

Programación con Java Script I

Sesión sincrónica 2

Controles de flujo



Bienvenida y actividad de bienestar

Duración: 10 minutos.

Nombre de la práctica: Comer con atención plena

Descripción de la práctica: Este ejercicio te servirá para practicar la atención plena en una actividad cotidiana, sin embargo, puedes extrapolar los principios a diferentes actividades de tu día a día.

Palabras clave: Mindfulness, meditación, gratitud

Instrucciones para el participante: ¿Cuándo fue la última vez que realmente disfrutaste cada bocado de tu comida? Es probable que hayas comido alimentos excelentemente preparados, de un gran sabor y calidad, pero ¿lo has disfrutado bocado a bocado?

Ryan Niemic, psicólogo y autor reconocido en Psicología Positiva, propone un ejercicio bastante sencillo para cultivar la atención plena. En lugar de comer apresuradamente, y casi en automático, date tiempo para hacer lo siguiente:

1. Una vez que tengas frente a ti tus alimentos, aprecia cada elemento de tu comida. No sólo pienses en los ingredientes, también reflexiona sobre todas las acciones y esfuerzo que permitieron que esa comida esté en tu plato.
2. Usa tus cinco sentidos para apreciar cada bocado, observa los colores, los olores, siente las texturas y sabores.
3. Después de cada bocado baja tus cubiertos. De esta manera comerás con mayor lentitud y te forzarás a repetir este proceso con atención plena.
4. Evalúa constantemente cuál es tu nivel de hambre y de saciedad, es decir, piensa si después de cada bocado aún estás hambriento o si ya estás cerca de la plenitud.
5. Al final reflexiona sobre el ejercicio y pregúntate: ¿cambió tu percepción y sabor de los alimentos?

Llevar a cabo este tipo de ejercicios en diferentes ámbitos puede ayudar en gran medida a cultivar la atención plena. ¡Practícalo!

Fuente: Tarragona, M. (s.f.). Comer con atención plena. Recuperado de <http://positivamente.com.mx/comer-con-atencion-plena/>

Actividad guiada

Parte 1

Duración: 75 minutos.

En los siguientes ejercicios vas a practicar el control de flujo en JavaScript aprendido en esta experiencia educativa.

Ejercicio 1

El primer ejercicio consistirá en realizar un programa que pida dos números y nos indique cuál es el mayor, el menor y si son iguales. Es importante mencionar que en este momento no es importante dónde guardes la información, sin embargo, es recomendable que tengas una carpeta especial para que vayas guardando los archivos creados.

1. En la carpeta que elegiste para guardar los archivos de tus ejercicios, crea el archivo index.html y carga el proyecto en tu entorno de trabajo (se explicará considerando que tienes instalado Visual Studio Code (VSC)). Para ello, es necesario que abras la carpeta en la cual creaste tu archivo.
2. Una vez que estés en el entorno de trabajo, abre el archivo index.html
3. Crea una estructura básica de 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">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Actividad guiada Parte 1

4. Instala el plugin emmet para que puedas utilizar los atajos, por ejemplo, para crear la estructura básica de un documento HTML solo se requiere teclear el símbolo !.
5. Modifica el título de la página para que diga “Control de Flujo con JavaScript – [Tu nombre]”.
6. Dentro del cuerpo del documento, con una etiqueta h1, coloca el título “Curso JavaScript” y coloca “Control de Flujo” en un párrafo.
7. Para visualizar el resultado, solo tienes que arrastrar el archivo al navegador o instalar el plugin Live Server en VSC. Tu página se debe ver así:



8. Crea, dentro de este nuevo directorio, un nuevo archivo js que se llame condicionales y asegúrate de referenciarlo correctamente en el archivo index.html.
9. Crea la variable numero1 y asígnale el valor que el usuario va a introducir a través de un prompt, puedes realizarlo mediante la siguiente instrucción:

```
var numero1 = parseInt(prompt('Introduce el primer número',0));
```


Actividad guiada

Parte 1

10. Crea una segunda variable **numero2** que diga lo mismo, pero la cadena deberá desplegar “Introduce el segundo número”.
11. Imprime en la consola los valores de las variables **numero1** y **numero2**.
12. Crea una sentencia **if** en la que compares si el número 1 es igual al número 2, en caso de que sean iguales, presenta en una alerta la cadena “Los números son iguales”. Para el alert, puedes hacerlo con esta instrucción:

```
alert("LOS NÚMEROS SON IGUALES");
```

13. En caso de que no se cumpla la condición, evalúa si el número 1 es mayor al número 2. Si es verdadero, genera dos alertas que tengan la cadena:
 - a. “EL NÚMERO MAYOR ES” +**numero1**
 - b. “EL NÚMERO MENOR ES” +**numero2**
14. En caso de que no se cumpla la condición, evalúa si el número 2 es mayor al número 1 y agrega las alertas correspondientes como en el caso anterior.
15. En caso de que no se cumpla ninguna de las condiciones anteriores, genera la alerta “INTRODUCE NÚMEROS CORRECTOS”.

Ahora, haz una validación para que la aplicación verifique si los valores introducidos son numéricos, en caso contrario, volverá a pedirlos.

1. Modifica tu código. Después de la declaración de las variables, agrega una sentencia para que, mientras se cumpla cualquiera de los siguientes supuestos, vuelva a pedir las variables:

Si numero1 <= 0 o **numero2** <= 0 o **numero1** no es numérico o **numero2** no es numérico

Actividad guiada

Parte 1

Ejercicio 2

Utiliza un bucle para mostrar la **suma** y la **media** de los números introducidos hasta introducir un número negativo y mostrar el resultado.

1. Crea un archivo llamado **suma_media.js**.
2. Modifica el documento index.html para apuntar al nuevo archivo.
3. Crea dos variables, una para **suma** y una para el **contador**, e inicialízalas en 0.
4. Declara un bucle para hacer mientras (**while**).
5. Dentro del bucle, en la parte de hacer, solicita al usuario "Introduce números hasta que uno sea número negativo". Asigna el valor que el usuario va a introducir a la variable **numero**.
6. Valida si el dato introducido es numérico, si no es numérico, asigna 0 a la variable **numero**.
7. En otro caso, pregunta si numero es mayor o igual a 0, si esto ocurre, suma a la variable suma el valor del **número** introducido e incrementa en uno al contador.
8. Fuera de la validación, imprime en consola el valor de **suma** y el valor del **contador**.
9. Muestra con una alerta "La suma de todos los números es: " el valor de **suma**.
10. A continuación, en la misma alerta, imprime el texto "La media de todos los números es: " (realiza la división de **suma** entre el **contador**).

Actividad guiada

Parte 1

Ejercicio 3

Haz un programa que muestre todos los números que están entre dos números introducidos por el usuario.

1. Crea el archivo intervalo.js.
2. Modifica el archivo index.html para que apunte al archivo recientemente creado.
3. Solicita dos números al usuario y asígnalos a las variables `numero1` y `numero2` respectivamente.
4. Imprime en el documento html con la etiqueta `h1`:

```
document.write("<h1>De "+numero1+" a "+numero2+" están estos números:</h1>")
```

5. Crea un bucle para recorrer desde el valor de la variable `numero1` hasta el valor de la variable `numero2`.
6. Dentro del bucle, imprime en el html desde el valor de **numero1**, incrementando de uno en uno, hasta llegar a **numero2**.

Ejercicio 4

Modifica el ejercicio anterior para mostrar en la consola solamente los números impares que hay entre los dos números introducidos por el usuario. Puedes hacerlo utilizando **while** y el operador **modal(%)**.

Consideraciones

Instructor: Refuerza los conocimientos de aquellos temas que consideres necesarios para que el aprendedor pueda realizar los ejercicios de esta Parte 1.

Receso

Duración: 10 minutos.

```
def __init__(self, **kwargs):
    self.name = kwargs.get("name")
    self.damage = kwargs.get("damage")
    self.armor = kwargs.get("armor")
    self.hit_points = kwargs.get("hp")
    self.current_hit_points = kwargs.get("hp")
    self.level = kwargs.get("level")

def attack(self, enemy: 'Unit') -> int:
    """
    Attack enemy unit. Return number of damage
    """
    damage_top_limit = self.damage + random.randint(0, 10)
    damage_bot_limit = self.damage - random.randint(0, 10)
    calculated_damage = random.randint(damage_bot_limit, damage_top_limit)
    if calculated_damage < 0:
        return 0
    enemy.current_hit_points -= calculated_damage
```


Actividad guiada

Parte 2

Duración: 75 minutos.

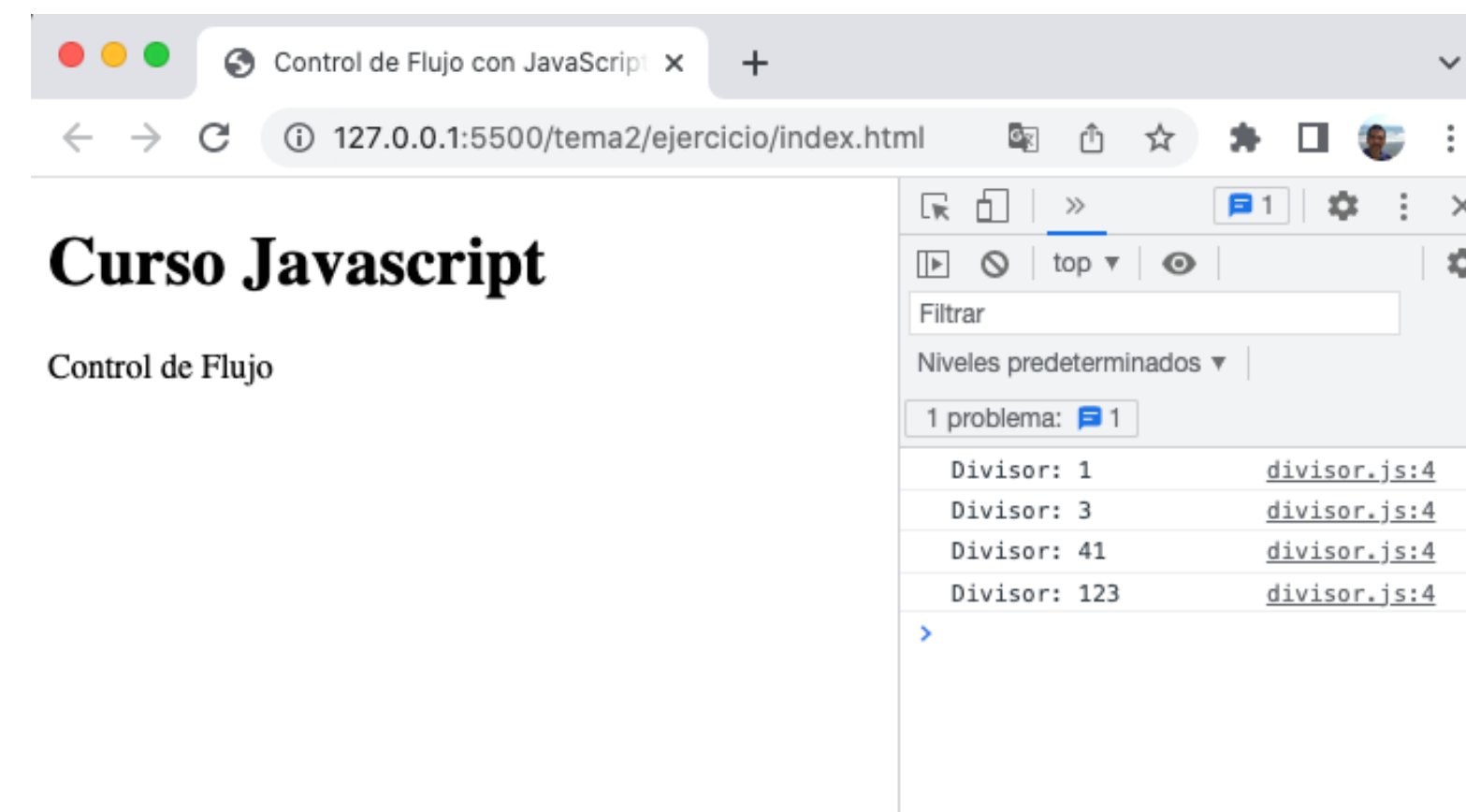
Utilizando el código HTML y la carpeta del ejercicio anterior, realiza las siguientes actividades:

Ejercicio 5

Muestra todos los números divisores de un número introducido en el prompt.

1. Crea un nuevo archivo js y nómbralo **divisor.js**.
1. Apunta el html para que tome la funcionalidad del nuevo archivo js.
2. Solicita al usuario un número y guárdalo en la variable **numero**.
3. Con un for que recorre desde 1 hasta numero, valida si el residuo de la división del número dado por el usuario dividido entre el número que itera es igual a 0, en ese caso, imprime en consola "Divisor: " + **i**, donde i es el número que está iterando.

Al final del ejercicio, tu programa debe entregar algo así (la prueba se realizó utilizando el número 123):



Actividad guiada

Parte 2

Ejercicio 6

Crea una calculadora en JavaScript.

1. Crea un nuevo archivo que se llame **calculadora.js**.
2. Realiza las modificaciones en el archivo html para que reconozca este nuevo archivo.
3. Solicita al usuario dos números a través del prompt.
4. Valida que los números sean mayores o iguales a 0 y que sean valores numéricos. En caso contrario, con un ciclo, vuelve a pedir los datos hasta que sean numéricos y mayores o iguales a 0.

5. Crea una variable que se llame **resultado** en la que vas a imprimir la siguiente cadena compuesta:

```
"La suma es: " +(numero1+numero2)+ " <br/>" +  
"La resta es: " +(numero1-numero2)+ " <br/>" +  
"La multiplicación es: " +(numero1*numero2)+ " <br/>" +  
"La división es: " +(numero1/numero2)+ " <br/>" ;
```

6. Crea una variable llamada resultadoCMD en la que vas a asignar la siguiente cadena compuesta:

```
"La suma es: " +(numero1+numero2)+ " \n" +  
"La resta es: " +(numero1-numero2)+ " \n" +  
"La multiplicación es: " +(numero1*numero2)+ " \n" +  
"La división es: " +(numero1+numero2)+ " \n" ;
```

7. Imprime en el documento el contenido de la variable resultado.
8. Imprime en una alerta el contenido de la variable resultadoCMD.
9. Imprime en la consola el contenido de la variable **resultadoCMD**.

Actividad guiada

Parte 2

Consideraciones

Instructor: Si consideras prudente, solicita al aprendedor que cambie el valor de alguna variable desde la consola y asegúrate de que lo pueda visualizar, esto para que aprenda que una variable puede cambiar de valor durante la ejecución de la aplicación.

Posiblemente el aprendedor no sepa utilizar GitHub y generar su cuenta.

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

Cierre

Duración: 10 minutos.

En esta actividad guiada aprendiste a solicitar información al usuario y validarla para que, dependiendo de la condición que se cumpla, el programa haga una u otra actividad. También pudiste poner en práctica la creación de ciclos para poder darle cierta inteligencia a tus aplicaciones.

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.