

---

# IIC2026

## Ayudantía 2 - HTML, CSS y JS

— Camila Basulto y Javiera Azócar —

---

# HTML & CSS

---

# HTML - Estructura completa

```
<!DOCTYPE html>
<html>
  <!-- Comentario -->
  <head>
    <meta charset="utf-8">
    <title>Título de documento completo</title>
  </head>
  <body>
    <h1> Un título </h1>
    <p> Un párrafo </p>
    <h2> Un subtítulo </h2>
    <p> Otro párrafo </p>
    <ul>
      <li>Primer elemento de lista.</li>
      <li>Segundo elemento de lista.</li>
      <li>Tercer elemento de lista.</li>
    </ul>
  </body>
</html>
```

▼ Título de documento completo × +

← → ↻ ⓘ Archivo

## Un título

Un párrafo

## Un subtítulo

Otro párrafo

- Primer elemento de lista.
- Segundo elemento de lista.
- Tercer elemento de lista.

# CSS

## Cascading Style Sheets

External CSS

```
<link href="estilo.css" rel="stylesheet">
```

```
1
2  ✓ body{background-color: ■rosybrown;
3      font-family: monospace;
4      font-size: 20px;
5  }
6
7  ✓ #identificador{
8      color: ■red;
9  }
10
11 ✓ .clase1{
12     font-size: 30px;
13 }
14
```

# Intro a JavaScript

---

# Intro a JavaScript

- Lenguaje de programación de web
- Es de alto nivel
- Se puede ejecutar en un navegador web y afecta el contenido de una página
- Se utilizará como lenguaje para construir visualizaciones a partir de datos



## En el HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo con JS</title>
  </head>

  <body id="raiz">
    <h1 id="principal">¡Presioname!</h1>
    <script src='programa.js' charset='utf-8'></script>
  </body>
</html>
```

**Programa a ejecutar,** aquí se incluye el código de JS

# PYTHON VS JAVASCRIPT

---



# Declaraciones de variables y funciones en Javascript

**const** : Nos permite declarar variables inmutables, osea no se puede cambiar.

**let** : Nos permiten crear variables mutables que si se puede cambiar.

## Javascript

```
JS prueba.js > ...
1  let variable = 1;
2  variable = 4;
3
4  // tambien es posible utilizar var, pero no es recomendable
5  var variable2 = 1;
6  variable2 = 4;
7
8  const variable3 = 1;
9  variable3 = 4; // error
10
11
```

## Python

```
prueba.py > ...
1  variable = 5
2
3  variable = 0
4
5  variable = "5"
6
```

# IF, ELSE IF & ELSE

## Python

```
testeo.py > ...  
1   x = 10  
2  
3   if x < 5:  
4       print("x es menor que 5")  
5   elif x == 5:  
6       print("x es igual a 5")  
7   else:  
8       print("x es mayor que 5")
```

### TERMINAL

x es mayor que 5

## Javascript

```
JS testeo.js > ...  
1   let x = 10;  
2  
3   if (x < 5) {  
4       console.log("x es menor que 5");  
5   } else if (x === 5) {  
6       console.log("x es igual a 5");  
7   } else {  
8       console.log("x es mayor que 5");  
9   }
```



# Ciclos

## Python

### FOR

```
12  for i in range(5):  
13      print(i)
```

## Javascript

```
11  for (let i = 0; i < 5; i++) {  
12      console.log(i);  
13  }
```

### WHILE

```
15  x = 0  
16  while x < 5:  
17      print(x)  
18      x += 1
```

```
15  let a = 0;  
16  while (a < 5) {  
17      console.log(a);  
18      a++;  
19  }
```

# LISTAS

## Python

```
testeo.py > ...  
1 lista = []  
2 lista.append(1)  
3 lista.append(2)  
4 lista.append(3)  
5 print(lista)  
6 print(lista[2])  
7 print(lista[0:2])
```

```
[1, 2, 3]  
3  
[1, 2]
```

## Javascript

```
JS testeo.js ×  
JS testeo.js > ...  
1 let lista = [];  
2 lista.push(1);  
3 lista.push(2);  
4 lista.push(3);  
5 console.log(lista);  
6 console.log(lista[2]);  
7 console.log(lista.slice(0, 2));
```

```
▶ (3) [1, 2, 3] testeo.js:5  
3 testeo.js:6  
▶ (2) [1, 2] testeo.js:7
```

# Funciones

## Javascript

```
JS prueba.js x
JS prueba.js > ...
1  function sumar9(numero) {
2      return numero + 9;
3  }
4  sumar9(6); // 15
5  sumar9(sumar9(6)); // 24
6
7  const sumar9 = function(numero) {
8      return numero + 9;
9  }
10  sumar9(6); // 15
11  sumar9(sumar9(6)); // 24
12
13  //Arrow functions
14  const sumar9 = (numero) => numero + 9;
15  sumar9(6); // 15
16
17  const algoMasComplicado = (arg1, arg2, arg3) => {
18      // descripción de la función
19      return valorResultado;
20  };
21  algoMasComplicado(1, 2, 3);
22
23  //LEER DATOS
24  d3.csv(LINK, parseoCaracteristicas).then(datos => {
25      datosFinales = datos;
26      joinDeDatos(datos);
27  })
```

## Python

```
+ prueba.py x
+ prueba.py > ...
1  def sumar9(numero):
2      return numero + 9
3
4  conjunto = [20, 5, 100, 38, 0, -4]
5  conjunto.sort(key= lambda x: -x)
6
```

## Python

```
a = 10
b = 5
max_value = max(a, b)
print(max_value)
```

```
lista = [1, 5, 3, 9, 2]
max_value = max(lista)
print(max_value)
```

## Javascript

```
let a = 10;
let b = 5;
let max_value = Math.max(a, b);
console.log(max_value);
```

```
let lista = [1, 5, 3, 9, 2];
let max_value = Math.max(...lista);
console.log(max_value);
```

# CONSOLA Y ELEMENTOS DE UNA PÁGINA

## Inspeccionar (ver los elementos de una página):

- Ctrl+Shift+I para Windows
- Cmd+Shift+I para Mac

## Ver consola (lo que imprime console.log)

- Ctrl+Shift+C para Windows
- Cmd+Shift+C para Mac

# Conexión a servidor...

(Hostear un servidor local)

---



## ... en un puerto de la computadora

- En el mismo directorio donde se encuentra el archivo HTML:

```
Command Prompt - python -m http.server 5000

D:\Repos\infovis>python -m http.server 5000
Serving HTTP on :: port 5000 (http://[::]:5000/) ...
Keyboard interrupt received, exiting.
```

abrir CMD: windows+R

Keyboard interrupt: ctrl+C

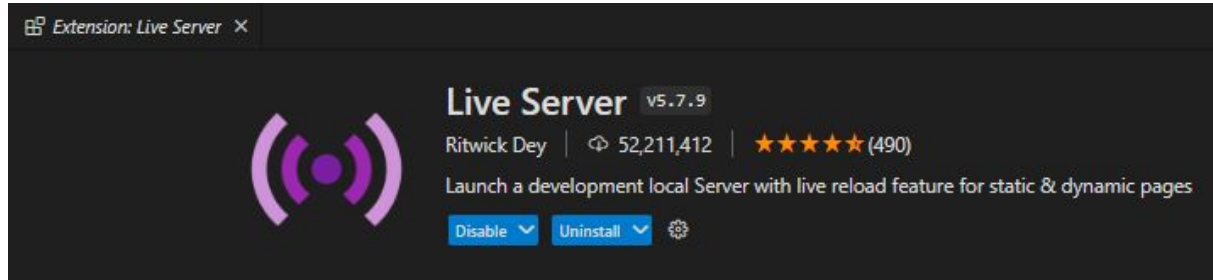
- En su buscador de preferencia:



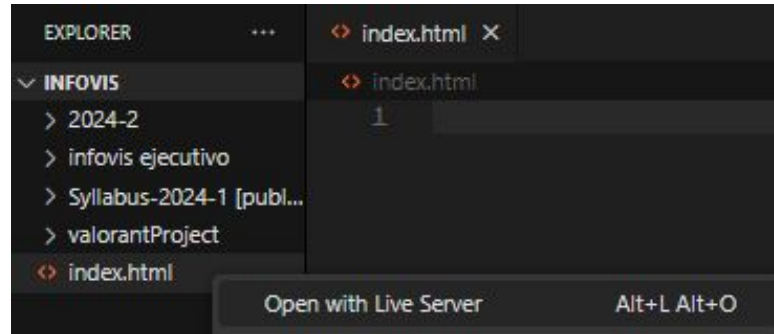
(puerto 5000 por convención)

## ... con Live Server de VS Code

- Instalar la extensión Live Server en VS Code:



- Abrir archivo con Open Live Server:



se abrirá automáticamente  
una ventana en el  
navegador predeterminado

---

---

# IIC2026

## Ayudantía 2 - HTML, CSS y JS

---

---