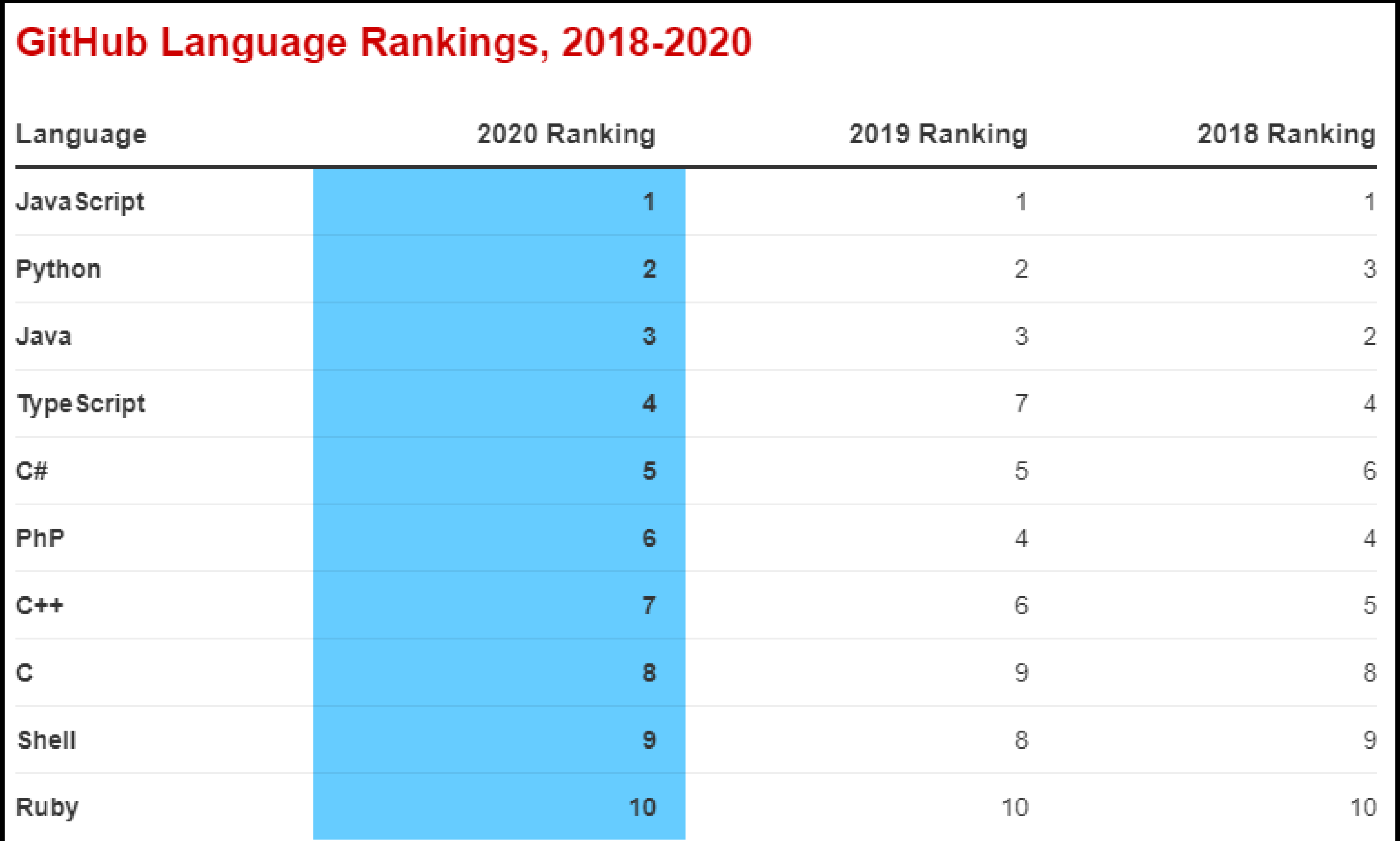
JavaScript

JavaScript is the world's most popular programming language.

Popularity of JS



GitHub Language Rankings, 2018-2020

Language	2020 Ranking	2019 Ranking	2018 Ranking
JavaScript	1	1	1
Python	2	2	3
Java	3	3	2
TypeScript	4	7	4
C#	5	5	6
PhP	6	4	4
C++	7	6	5
C	8	9	8
Shell	9	8	9
Ruby	10	10	10

JavaScript

- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.
- It is fun to learn.
- Adding interactive behavior to web pages
- Creating web and mobile apps
- Building web servers and developing server applications
- Game Development
- Example: ____________________🚗_ (bruno-simon.com)

3D Games - Play 3D Games on CrazyGames

Getting Started

Resources You Need?

1. Browser
2. Bss Kaafi hai

Steps to Code

Open Browser

Press F12

Go to console

Basics of JS

Power of JS

- With the object model, JavaScript gets all the power it needs to create dynamic HTML:
- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

- NaN (Not a number)
- undefined

Data Types

numbers, strings, objects and more:

Variable Declaration

-
  ```
  var x, y, z;   // Statement 1
  x = 5;         // Statement 2
  y = 6;         // Statement 3
  z = x + y;     // Statement 4
  ```

- let a, b, c;

- const a, b, c;

- Var can be redeclared, reassigned

- Let can only be reassigned

- Const nothing can be done

- Variables declared with the var keyword cannot have **Block Scope**.
- Variables declared inside a block **{ }** can be accessed from outside the block.

```
{
  var x = 2;
}
// x CAN be used here
```

- Variables declared with the let keyword can have **Block Scope.**

```
{
  let x = 2;
}
// x can NOT be used here
```

-
  ```
  // You can create a const object:
  const car = {type:"Fiat", model:"500", color:"white"};

  // You can change a property:
  car.color = "red";

  // You can add a property:
  car.owner = "Johnson";
  ```

-
  ```
  const car = {type:"Fiat", model:"500", color:"white"};
  car = {type:"Volvo", model:"EX60", color:"red"};    // ERROR
  ```

-
  ```
  // You can create a constant array:
  const cars = ["Saab", "Volvo", "BMW"];

  // You can change an element:
  cars[0] = "Toyota";

  // You can add an element:
  cars.push("Audi");
  ```

-
  ```
  const cars = ["Saab", "Volvo", "BMW"];
  cars = ["Toyota", "Volvo", "Audi"];    // ERROR
  ```

- var x = 16 + "Volvo";
- 16Volvo
- var x = 16 + 4 + "Volvo";
- 20Volvo
- typeof(null)
- object
- null === undefined
  null == undefined
- False
- true

JavaScript Functions

- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).

-
  ```
  function myFunction(p1, p2) {
    return p1 * p2;   // The function returns the product of p1 and p2
  }
  ```

-
  ```
  function name(parameter1, parameter2, parameter3) {
    // code to be executed
  }
  ```
- **Function Invocation:**
- The code inside the function will execute when "something" **invokes** (calls) the function:
- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Events

- *<element event=**'some JavaScript'**>*
- <button onclick="displayDate()">The time is?</button>
- onchange, onclick, onmouseover, onmouseout, onkeydown, onload.

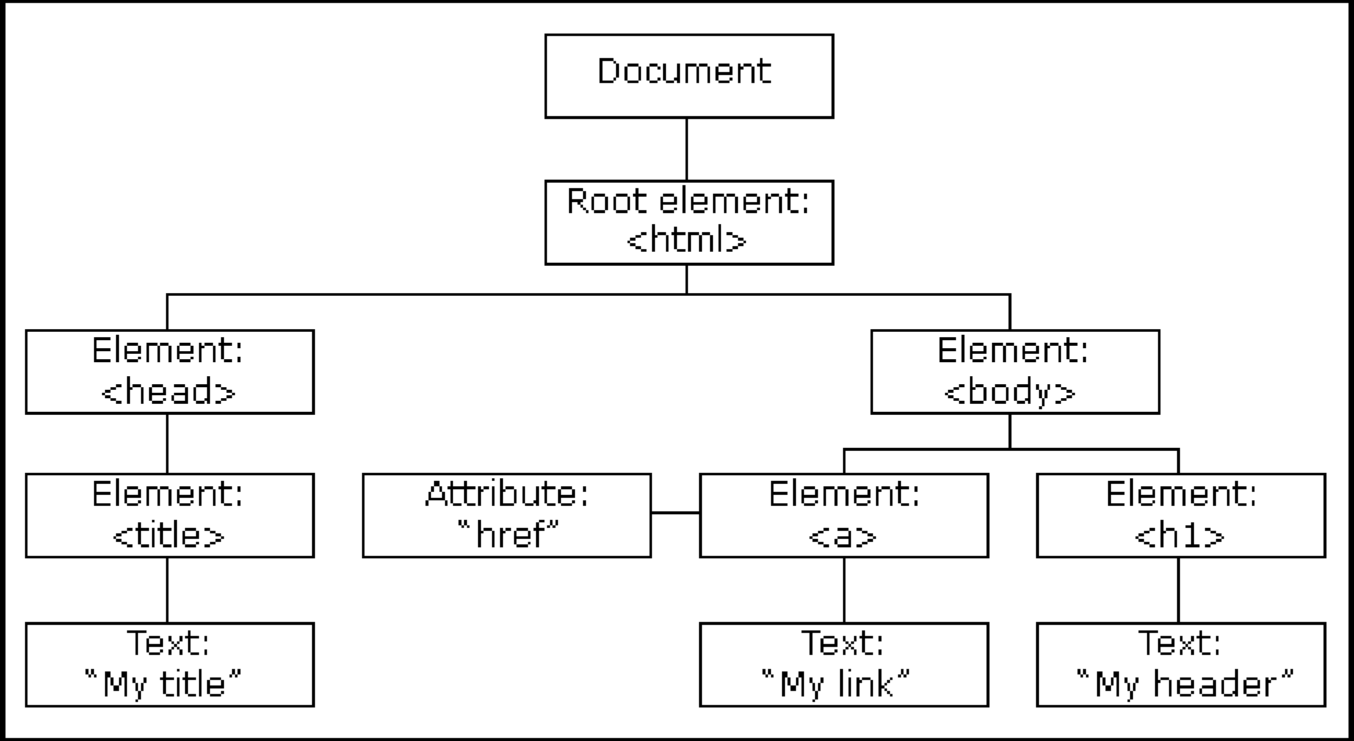
What can JavaScript Do?

- Event handlers can be used to handle and verify user input, user actions, and browser actions:
- Things that should be done every time a page loads
- Things that should be done when the page is closed
- Action that should be performed when a user clicks a button
- Content that should be verified when a user inputs data
- Many different methods can be used to let JavaScript work with events:
- HTML event attributes can execute JavaScript code directly
- HTML event attributes can call JavaScript functions
- You can assign your own event handler functions to HTML elements
- You can prevent events from being sent or being handled…

JS in HTML

JavaScript Can Change HTML Content

The HTML DOM Tree of Objects



Document

Root element: <html>

Element: <head>

Element: <title>

Text: "My title"

Element: <body>

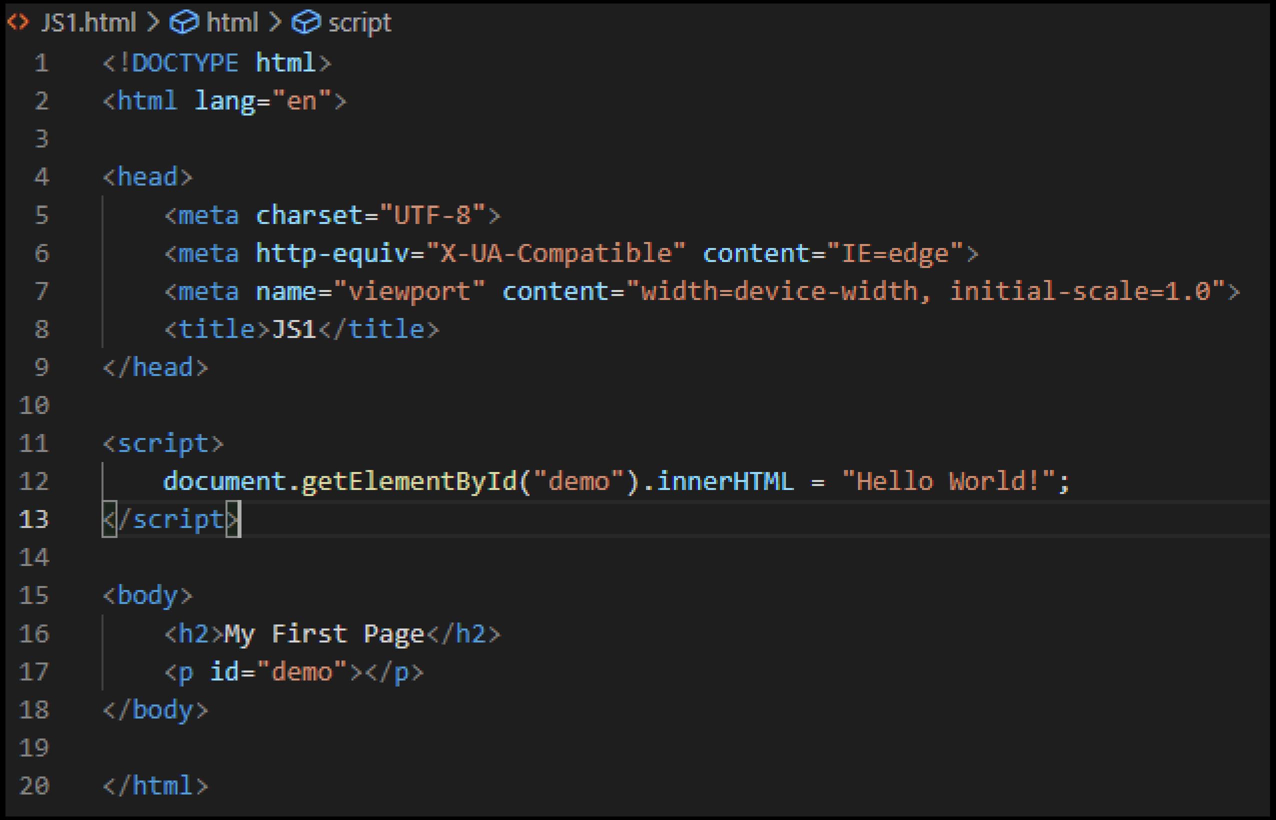
Attribute: "href"

Element: <a>

Text: "My link"

Element: <h1>

Text: "My header"



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JS1</title>
</head>

<script>
    document.getElementById("demo").innerHTML = "Hello World!";
</script>

<body>
    <h2>My First Page</h2>
    <p id="demo"></p>
</body>

</html>
```

First JS Program

document.getElementById("demo").innerHTML = "Hello World!";

The HTML DOM Document Object

- document.getElementById(Id)
- document.getElementsByTagName(*name*)
- document.getElementsByClassName(*name*)

- *element*.innerHTML = *new html content*
- *element.attribute* = *new value*
- *element*.style.*property* = *new style*

-
  ```
  var para = document.createElement("p");
  var node = document.createTextNode("This is new.");
  para.appendChild(node);

  var element = document.getElementById("div1");
  element.appendChild(para);
  ```