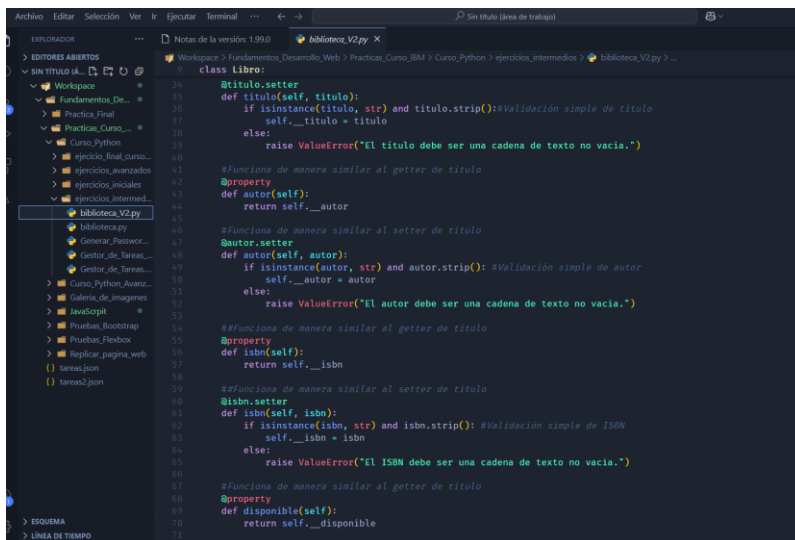


Lenguajes de Servidores Web



```
class Libro:
    @titulo.setter
    def titulo(self, titulo):
        if isinstance(titulo, str) and titulo.strip(): #Validación simple de titulo
            self.__titulo = titulo
        else:
            raise ValueError("El titulo debe ser una cadena de texto no vacia.")
    #Funciona de manera similar al getter de titulo
    @property
    def autor(self):
        return self.__autor
    #Funciona de manera similar al setter de titulo
    @autor.setter
    def autor(self, autor):
        if isinstance(autor, str) and autor.strip(): #Validación simple de autor
            self.__autor = autor
        else:
            raise ValueError("El autor debe ser una cadena de texto no vacia.")
    #Funciona de manera similar al getter de titulo
    @property
    def isbn(self):
        return self.__isbn
    #Funciona de manera similar al setter de titulo
    @isbn.setter
    def isbn(self, isbn):
        if isinstance(isbn, str) and isbn.strip(): #Validación simple de ISBN
            self.__isbn = isbn
        else:
            raise ValueError("El ISBN debe ser una cadena de texto no vacia.")
    #Funciona de manera similar al getter de titulo
    @property
    def disponible(self):
        return self.__disponible
```

Francisco Javier Otero Herrero

Grupo ATU

08-04-2025

Lenguajes de Servidores Web

Lenguajes de Servidores Web

Contenido

<i>Actividad 1:</i>	3
<i>Lenguajes del lado del Cliente →</i>	3
<i>HTML (HiperText Markup Language):</i>	3
<i>CSS (Cascading Style Sheets):</i>	4
<i>JavaScript:</i>	5
<i>TypeScript:</i>	5
<i>Lenguajes del lado del Servidor →</i>	6
<i>PHP:</i>	6
<i>Python:</i>	6
<i>Java:</i>	7
<i>Node.js:</i>	7
<i>Ruby:</i>	7

Lenguajes de Servidores Web

❖ Actividad 1:

En esta actividad vamos a definir los principales lenguajes del lado del cliente y del lado del servidor.

✓ Lenguajes del lado del Cliente →

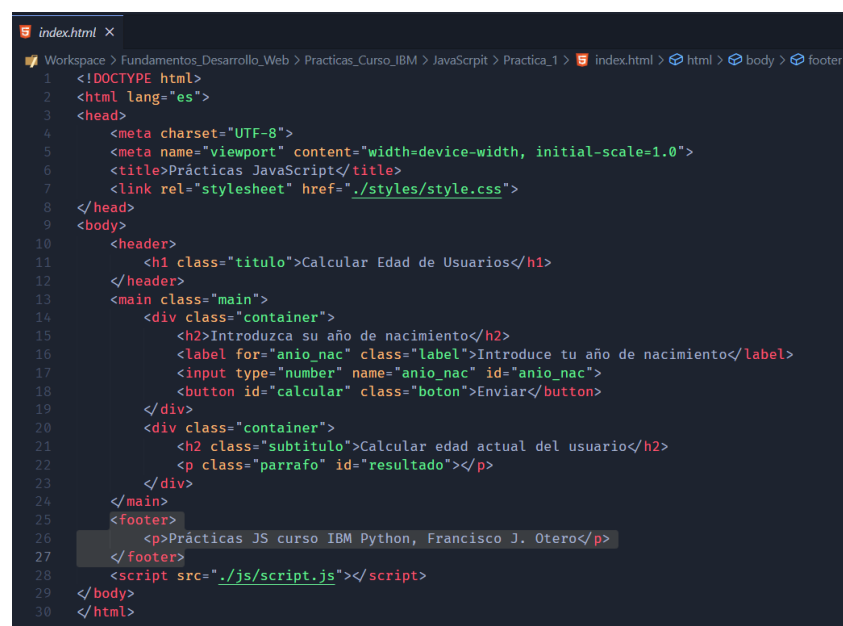
Estos lenguajes se ejecutan en el navegador del usuario. Son fundamentales para mostrar correctamente una página web, darle diseño visual y permitir interacción con el usuario.

- *HTML (HyperText Markup Language):*

Lenguaje de marcado que estructura el contenido de una página web. **HTML** no es un lenguaje de programación, sino un lenguaje de marcado que permite estructurar el contenido de una página web. Define elementos como encabezados, párrafos, imágenes, enlaces, formularios, tablas, etc.

- Es el esqueleto de cualquier página web.
- Funciona junto a CSS y JavaScript para crear páginas completas.
- Es interpretado por todos los navegadores web.

Es un lenguaje el cual he usado y uso mucho, se podría decir que prácticamente a diario lo utilizo.



```
index.html x
Workspace > Fundamentos_Desarrollo_Web > Practicas_Curso_IBM > JavaScript > Practica_1 > index.html > html > body > footer
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Prácticas JavaScript</title>
7   <link rel="stylesheet" href="./styles/style.css">
8 </head>
9 <body>
10   <header>
11     <h1 class="titulo">Calcular Edad de Usuarios</h1>
12   </header>
13   <main class="main">
14     <div class="container">
15       <h2>Introduzca su año de nacimiento</h2>
16       <label for="anio_nac" class="label">Introduce tu año de nacimiento</label>
17       <input type="number" name="anio_nac" id="anio_nac">
18       <button id="calcular" class="boton">Enviar</button>
19     </div>
20     <div class="container">
21       <h2 class="subtitulo">Calcular edad actual del usuario</h2>
22       <p class="parrafo" id="resultado"></p>
23     </div>
24   </main>
25   <footer>
26     <p>Prácticas JS curso IBM Python, Francisco J. Otero</p>
27   </footer>
28   <script src="./js/script.js"></script>
29 </body>
30 </html>
```

Lenguajes de Servidores Web

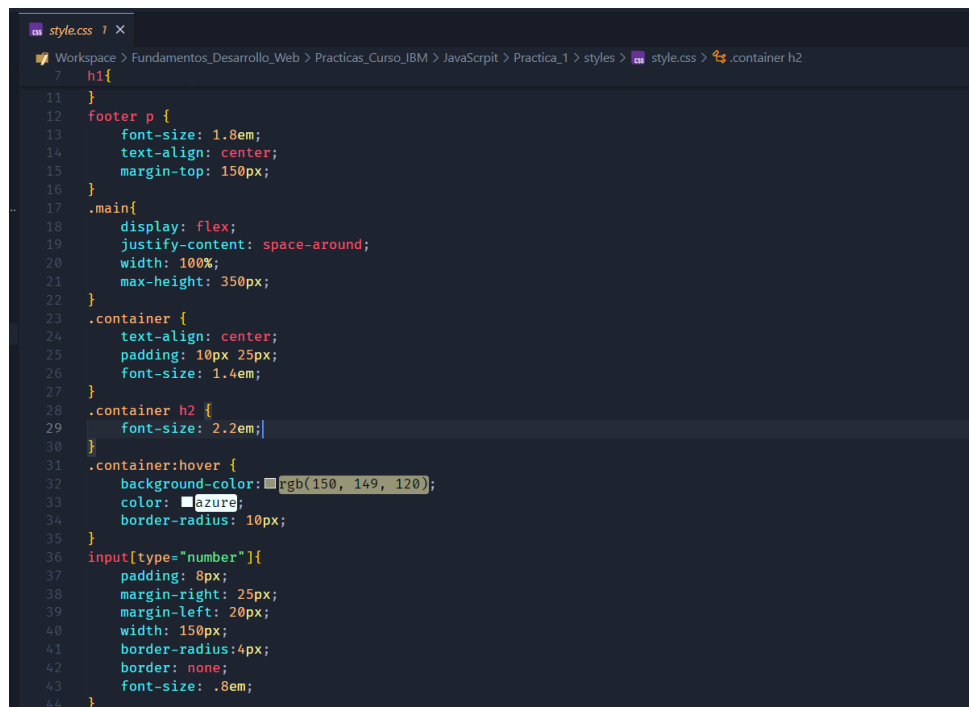
- *CSS (Cascading Style Sheets):*

CSS es el lenguaje que se usa para dar estilo y diseño visual a las páginas web construidas en **HTML**. Permite definir colores, tamaños, fuentes, márgenes, posiciones de elementos, efectos visuales, animaciones, etc.

Puede escribirse de forma interna (en la misma página), externa (en archivos .css) o en línea (directamente sobre elementos HTML).

Es clave para que la web sea atractiva, responsiva y usable en diferentes dispositivos.

Es un lenguaje el cual también he usado y uso mucho.



```
style.css 1 x
Workspace > Fundamentos_Desarrollo_Web > Practicas_Curso_IBM > JavaScript > Practica_1 > styles > style.css > .container h2
7   h1{
11  }
12  footer p {
13      font-size: 1.8em;
14      text-align: center;
15      margin-top: 150px;
16  }
17  .main{
18      display: flex;
19      justify-content: space-around;
20      width: 100%;
21      max-height: 350px;
22  }
23  .container {
24      text-align: center;
25      padding: 10px 25px;
26      font-size: 1.4em;
27  }
28  .container h2 {
29      font-size: 2.2em;
30  }
31  .container: hover {
32      background-color: rgb(150, 149, 120);
33      color: azure;
34      border-radius: 10px;
35  }
36  input[type="number"]{
37      padding: 8px;
38      margin-right: 25px;
39      margin-left: 20px;
40      width: 150px;
41      border-radius: 4px;
42      border: none;
43      font-size: .8em;
44  }
```

Lenguajes de Servidores Web

- **JavaScript:**

Lenguaje de programación que permite dotar de dinamismo e interactividad a las páginas web (validación de formularios, menús, sliders, etc.).

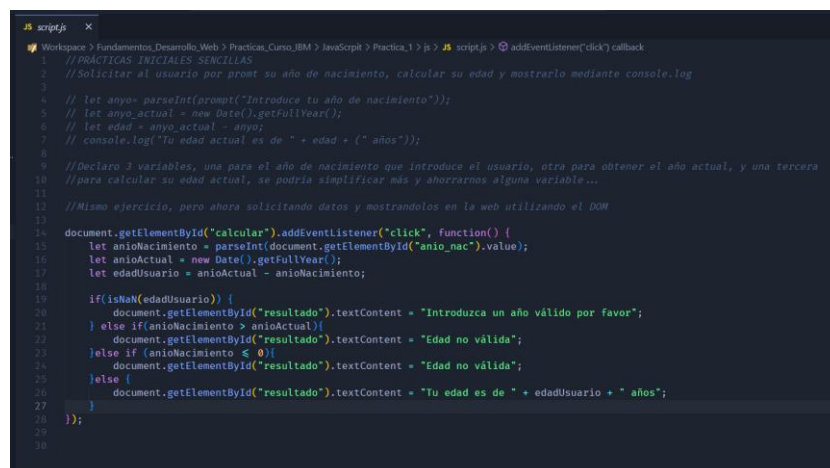
JavaScript es un lenguaje de programación que permite hacer páginas interactivas y dinámicas.

Se ejecuta directamente en el navegador del usuario.

Se utiliza para cosas como validación de formularios, menús dinámicos, sliders, carritos de compra, efectos interactivos, etc.

Puede trabajar junto a HTML y CSS, y hoy en día también se usa en el servidor mediante **Node.js**.

Es un lenguaje el cual también he usado y uso mucho.



```

1 //PRÁCTICAS INICIALES -SENCILLAS
2 //Solicitar al usuario por prompt su año de nacimiento, calcular su edad y mostrarlo mediante console.log
3
4 // let anyo= parseInt(prompt("Introduce tu año de nacimiento"));
5 // let anyo_actual = new Date().getFullYear();
6 // let edad = anyo_actual - anyo;
7 // console.log("Tu edad actual es de " + edad + (" años"));
8
9 //Declaro 3 variables, una para el año de nacimiento que introduce el usuario, otra para obtener el año actual, y una tercera
10 //para calcular su edad actual, se podría simplificar más y ahorrarnos alguna variable...
11
12 //Mismo ejercicio, pero ahora solicitando datos y mostrándolos en la web utilizando el DOM
13
14 document.getElementById("calcular").addEventListener("click", function() {
15     let anioNacimiento = parseInt(document.getElementById("anio_nac").value);
16     let anioActual = new Date().getFullYear();
17     let edadUsuario = anioActual - anioNacimiento;
18
19     if(isNaN(edadUsuario)) {
20         document.getElementById("resultado").textContent = "Introduzca un año válido por favor";
21     } else if(anioNacimiento > anioActual){
22         document.getElementById("resultado").textContent = "Edad no válida";
23     } else if (anioNacimiento <= 0){
24         document.getElementById("resultado").textContent = "Edad no válida";
25     } else {
26         document.getElementById("resultado").textContent = "Tu edad es de " + edadUsuario + " años";
27     }
28 });
29
30

```

- **TypeScript:**

Superset de JavaScript que añade tipado estático y características de programación orientada a objetos desarrollado por Microsoft. Se compila a JavaScript para ser usado en el navegador. Muy utilizado en grandes aplicaciones web modernas, como las hechas con Angular. Lenguaje que desconozco.

Lenguajes de Servidores Web

✓ Lenguajes del lado del Servidor →

Estos lenguajes no son visibles para el usuario. Se ejecutan en el servidor y se encargan de la lógica interna, gestión de datos y generación dinámica del contenido antes de enviarlo al navegador.

- *PHP (Hypertext Preprocessor):*

PHP es uno de los lenguajes de servidor más populares y usados. Está especialmente diseñado para el desarrollo web. Se integra muy bien con bases de datos como **MySQL**. Es fácil de aprender y muy extendido en alojamientos web. Es el lenguaje detrás de gestores de contenido como **WordPress, Joomla, Moodle**, etc.

Es un lenguaje que he utilizado en el pasado, el cual hace bastante que no practico, por lo tanto, se puede decir que estoy algo desactualizado.

- *Python:*

Python es un lenguaje versátil, claro y fácil de leer, cada vez más usado en desarrollo web. Tiene frameworks potentes como **Django o Flask** que permiten crear aplicaciones web completas y seguras. Ideal para combinar con análisis de datos, inteligencia artificial y automatizaciones. Muy apreciado en proyectos donde se requiere rapidez y limpieza en el código.

Por culpa de Python es que estoy algo desactualizado en PHP, ya que llevo unos meses haciendo formación en este lenguaje. Ya aprobé el certificado de iniciación, ahora estoy empezando el de nivel avanzado, todo ello con **IBM y SkillBuild**.

```
ejercicio6.py X
Workspace > Fundamentos, Desarrollo, Web > Practicas, Curso IBM > Curso Python > ejercicios_iniciales > ejercicio6.py > ...
1 #EJERCICIO 6: Adivina el número.
2 #Escribir un programa que genere un número aleatorio entre 1 y 10 y le pida al usuario que lo adivine.
3 #El programa debe indicar si el número es mayor o menor que el ingresado.
4
5 #Importar metodo Random
6 import random
7
8 #Generar el número aleatorio entre 1 y 10
9 numero_secreto = random.randint(1,10)
10
11 #Ahadinos una variable para indicarle al usuario el nº de intentos que lleva
12 intento = 0
13
14 #Bucle infinito
15 while True:
16     #A cada iteracion intento sumará 1
17     intento +=1
18     #Solicitar al usuario un número
19     numero = int(input(f"Intento nº {intento}: Introduzca un número entre 1 y 10 para adivinar el nº secreto: "))
20     #Condicional para ver si el nº del usuario es mayor o menor al número aleatorio generado
21     if numero == numero_secreto:
22         #Si adivina el número
23         print(f"Correcto! Has adivinado el número que era: {numero_secreto}")
24         print("Fin Del JUEGO")
25         #Romper el bucle infinito while
26         break
27     #Si el número del usuario es menor le damos una pista
28     elif numero < numero_secreto:
29         print("Te has quedado corto... el número es mayor, prueba otra vez")
30     #Si el número del usuario es mayor le damos una pista
31     else:
32         print("Te has pasado... el número es menor, prueba otra vez")
```

Lenguajes de Servidores Web

- *Java:*

Java es un lenguaje muy robusto, orientado a objetos y multiplataforma. Usado en entornos empresariales de gran escala. Tiene frameworks como **Spring** que permiten crear aplicaciones web potentes, seguras y escalables. Aunque requiere más configuración, es ideal para aplicaciones críticas y de alto rendimiento.

Lenguaje el cual practique durante el primer año que curse el módulo de grado superior de **Desarrollo de Aplicaciones Web**. No fue un lenguaje al cual guarde especial cariño, ya que me costó mucho aprenderlo y fue el culpable de que estuviera a punto de dejar el curso, por suerte no lo hice.

- *Node.js:*

Node.js permite utilizar **JavaScript** en el servidor, lo que permite unificar el lenguaje en cliente y servidor. Basado en un modelo asíncrono muy eficiente para aplicaciones en tiempo real. Ideal para chats, servicios de streaming o APIs REST. Usado junto a frameworks como **Express**.

- *Ruby:*

Ruby es un lenguaje dinámico y elegante, y su framework **Ruby on Rails** es muy popular. Facilita el desarrollo rápido de aplicaciones web. Famoso por su filosofía de “convención sobre configuración”. Menos usado que otros hoy en día, pero aún vigente para proyectos web bien estructurados.

Para finalizar esta práctica, voy a dejar un enlace a **GitHub**, en la cual tengo algunas prácticas y pequeños proyectos realizados con algunos de los lenguajes mencionados en esta actividad.

<https://github.com/FranciscoJOtero/Curso-IBM-Iniciacion-Python>