

LEARN WEB

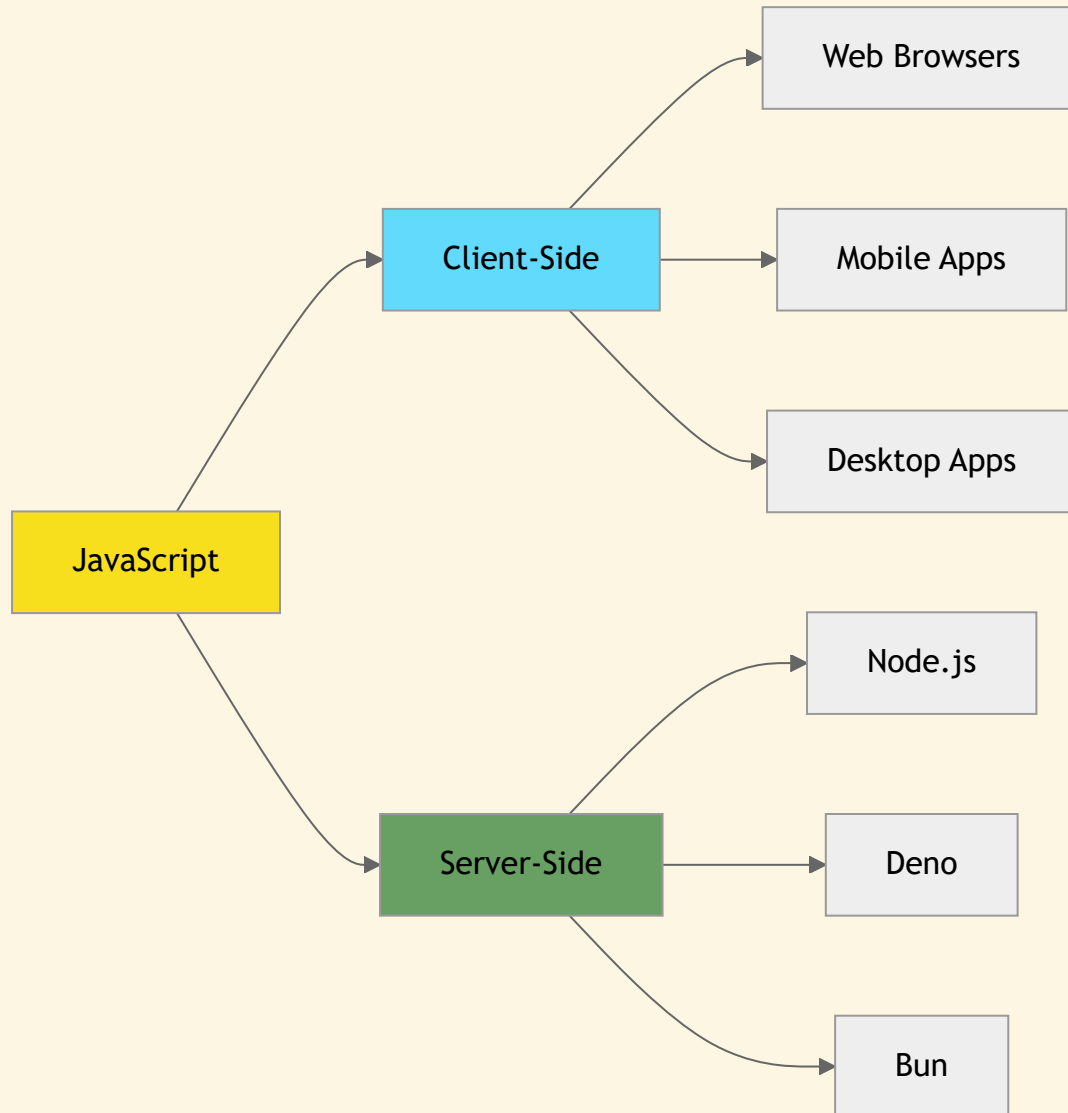
ale66

JAVASCRIPT

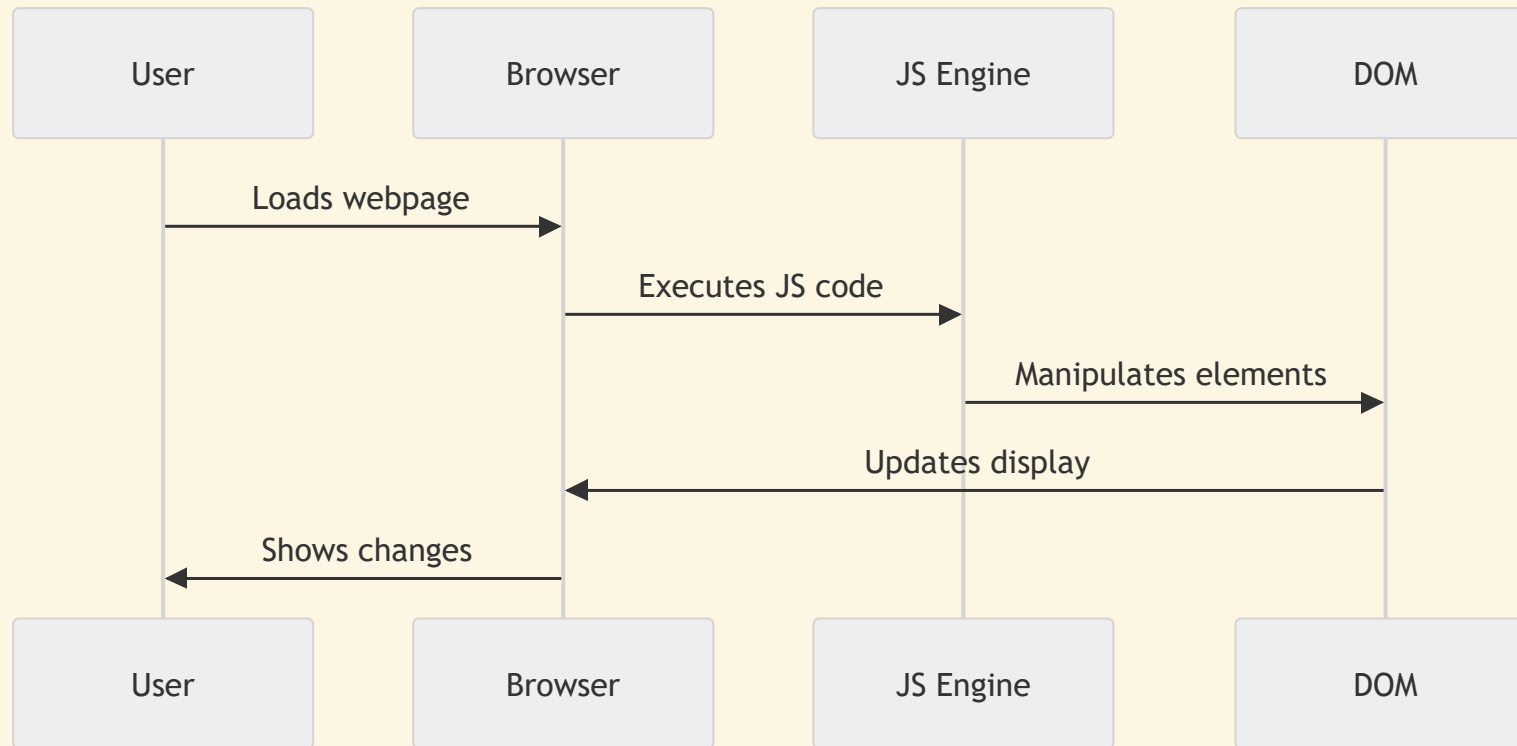
JS

- animates web pages
- tailors 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

```
1 // Variables
2 let name = "Alice";
3 const age = 25;
4
5 // Functions
6 function greet(person) {
7     return "Hello, " + person + "!";
8 }
9
10 // Calling a function
11 console.log(greet(name));
```

notice ; as line terminator

HELLO WORLD! IN JS

```
1 <html>
2 <head></head>
3 <body>
4   <h1 id="greeting">Welcome</h1>
5
6   <button onclick="changeGreeting()">Click Me</button>
7
8   <script>
9     function changeGreeting() {
10       document.getElementById('greeting').textContent =
11         'Hello, JavaScript!';
12     }
13   </script>
14 </body>
15 </html>
```

Result: Button click changes the heading text

EXAMPLE 2: INTERACTIVE COUNTER

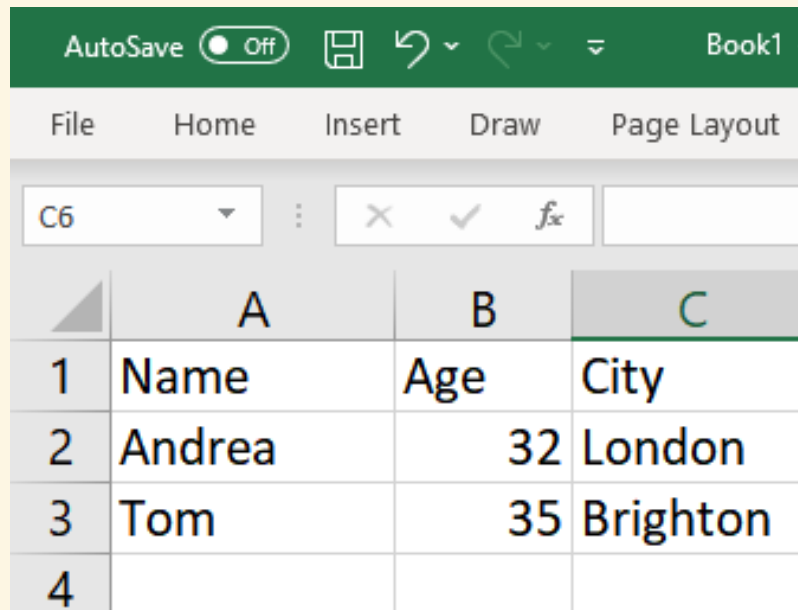
```
1 <html>
2 <head></head>
3 <body>
4   <h1>Count: <span id="count">0</span></h1>
5   <button onclick="increment()">Increment</button>
6   <button onclick="decrement()">Decrement</button>
7   <button onclick="reset()">Reset</button>
8
9   <script>
10     let count = 0;
11
12     function increment() {
13       count++;
14       updateDisplay();
15     }
16
17     function decrement() {
18       count--;
```


A STEP BACK: THE BASICS

JS VARIABLES

A JS variable is a symbolic name for some content, the *value*, that is kept in the browser's (volatile) memory

In spreadsheets, cells are variables

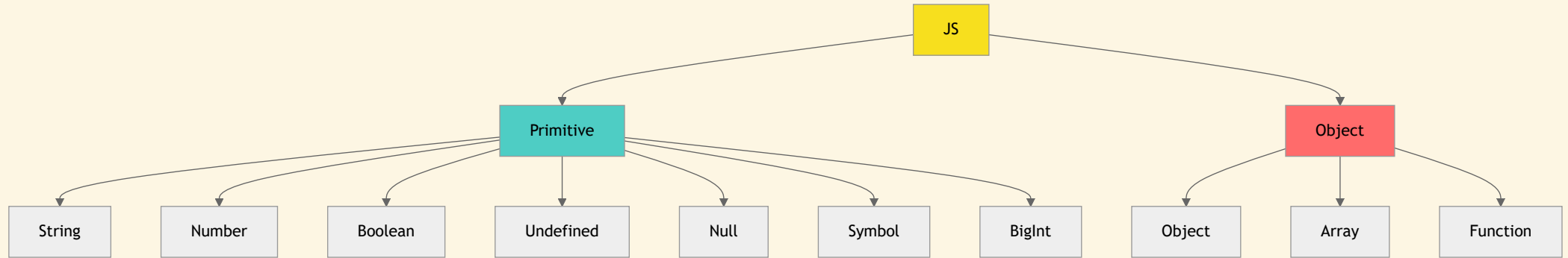


The screenshot shows a Google Sheets interface with a table. The table has three columns: Name, Age, and City. The first row contains the headers. The second row contains 'Andrea', '32', and 'London'. The third row contains 'Tom', '35', and 'Brighton'. The fourth row is empty. The cell B2, which contains the value '32', is highlighted in yellow. The interface includes a top bar with 'AutoSave Off', a menu bar with 'File', 'Home', 'Insert', 'Draw', and 'Page Layout', and a formula bar showing 'C6'.

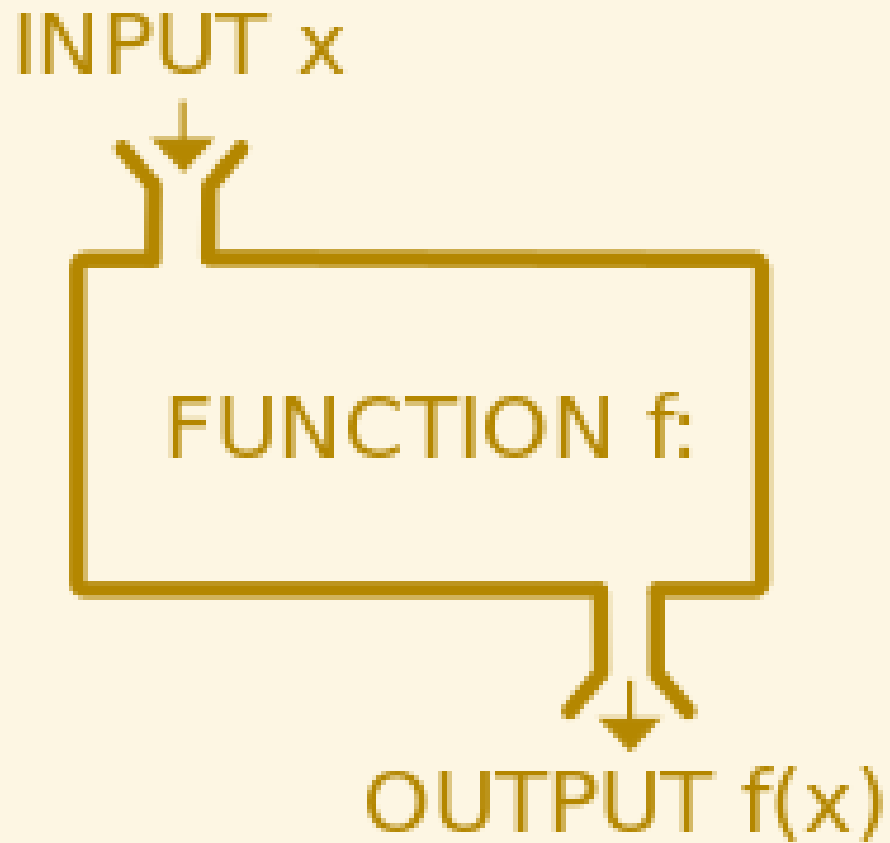
	A	B	C
1	Name	Age	City
2	Andrea	32	London
3	Tom	35	Brighton
4			

B2 = 32 is a variable with name **B2**, content **32** and type **int**

DATA TYPES



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*

- 1 The higher the temperature the quicker pizza cooks.
- 2
- 3 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

```
1  /* Convert Italian exam marks into percentages */
2  function marks2pc(marks) {
3
4      let converted = (marks / 30) * 100
5
6      // Math.round() is a 'foreign' function that rounds up 50.65 --> 51 etc.
7      let pc = Math.round(converted);
8
9      return pc;
10 }
```

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

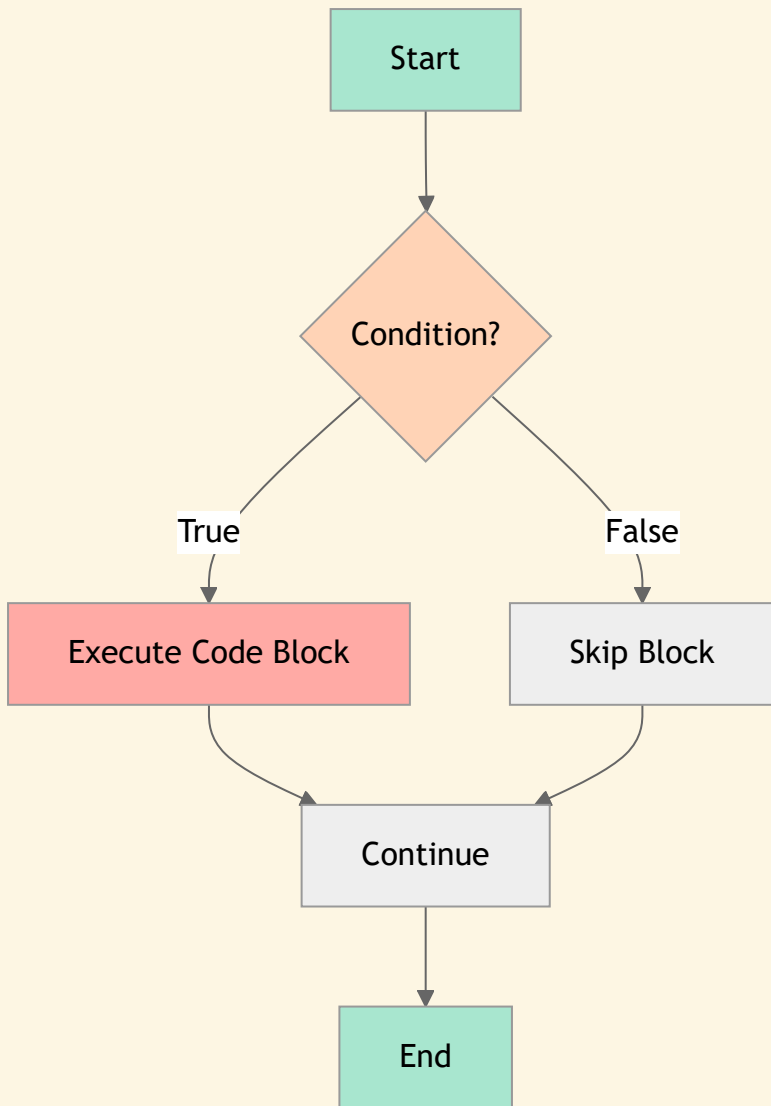
`pc` is the *return value* of the f.

OBSERVATIONS

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

```
1  /* convert and show on the page the UK version of the Webcomm marks */  
2  let my_marks = 27;  
3  
4  let uk_marks = marks2pc(my_marks);  
5  
6  document.getElementById("convertedMarks").innerHTML = uk_marks;
```

CONTROL FLOW



CONDITIONAL STATEMENTS

```
1  let age = 18;
2
3  if (age >= 18) {
4    console.log("You are an adult");
5  } else if (age >= 13) {
6    console.log("You are a teenager");
7  } else {
8    console.log("You are a child");
9  }
10
11 // Ternary operator
12 let status = age >= 18 ? "adult" : "minor";
```

EXAMPLE 3: AGE CHECKER

```
1 <html>
2 <head></head>
3 <body>
4   <h1>Age Verification</h1>
5   <input type="number" id="ageInput" placeholder="Enter your age">
6   <button onclick="checkAge()">Check</button>
7   <p id="result"></p>
8
9   <script>
10     function checkAge() {
11       const age = document.getElementById('ageInput').value;
12       const result = document.getElementById('result');
13
14       if (age === '') {
15         result.textContent = 'Please enter your age';
16       } else if (age < 13) {
17         result.textContent = 'You are a child';
18       } else if (age < 18) {
```

ITERATION

BASIC IDEA

We need to operate over sequences/collection of atomic data

Example: column operations in spreadsheets

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

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

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

then pull the formula over the whole column.

ITERATIONS, A

```
1 // For loop
2 for (let i = 0; i < 5; i++) {
3   console.log(i);
4 }
```

ITERATIONS, B

```
1 // While loop
2 let count = 0;
3 while (count < 5) {
4   console.log(count);
5   count++;
6 }
```

INDEXED DATA

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

```
1 let fruits = ["apple", "banana", "cherry"];  
2  
3 console.log(fruits[0]); // "apple"  
4 console.log(fruits[1]); // "banana"  
5 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

```
1 let word = "Hello";  
2  
3 console.log(word[0]); // "H"  
4 console.log(word[1]); // "e"  
5 console.log(word[4]); // "o"
```

THE LENGHT

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

```
1 let colors = ["red", "green", "blue"];
2
3 let name = "JavaScript";
4
5 console.log(colors.length); // 3
6 console.log(name.length);   // 10
```

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

PRACTICE

```
1 let numbers = [10, 20, 30, 40, 50];  
2  
3 let message = "Code";  
4  
5 // What will these output?  
6 numbers[3]  
7 message[0]  
8 numbers[numbers.length - 1]
```

ITERATIONS, C

```
1 // fruits is an array of strings
2 const fruits = ['apple', 'banana', 'orange'];
3 for (const fruit of fruits) {
4   console.log(fruit);
5 }
```



```
1  /* convert and show on the page the UK version of the marks */  
2  for (const m of my_italian_exam_marks){  
3      let uk_marks = marks2pc(m);  
4      console.log(uk_marks)  
5  }
```

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 🤔

