



Módulo 2: Lenguaje de Guión

Curso de Integración de Componentes Software Web

Características del Lenguaje de Guión

- Los lenguajes de guión son interpretados y orientados a eventos.
 - Interpretados: Ejecutados línea por línea.
 - Orientados a eventos: Responden a acciones del usuario.
- Importancia: Hacen las páginas web interactivas.

Ejemplos:

JavaScript ejecuta código cuando se hace clic en un botón:

```
<button onclick='alert("Hola")'>Haz clic</button>
```

Relación entre Scripting y HTML

- HTML estructura la web, el scripting añade interactividad.
- Scripts pueden cambiar el contenido HTML, CSS y responder a acciones.
- Ventajas: Dinamismo sin recargar la página.

Ejemplos:

HTML + JS:

```
<p id='demo'></p>
```

```
<script>document.getElementById('demo').innerHTML='Hola!';</script>
```

Sintaxis del Lenguaje de Guión

- Reglas básicas:
 - Variables: var, let, const.
 - Operadores: +, -, *, /, &&, ||.
 - Estructuras: if/else, for, while.
- Ejemplo: Calcular el área de un rectángulo.

Ejemplos:

```
let base=5, altura=3;  
let area=base*altura;  
alert('Área: '+area);
```

Sintaxis del Lenguaje de Guión

- **Valores**: 'Chris' es un valor.
- **Expresiones**: Las expresiones son unidades de código que se evalúan y se convierten en valores. Por ejemplo, la siguiente es una expresión: `normalName + 'Script'`. Cuando `normalName` tiene el valor 'Chris', esta expresión se evalúa como `ChrisScript`.
- **Operadores**: Los operadores hacen algo con los valores. El operador `+` (conocido como operador de *concatenación*) une el valor de `normalName` y el valor literal 'Script'.
- **Palabras clave**: Las palabras clave son partes de JavaScript que tienen un significado especial y hacen que JavaScript haga algo. La palabra *let* es una palabra clave que JavaScript que almacene el valor a la derecha del operador `=` (asignación) utilizando el nombre del lado izquierdo de `=`.

Ejemplo:

```
let normalName = 'Chris';  
  
let javascriptName = normalName + 'Script';  
  
console.log(Su nombre JavaScript es ' + javascriptName);
```

Sintaxis del Lenguaje de Guión

- 1. JavaScript distingue entre mayúsculas y minúsculas**
- 2. JavaScript ignora los espacios en blanco**
- 3. Los programadores de JavaScript utilizan camelCase y guiones bajos**

Integración de Scripts en Web

- Maneras de incluir scripts:
 - Interno: Dentro de `<script>` en HTML.
 - Externo: En un archivo `.js` separado.
 - Ubicación: Head o Body.

Ejemplos:

```
<script src='app.js'></script>
```

 para archivos externos.

Ejecutar JavaScript a partir de atributos de evento HTML

- Los atributos de evento de HTML se introdujeron en el lenguaje al principio de la vida de JavaScript. Su propósito es detectar eventos que ocurren a elementos HTML y ejecutar código JavaScript cuando esos eventos ocurren.
- Este es el aspecto de JavaScript adjunto a un atributo de evento HTML:

Ejemplos:

```
<button onclick="alert('Bienvenido a JavaScript')"> Haga clic aquí para ver un mensaje importante </button>.
```


Ejecutar HTML dentro de elementos de script

- El segundo método para incluir JavaScript en HTML se denomina JavaScript *en línea* .
- Para utilizar JavaScript en línea, coloque su JavaScript entre una etiqueta `<script>` de apertura y una etiqueta `</script>` de cierre en su documento HTML.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>
    <title>JavaScript en línea</title>
  </head>
  <body>
    <script>
      let personName = prompt('¿Cual es tu nombre?');
      alert('Hola, ' + personName);
    </script>
  </body>
</html>
```

Incluir archivos JavaScript en su HTML

- El tercer método para ejecutar JavaScript en una página web consiste en importar un archivo que contenga código JavaScript en el documento HTML. Para ello se utiliza el atributo `src` del elemento `script`.
- Esta es, con diferencia, la forma más habitual de utilizar JavaScript en las páginas web. Para utilizar este método, cree un archivo que contenga su JavaScript y guárdelo con la extensión `.js`

Ejecutar HTML dentro de elementos de script

```
const ball = document.getElementById('ball');
document.addEventListener('keydown', handleKeyPress);
let position = 0;
function handleKeyPress(e) {
  if (e.code === 'ArrowLeft') {
    position = position - 10;
  }
  if (e.code === 'ArrowRight') {
    position = position + 10;
  }
  if (position < 0) {
    position = 0;
  }
  refresh();
}
function refresh() {
  ball.style.left = position + 'px';
}
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>
    <title>Document</title>
    <style>
      #ball {
        background-color: red;
        border-radius: 50%;
        width: 20px;
        height: 20px;
        position: relative;
      }
    </style>
  </head>
  <body>
    <div id="ball"></div>
    <script src="gamelet.js"></script>
  </body>
</html>
```

Prácticas del Módulo 2



Actividades sugeridas:



Haz que la bola se mueva una distancia mayor con cada pulsación de tecla.



Cambia las teclas que hacen que la bola se mueva.

Puedes encontrar una referencia completa de todos los códigos de teclas [aquí](#)



Haz que al pulsar las flechas arriba y abajo la bola se mueva arriba y abajo.

Sugerencia: Debe cambiar el valor de `style.top` para ajustar la posición vertical.

Prácticas del Módulo 2



Actividades sugeridas:



1. Crear un script que muestre la hora actual.



2. Agregar un botón HTML que cambie el color de fondo.



3. Escribir un script que abra una alerta al cargar la página.