Fundamentos JavaScript
Sesión 1 Semana 2

Estructura Switch case

Condicional de selección que nos permite realizar y analizar múltiples decisiones de forma eficiente.

En ocasiones es recomendable utilizar mejor un switch y no varios if ELSE.

```
switch (expresión) {
  case valor1:
    //Declaraciones ejecutadas cuando el resultado de expresión coincide con el valor1
    [break;]
  case valor2:
    //Declaraciones ejecutadas cuando el resultado de expresión coincide con el valor2
    [break;]
  ...
  case valorN:
    //Declaraciones ejecutadas cuando el resultado de expresión coincide con valorN
    [break;]
  default:
    //Declaraciones ejecutadas cuando ninguno de los valores coincide con el valor de la expresión
    [break;]
}
```

Case: El valor del case lo comparamos con la expresión.

Interrupción Break: Esta declaración está asociada a cada una de las etiquetas case. Este nos asegura que el programa salga del case

Default: Indicamos un mensaje cuando ningún valor coincide con el valor de la expresión

```
/*para comparar dentro de un condicional hacerlo con
triple igual ===*/

let dia = 5 // 1 lunes, 2= martes
//0 es domingo
switch(dia){
    case 0:
    console.log('Es domingo');
    break
    case 1:
    console.log("Lunes");
    break;
    case 2:
    console.log("Martes");
    break;
    case 3:
    console.log("Miércoles");
    break;
    case 4:
    console.log("Jueves");
    break;
    case 5:
    console.log("Viernes");
    break;
    case 6:
    console.log("Sábado");
    break;
    default:
    console.log("El día ingresado no existe");
    break;
}
```

switch2.js > expr

```
let expr = 'Platanos';

switch (expr) {
    case 'Naranjas':
      console.log('El kilogramo de naranjas cuesta $10000.');
      break;
    case 'Manzanas':
      console.log('El kilogramo de manzanas cuesta $5000');
      break;
    case 'Platanos':
      console.log('El kilogramo de platanos cuesta $6000');
      break;
    case 'Cerezas':
      console.log('El kilogramo de cerezas cuesta $7000.');
      break;
    case 'Mangos':
    case 'Papayas':
      console.log('El kilogramo de mangos y papayas cuesta $6000');
      break;
    default:
      console.log('Lo lamentamos, por el momento no disponemos de ' + expr);
  }
```

Segmento de ejercicios + socialización 1

Realizar una estructura de selección switch case para los meses del año

```html
<div>
  <label>Mes</label>
  <input id="input" type="text">
</div>
```

```
function calcular() {
    let mes = Number(document.getElementById('input').value);
    switch (mes) {
        case 1:
            alert("El mes es enero");
            break;
        case 2:
            alert("El mes es febrero");
            break;
        case 10:
            alert("El mes es octubre");
            break;
        default: alert("El mes no es enero, febrero ni octubre");
            break;
    }

}
```

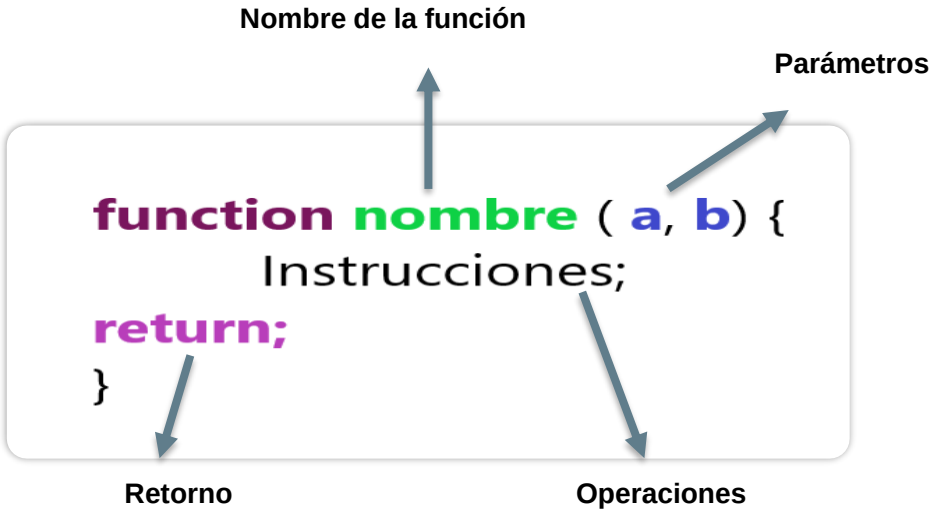
Subprogramas

Conjuntos de instrucciones que realizan una labor específica.

En términos generales, una función es un "subprograma"

Funciones en JavaScript (Objetos de primera clase)

Las funciones son una forma de **agrupar código que vamos a usar varias veces permitiéndonos además, pasar diferentes valores para obtener diferentes resultados.**



Las funciones **no** son lo mismo que los procedimientos.

Una función siempre devuelve un valor, pero un

procedimiento, puede o no puede devolver un valor.

En JavaScript, las funciones son objetos de primera clase,

es decir, son objetos y se pueden manipular y transmitir al igual que cualquier otro objeto.

Concretamente son objetos **Function**

Para qué sirven las funciones

Es un conjunto de instrucciones que realizan una tarea o calculan un valor con el fin de no tener que escribir el código varias veces. Se crea una función y luego se invoca.

Tipos de funciones

Declaración de una función sin retorno

```
// Declaración de una función sin retorno
function imprimirMensaje(){
  console.log('Hola, soy una función sin retorno');
}
 //Invocación de la función - llamado
 imprimirMensaje();
```

Declaración de una función con retorno

```
//Declaración de una función con retorno
function calcularPromedio(num1, num2, num3){
    let promedio = (num1+num2+num3)/3;
    return 'El promedio es: ' + promedio;
}

console.log(calcularPromedio(35,10,9));
```

Otra manera de plantear la función anterior

```
//Declaración de una función con retorno
function calcularPromedio(num1, num2, num3){
    let promedio = (num1+num2+num3)/3;
    return `El promedio es: ${promedio}`;
}

console.log(calcularPromedio(35,10,9));
```

Plantillas literales `` permiten combinar cadenas, expresiones y variables de una manera más sencilla

Asignar la función a una variable

```
//Asignar la función a una variable
const calcula = calcularPromedio;
console.log(calcula(45, 35, 62));
```

Expresión de una función o función anónima

```
//Expresión de una variable
const calculaSuma = function(num1,num2){
  const suma = num1 + num2;
  return suma;
}
console.log(`El resultado de la suma es: `,calculaSuma(10,50));
```

Funciones de tipo Flecha

```
//Función de tipo Flecha
const calcularResta = (num1,num2) => {
    const resta = num1 - num2;
    return resta;
}
console.log('El resultado de la resta es: ' + calcularResta(25,10));
```

Funciones Flecha con un solo parámetro y sola expresión

```
//Función Flecha con un solo parámetro y sola expresión
const multiplicar = x => x**3;
console.log(multiplicar(10));
```

Ejemplo

Por ejemplo, si el usuario hace clic en un botón en una página web, es posible que desee responder a esa acción mostrando un cuadro de información

Función como parámetro

```
//Función como parámetro en otra función
const avisaUsuario = (fun,x) => {
  alert(fun(x));
}

const saludaUsuario = (nombre) => {
  return `Hola ${nombre}`;
}

avisaUsuario(saludaUsuario,'Jenny');
```

Otra manera

```
//Función como parámetro en otra función
const avisaUsuario = (fun,x) => {
  alert(fun(x));
}

const saludaUsuario = (nombre = 'Jenny') => {
  return `Hola ${nombre}`;
}

avisaUsuario(saludaUsuario);
```

```
//Función como parámetro en otra función
const avisaUsuario = (fun,x,v) => {
  alert(fun(x,v));
}

const saludaUsuario = (nombre,apellido) => {
  return `Hola ${nombre} ${apellido}`;
}

avisaUsuario(saludaUsuario,'Jenny','Montoya');
```

Eventos

Los eventos son acciones que suceden en el sistema que está programando y que el sistema le informa para que pueda responder de alguna manera si lo desea.

Podemos identificar como eventos cuando el usuario mueve el mouse, clic sobre algún elemento, se escribe algo en un input, cuando el usuario borra algo, cuando el usuario redimensiona una ventana.

element.addEventListener

Registra un evento a un objeto en específico.

controlador de eventos al documento

Escuchador de eventos

Eventos del mouse

Evento click

index.html > html > body > button#boton

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/styles/style.css">
    <title>Document</title>
</head>
<body>

    <button id="boton" >Presióname</button>
    <br>
    <div id="caja">Hola soy una caja</div>

    <script src="/scripts/script.js"></script>
</body>
</html>
```

```
/*Evento click*/

var boton = document.querySelector('#boton');
boton.addEventListener('click',function(){
    alert('Este es el evento click');
});
```

Evento mouseover

```
//Mouse Over -- cuando se pasa por encima de un botón realiza una acción
//ejemplo cambiar color

var boton = document.querySelector('#boton');

boton.addEventListener('mouseover',function(){
    boton.style.background = 'aquamarine';
});
```

Evento mouseout

```
//Mouse out
//captura el evento cuando sales de pasar por encima del botón realiza

var boton = document.querySelector('#boton');

boton.addEventListener('mouseout',function(){
    boton.style.background = 'blue';
});
```

Eventos presión tecla y foco

index.html > html > body

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/styles/style.css">
    <title>Document</title>
</head>
<body>

    <button id="boton" >Presióname</button>
    <br>
    <div id="caja">Hola soy una caja</div>


    <form>
        <input type="text" name="nombre" id="campo_nombre">
    </form>

    <script src="/scripts/script.js"></script>
</body>
</html>
```

Focus

```
//Focus
//hace foco en el campo
var input = document.querySelector('#campo_nombre');
input.addEventListener('focus',function(){
    console.log('Estoy dentro del input');
});
```

Blur

```
//Blur
//Cuando salimos del foco
var input = document.querySelector('#campo_nombre');
input.addEventListener('blur',function(){
    console.log('Estoy fuera del input');
});
```

keydown

```
//Keydown
//cuando pulsamos una tecla
//sirve para capturar cosas del teclado
//String.fromCharCode
var input = document.querySelector('#campo_nombre');
input.addEventListener('keydown',function(event){
    console.log('Pulsando la tecla', String.fromCharCode(event.keyCode));
});
```

keypress

```
//keypress
//tecla luego de presionada
var input = document.querySelector('#campo_nombre');
input.addEventListener('keypress',function(event){
    console.log('Tecla presionada', String.fromCharCode(event.keyCode));
});
```

keyup

```
8
9   //keyup
0   //captura el evento cuando levanto el dedo de la tecla
1
2   var input = document.querySelector('#campo_nombre');
3   input.addEventListener('keyup',function(event){
4       console.log('Tecla soltada', String.fromCharCode(event.keyCode));
5   });
6
7
```

load

```
//Evento load
//Cuando la página esté cargada se ejecuta el código js que accede al DOM
// y a los elementos html

window.addEventListener('load', () => {



    /*Evento click*/
    var boton = document.querySelector('#boton');
    boton.addEventListener('click', function () {
        alert('Este es el evento click');
    });

    //Mouse Over -- cuando se pasa por encima de un botón realiza una acción
    //ejemplo cambiar color

    //var boton = document.querySelector('#boton');

    boton.addEventListener('mouseover', function () {
        boton.style.background = 'aquamarine';
    });
```

Referencia

https://developer.mozilla.org/es/docs/Web/Events

https://developer.mozilla.org/es/docs/Web/API/Window/localStorage

Fundamentos de JavaScript
Sesión 1 Semana 2