

Programación con JavaScript II

Sesión sincrónica 4

Scope y this



Bienvenida y actividad de bienestar

Duración: 10 minutos.

Nombre de la práctica: Felicidad en el trabajo.

Descripción de la práctica: Reflexionarás sobre las distintas dimensiones de tu vida cotidiana, enfocando el análisis a cómo fomentar un estado de ánimo y relaciones positivos en el ámbito laboral.

Palabras clave: Involucramiento, emociones positivas y relaciones positivas.

Instrucciones para el aprendedor:

Elegir conscientemente maneras de incrementar la felicidad en el trabajo puede hacer la diferencia en cómo nosotros nos sentimos y qué tan bien nos desempeñamos. En lugar de quejarnos del trabajo, ¿por qué no pensar en cómo podemos obtener mayor felicidad de lo que hacemos?

Estar más involucrados en lo que hacemos contribuye a nuestra felicidad y bienestar y nos lleva a un mejor desempeño y productividad. A manera de reflexión, responde las siguientes preguntas que están enfocadas a distintas dimensiones de tu vida:

- **Dar:** ¿Cómo estoy apoyando a mis colaboradores, compañeros, líderes, proveedores y clientes?
- **Relaciones:** ¿Cómo puedo mejorar mis relaciones en el trabajo?, ¿cómo logro un balance entre la vida laboral y familiar?
- **Ejercicio:** ¿Cómo puedo integrar la actividad física dentro de mis actividades diarias?, ¿cómo me aseguro de que estoy comiendo bien y descansando lo suficiente?
- **Conciencia:** ¿Cómo puedo construir momentos de atención plena en mi día laboral?
- **Ensayo:** ¿Qué habilidades estoy construyendo?, ¿qué cosas nuevas he experimentado?
- **Dirección:** ¿Cuáles son mis metas laborales hoy, esta semana, este año?, ¿cómo caben y contribuyen estas con mis metas de vida y me ayudan a desarrollar mis competencias en la construcción de mis relaciones y cómo contribuyo con lo anterior a ayudar a otros?, ¿cómo se pueden alinear mis metas laborales con las de mi equipo y la organización?
- **Resiliencia:** ¿Cuáles son mis tácticas para lidiar con los retos difíciles en el trabajo?, ¿me estoy enfocando en lo que puedo controlar?, ¿necesito pedir ayuda a otros?, ¿hay alguien a mi alrededor que requiere de mi ayuda?
- **Emoción:** ¿Qué cosas, aunque sean pequeñas, puedo encontrar que me pueden hacer sentir bien en mi trabajo hoy?, ¿qué me ha hecho sonreír?

Fuente: Tomado del catálogo de actividades para profesores.

Actividad guiada

Parte 1

Duración: 75 minutos.

En este tema pondrás en práctica algunas de las nuevas características de JavaScript.

1. Crea una clase llamada **Pelicula**. La clase recibirá un objeto al momento de instanciarse con los siguientes datos:
 - ID de la película en IMDB
 - Título
 - Director
 - Año de estreno
 - País o países de origen
 - Géneros
 - Calificación en IMDB
2. Realiza las siguientes validaciones:
 - Todos los datos del objeto son obligatorios.
 - El ID IMDB debe tener nueve caracteres, los primeros dos son letras y los siete restantes son números.
 - El título no debe rebasar los 100 caracteres.
 - El nombre del director no debe rebasar los 50 caracteres.
 - El año del estreno debe ser un número entero de cuatro dígitos.
 - El país o países deben ser introducidos en forma de arreglo.
 - Los géneros deben ser introducidos en forma de arreglo.
 - Los géneros deben estar dentro de los géneros aceptados.
3. Crea un método estático que devuelva los géneros aceptados.
4. Valida que la calificación sea un número entre 0 y 10, pudiendo tener decimales de una posición.
5. Crea un método que devuelva toda la ficha técnica de la película.
6. A partir de un arreglo con la información de tres películas, genera tres instancias de la clase de forma automatizada e imprime la ficha técnica de cada película.

Actividad guiada

Parte 1

- Géneros aceptados: Acción, Adultos, Animación, Autobiografía, Aventura, Ciencia ficción, Cine negro, Comedia, Concursos de televisión, Crimen, Deportes, Documental, Drama, Familiar, Fantasía, Guerra, Historia, Musicales, Misterio, Noticias, Películas de vaqueros, Programa de entrevistas, Reality shows, Romance, Suspenso, Terror.

Para cumplir con los requerimientos anteriores, realiza lo siguiente:

1. Crea un documento html en el directorio que hayas destinado para tus archivos de la sesión guiada 3 del curso Programación con JavaScript II.
2. En ese mismo directorio, crea una carpeta en donde guardarás los archivos js.
3. En esta carpeta, crea el archivo **Pelicula.js**.
4. Enlaza el archivo js al archivo **index.html**.
5. Crea la clase **Pelicula**.
6. Dentro del cuerpo, crea un constructor que recibirá como parámetro un arreglo integrado por {id, titulo, director, estreno, pais, generos, calificacion}.
7. Para cada uno de los objetos recibidos, crea un atributo de la clase `this[parametro_recibido]`, que es igual a cada uno de los parámetros recibidos. Por ejemplo: **this.id=id**; y así sucesivamente para cada uno de los parámetros.
8. Se va a requerir validar alguna cadena. Para eso crea la función genérica **validarCadena** y pásale dos parámetros:
 - a. La propiedad que se va a evaluar y el valor en cuestión.
 - b. Valida lo siguiente en el cuerpo de la función:
 - i. Si el valor es nulo, imprime en la consola una alerta que indique ``${propiedad} "${valor}" está vacío``.
 - ii. Si el tipo de dato no es cadena, envía un error en la consola que diga ``${propiedad} "${valor}" ingresado no es una cadena de texto``.
 - iii. En caso de que pase las validaciones, retorna un valor **true**.

Actividad guiada

Parte 1

9. Para validar que el IMDB ID cumpla con la regla de formato (las dos primeras posiciones sean letras y los siguientes sean números), realiza lo siguiente:
 - a. Crea un método `validarIMDB` que reciba el parámetro `id`:
 - i. Utilizando la función **`validarCadena`**, valida el `id`:
 1. Si la validación regresa **`true`**, realiza una nueva validación en la que, a través de la siguiente expresión regular `(!/^([a-z]){2}([0-9]){7}$/.test(id))`, verifiques que la cadena tenga dos letras, seguidas de siete números. Si la validación no es correcta, envía el mensaje ``IMBD id ${id} no es válido, debe tener nueve caracteres, los dos primeros deben ser letras minúsculas y los siete restantes números``.
 - b. Mándalo a llamar dentro del constructor para que cuando se instancie, se realice la validación.
 - c. Valida el funcionamiento creando una constante **`pele`** en la que se instancie a la clase **`Pelicula`**.
10. Para validar la longitud de la cadena **`titulo`** y **`director`**, crea un método **`validarLongitudCadena`** que reciba como parámetros la propiedad `titulo` o nombre **`del director`**, el valor como tal y la **`longitud`** que debe tener la propiedad.
 - a. En el cuerpo del método, valida si la longitud del valor es mayor al parámetro `longitud`, en caso de ser verdadero, envía a la consola el error ``${propiedad} "${valor}" excede el número de caracteres permitidos (${longitud})``;
 - b. Retorna un valor **`true`**.

Consideraciones:

Aprendedor: Asegúrate de saber utilizar las ***template strings*** y las expresiones regulares para que puedas realizar la implementación correcta en tu código.

Instructor: Es probable que algunos aprendedores tengan dudas respecto a las funciones regulares, identifícalas y fortalece su conocimiento al respecto.

Receso

Duración: 10 minutos.

Programación con JavaScript II

Actividad guiada

Parte 2

Duración: 75 minutos.

1. Crea el método **ValidarTitulo** que reciba en el parámetro la variable titulo. En el cuerpo del método, valida el título utilizando el método **validarCadena** con los parámetros ("Título", titulo). Si la validación se cumple, crea una nueva validación utilizando el método **validarLongitudCadena** que reciba los parámetros ("Título", titulo, 100). Si la validación se cumple, entonces regresa el valor true, en caso contrario, muestra el error en la consola.
2. Crea el método **validarDirector** que recibirá como parámetro la variable director utilizando los mismos métodos, solamente que **validarCadena** recibirá como parámetros ("Director", director) y **validarLongitudCadena** recibirá ("Director", director, 50).
3. Ambos métodos deberán ser invocados desde el constructor.
4. Para validar que el año de publicación y la calificación sean números, crea la función **validarNumero** que recibirá como parámetros la propiedad y el valor.
 - a. En el cuerpo de la función, verifica:
 - i. Que el valor no esté vacío. En caso de que esté vacío, imprime un warning en la consola ``${propiedad} "${valor}" está vacío``.
 - ii. Que el tipo de dato sea numérico, en caso contrario, imprime en la consola el error ``${propiedad} "${valor}" ingresado no es un número``.
 - iii. Si todo es correcto, el método retorna true.
5. Para validar la longitud del año, crea el método **validarEstreno** y recibe como parámetro la variable estreno.
 - a. En el cuerpo deberás utilizar el método **validarNumero** con los parámetros ("Año de estreno", estreno). Si se cumple la condición, valida con una expresión regular que solo se acepten números del 0 al 9 y que la longitud sea cuatro.
 - b. Si la condición no se cumple, imprime un error en la consola con el mensaje ``Año de estreno "${estreno}" no es válido, debe ser un número de 4 dígitos``.
 - c. Invoca el método desde el constructor.
6. Para validar que el país o países y el género pasen como arreglo, crea un método **validarArreglo** que reciba como parámetros **propiedad y valor**.
 - a. Valida que el valor no esté vacío, si no es así, envía un warning a la consola.
 - b. Valida que el valor pasado como parámetro sea del tipo Array, en caso contrario, envía un mensaje de error a la consola.
 - c. Valida que la longitud del parámetro valor no sea 0, en caso de que así sea, envía un mensaje de error a la consola.
 - d. Itera el arreglo y valida que los elementos del arreglo sean cadenas, si no es el caso, envía un mensaje de error a la consola.
 - e. Si las validaciones se ejecutan correctamente, retorna **true**.
7. Para validar el país, crea el método **validarPais** que recibe el parámetro **pais** y en el cuerpo del método solamente invoca la función **validarArreglo** con los parámetros ("Pais", pais) y manda a llamar este método desde el constructor.

Actividad guiada

Parte 2

8. Para crear el método estático que devuelva los géneros aceptados:
 - a. Declara el “obtenedor” **listaGeneros** de tipo estático que en su cuerpo debe tener el arreglo de los géneros.
 - b. Declara el método estático **generosAceptados**, consumirá el atributo estático y retornará en la consola un info con el mensaje ` Los géneros aceptados son \${Pelicula.listaGeneros.join(",")} `.
 - c. Para ejecutar este método, solo invócalo de la forma **Clase.metodo()**.
9. Para validar los géneros, crea el método **validarGeneros** que recibe como parámetro la **variable generos**. En el cuerpo del método va a realizar lo siguiente:
 - a. Validar que los géneros sean un arreglo, utilizando la función **validarArreglo** con los parámetros (“Generos”, generos).
 - b. Si la validación se cumple, se debe iterar el arreglo **generos** con un **for/of**.
 - i. Dentro del cuerpo del for/of, verifica que el elemento iterado se encuentre dentro del arreglo listaGeneros utilizando el método **includes** de los arreglos. En caso de que no se cumpla la condición, imprime un error en la consola indicando que los arreglos son incorrectos, seguido de los arreglos en forma de cadena.
 - ii. Después del mensaje, imprime la lista de géneros aceptados. Recuerda que los **getters** estáticos no se invocan con this, sino que se deben invocar como **Pelicula.listaGeneros()**.
10. Para validar que la calificación sea un número, inclusive con decimales, crea un método validaCalificacion y recibe como parámetro la variable **calificacion**. Dentro del cuerpo de la función, valida lo siguiente:
 - a. Que el parámetro **calificacion** sea un número, utilizando el método **validarNumero**.
 - b. Si la calificacion es menor a 0 o es mayor a 10, envía a la consola un mensaje de error indicando que la **calificacion** tiene que estar en el rango de 0 a 10. En caso contrario, fija que la variable **calificacion** solo pueda aceptar hasta 1 decimal.
 - c. Invoca el método desde el constructor.
11. Para revolver toda la ficha técnica de la película, crea el método **fichaTecnica** y envía a la consola con un info todas las variables.
12. Invoca este método de la forma **pelicula.fichaTecnica()**.
13. Para generar las tres instancias de manera automatizada, crea un arreglo con tres películas (cada una de las películas es un objeto que contiene **id**, **titulo**, **director**, **estreno**, **pais**, **generos**, **calificacion**).
 - a. Utiliza el método **forEach**, el cual recibirá como parámetro una función flecha que crea una instancia de la clase **Pelicula** y que invoca al método **fichaTecnica**.

Actividad guiada

Parte 2

Consideraciones:

Instructor: Refuerza con los alumnos el uso de cada uno de los conceptos aprendidos en el tema.

Aprendedor: Asegúrate de saber utilizar las template strings y las expresiones regulares para que puedas realizar la implementación correcta en tu código.

Cierre

Duración: 10 minutos.

En este ejercicio pudiste poner en práctica algunos conceptos ya aprendidos en otras secciones, pero es importante resaltar el uso del scope en la clase desarrollada y en cada uno de sus métodos. También se implementó el uso de this, utilizándolo en al menos dos de las formas más comunes. ¿Las pudiste identificar?

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.