
Actividades UD3



15/12/2025
CURSO:2ºDAW

Desarrollo entorno cliente
Celia Caravaca Vega

Índice

ACTIVIDADES UD3:	1
Preguntas Teóricas	1
1. Funciones clausura (closures):	1
2. Funciones callback:	2
3. Funciones de orden superior (HOF):	2
4. Funciones generadoras:	2
5. Expresiones regulares:	2

ACTIVIDADES UD3:

Actividades unidad 3: Funciones

Callback, Closure, HOF, Generadoras y expresiones regulares.

Preguntas Teóricas

1. Funciones clausura (closures):

a) Explica con tus palabras qué es una función clausura en JavaScript.

- Es una función que aunque la función padre termine, las variables creadas dentro de esta se mantienen y puedes acceder a través de la función hija y manipular los valores llamando a la función hija.

b) ¿Por qué las variables dentro de una clausura continúan existiendo después de ejecutar la función que las define?

- Porque al crear funciones dentro de funciones que hacen referencia a las variables de la padre se crea referencia y no se pueden borrar por que aun están siendo usadas.

c) Cita un ejemplo donde sería útil usar una clausura.

```
function iniciarContador() {  
    let i = 0;  
  
    const id = setInterval(a, 1000);  
  
    function a() {  
        i = i + 1;  
        console.log(i);  
    }  
  
    return function parar() {  
        clearInterval(id);  
    };  
}  
  
const detener = iniciarContador();  
  
// cuando quieras parar:  
detener();
```

2. Funciones callback:

a) Define qué es una función callback y su utilidad en programación.

- Una función callback es una función que se le pasa como parámetro a otra función.
- Su utilidad puede ser por legibilidad del código.

b) ¿Cuál es la diferencia entre pasar una función como callback y llamarla directamente dentro de otra función?

- Cuando la llamamos dentro de una función solo se puede ejecutar una función no puedes cambiar que se ejecutará desde afuera, pero cuando pasas una función con callback pasas la función como parámetro por lo que es reutilizable y le puedes pasar distintos tipos de funciones y dependiendo se pueden hacer más cosas.

3. Funciones de orden superior (HOF):

a) ¿Qué significa que una función sea de orden superior?

- Significa que cumple al menos una de estas dos condiciones:
- 1. Recibe una o más funciones como parámetro.
- 2. Devuelve una función como resultado.

b) ¿Qué ventajas tienen las funciones de orden superior frente a funciones normales?

- Código más limpio y corto.
- Reutilización de código.
- Más flexibilidad.
- Código más fácil de leer.

4. Funciones generadoras:

a) Explica qué es una función generadora y qué significa la palabra clave yield.

- Una función generadora es una función que no devuelve los valores de golpe, sino que se queda en pausa y va dando los uno por uno (lleva un asterisco en function*).
- yield es como el return devuelve un valor pero pausa la función y cuando vuelves a llamar a la función sigue por donde se quedó.

b) ¿Cómo se pueden obtener los valores de una función generadora?

- Primero lo haces como si fuera una función normal y usas la variable con el método .next() para recibir un objeto compuesto por valor y si ha terminado.

5. Expresiones regulares:

a) ¿Qué es una expresión regular y para qué se utiliza?

- Es un patrón que sirve para comparar si el texto coincide con el patrón.

b) Explica qué hacen los flags g, i y m.

- g: Busca todas las coincidencias, no solo la primera.
- i: No distingue entre mayúsculas y minúsculas.
- m: (Multilínea) aunque tenga saltos de línea el ^ y \$ hacen que funcione por línea.

c) ¿Cuál es la diferencia entre los métodos test() y exec() de un objeto RegExp?

- test(): Devuelve un boolean si el patrón coincide o no con el texto.
- exec(): Devuelve un array con información de la coincidencia (el texto, posición, grupos...) o null si no coincide.