

Comparación Editores

Variables

Let o var declaran variables.

Creación de referencias a los elementos del DOM

Let párrafo en este caso crearía un querySelector del único p que tenemos.

```
let parrafo = document.querySelector("p");
```

Window.onload sirve para decirle al archivo que cargue el script una vez se haya cargado lo demás.

```
window.onload = function() {  
}
```

Asignación de manejadores de eventos: addEventListener

El selector querySelector seleccionará con lo que queremos interactuar.

Existe querySelector a secas que solo cogería un elemento y también existe querySelectorAll que cogería todos los elementos con el mismo nombre.

Asignación de que se ejecute un evento (que se haga click, mas una función)

```
constante.addEventListener("click", función)
```

En este ejemplo se selecciona el botón y al hacerle click cambiaría de color

```
document.querySelector("button").addEventListener("click", cambiaColor);
```

Declaramos una constante de dígitos que seleccionara todos los dígitos que existan a través de .digito que es el nombre de la clase que tiene asignado. Recorrer el array de los dígitos que van del 0 al 9. Se le añade el evento click y ejecutara la función "procesaDigito"

```
const digitos = document.querySelectorAll(.digito);  
for (let digito of digitos) {  
  digito.addEventListener("click", procesaDigito);  
}
```

Creación de elementos de forma dinámica con innerHTML

Creamos una constante y lo asignamos a un div con un document.querySelector("div");

Para añadir un div podríamos apuntar a esa constante nombre.innerHTML+=<div>Contenido</div>

Objetos en js

Se puede crear objetos parecidos a los de java:

```
var persona = {  
  nombre: "pepe",  
  edad: 24,  
  codigo postal: 33200,  
  
  saluda: function() {  
    console.log("saludos de", this.nombre)  
  }  
};  
persona["codigo postal"]  
console.log(persona.saluda());
```

StringBuilders

Muy interesantes los `StringBuilders`, en los cuales se ponen las comillas invertidas para construirlos

En este caso podemos concatenar `Strings` con propiedades de un objeto llamado `"persona"`

```
var salida = `La persona: \t ${persona.nombre} tiene ${persona.edad} años \n y vive en  
${persona.direccion}`;
```

Estos dos no serían el mismo tipo de `String`, ya que el primero si lo sería pero el segundo sería un objeto

```
var tipoPrimitivo = "ejemplo de cadena";
```

```
var objString = new String("Cadena como objeto String");
```

`.trim()` en `strings` eliminaría espacios en blanco

`.replace()` para que sustituya caracteres

Los `Strings` se pueden recorrer como un array con un bucle `for`