

# **Intermedio: definiciones y consejos**

# Repaso rápido

## Valores: datos que entiende Javascript

- Se pueden combinar mediante operaciones (+, -, &&, ...)
- Se pueden almacenar en variables
- Siempre son de un tipo (número, cadena, booleano, función, objeto o **undefined**)
- Se pueden pasar como parámetros a funciones
- Se pueden devolver como valor de retorno de funciones
- Se interpretan como “verdadero” o “falso”...
  - Falso: 0, “”, null, undefined, false
  - Verdadero: cualquier otra cosa

# Repaso rápido

Expresiones: valores o combinaciones de valores

- Un valor es una expresión
- Una invocación a una función es una expresión
- Una combinación expresiones es una expresión
- Se utilizan como condiciones de **if**, **while**, **for**
- Se utilizan al invocar una función (parámetros)
- Se utilizan después de **return** como valor de retorno
- Se utilizan a la derecha de una asignación a una variable

# Repaso rápido

Variables: almacenes con nombre para valores

- Se crean con la palabra clave **var**
- Cuando aparecen a la izquierda de una asignación (=)
  - El valor que guardan se sustituye por el valor de la expresión a la derecha del =
  - El valor anterior se pierde
- Cuando aparecen en una expresión, se sustituyen por el valor que almacenan
- El nombre de una variable puede contener cualquier caracter alfanumérico, \$ y \_

# Repaso rápido

Sentencia: una instrucción, generalmente una línea de código, que representa una acción o una condición

- No se resuelven en un valor
- No son expresiones
  - Las expresiones suelen ser una parte de una sentencia
- Por ejemplo, son sentencias:

```
if (...) { ... }
```

```
while (...) { ... }
```

```
var variable;
```

```
function miFuncion() { ... }
```

# Repaso rápido

Bloque de código: Una serie de expresiones o secuencias encerradas entre { y }

- Aparecen, por ejemplo, en

```
if (...) { /* bloque */ } else { /* bloque */ }
```

```
while (...) { /* Esto es un bloque */ }
```

```
function miFuncion (...) { /* Esto es un bloque */ }
```

# Repaso rápido

Comentarios: textos explicativos que Javascript ignora

- Comentarios de línea:

Empiezan con `//` y se extienden hasta el final de la línea

```
var prueba; // esto es un comentario
```

- Comentarios de bloque:

Empiezan con `/*` y terminan con `*/`

```
var prueba;
```

```
/* todo esto
```

```
es un comentario */
```

# Punto y coma

- Se recomienda escribir un punto y coma al final de cada sentencia
  - Es decir, al final de cada línea
- NO se escribe punto y coma:
  - Detrás del cierre `}` de un bloque
  - En el interior de una expresión
- Es opcional
- Permite escribir varias sentencias en la misma línea

```
alert("hola"); alert("adios");
```

# Indentación

Para mejorar la legibilidad del código, se recomienda indentar (tabular, añadir espacios a la izquierda) el código cada vez que se abre un nuevo bloque.

```
if (false) {  
  while (true) {  
    if (true) {  
      alert("cada nivel indentado!");  
    } else {  
      alert("los } vuelven al nivel de su }");  
    }  
  }  
  console.log("dentro del primer if");  
}
```

# Indentación

Para mejorar la legibilidad del código, se recomienda indentar (tabular, añadir espacios a la izquierda) el código cada vez que se abre un nuevo bloque.

```
if (false) {  
  while (true) {  
    if (true) {  
      alert("cada nivel indentado!");  
    } else {  
      alert("los } vuelven al nivel de su }");  
    }  
  }  
  console.log("dentro del primer if");  
}
```