Funciones

{desafío} latam_

Activen las cámaras los que puedan y pasemos asistencia

- Objetivo de la sesión
- Activación de conceptos

- Conceptualización
- Ejercitación

- Reflexión

Inicio

{desafío}
latam_

Objetivos

/* Crear y llamar funciones */

/* Crear funciones que reciban parámetros */

/* Llamar funciones desde el onclick de un elemento */

/* Crear funciones que devuelvan valores */

Activación de conceptos

Contesta la pregunta correctamente y gana un punto

Instrucciones:

- Se realizará una pregunta, el primero en escribir “YO” por el chat, dará su respuesta al resto de la clase.
- El docente validará la respuesta.
- En caso de que no sea correcta, dará la oportunidad a la segunda persona que dijo “Yo”.
- Cada estudiante podrá participar un máximo de 2 veces.
- Al final, el/la docente indicará el 1º, 2º y 3º lugar.
- Esta actividad no es calificada, es solo una dinámica para recordar los conceptos clave para abordar esta sesión.

{desafío} latam_

Activación de conceptos

¿Qué se obtiene al comparar 2 === '2' ?

{desafío}
latam_

Activación de conceptos

¿Qué se obtiene al comparar true || false?

{desafío}
latam_

Activación de conceptos

¿Qué se obtiene al comparar true && false?

{desafío}
latam_

Activación de conceptos

¿Qué se obtiene al comparar true && false == false? ¿En qué orden se resuelve?

Activación de conceptos

```
a = 550
if (a < 576){
  console.log("xs")
}
if(a < 768){
  console.log("sm")
}
if(a < 962){
  console.log("md")
}
else{
  console.log("xl")
}
```

¿Qué se muestra en la consola?

{desafío} latam_

Activación de conceptos

Primer lugar:
_____

Segundo lugar:
_____

Tercer lugar:
_____

{desafío} latam_

Desarrollo

{desafío}
latam_

/* Introducción a funciones */

{desafío}
latam_

Introducción a funciones

¿Qué es una función?

- Las funciones son conjuntos de instrucciones que podemos programar una vez y utilizarlas cada vez que necesitemos.
- Las funciones tenemos que crearlas y luego usarlas (llamarlas)

```
function aprenderHaciendo(){
 // ¡Manos al código!
}
```

Introducción a funciones

Conceptos básicos

- Creamos una función 1 sola vez
- La utilizamos tantas veces como sea necesario.
- Usar, llamar e invocar son sinónimos. Usualmente escucharemos términos como invocar a la función o la función invocada.
- Al momento de crearlas, usualmente ocuparemos los términos definir una función o crearla.

Introducción a funciones

Creando nuestra primera función

```
/* Creamos la función */
function pintar_negro(){
  elemento = document.querySelector("body")
  elemento.style.backgroundColor = "black"
}

/* LLamamos a la función */
pintar_negro();
```

Introducción a funciones

Creando nuestra primera función

```
<!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" />
    <title>Document</title>
  </head>
  <body>
    <script>
      /* Creamos la función */
      function pintar_negro() {
        elemento = document.querySelector("body");
        elemento.style.backgroundColor = "black";
      }

      /* LLamamos a la función */
      pintar_negro();
    </script>
  </body>
</html>

```

- Dentro de la página web crear la función pintar_rojo y la función pintar_amarillo y luego llamarlas.

Ejercicio
¡Manos al teclado!

Objetivos

/* Crear y llamar funciones */ 

/* Crear funciones que reciban parámetros */

/* Llamar funciones desde el onclick de un elemento */

/* Crear funciones que devuelvan valores */

{desafío} latam_

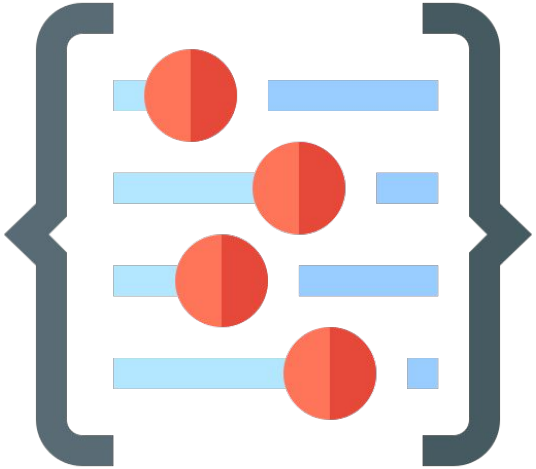
/* Funciones y parámetros */

{desafío}
latam_

Funciones y parámetros

¿Qué son los parámetros?

- Los parámetros son valores que puede recibir una función que nos permite hacerla mucho más flexible.



Funciones y parámetros

Ejemplo de uso

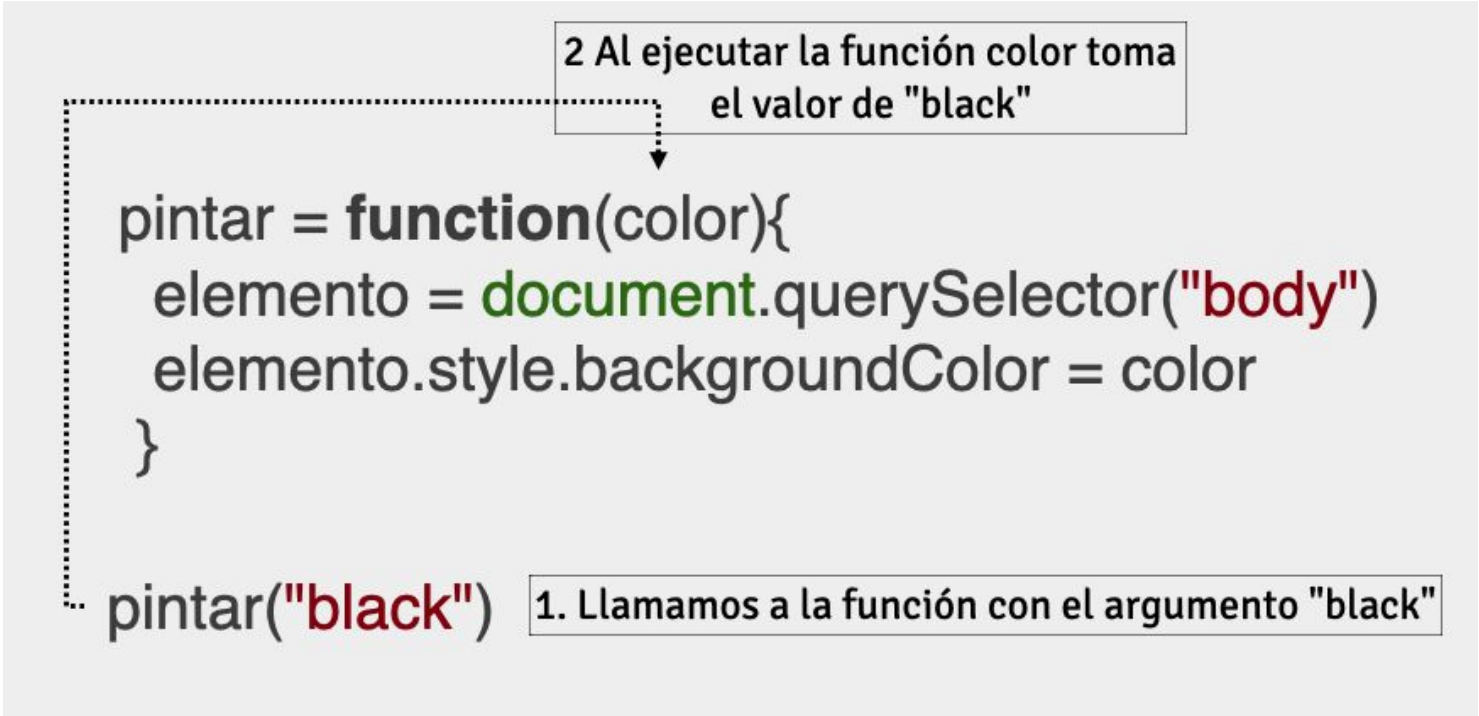
Por ejemplo podemos modificar la función pintar que creamos previamente para que reciba un color y luego utilizar este color para pintar la página.

```
pintar = function(color){
  elemento = document.querySelector("body")
  elemento.style.backgroundColor = color
 }

 pintar("black");
 pintar("red");
 pintar("yellow");
```

Funciones y parámetros

Parámetros y argumentos



Funciones y parámetros

Funciones con múltiples parámetros

Las funciones pueden tener más de un parámetro, para lograrlo simplemente tenemos que separarlos con coma al momento de definir la función.

```
funcionMultiplesParametros(par1, par2, par3)
```

Al llamar a la función tenemos que pasar los valores en el mismo orden que están definidos los parámetros.

```
funcionMultiplesParametros("azul", "#id-2", 5)
```

¡Manos a la obra!

¿Qué se muestra en la consola del inspector de elementos?

```
funcionMultiplesParametros(par1, par2, par3)
{
  console.log(par2)
}

funcionMultiplesParametros(3, 2, 1)
```

Pregunta

{desafío} latam_

Funciones y parámetros

Parámetros con valores por defecto

```
pintar = function(color = "black"){
  elemento = document.querySelector("body")
  elemento.style.backgroundColor = color
 }

 pintar();
 pintar("red");
 pintar("yellow");
```

¡Manos a la obra!

¿Qué se muestra en la consola del inspector de elementos?

```
funcionMultiplesParametros(par1, par2 = 2, par3 = 2 )
{
  console.log(par2 + par3)
}

funcionMultiplesParametros(3)
```

Pregunta

{desafío} latam_

Objetivos

/* Crear y llamar funciones */ ✅

/* Crear funciones que reciban parámetros */ ✅

/* Llamar funciones desde el onclick de un elemento */

/* Crear funciones que devuelvan valores */

{desafío} latam_

/* Funciones y DOM */

{desafío}
latam_

- Dentro de onclick llamaremos a la función, la función debe estar definida dentro de script. El script puede estar en el mismo archivo o en otro.

```
<button onclick="pintar('black')"> Pintar negro</button>
<script>
  function pintar(color) {
    elemento = document.querySelector("body")
    elemento.style.backgroundColor = color
  }
</script>
```

Funciones

Reutilizando funciones

- Ahora que onclick llama a la función se hace sencillo reutilizar la lógica.

```
<button onclick="pintar('black')"> Pintar negro</button>
<button onclick="pintar('red')"> Pintar rojo</button>
<button onclick="pintar('green')"> Pintar verde</button>
<script>
  function pintar(color) {
    elemento = document.querySelector("body");
    elemento.style.backgroundColor = color;
  }
</script>
```

Ejercicio guiado

Demostración

- En una página nueva pondremos 3 imágenes, al hacer click en una de ellas le agregaremos bordes.

Funciones

Solución al ejercicio guiado

```
<img id="img-1" src="http://placekitten.com/g/300/300" alt="" onclick="agregarBordes('img-1')">
<img id="img-2" src="http://placekitten.com/g/300/300" alt="" onclick="agregarBordes('img-2')">
<img id="img-3" src="http://placekitten.com/g/300/300" alt="" onclick="agregarBordes('img-3')">

<script>
  /* Creamos la función */
  function agregarBordes(elementId) {
    elemento = document.querySelector('#' + elementId);
    elemento.style.border="dashed 3px brown"
  }
</script>
```

Ejercicio

- Modifica la página anterior para que la función adicionalmente al elemento pueda recibir un color.
- Especificar distintos colores al llamar a la función en el onlick

Ejercicio
¡Manos al teclado!

{desafío} latam_

Objetivos

/* Crear y llamar funciones */ ✅

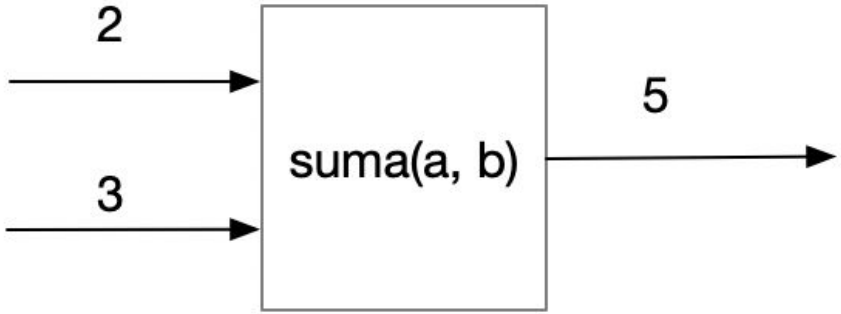
/* Crear funciones que reciban parámetros */ ✅

/* Llamar funciones desde el onclick de un elemento */ ✅

/* Crear funciones que devuelvan valores */

{desafío} latam_

El retorno de las funciones



Las funciones reciben valores y pueden además devolver valores

Funciones

Retorno

```
function suma(a, b){
  return a + b
}

alert(suma(2,3)) /* Opción 1 */
resultado = suma(2,3) /* Opción 2 */
console.log(resultado)
```

Funciones

No todas las funciones devuelven un valor

```
» function suma (a, b){
    alert (a + b)
}
suma(2, 3)
← undefined
```

Funciones

Solo aquellas que tienen return

```
function suma (a, b) {
    return a + b
}
suma(2, 2)
```

Ejercicio

Nos piden crear una función llamada getBkgColor para obtener el color de fondo de un elemento web. O sea debe buscar un elemento a partir de un selector (#id o .class) entregado y debe devolver el color de fondo del elemento.

- ¿Cuál es/son los parámetros de entrada de la función?
- ¿Qué valor deberíamos devolver?
- Ahora creamos la función

Ejercicio
¡Manos al teclado!

Funciones

Solución al ejercicio

```
<div id="el-1" class="element" style="background-color: red"></div>
<div id="el-2" class="element" style="background-color: yellow"></div>
<script>
  function getBkgColor(selector){
    ele = document.querySelector(selector)
    return ele.style.backgroundColor
  }
  getBkgColor("#el-1") /* Probamos nuestra función */
</script>
```

Ejercicio

Nos piden crear una función getValue que obtenga el valor de un input a partir de un selector y devuelva un texto que diga "mucho" si el valor es mayor que un parámetro, "exacto" si el valor es igual al parámetro o "muy poco" si el valor es menor al parámetro

- Antes de hacer el ejercicio discutir:
    - ¿Cuál/es es/son los parámetros de entrada de la función?
    - ¿Qué valor deberíamos devolver?

Ejercicio
¡Manos al teclado!

{desafío} latam_

Objetivos

/* Crear y llamar funciones */

/* Crear funciones que reciban parámetros */

/* Llamar funciones desde el onclick de un elemento */

/* Crear funciones que devuelvan valores */

{desafío} latam_

Cierre

{desafío}
latam_

Reflexionemos

¿Existe algún concepto que no hayas comprendido?

{desafío} latam_

¿Qué sigue?

- Revisar la guía que trabajarán de forma autónoma.
- Revisar en conjunto el desafío.

{desafío} latam_

{desafío} latam_

Academia de talentos digitales