«Talento Tech»

Desarrollo Web 2

Clase 06

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase N° 6 | Objetos

Temario:

- Objetos
- Métodos en objetos
- Repaso: HTML y CSS

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Objetos

Un objeto es una colección de propiedades, donde cada una de ellas posee una clase **(o KEY)** y su respectivo valor **(o VALUE)**.

Creando Un Objeto:
Imaginemos que tenemos una persona o un auto, que tienen distintas características. Si queremos darle distintas propiedades a nuestro objeto, se las podemos asignar de esta manera:

```
    let miJugador ={
    nombre: "Messi",
    edad: 36,
    posicion:10
}
console.log(miJugador.nombre);
```

Resultado por consola:
Messi

Editando Una Propiedad:
Llamamos inicialmente al nombre de nuestro jugador “Messi”. Luego, modificamos la propiedad y le asignamos un nuevo valor. Como vemos en este ejemplo, podemos acceder a editar propiedades de nuestros objetos.

```
let miJugador ={
    nombre: "Messi",
    edad: 36,
    posicion:10
}
miJugador.nombre = "Lionel"
console.log(miJugador.nombre);
```

Resultado por consola:
Lionel

Agregando Una Propiedad:

```
let miJugador ={
    nombre: "Messi",
    edad: 36,
    posicion:10
}
miJugador.estatura = "1.69m";
console.log(miJugador.estatura);
```

Resultado por consola:
1.69m

Si bien no mencionamos la estatura inicialmente como propiedad, podemos añadirla muy fácilmente.

Eliminando Una Propiedad:
Usando “Delete” podemos eliminar la propiedad que deseemos.

```
let miJugador = {
    nombre: "Messi",
    edad: 36,
    posicion:10
}
delete miJugador.edad;
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Recorriendo nuestro objeto con for

Usando el **bucle for** podemos recorrer todo nuestro objeto de igual manera que lo hacemos con los **arrays**.
Se crea un **array** nuevo, y a cada posición del array le agregamos propiedades distintas.
Esto se vería gráficamente así:

arrayJugadores					
POSICION 0			POSICION 1		
ID	NOMBRE	POSICION	ID	NOMBRE	POSICION
1	LIONEL MESSI	10	2	LUCIANA AYMAR	8

El código se escribiría de esta manera:

```
let arrayJugadores = [
    {
        id: 1,
        nombre: "Lionel Messi",
        posicion: "10"
    },
    {
        id:2,
        nombre: "Gabriel López",
        posicion: "8"
    }
];
for(let i = 0; i <= arrayJugadores.length; i++){
    console.log(arrayJugadores[i].nombre);
    console.log(arrayJugadores[i].posicion);
    console.log(" ");
}
```

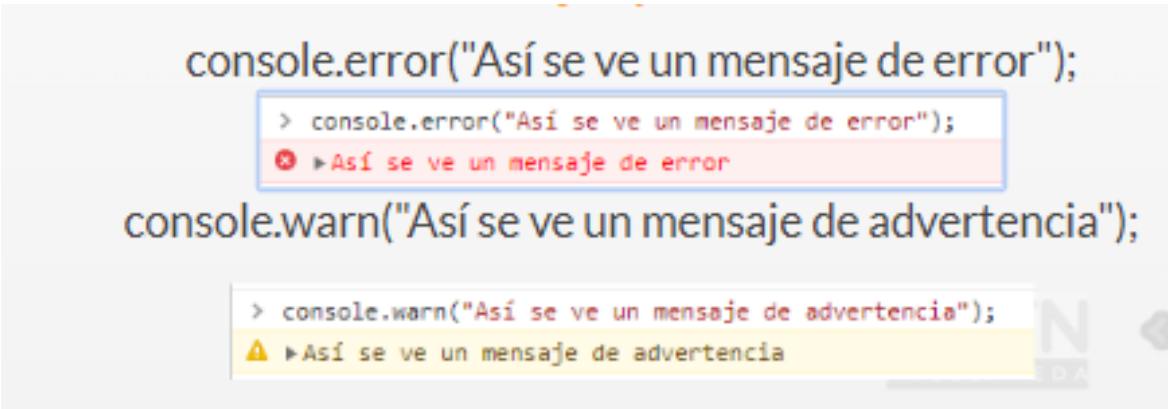
Resultado por consola:
Lionel Messi
10

Gabriel López
8

Métodos en Objetos Primitivos

Como vimos, al desarrollar un programa tenemos la posibilidad de crear nuestros propios objetos, como vimos en el tema anterior.
Pero también contamos con una gran cantidad de objetos que ya se encuentran creados por defecto, y cuentan con sus correspondientes propiedades y métodos.

Hasta ahora, estuvimos operando con algunos de esos métodos, sin notarlo.
Por ejemplo, cada vez que usamos el console.log, estamos usando un **objeto** llamado **console**, que se encarga de imprimir información. Este objeto cuenta con muchos métodos.
Uno de ellos es el **.log**, que ya conocemos.
Sin embargo, no es el único método del objeto console.
Aquí podemos ver más métodos, como el **.error** (que notifica sobre fallas) o el **.warn** (utilizado principalmente para realizar advertencias).



Buenos Aires aprende
Agencia de Habilidades para el Futuro

Objeto Window

Ya hemos utilizado los métodos alert(Muestra una ventana emergente con un mensaje de alerta) y prompt(Muestra una ventana emergente con un mensaje y permite al usuario ingresar un valor.).
El objeto window tiene varios otros métodos útiles como:
confirm(): Muestra una ventana emergente con un mensaje y opciones para confirmar o cancelar.

- **open():** Abre una nueva ventana o pestaña del navegador.
- **close():** Cierra la ventana actual.
- **postMessage():** Envía un mensaje a otra ventana o pestaña abierta.

Objeto Number

El objeto Number, también global, nos provee métodos para trabajar con números, tales como:

.isNaN: es utilizado para determinar si un valor es de tipo NaN (Not a Number). Este método retorna true si el valor proporcionado es NaN, y false en caso contrario

.toFixed(): es utilizado en JavaScript para formatear un número con una cantidad específica de dígitos decimales y devolverlo como una cadena de texto.

.toString(): se utiliza para convertir un valor en su representación de cadena de texto.

.parseInt(): se utiliza para analizar una cadena y devolver un número entero basado en su contenido.

.parseFloat(): se utiliza para analizar una cadena y devolver un número de punto flotante (número decimal) basado en su contenido.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Ejemplo:

```
//Objeto del producto

const producto = {

    nombre: "Camiseta",

    precio: 19.99,

    cantidad: 2,

};

//Convierte el precio y cantidad a string utilizando .toString()

const precioString = producto.precio.toString();

const cantidadString = producto.cantidad.toString();

//Muestra el precio y cantidad como string en un mensaje

console.log(`Producto: ${producto.nombre}`);

console.log(`Precio: ${precioString}`);

console.log(`Cantidad: ${cantidadString}`);

function calcularPrecioTotal(cantidad){

    const precioUnitario = 15.99;

    const cantidadFloat = parseFloat(cantidad);

    if(isNaN(cantidadFloat)){

        console.log("Por favor, ingresa una cantidad válida")

    } else {

        const total = precioUnitario * cantidadFloat;

        console.log(`Total a pagar: ${total.toFixed(2)}`);
```

```
    }

}

const inputCantidad = prompt("Ingresa la cantidad de productos: ");

calcularPrecioTotal(inputCantidad);
```

Repaso HTML y CSS

¿Qué es el desarrollo front-end?

El desarrollo front-end es el proceso de crear y desarrollar la parte visible de un sitio web o aplicación web. Se centra en la implementación de la interfaz de usuario y la interacción del usuario con ella. Implica el uso de los siguientes lenguajes
Lenguaje de marcado, como HTML, para estructurar y organizar el contenido
Lenguajes de estilo, como CSS, para diseñar y dar estilo a la apariencia visual.
Lenguajes de programación, como JavaScript, para agregar interactividad y funcionalidad a la interfaz.
En Desarrollo Web 1, aprendimos sobre HTML y CSS, encargados de la parte estética y visual de nuestra página web.

HTML: Es un lenguaje que permite definir y darle estructura a las páginas web. Sin embargo, esta estructura no es suficiente para hacer que una página luzca bien y sea interactiva, por lo que necesitas otras tecnologías como CSS y JavaScript.

Estructura de HTML

El contenido de un elemento **HTML** está encerrado dentro de las etiquetas de inicio y final, por ejemplo, **<h1>**

Buenos Aires aprende
Agencia de Habilidades para el Futuro

```
<p> Soy un párrafo</p>
```

Apertura de etiqueta	Contenido	Cierre de etiqueta

CSS: El CSS es lo que se llama un lenguaje de hojas de estilo en cascada y se utiliza para darle estilos a elementos escritos en un lenguaje de marcado como HTML.

Estructura de CSS

CSS se basa en un sistema de reglas que determina cómo se aplican los estilos a los elementos HTML. Cada regla está compuesta por un selector, que identifica los elementos a los que se aplicará el estilo, y una serie de propiedades y valores, que definen cómo se verán esos elementos.

```
selector {
    propiedad:valor;
}
```

¿Qué tipos de selectores en CSS existen?

- **Selector de etiquetas**
  Este selector seleccionará elementos HTML según su nombre.

- **Selector de clase**
  El selector de clases selecciona elementos HTML con un atributo de clase específico. Para hacerlo, escribimos un punto (.), seguido del nombre de la clase.

- **Selector de id**
  Este selector utiliza el atributo id que el elemento HTML posee para seleccionarlo. Se escribe el carácter numeral o hashtag (#) seguido de la identificación o id del elemento.
  Se le denomina estilos en cascada porque se lee, procesa y aplica el código desde arriba hacia abajo

❓ Pregunta de reflexión

En la clase de hoy aprendimos a recorrer arrays con el bucle for, pero….
¿Se te ocurre alguna manera de implementar lo aprendido en tu página web utilizando arrays?

En el Desarrollo Front-End;
¿Podes mencionar al menos dos situaciones en las que la combinación de HTML, CSS y JavaScript sería esencial para lograr una experiencia de usuario mejorada en un sitio web?

Buenos Aires aprende
Agencia de Habilidades para el Futuro

📝Desafío #6

1. Creá un objeto llamado **producto** que represente un producto de un **e-commerce**.
2. Asignar propiedades como nombre, precio, y cantidad disponible.
3. Inicialmente, establece el **nombre** del producto como "Camiseta".
4. Luego, modifica la propiedad del producto para cambiar su **nombre** a "Zapatillas".
5. Ahora vamos a añadir la **propiedad “categoría”** al objeto producto con un valor que represente la categoría del producto (por ejemplo, "ropa" o "calzado").
6. Por último, elimina la **propiedad cantidadDisponible** del objeto producto usando el **operador delete**, simulando que el producto está agotado.


Buenos Aires
aprende
Buenos
Aires
Ciudad