Programación con Java Script I

Sesión sincrónica 3

Funciones

Bienvenida y actividad de bienestar

Duración: 10 minutos.

Nombre de la práctica: Comparte felicidad
Descripción de la práctica: Durante esta actividad compartirás fotografías con diferentes personas, procurando propiciar en ellas emociones positivas.
Palabras clave: Fortalezas de carácter, amor, asombro, emociones positivas
Instrucciones para el participante:

1. Toma fotografías a escenas o instantes que creas que puedan causar emociones positivas a algún familiar o amigo.
2. Comparte las fotografías con dichas personas, esto lo puedes hacer de manera privada (a través de algún programa de mensajería instantánea) o bien públicamente (Facebook).
3. Realiza este durante una semana.
4. Al final del periodo, platica con las personas a quienes les enviaste las fotografías, pregúntales qué les hizo sentir.
5. La intención es que los participantes, al final del periodo establecido para la actividad, se tomen un momento de platicar con las personas a quienes se lo mandaron y sepan qué les hizo sentir.
6. Por último, reflexiona, ¿de qué manera te hizo sentir a ti el compartir dichas imágenes? Piensa cómo esto beneficia tu vida cotidiana.

Fuente: Yu ChenEmail author, Gloria Mark and Sanna Ali. (2016). Promoting Positive Affect through Smartphone Photography. Recuperado de: http://psywb.springeropen.com/articles/10.1186/s13612-016-0044-4

Actividad guiada Parte 1

Duración: 75 minutos.

En los siguientes ejercicios vas a practicar la creación de las funciones aprendidas en esta experiencia educativa.

Ejercicio 1

1. Crea un archivo js en blanco, se sugiere que le pongas de nombre main.js.
2. Crea un archivo html con la estructura básica.
3. Agrega un link desde el html hacia el js.

El archivo html se debería ver de la siguiente forma:

```
<!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>Control de Flujo con JavaScript - Juan Perez</title>
<script type="text/javascript" src="./js/divisor.js"></script>
</head>
<body>
<h1>Curso Javascript</h1>
<p>Control de Flujo</p>
</body>
</html>
```

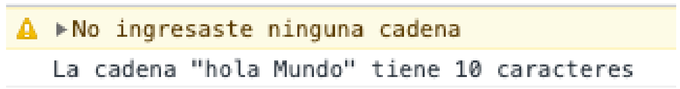
Actividad guiada Parte 1

En el archivo js realiza lo siguiente:
Programa una función que cuente el número de caracteres de una cadena de texto, por ejemplo, **miFunción**("Hola Mundo") devolverá: La cadena “Hola Mundo” tiene 10 caracteres.

1. Crea la función contar caracteres.
2. Como parámetro, asigna un valor por defecto (cadena vacía) para que, si no se pasa alguna cadena, le sea asignado este valor.
3. Utiliza una validación para que, si no se pasa ningún carácter, se ponga un menaje de **warning** en la consola indicando “No ingresaste ninguna cadena”. En caso contrario, indicar con el texto: “La cadena ${cadena} tiene ${numero} caracteres”.
4. Manda a llamar la función con los siguientes parámetros:

```
contarCaracteres();
contarCaracteres("hola Mundo");
```

5. La salida en la consola deberá tener la siguiente apariencia:



6. Convierte esta función en una función flecha asignándola a una constante, de manera que sea una función expresada y utilice el operador ternario.
7. Prueba nuevamente la función y debería tener el mismo resultado.

Actividad guiada Parte 1

Ejercicio 2

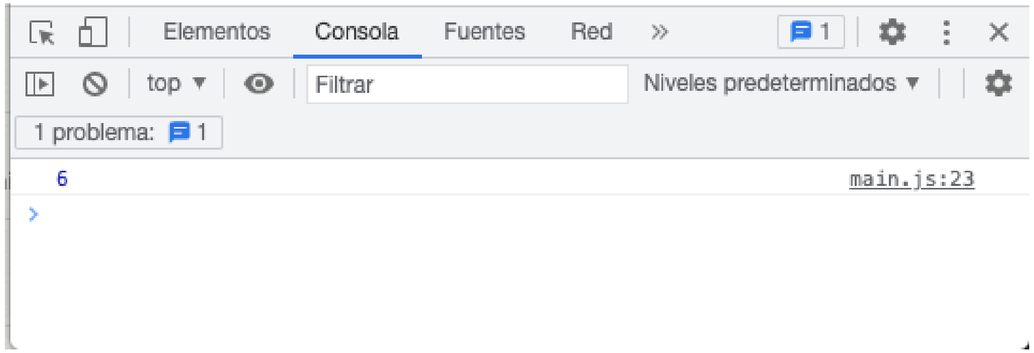
Crea funciones anónimas:

1. Declara una función anónima en una variable llamada **pelicula.**
2. Haz que la función regrese la cadena “El nombre de la película es: " más el nombre de la película proporcionada por el usuario.
3. Invoca a la función desde la consola de esta manera: pelicula(“Avengers”).
4. Tu resultado tendría que ser similar al siguiente:

El nombre de la película es Avengers

Callback

1. Crea una función que se llame **sumame**, a la que le pases como argumento dos números enteros.
2. Crea una variable que se llame **suma** dentro de la función y que realice la **suma** de los dos argumentos.
3. Imprime en la consola el contenido de la variable suma.
4. Mándala llamar pasando dos parámetros. Para este ejemplo se utilizó **sumame**(2,4), esta tendría que ser la salida:



Actividad guiada
Parte 1

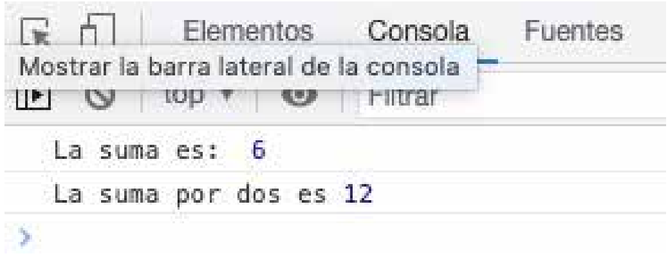
5. Conviértela en una función **Callback**, agregando dos **Callback** en los parámetros, por ejemplo, suma**Ymuestra** y **sumaPorDos.**
6. Usa la función callback **sumaYmuestra** y envíale el valor de suma.
7. Usa la función callback **sumaPorDos** y multiplica el resultado de la **suma** por 2.
8. Manda llamar a la función **sumame** con cuatro parámetros:

   a. En el primero ingresa un número entero.
   b. En el segundo argumento ingresa otro número entero.
   c. En el tercero ingresa un callback, que puede ser una función anónima que recibe como parámetro el **resultado** de suma y lo que hará es mandar a la consola la cadena 'La suma es', resultado.
   d. En el cuarto parámetro ingresa una función anónima que recibe como parámetro **resultado** y lo que hará es mandar a la consola 'La suma por dos es:', resultado.

```
sumame(2,4,function(resultado){
console.log('La suma es: ', resultado);
}, function (resultado){
console.log('La suma por dos es: ', resultado)
});
```

Actividad guiada
Parte 1

9. El resultado de tu código debería verse así:



En la llamada de la función, convierte las dos funciones que pasaron como parámetro en funciones flecha.

Consideraciones

Reforzar los conocimientos de aquellos temas que consideres necesarios para que el aprendedor pueda realizar los ejercicios de la parte 1.

Receso

Duración: 10 minutos.

Universidad Tecmilenio

Actividad guiada
Parte 2

Duración: 75 minutos.

Utilizando el código desarrollado HTML y la carpeta del ejercicio anterior, realiza las siguientes actividades:

Ejercicio 3

Crea un programa que calcule el factorial de un número (El factorial de un número entero positivo n se define como el producto de todos los números enteros positivos desde 1 hasta n), por ejemplo, factorial(5) devolverá 120 (=1*2*3*4*5).

1. Crea un nuevo archivo js y nómbralo factorial.js.
2. Apunta el html para que tome la funcionalidad del nuevo archivo js.
3. En el archivo js, crea una función expresada llamada factorial que haga las siguientes validaciones:

   a. Pedir al usuario que ingrese un número y evalúa si se ingresó con undefined, en caso de que sí, imprime un warning en la consola que le indique al usuario “No ingresaste un número”.
   b. Valida que el valor que se está ingresando es un número, en caso de que no lo sea, imprime un error en la consola que diga ‘el valor de “$(variable)” ingresado NO es un número’.
   c. Valida que, si el número ingresado es igual a 0, se envíe a la consola un error que diga “El número no puede ser 0”.
   d. Valida el número con Math.sign, en caso de ser negativo, imprime un error en la consola que diga “El número no puede ser negativo”.
   e. Declara una variable factorial con valor igual a 1.
   f. Crea un ciclo en donde tomes el valor del número ingresado por el usuario y se lo asignes a la variable que controla el ciclo, en cada iteración le debes restar 1 y harás esto mientras que la variable que controla el ciclo sea mayor que uno.
   g. Dentro del ciclo, multiplica la variable factorial por la variable que controla el ciclo y asigna ese nuevo valor a la misma variable factorial.
   h. Cuando termine el ciclo, informa a la consola la siguiente cadena `El factorial del número ${variable) es $(factorial)´.

Actividad guiada
Parte 2

4. Realiza las siguientes pruebas:

```
factorial(0);
factorial("5");
factorial(1,2,3);
factorial(0);
factorial(-5);
factorial(5);
factorial(8);
```

Ejercicio 4

Crea una función que valide que un texto sea un email válido. Regresará verdadero o falso según corresponda:

1. Crea un nuevo archivo js y nómbralo **email.js.**
2. Apunta el html para que tome la funcionalidad del nuevo archivo js.
3. En el archivo js, crea una función expresada llamada **validarEmail** que reciba como argumento email e inicialízalo como una cadena vacía.
4. Dentro de la función, valida lo siguiente:
   a. Que se haya ingresado una cadena, en caso contrario, informa a la consola con un warning “No ingresaste un email”.
   b. Valida que la cadena introducida sea del tipo ***string***, en caso contrario, regresa un error en la consola con la cadena `el valor de “$(variable)” ingresado NO es una cadena de texto’.
   c. Declara una variable **expReg** e iguálala a una expresión regular:

Actividad guiada
Parte 2

```
let expReg=/[a-z0-9]+(\.[_a-z0-9]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,15})/i.test(email);
```

5. Realiza al menos las siguientes pruebas:

```
validarEmail();
validarEmail(34);
validarEmail("usuario.apellido@univeridad");
validarEmail("nombre_apellido@universidad.edu");
```

Consideraciones

Aprendedor: Algunas de las validaciones las puedes optimizar utilizando swift, si consideras prudente, utilízalo.

Para generar el salto de línea en la alerta, se sugiere el salto de línea que en JavaScript se representa con \n.

Recuerda que al término de tus actividades siempre es importante que las subas a tu repositorio remoto en GitHub.

Cierre

Duración: 10 minutos.

En esta actividad guiada pudiste practicar con los tipos de funciones más utilizados en el mundo de la programación en JavaScript: las funciones “normales”, que usarás cada día; las callback, que tienen un comportamiento un poquito diferente, ya que se pueden llamar a sí mismas desde dentro de la función; y las funciones flecha, que actualmente están sustituyendo a las funciones en donde utilizas la palabra función, a medida que avances, podrás ver que están acaparando el código JavaScript y en otros lenguajes más avanzados como TypeScript.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.