

DOM JavaScript

Document Object Model

Es el "puente" entre el código JavaScript y el HTML, transformando las etiquetas en objetos que se pueden controlar para crear interactividad, como cambiar texto al hacer clic en un botón o actualizar contenido sin recargar la página.

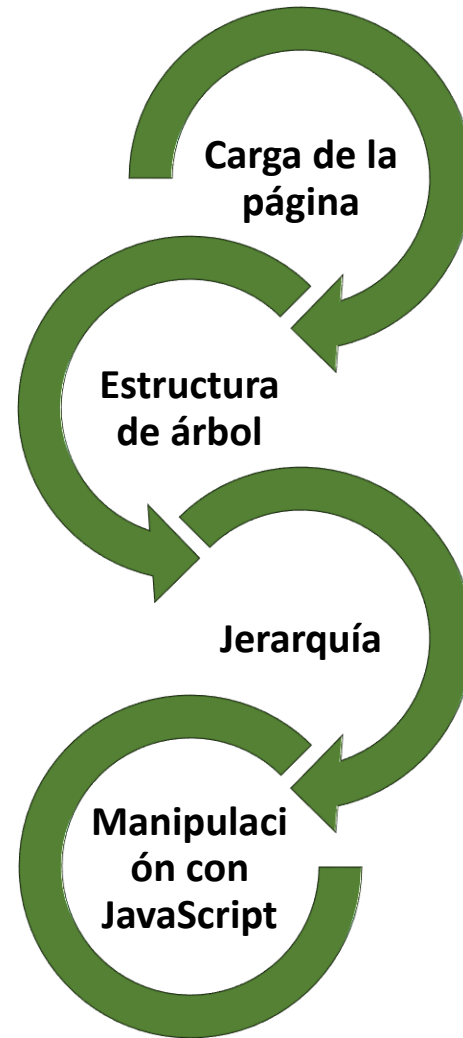
En resumen:

Interfaz que representa una página HTML como un árbol de objetos

DOM JavaScript

Cada etiqueta HTML (como <html>, <head>, <body>, <p>, <div>), atributo y texto se convierte en un "nodo" en este árbol.

- Seleccionar: Encontrar elementos (por ID, clase, etiqueta).
- Crear/Eliminar: Añadir o quitar elementos.
- Modificar: Cambiar texto, atributos (como src, href), clases CSS o estilos.
- Eventos: Ejecutar código en respuesta a acciones del usuario (clics, movimientos del ratón)



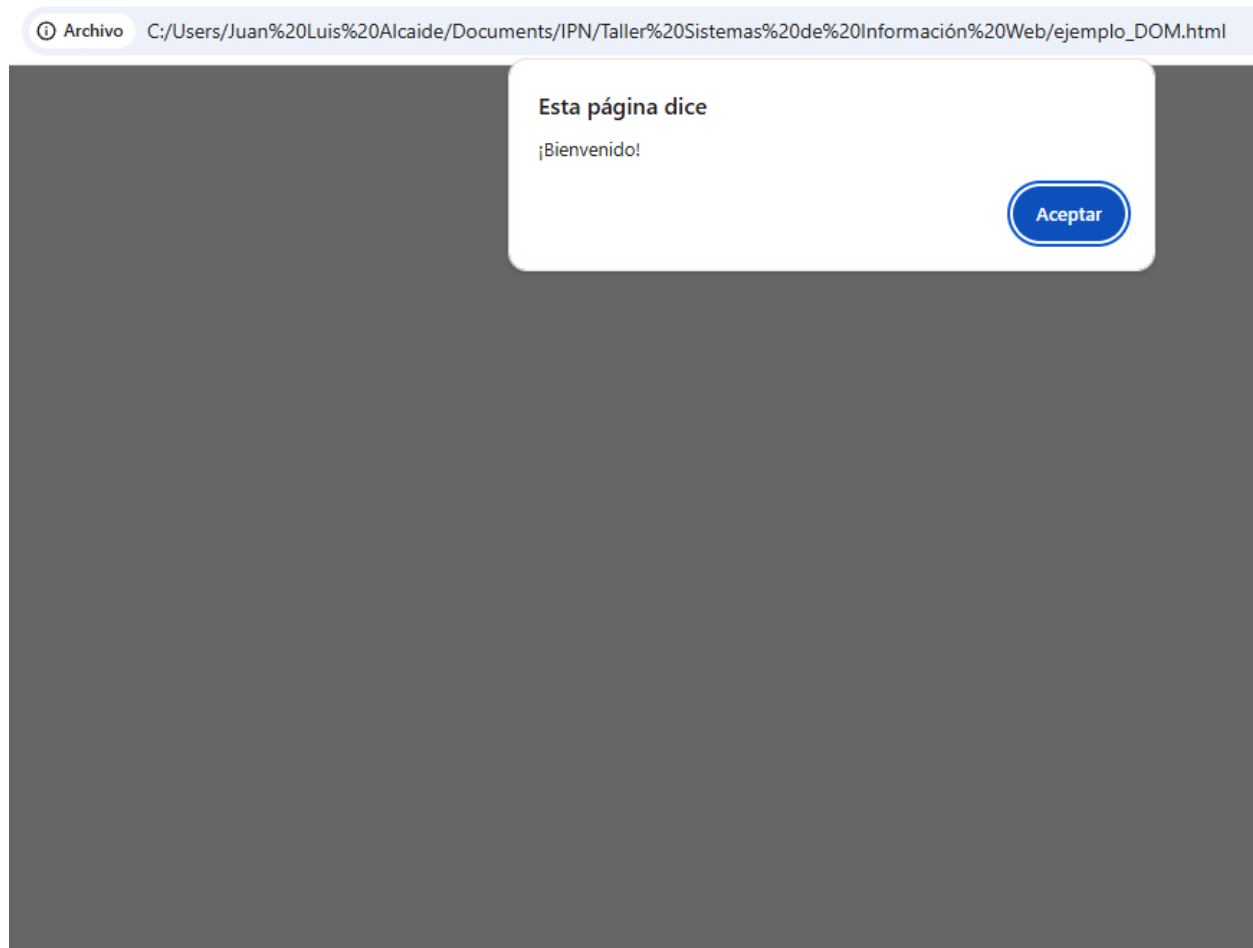
El navegador carga una página HTML, crea una representación de árbol en memoria llamada DOM.

Los nodos están interconectados, reflejando la estructura anidada del HTML (un <p> dentro de <body> es un hijo).

DOM JavaScript

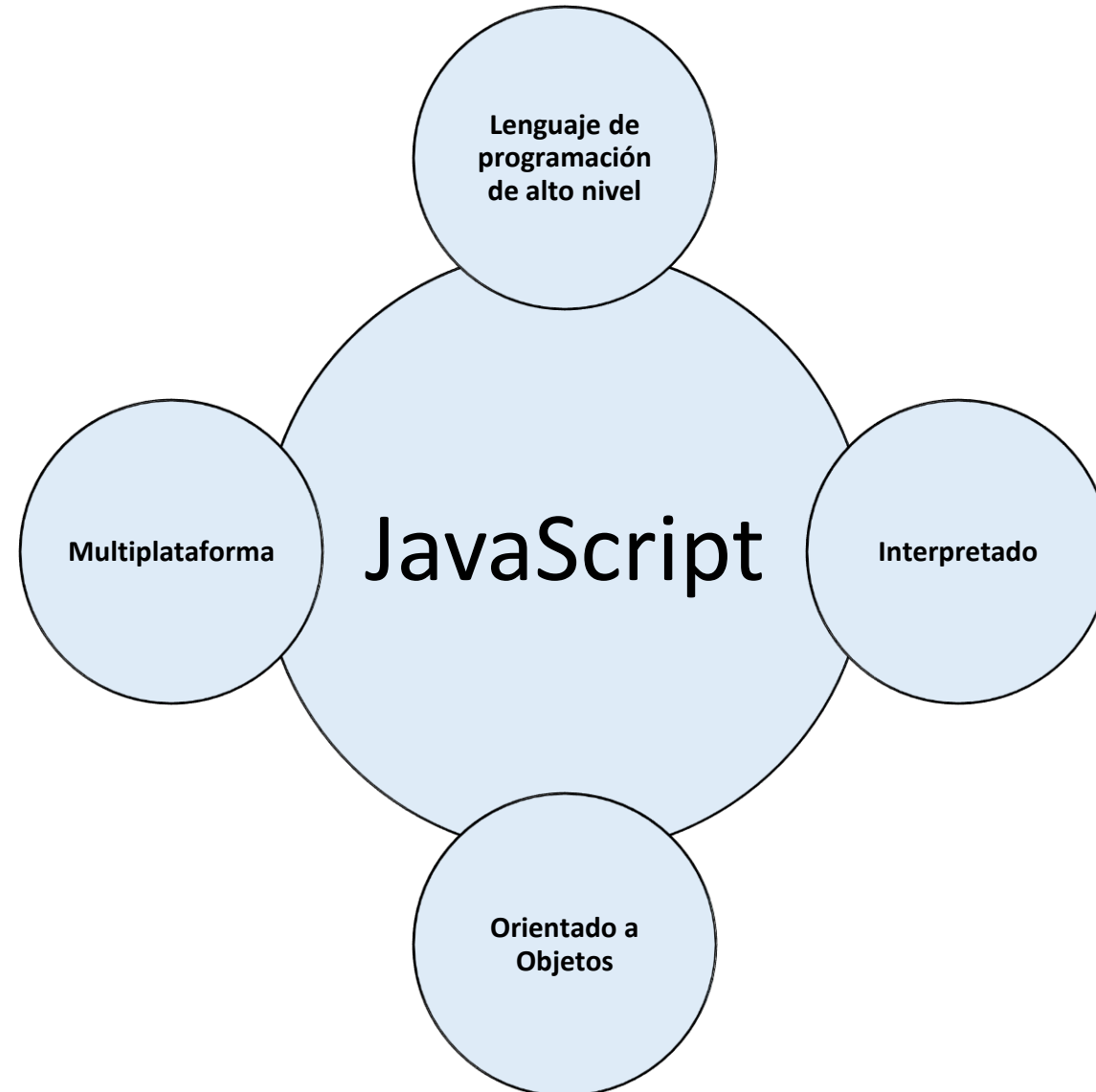
Sintaxis:

`<script> instrucciones JavaScript </script>`



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo DOM</title>
</head>
<body>
  <div >
    <script>
      alert("¡Bienvenido!");
    </script>
  </div>
</body>
</html>
```

Características JavaScript



Crea interactividad en sitios web (lado del cliente) y aplicaciones servidor (Node.js)

Sintaxis JavaScript

Case sensitive - Sensible a mayúsculas/minúsculas:

var, Var y VAR son tres cosas distintas.

Punto y coma (;):

Se utilizan para finalizar sentencias.

Variables:

Se declaran usando las palabras clave let (para valores que cambian), const (para valores constantes) o, de forma tradicional, var.

Sintaxis JavaScript

Tipado Dinámico:

Las variables no están ligadas a un tipo de dato específico y pueden cambiar de tipo durante la ejecución.

Funciones:

Bloques de código reutilizables definidos con la palabra reservada function.

Unicode:

Soporte completo para caracteres Unicode, lo que permite usar tildes o caracteres especiales en nombres de variables.

Sintaxis JavaScript

document.write(): Escribe directamente en el documento HTML.

Sintaxis:

```
document.write("Taller de Sistemas de Información Web");  
document.write(variable);
```

console.error(): Muestra mensajes identificados como.

Sintaxis:

```
console.error("No se puede realizar la consulta");
```

```
try {  
  let data = JSON.parse("invalid-json");  
}  
catch (error) { console.error("Error al parsear datos:", error);  
}
```

Sintaxis JavaScript

Instrucciones para mostrar información

console.log(): Imprime mensajes o valores de variable.

Sintaxis:

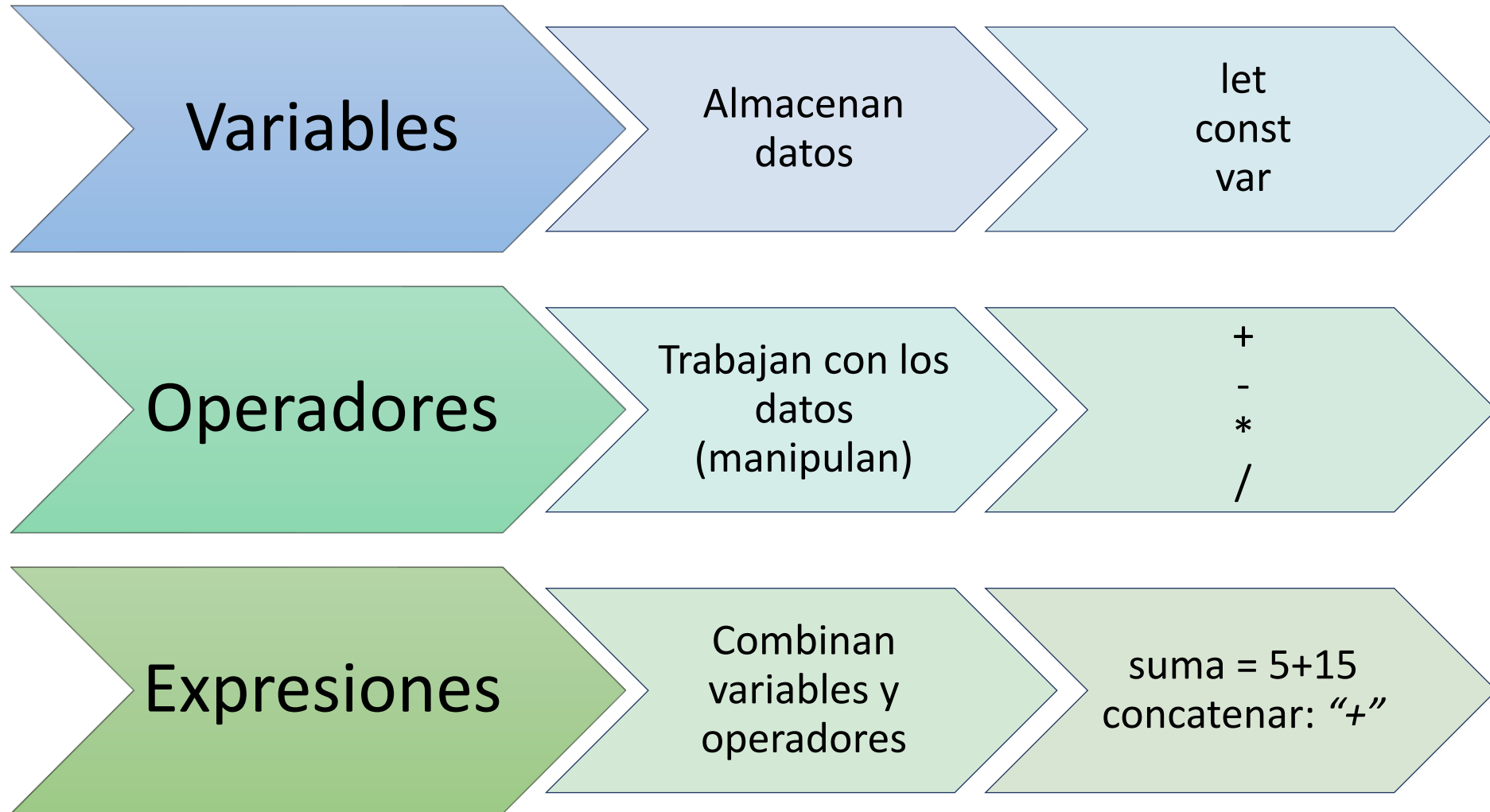
```
console.log("Taller de Sistemas de Información Web");  
console.log(variable);
```

alert(): Muestra una ventana emergente en el navegador.

Sintaxis:

```
alert("¡Bienvenido!");
```

Variables, operadores, expresiones JavaScript



Variables, operadores, expresiones JavaScript



Sintaxis:

```
let edad = 25;  
const nombre = "Juan";
```

Variables, operadores, expresiones JavaScript



Sintaxis:

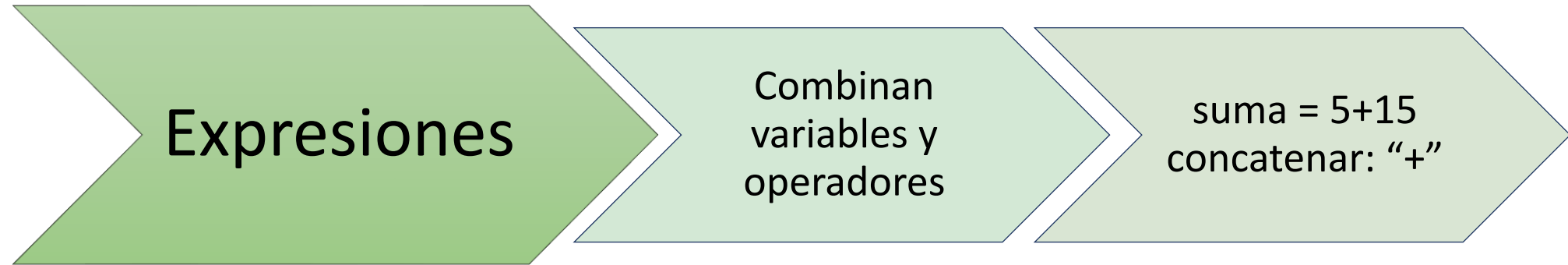
let suma = 25+10;

let resta = 25-10;

*let multiplicación = 25*10;*

let división = 25/10;

Variables, operadores, expresiones JavaScript



Sintaxis:

```
let suma=25+10;
```

```
let resta=25-10;
```

```
let multi=25*10;
```

```
let division=25/10;
```

```
document.write("El resultado de la suma es: <br>" + suma);
```

Variables, operadores, expresiones JavaScript



Archivo C:/Users/Juan%20Luis%20Alcaide/Documents/IPN/Taller%20Sistemas%20de%20Información%20Web/ejemplo_variables_operadores_expresiones.html



El resultado de la suma es:

35

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo variables</title>
</head>
<body>

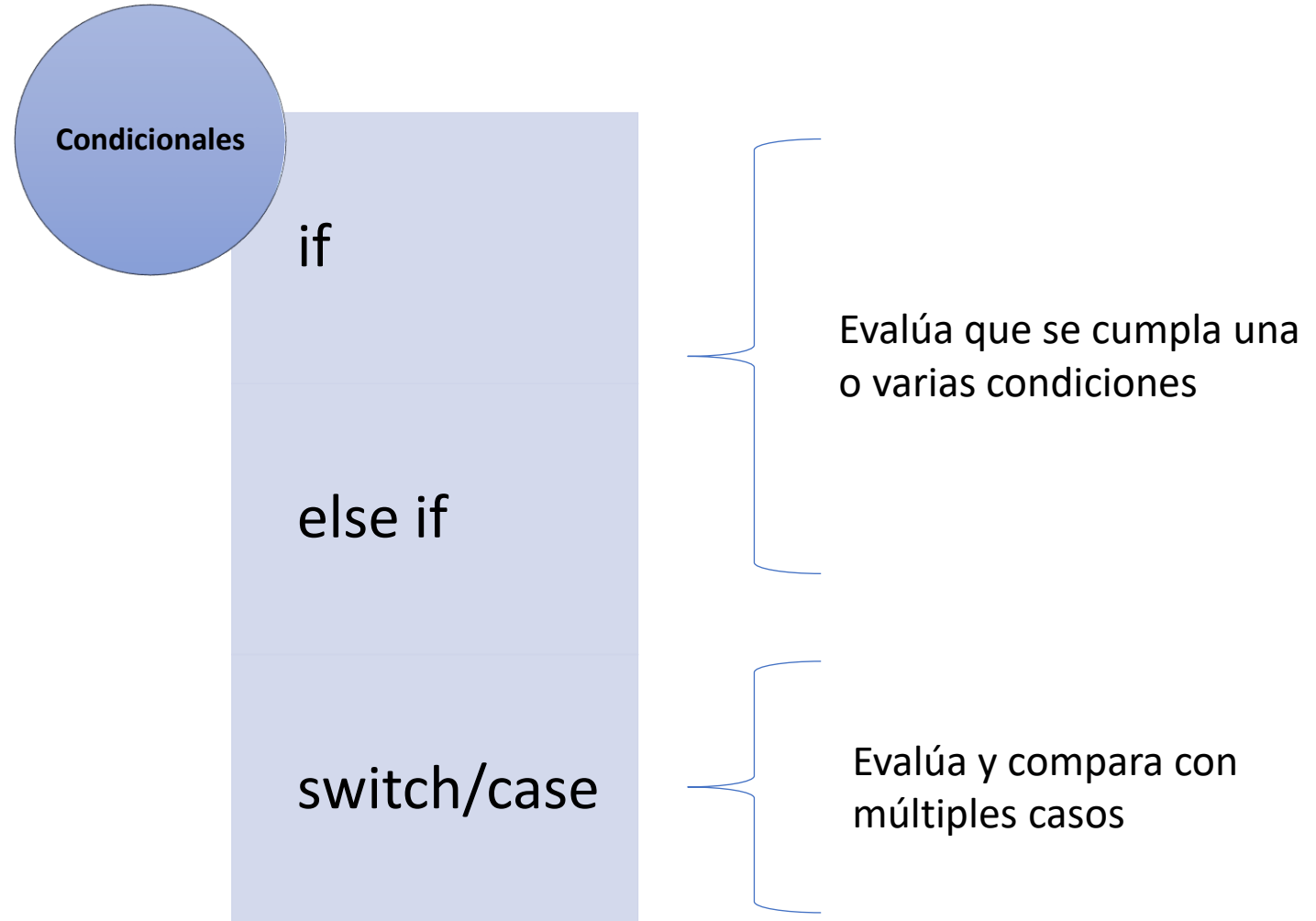
  <div >

    <script>
      let suma=25+10;
      let resta=25-10;
      let multi=25*10;
      let division=25/10;

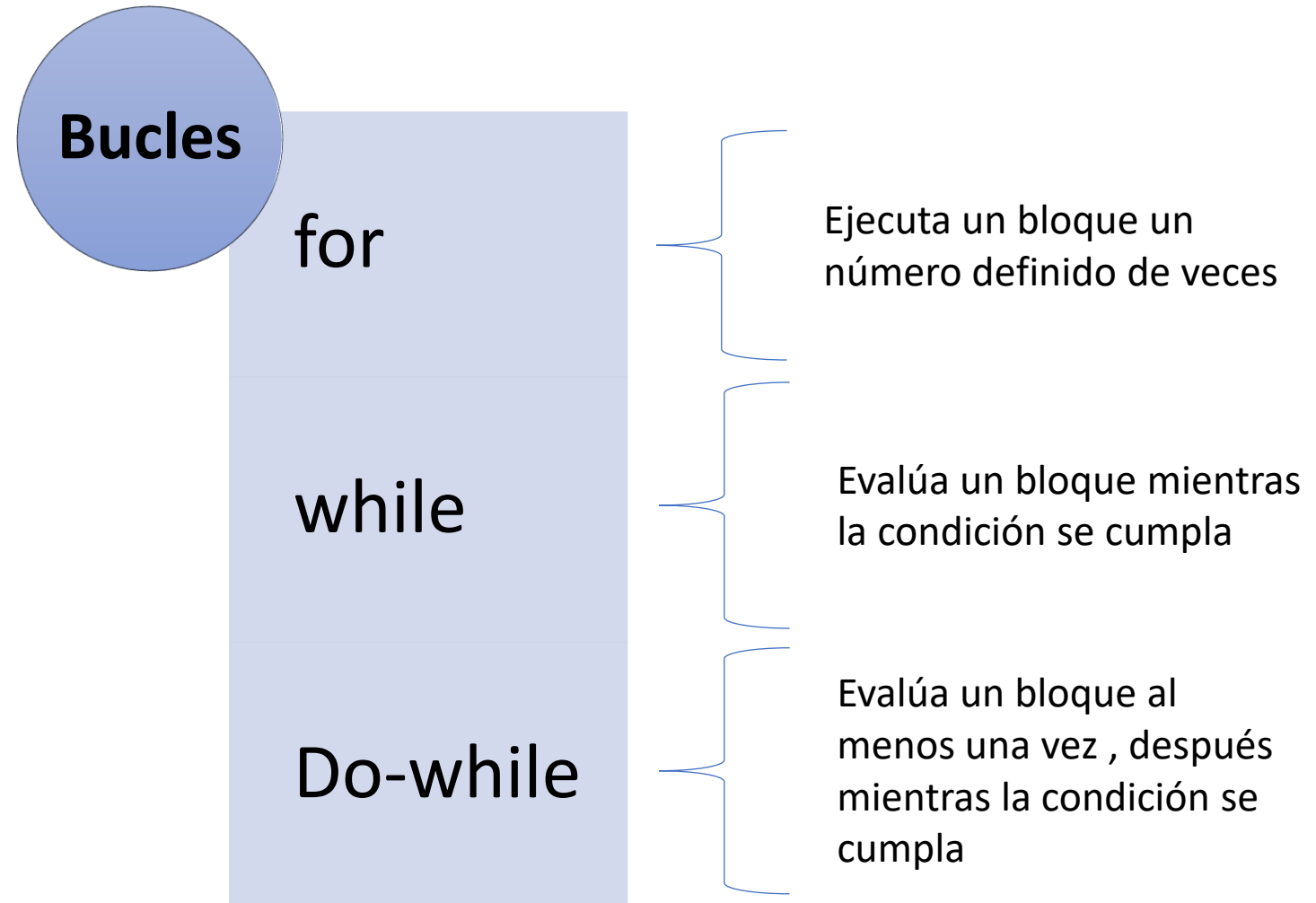
      document.write("El resultado de la suma es: <br>" + suma);
    </script>

  </div>
</body>
</html>
```

Sentencias de selección JavaScript



Sentencias de iteración JavaScript



Arreglos en JavaScript

Colecciones ordenadas de datos

- a. Almacenan diferentes tipos de datos
- b. Son dinámicos: su tamaño puede variar

Sintaxis

```
let carnes = ['pescado', 'pollo', 'res'];
```

```
let colores = new Array('rojo', 'verde', 'azul');
```