marks */
let my_marks = 27;

let uk_marks = marks2pc(my_marks);

document.getElementById("convertedMarks").innerHTML = uk_marks;
```

CONTROL FLOW


Start
Condition?
True
False
Execute Code Block
Skip Block
Continue
End

CONDITIONAL STATEMENTS

```javascript
let age = 18;

if (age >= 18) {
  console.log("You are an adult");
} else if (age >= 13) {
  console.log("You are a teenager");
} else {
  console.log("You are a child");
}

// Ternary operator
let status = age >= 18 ? "adult" : "minor";
```

EXAMPLE 3: AGE CHECKER

```html
<html>
<head></head>
<body>
  <h1>Age Verification</h1>
  <input type="number" id="ageInput" placeholder="Enter your age">
  <button onclick="checkAge()">Check</button>
  <p id="result"></p>

  <script>
    function checkAge() {
      const age = document.getElementById('ageInput').value;
      const result = document.getElementById('result');

      if (age === '') {
        result.textContent = 'Please enter your age';
      } else if (age < 13) {
        result.textContent = 'You are a child';
      } else if (age < 18) {
```

ITERATION

BASIC IDEA

We need to operate over sequences/collection of atomic data

Example: column operations in spreasheets

```
=AVERAGE(A1:A100)
```

```
=AVERAGEIF(A1:A100, ">0")
```

```
=ROUND(A1, 2)
```

then pull the formula over the whole column.

ITERATIONS, A

```
// For loop
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

ITERATIONS, B

```javascript
// While loop
let count = 0;
while (count < 5) {
  console.log(count);
  count++;
}
```

INDEXED DATA

A sequence of values stored in a variable that can be accessed individually by means of their **position** (index)

```
let fruits = ["apple", "banana", "cherry"];

console.log(fruits[0]);  // "apple"
console.log(fruits[1]);  // "banana"
console.log(fruits[2]);  // "cherry"
```

- use of square brackets
- indices start at **0**
- each element has a unique position
- two main types: **arrays** and **strings**

STRINGS

Text treated as a sequence of keyboard characters

Same indexing as arrays

```
let word = "Hello";

console.log(word[0]);  // "H"
console.log(word[1]);  // "e"
console.log(word[4]);  // "o"
```

THE LENGHT

Both arrays and strings have a **length** property

```javascript
let colors = ["red", "green", "blue"];

let name = "JavaScript";

console.log(colors.length);  // 3
console.log(name.length);    // 10
```

Last element is always at index: `length-1`

PRACTICE

```javascript
let numbers = [10, 20, 30, 40, 50];

let message = "Code";

// What will these output?
numbers[3]
message[0]
numbers[numbers.length - 1]
```

ITERATIONS, C

```javascript
// fruits is an array of strings
const fruits = ['apple', 'banana', 'orange'];
for (const fruit of fruits) {
  console.log(fruit);
}
```

```
/* convert and show on the page the UK version of the marks */
for (const m of my_italian_exam_marks){
  let uk_marks = marks2pc(m);
  console.log(uK_marks)
}
```

Copy and run it on pythontutor.com

`console.log()` and `window.alert()` are simple ways to print out results.

OBSERVATIONS

Functions should be defined every time a block of code is required to appear more than once:

- improve readability
- improve maintainance

JS is probably the hardest programming language for learners 😳