

## Clase 1 Introducción a la programación (Conceptos claves)

### Variables

Los programas usan los datos de acuerdo a las instrucciones que lo forman. Vamos al ejemplo del saludo, si quiero que salude a alguien de nombre Jose, escribiría un programa como:

**escribe "Hola José"**

Pero si quiero que salude a María lo cambiaría por

**escribe "Hola María"**

¿Un programa para cada nombre diferente? No tiene mucho sentido. Esto se soluciona con las **variables**: elementos con nombre donde se guardan los datos.

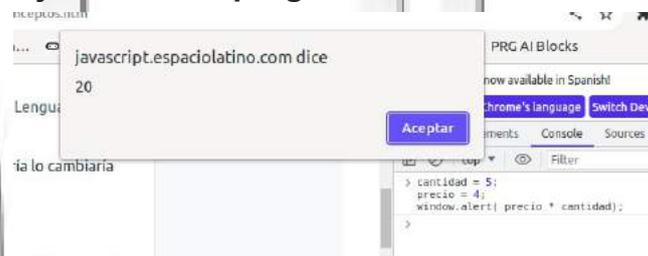
En los ejemplo previo usaría una **variable**, que podría llamar "*nombre*", para guarda el dato "**Juan**".

Para guardar un valor en una variable se usa el símbolo "=" (no es el de las matemáticas). Es el usado habitualmente en los lenguajes de programación.

Un ejemplo:

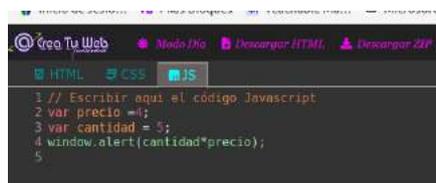
```
cantidad = 5;  
precio = 4;  
echo cantidad*numero;  
0  
cantidad = 5;  
precio = 4;  
window.alert( precio * cantidad);
```

**Una variable es un espacio en memoria que permite guardar un dato y poderlo modificar durante la ejecución del programa.**



*Las variables son como cajas con nombre capaces de almacenar un valor en cada momento.*

En el ejemplo anterior podemos ver dos variables, una llamada "**cantidad**" y otra "**precio**", luego se realiza una operación y se muestra en pantalla (Puedes usar el editor de líneas "<https://creatuweb.espaciolatino.com/ediweb/>" para escribir el código).



¿Como ha funcionado este script?

1. Guarda 5 en la variable cantidad
2. Guarda 4 en la variable precio
3. Escribe el resultado de cantidad \* precio

## Funciones

Si miramos el ejercicio siguiente, para calcular el precio en dólares de cada producto, si fueran muchos, tendrías que escribir una y otra vez el código de arriba. Un poco repetitivo, serían muchas líneas de código muy parecidas. Pues bien, cualquier lenguaje de programación nos ofrece una solución: las **funciones**.

Las instrucciones del ejemplo podrían encerrarse en un bloque con un nombre, por ejemplo **precio()** y tendrías una función.

```
function precio(cantidad, precio){  
var resultado = cantidad*precio;  
return resultado;  
}  
window.alert(precio(5,4)+"\n"+precio(4,10));
```

Si ejecutas este ejemplo verás que aparecen dos valores 20 (5\*4) y 40 (4\*10), pero no has tenido que repetir las instrucciones de multiplicar.

Si quieres calcular un precio en dólares **llamas a la función** y le pones entre paréntesis dos números para que la función calcule el producto de ambos y nos devuelva el resultado.

Otro ejemplo:

```
function precio(precio,cotizacion){  
var resultado = precio/cotizacion;  
return resultado;  
}  
window.alert(precio(120,40)+"\n"+precio(160,40));
```



## Objetos

Los **objetos** son una extensión del concepto de variable.

Se trata de unas estructuras que permite englobar datos y funciones relacionadas entre sí de alguna manera.

- Un objeto puede almacenar valores (propiedades)
- Un objeto puede almacenar funciones (métodos)
- Un objeto puede contener otros objetos

En los ejemplos anteriores de esta clase vimos el código **window.alert()** para mostrar un resultado del programa. Pues **window** es un objeto y **alert()** es un método de ese objeto.

### Crear un objeto vacío

```
const objeto = {}; // Crea un objeto vacío.  
console.log(objeto); // Permite mostrar el contenido del objeto.  
{} // Resultado de mostrar un objeto vacío.
```

### Crear un objeto con propiedades

```
const jugador = {  
  nombre: "Juan",  
  edad: 30,  
  goles: 10,  
};  
console.log(jugador.nombre);  
Juan
```

En el ejemplo del jugador definimos **un objeto jugador** que tiene tres **propiedades** para guardar el nombre, edad y los goles.

### Crear un objeto con propiedades y métodos

```
const jugador = {  
  nombre: "Juan", //Es una propiedad  
  edad: 30, //Es una propiedad  
  talk: function () {return 10;} //Es un método  
};  
console.log(jugador.nombre); // Muestra "Juan"  
console.log(jugador.talk()); // Muestra "10"
```

```
Juan  
10
```

Si queremos mostrar en una ventana el resultado sustituimos **console.log** por **window.alert**.

### Si queremos agregar una propiedad a un objeto

```
const jugador = {}; //Crea un objeto vacío  
jugador.edad = 30; // Agrega la propiedad "edad"  
jugador["nombre"] = "Juan"; // Agrega la propiedad "nombre" Otra manera
```

Existen algunos casos donde debemos utilizar paréntesis rectos, como cuando hay espacios en el nombre de la propiedad.

Marcelo

Rebellato