

Programación con Java Script I

Sesión sincrónica 8

# Manipulación del DOM



# Bienvenida y actividad de bienestar

**Duración: 10 minutos.**

**Nombre de la práctica:** Tus fortalezas en los ojos del otro

**Descripción de la práctica:** En la práctica podrás reflexionar sobre la percepción que otros tienen sobre tus fortalezas de carácter.

**Palabras clave:** Fortalezas de carácter

**Instrucciones para el participante:** ¿Recuerdas alguna ocasión en la que hablaste con algún colega y éste te reveló algo positivo que piensa de ti?

Cuando esto ocurre usualmente deja huella en nuestros comportamientos y acciones pues nos damos cuenta de que las personas tienen percepciones sobre nuestras fortalezas que nosotros mismos no vislumbramos. Haz lo siguiente:

1. Piensa sobre alguna vez que algún compañero de trabajo te compartió lo que piensa de ti y que te haya sorprendido.
2. Piensa, ¿qué fue lo que te llamó más la atención?, ¿qué fortalezas vio en ti que pensaste no tenías tan desarrolladas?
3. Por último, señala en un texto por qué consideras que esta revelación te causó tanto impacto, así como la manera en que te ayudó a cultivar tus fortalezas de carácter.

Fuente Niemic, R. (2016). How to Assess Your Strengths: 5 Tactics for Self-Growth. Recuperado de <https://www.psychologytoday.com/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth?platform=hootsuit>

# Actividad guiada

## Parte 1

**Duración: 75 minutos.**

Existen muchas formas de propiciar que las personas que te rodean experimenten emociones positivas y, aunque no lo creas, esto lo puedes lograr a través de actividades que realizas cotidianamente. Por ejemplo, tomando fotografías y compartiéndolas.

Haz lo siguiente:

1. Diariamente toma una o dos fotografías de instantes, cosas o personas, que te hayan hecho feliz ese día.
2. Repite esto durante una semana.
3. Luego que termine la semana, haz un collage con las fotografías que tomaste.
4. Analiza las fotografías y reflexiona sobre todos los momentos que te han dado felicidad:
  - a. Identifica qué es lo que te hace feliz usualmente.
  - b. Reflexiona si fueron momentos o personas.
5. A manera de cierre, reflexiona ¿de qué te sirve conocer y establecer claramente qué te hace feliz?



## Actividad guiada

### Parte 1

En el siguiente ejercicio vas a practicar los conceptos de eventos aprendidos en esta experiencia educativa, a través de la creación de un par de secciones en la aplicación que tiene un menú de hamburguesa con cinco secciones, que serán accedidas desde el menú.

En la sección 2, crearás la funcionalidad necesaria para poder mover una pelota con el teclado. Para alcanzar nuestro objetivo, realiza las siguientes actividades:

1. En el archivo index.html, modifica el contenido del link que apunta a la #seccion 2, asignándole la cadena Eventos del teclado.
2. En la sección con id seccion2:
  - a. Modifica el contenido de la etiqueta h2 y colócale Eventos del teclado.
3. En la carpeta js, crea el archivo con el nombre teclado.js y haz lo siguiente:
  - a. Asigna a la constante d la clase document.
  - b. Exporta la función shortcuts que va a recibir el evento representado con la letra e.
  - c. Imprime en la consola el tipo de evento recibido.
  - d. Imprime en la consola también el evento recibido.
4. En el archivo index\_dom.js, realiza:
  - a. La importación de la función shortcuts desde el archivo "ruta\_del\_archivo/teclado.js".
  - b. Para desencadenar el evento, utiliza addEventListener y pásale como parámetros el tipo de evento del teclado que se está escuchando ("keydown") y, como segundo parámetro, envíale una función flecha en la que invoques a la función shortcuts que recibe como parámetro al evento.
5. Para probar esta funcionalidad solo es necesario ejecutar la página, hacer clic en cualquier parte de esta y posteriormente oprimir cualquier tecla. En la consola debe informarse un objeto del tipo KeyboardEvent con la información del evento.
6. Desencadena otras propiedades en la función shortcuts para que puedas analizar su comportamiento (key, keyCode, ctrlKey, altKey, shiftKey).
7. También puedes probar con otros eventos como keyup o keypress.
8. Una vez que hayas comprobado la funcionalidad de los eventos y los comportamientos, crea una secuencia shortcut para que, si el usuario oprime esas teclas, se ejecute un alert:
  - a. Valida si las teclas presionadas son la "a" y la tecla "alt".
  - b. En el cuerpo de la validación, crea una alerta con el texto "Has lanzado una alerta con el teclado".
9. Crea otra validación para que valide con la tecla "c" y "alt" y crea un confirm con el texto "Has lanzado una confirmación con el teclado".
10. Crea otra validación con la tecla "p" y "alt" y en el cuerpo de la validación crea un prompt con el texto "Has lanzado un aviso con el teclado".

## Actividad guiada

### Parte 1

Ahora que conoces más acerca de los eventos del teclado, crea el escenario en donde se va a mover la pelota que vamos a controlar:

1. En el archivo index.html:
  - a. En la sección con el id seccion2, agrega la etiqueta article con la clase stage.
  - b. En el cuerpo de article, crea un div con la clase ball.
2. En el archivo css, crea lo siguiente:
  - a. Crea un comentario que diga eventos del teclado.
  - b. A la clase stage asigna:
    - i. Display con el valor flex.
    - ii. Una altura mínima de la mitad del view port.
    - iii. Como color de fondo el valor que se tiene en la variable --second-color.
  - c. A la clase ball:
    - i. Margen automático.
    - ii. Altura de 4rem.
    - iii. Alto de 4rem.
    - iv. Un redondeado de esquinas del 50%.
    - v. Como color de fondo el valor que se encuentra en la variable --main-color.
3. En el archivo teclado.js:
  - a. Crea las variables "x" y "y", las cuales deben estar inicializadas en 0.
  - b. Exporta la función moveBall:
    - i. Pásale como parámetros el evento, el selector de la pelota y el selector del escenario.
    - ii. Crea variables para el DOM, tanto la pelota (\$ball), y asígnale al querySelector el valor del selector que está guardado en la variable ball.
    - iii. Realiza lo mismo para el escenario (\$stage), pero utilizando la variable stage.
    - iv. Crea una variable limitsBall para establecer los límites de la pelota con la función getBoundingClientRect() de la variable \$ball.
    - v. Crea una variable limitsStage para establecer los límites del stage con la función getBoundingClientRect() de la variable \$stage.

## Actividad guiada

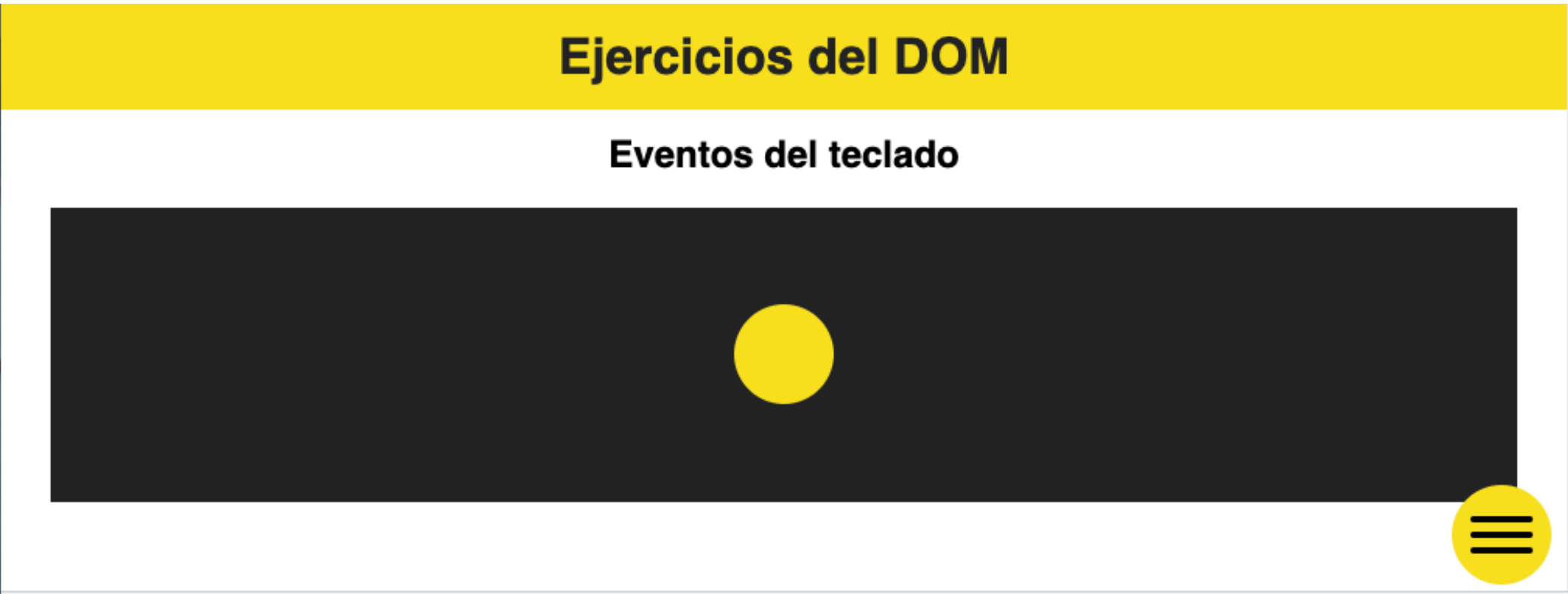
### Parte 1

- c. Con un switch case, evalúa el contenido del keyCode en el evento:
  - i. Cuando el keyCode vale 37, valida si el límite izquierdo de la pelota es mayor que el límite izquierdo del escenario, en ese caso:
    - 1. Deshabilita el comportamiento del evento.
    - 2. Resta 1 a la variable "x".
  - ii. Cuando el keyCode vale 38, valida si el límite superior de la pelota es mayor que el límite superior del escenario, en ese caso:
    - 1. Deshabilita el comportamiento del evento.
    - 2. Resta 1 a la variable "y".
  - iii. Cuando el keyCode vale 39, valida si el límite derecho de la pelota es menor que el límite derecho del escenario, en ese caso:
    - 1. Deshabilita el comportamiento del evento.
    - 2. Suma 1 a la variable "x".
  - iv. Cuando el keyCode vale 40, valida si el límite inferior de la pelota es mayor que el límite derecho del escenario, en ese caso:
    - 1. Deshabilita el comportamiento del evento.
    - 2. Resta 1 a la variable "y".
  - v. En otro caso no debe hacer nada.
- 4. En el archivo index\_dom.js:
  - a. Importa la función moveBall.
  - b. Ejecuta la función moveball en el addEventListener keydown y pásale como parámetros el evento, el selector de la pelota y el selector del escenario.

# Actividad guiada

## Parte 1

Al final de tu práctica, tu página debe lucir así:



## 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, self.damage)
    damage_bot_limit = self.damage - random.randint(0, self.damage)
    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.

Ahora vamos a trabajar con la sección 3 de la página del menú hamburguesa, en donde vamos a crear una cuenta regresiva.

1. En el archivo html:
  - a. Modifica en el menú el contenido link con el id **#seccion3** y colócale cuenta Regresiva.
  - b. Modifica el título de la sección con id **seccion3** y colócale Cuenta regresiva.
  - c. Crea una div vacía con el id **countdown**.
2. Crea el archivo **cuenta\_regresiva.js** en la subcarpeta dom de la carpeta js y realiza lo siguiente:
  - a. Declara una constante y asígnale la clase document.
  - b. Para crear la cuenta regresiva, exporta por default la función:
    - i. Nómbrala countdown.
    - ii. Pásale como parámetros el id, la fecha límite (limitDate) y el mensaje final (finalMessage).
    - iii. Declara dos variables del DOM:
      1. \$countdown, que va a obtener lo que viene en el primer parámetro (id) mediante el metodo getElementById.
      2. CountdownDate es igual a un nuevo objeto Date que recibe como parámetro la fecha límite y es necesario expresarla en milisegundos con el método getTime().
3. Crea una variable contDownTempo que va a ser igual a un setInterval, en la que:
  - a. El intervalo de actualización será cada segundo.
  - b. Existirá una variable now en la que se va a obtener la fecha de ahora.
  - c. Existirá una variable limitTime que va a ser igual a la resta de la fecha solicitada menos la fecha actual.
  - d. También existirán variables (*days, hours, minutes, seconds*) para la conversión del valor de la variable limitTime a esas unidades de tiempo. Para ello, es necesario obtener un factor de conversión de milisegundos a la unidad deseada, redondear el resultado y hacer los ajustes estéticos necesarios, por ejemplo, expresar las horas, minutos y segundos con dos dígitos siempre y los que consideres necesarios.

## Actividad guiada

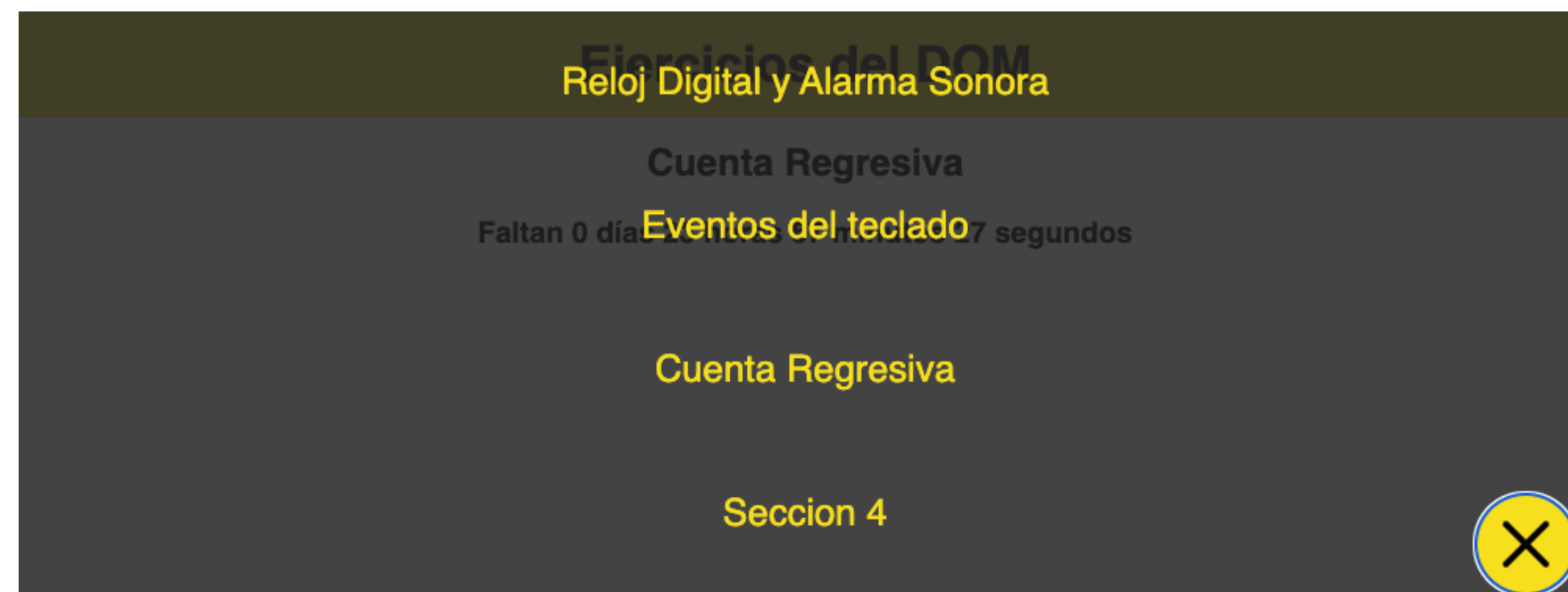
### Parte 2

- e. Además, codifica una expresión en la que se indicará que el contenido HTML de `$countDown` tendrá el siguiente mensaje (creado con template strings) `<h3>Faltan ${days} días ${hours} horas${minutes} minutos ${seconds} segundos</h3>`.
- f. Para presentar el mensaje, valida que la variable `limitTime` sea menor a cero:
  - i. Limpia el contenido de la variable `countDownTempo`.
  - ii. Asigna al contenido HTML de la variable `$countDown` el mensaje final en un `h3` y creado con template strings.

#### 3. En el archivo **index\_dom.js**:

- a. Importa la variable `countDown` desde **“ruta\_del\_archivo/cuenta\_regresiva.js”**.
- b. En el `DOMContentLoaded`, invoca a la función con sus tres parámetros:
  - i. El nombre del selector (“`countdown`”).
  - ii. Fecha válida que acepte JavaScript. Ejemplo “Oct 09, 2021 00:00:01”.
  - iii. El mensaje que se va a desplegar. Ejemplo: “Feliz Cumpleaños!!”.

Al final de tu práctica, tu archivo debe verse así:



#### Consideraciones

Instructor, refuerza los conocimientos de aquellos temas que consideres necesarios para que el aprendedor pueda realizar este ejercicio.

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.