

JS-Scripts-Tampermonkey

- **Strict mode:**
 - Variante restringida de JavaScript
 - Alternativa al modo normal o "sloppy mode"
 - La semántica es diferente
 - Puede funcionar de forma diferente en los navegadores
 - Se usa automáticamente en los módulos y las clases
 - Cambios:
 - Elimina fallos silenciosos para que lancen errores
 - Soluciona fallos que dificulta la optimización
 - Prohíbe sintaxis que pueda ser definida en futuras versiones de ECMAScript
 - **USO:**
 - Para todo el script

"use strict";

const v = "Hi! I'm a strict mode script!";

- En una función:

```
function myStrictFunction() {  
  "use strict";  
  function nested() {  
    return "And so am I!";  
  }  
  return `Hi! I'm a strict mode function! ${nested()}`;  
}  
function myNotStrictFunction() {  
  return "I'm not strict."  
}
```

- **DIFERENCIAS:**
 - No se puede usar en funciones con parámetros por defecto

```
function sum(a = 1, b = 2) {  
  "use strict";  
  return a + b;  
}
```

```
}  
// Uncaught SyntaxError: Illegal 'use strict' directive in function with non-simple  
parameter list
```

- No se pueden duplicar parámetros en funciones

```
function sum(a, a) {  
  "use strict";  
  return a + a;  
}
```

```
// Uncaught SyntaxError: Duplicate parameter name not allowed in this context
```

- No se pueden usar variables sin declarar

```
"use strict";  
variable = 12;
```

```
//Uncaught ReferenceError: variable is not defined
```

- MAS: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode
- **Módulos:**
 - Separar el código en bloques que se puedan importar si es necesario
 - Muchas bibliotecas y frameworks hacían uso de ellos: CommonJS, RequireJS
 - Soportado en la mayor parte de navegadores modernos
 - Uso de import y export
 - Archivos con extensión .mjs (aunque puede ser también .js)
 - Más información: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>
 - Características:
 - Automáticamente usa el modo estricto
 - Para usar otros módulos necesito usar un módulo
 - No se puede usar import en un script
 - Sólo se ejecutan una vez, aunque se hayan referenciado múltiples veces
 - Las características importadas no están disponibles en el scope global
 - Las variables globales sí que están disponibles en los módulos

- Trabajar con módulos:

<head>

...

<script type="module" src="main.js"> </script>

</head>

- No hace falta añadir defer porque los módulos lo aplican automáticamente
- 1. export
- Exportar las características del módulo: funciones, var, let, const, clases
- Tienen que ser elementos top-level

```
// Archivo modules/square.js
export const name = "square";
export function draw(ctx, length, x, y, color) {
  ctx.fillStyle = color;
  ctx.fillRect(x, y, length, length);
  return { length, x, y, color };
}
```

- También se pueden agrupar al final del módulo

```
// Archivo modules/square.js
const name = "square";
function draw(ctx, length, x, y, color) {
  ctx.fillStyle = color;
  ctx.fillRect(x, y, length, length);
  return { length, x, y, color };
}
export {name, draw};
```

- import: Importar características de un módulo

```
// Archivo main.js
import {name, draw} from "./modules/square.js";
console.log(name);
```

- Los elementos importados se pueden renombrar con as

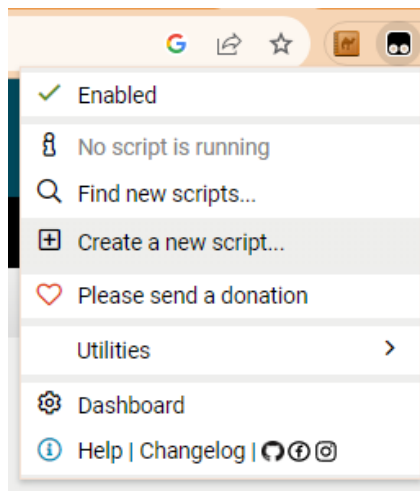
```
// Archivo main.js
import {name as nombre, draw} from "./modules/square.js";
console.log(nombre);
```

- Podemos importar todo dentro de un objeto

```
// Archivo main.js
import * as Square from "./modules/square.js";
console.log(Square.name);
```

- **Tampermonkey:**
 - Extensión del navegador
 - Disponible para múltiples navegadores
 - Permite ejecutar JS en la web que quieras y hacer modificaciones:
 - Eliminar elementos molestos
 - Añadir características que queramos
 - Permite buscar scripts de otros usuarios
 - <https://www.tampermonkey.net/>

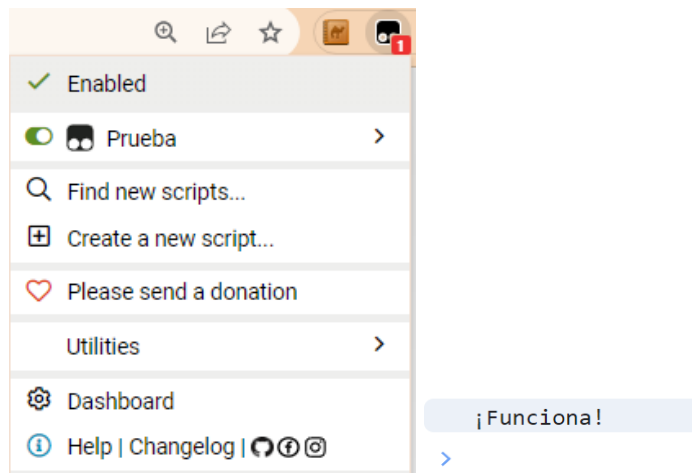
- **CREAR UN NUEVO SCRIPT:**



- Cambiar @match por la URL de la web que queramos
 - Añadir * para que sirva en todo el dominio
- Opcionalmente cambiar @name para asignarle un nombre
- Añadir un console.log con el mensaje que sea dentro de function:

```
(function() {
'use strict';
console.log("¡Funciona!");
})();
```

- Guardar el fichero (File > Save)
- Al recargar la página debería haberse ejecutado.



Ejercicio:

De una web que uses habitualmente:

Piensa en algo que querrías eliminar

Piensa en algo extra que te gustaría que tuviera

Crea un script con Tampermonkey para que solucione algo de lo anterior

Añade un comentario en el script comentando lo que consigues