

Programación funcional

Variables en

- Programación estructurada: guardan valores atómicos
- Programación orientada a objetos: guardan objetos
- Programación funcional: guardan código ejecutable

Variables de tipo función

- Javascript puede manipular variables de tipo función
- Guardan un conjunto de instrucciones
- Pueden ejecutarse

Creación

- `let f = function () { // instrucciones }`
- `let f = Function(" // instrucciones")` — `Function()` es equivalente a `eval()`, pero devuelve una variable para ejecutarla posteriormente

Ejecución

- Se puede ejecutar una variable de tipo función invocandola con `()`

Funciones de orden superior

- Función que posee parámetros o retorno de tipo función
- `forEach` de los arreglos recibe una variable de tipo función y la repite en cada casillero del arreglo

Funciones flecha

- También conocidas como arrow functions
- Permiten redactar una función con una flecha (`->`) en lugar de la instrucción `function`
- La sintaxis es: `(parametros) -> { bloque de código }`
- Se pueden utilizar para
 - asignar variables de tipo función
 - ★ pasarlas como parámetros a funciones de orden superior
- Si la función flecha tiene un único parámetro se pueden omitir los paréntesis
- Si la función flecha consiste únicamente de una instrucción `return` se pueden omitir las llaves y la palabra clave `return`

Programación funcional en arreglos

- `forEach`: recorre y repite una función
- `map`: ejecuta una función en todos los elementos y retorna un nuevo arreglo con los resultados
- `filter`: ejecuta una función booleana (predicado) y retorna un nuevo arreglo con los elementos para los que la función retornó `true`
- `find`: aplica un predicado y devuelve el primer valor que lo cumple.
- `findIndex`: igual que `find`, devuelve el índice
- `some`: devuelve verdadero si algún elemento cumple con un predicado
- `every`: devuelve verdadero si todos los elements cumplen con un predicado
- `reduce`: aplica una función de reducción o pliegue.