DESARROLLO WEB FULL STACK

Funciones Javascript

- Funciones como parámetro.
- Clausuras.
- Funciones flecha.

Funciones como parámetro.

En ocasiones vamos a necesitar aplicar diferentes funciones dentro de una función.

Para esto pasamos la **firma** de la función por parámetro.

```
function ponerleOnda(mensaje) {
  return mensaje + '!!!!';
}

function bajarleOnda(mensaje) {
  return mensaje + '...';
}

function mostrarMensaje(forma, mensaje) {
  console.log(forma(mensaje))
}

mostrarMensaje(ponerleOnda, 'Hola');
```

Clausura o Closure

Una *clausura o closure* es la combinación de una función y el ámbito en el que se declaró dicha función. Es decir, son funciones que manejan variables independientes.

En otras palabras, la función definida en la clausura "recuerda" el ámbito en el que se ha creado.

+info:
https://developer.mozilla.org/es/docs/Web/JavaScript/Closures

Ejemplo de clausura

Supongamos que queremos hacer un contador que se mueva a intervalos definidos.

Esto no va a funcionar como queremos porque cada vez que llamemos a contador acumulado es 0 nuevamente.

Necesitamos mantener acumulado en el valor final siempre, necesitamos “encerrarlo”, clausurarlo.

```
function contador(intervalo) {
  var acumulado = 0;
  acumulado = acumulado + intervalo;

  return acumulado;
}

console.log(contador(2));
console.log(contador(2));
console.log(contador(2));
```

Ejemplo de clausura

Como vimos en nuestra definición una clausura se realiza con una función que atrapa el ámbito en el que fué declarada.

Ahora si, nuestra variable queda atrapada dentro de la función que se retorna.

```
function contador(intervalo) {
 var acumulado = 0;
 function acumular() {
  return acumulado = acumulado + intervalo;
 }

 return acumular;
}

var saltar = contador(2);

console.log(saltar());
console.log(saltar());
console.log(saltar());
```

Funciones flecha

La expresión de función flecha tiene una sintaxis más corta que una expresión de función convencional.

Las funciones flecha siempre son anónimas.

+info:
https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Funciones/Arrow_functions

Funciones flecha

Tienen una sintaxis más corta.

También tienen sus propias reglas y convenciones.

```
// Función convencional
var nombre = function(arg1, arg2) {
  return arg1 + arg2;
}

// Función flecha
var nombre = (arg1, arg2) => {
  return arg1 + arg2;
}
```

Funciones flecha

Convenciones de escritura

```
// Si tiene solo un argumento no se
escriben los paréntesis
var nombre = arg => {
  return arg;
}

// Si no tiene argumentos van los
paréntesis
var nombre = () => {
  return 'Juan';
}

// Tiene return implícito
var nombre = arg => arg;
```