



Universidad  
**Tecmilenio**®



Te invitamos a que realices el siguiente ejercicio mental, el cual tomará cinco minutos y servirá para obtener una mejor claridad en los conceptos que aprenderemos el día de hoy.

Mindfulness con gratitud.

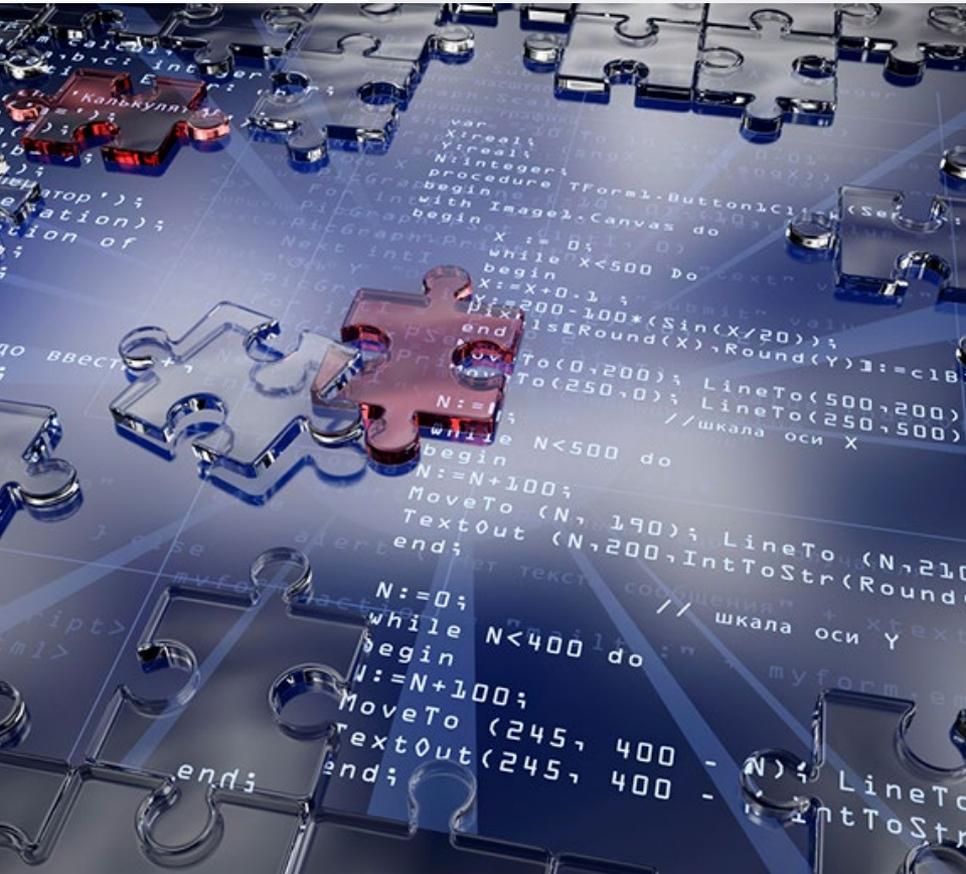
Liga: <https://youtu.be/qSfjmeM65As>



# Interactividad y diseño multimedia

## 9. Estructura y navegación en páginas HTML5

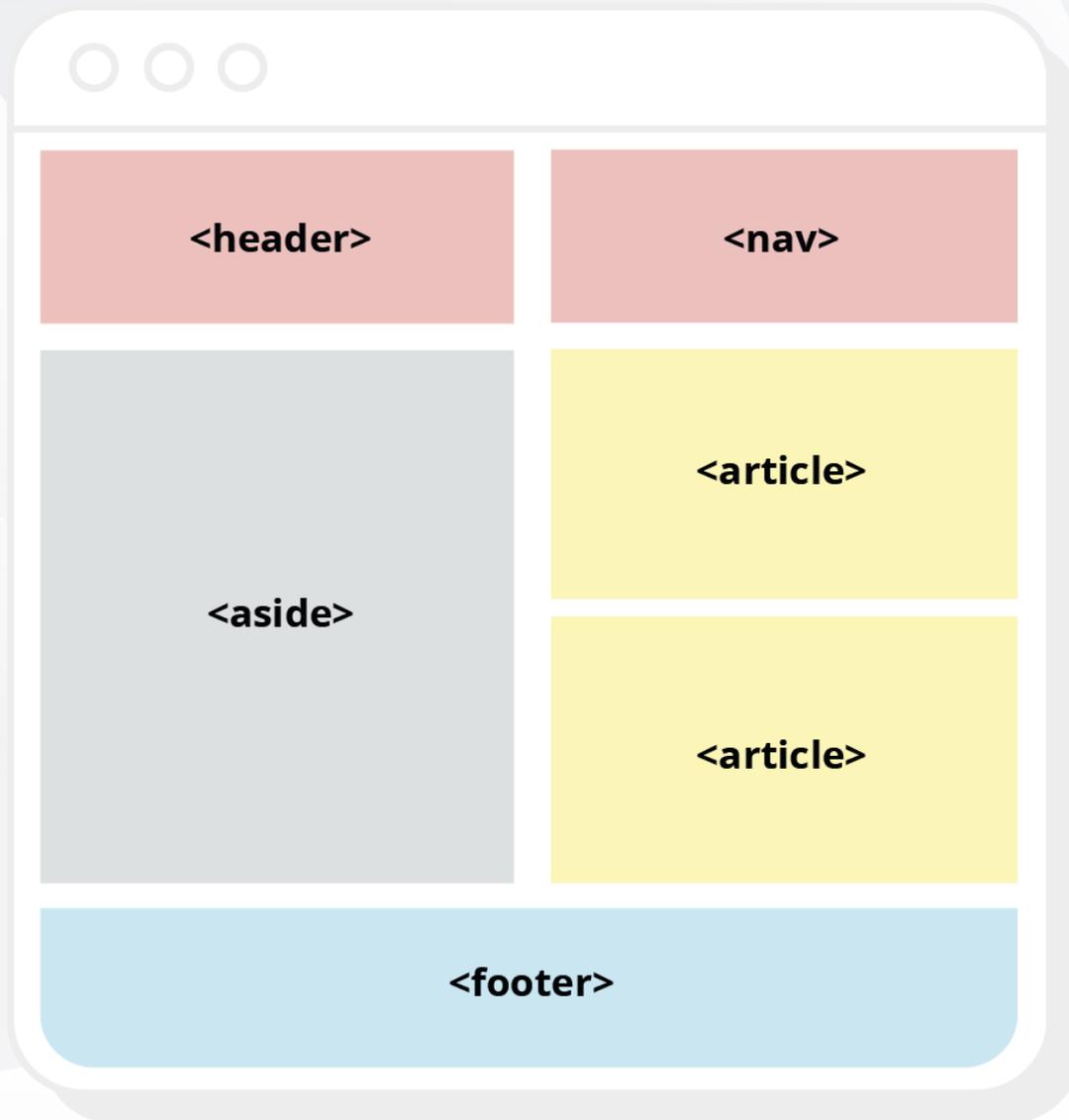




HTML5 incluye nuevos elementos que te permiten marcar su contenido y presentar una mejor estructura para tus documentos en comparación con las versiones anteriores de HTML.

Una de las tareas más comunes en una página es identificar diferentes áreas del documento; la barra de navegación, el encabezado, el pie de página, entre otros.





## Estructura de documentos en HTML5

**<header>**

Define una región de encabezado de una página o sección.

**<footer>**

Define una región de pie de página de una página o sección.

**<nav>**

Define una región de navegación de una página o sección.

**<section>**

Define una región lógica de una página o una agrupación de contenido.

**<aside>**

Contiene información relacionada con el contenido circundante.

**<article>**

Se usa para contenido independiente que se puede consumir independientemente, como una entrada de blog.



**01**

Modificar el contorno  
del documento

**02**

Marcado de fechas y  
horarios

**03**

Resaltar texto para  
referencia

**04**

Marcado de letra  
pequeña

**05**

Marcado de figuras  
y leyendas



- Reflexiona y responde lo siguiente.
  - a) ¿Para qué sirve el encabezado en HTML 5?
  - b) ¿Para qué sirve el footer en HTML 5?
  - c) Menciona un ejemplo de cada uno.





Usar los elementos estructurales de HTML5 te ayuda a hacer que tu marcado sea más semántico, pero también ayuda a definir los principales hitos en el documento.

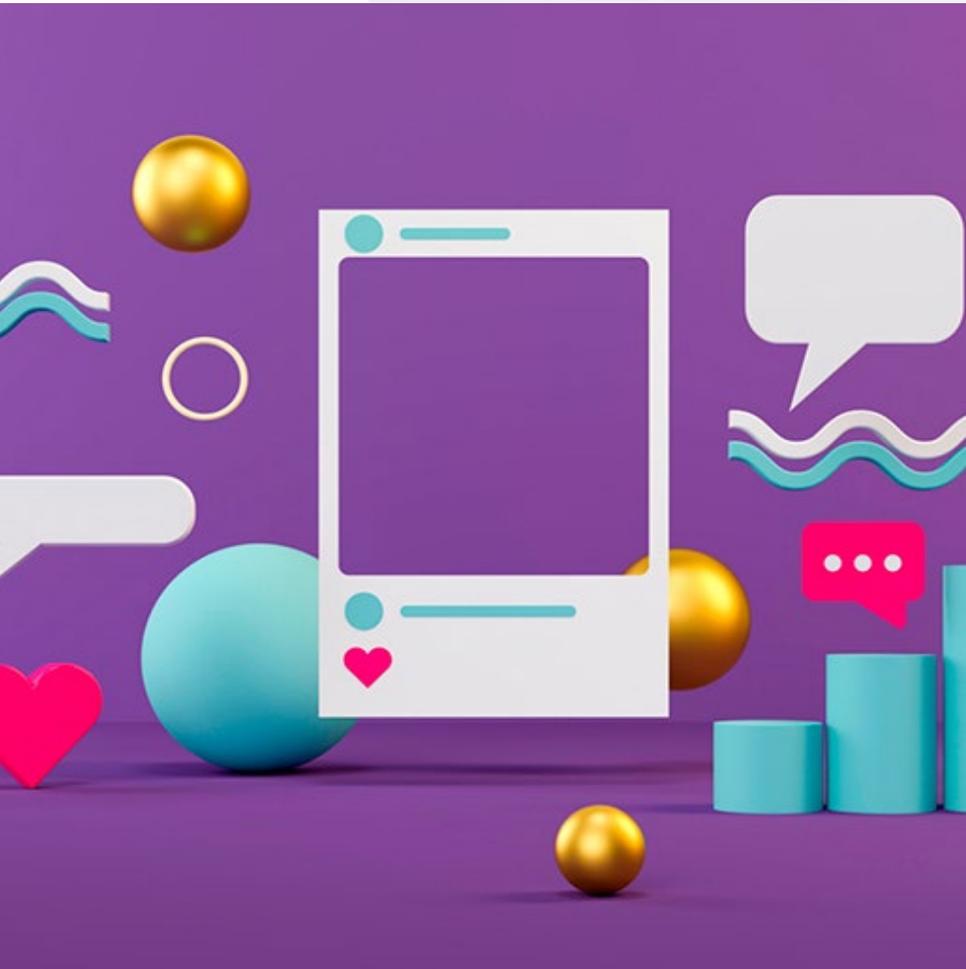
Considera qué tan importante es para los usuarios de lectores de pantalla y las personas que navegan con el teclado para poder saltar a diferentes áreas de la página como la navegación.



# Interactividad y diseño multimedia

## 10. Creando gráficos con HTML5





Los gráficos interactivos de alta resolución son una parte clave de la mayoría de las aplicaciones modernas. Los gráficos pueden ayudar a mejorar la experiencia del usuario al proporcionar un aspecto visual al contenido, haciendo que el sitio web sea más atractivo y fácil de usar. La interactividad permite que los elementos gráficos de un sitio web se adapten y respondan a los comentarios del usuario o los cambios en el entorno y es otro elemento importante para mantener la atención del usuario y su interés en el contenido.



## SVG

- Esta tecnología usa una forma de notación basada en XML.
- Está estrechamente relacionado con HTML.
- Permite agregar contenido gráfico interactivo en tus páginas web.
- Comprende un conjunto de elementos que encierra en un elemento `<svg>`.



## SVG en una página

### Ejemplo de un rectángulo

```
<svg width="200" height="200">  
<rect x="10" y="20" width="100" height="80" stroke="red" fill="#ccc" />  
</svg>
```



## Rutas SVG

"d" significa datos

"M" significa move to

"L" significa line to

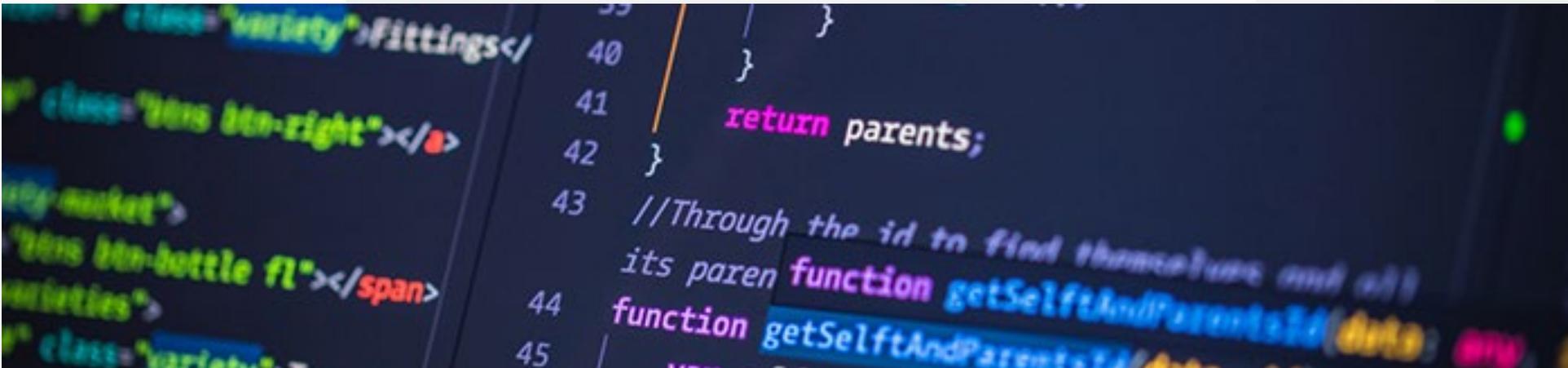
"Q" significa curva cuadrática

"Z" significa cerrar la ruta



- Reflexiona y responde lo siguiente.
  - a) ¿Qué son los elementos SVG?
  - b) ¿Qué beneficios brinda usar SVG en HTML?





En este tema has visto cómo SVG en HTML5 proporciona una forma poderosa de crear aplicaciones con gráficos bidimensionales interactivos.

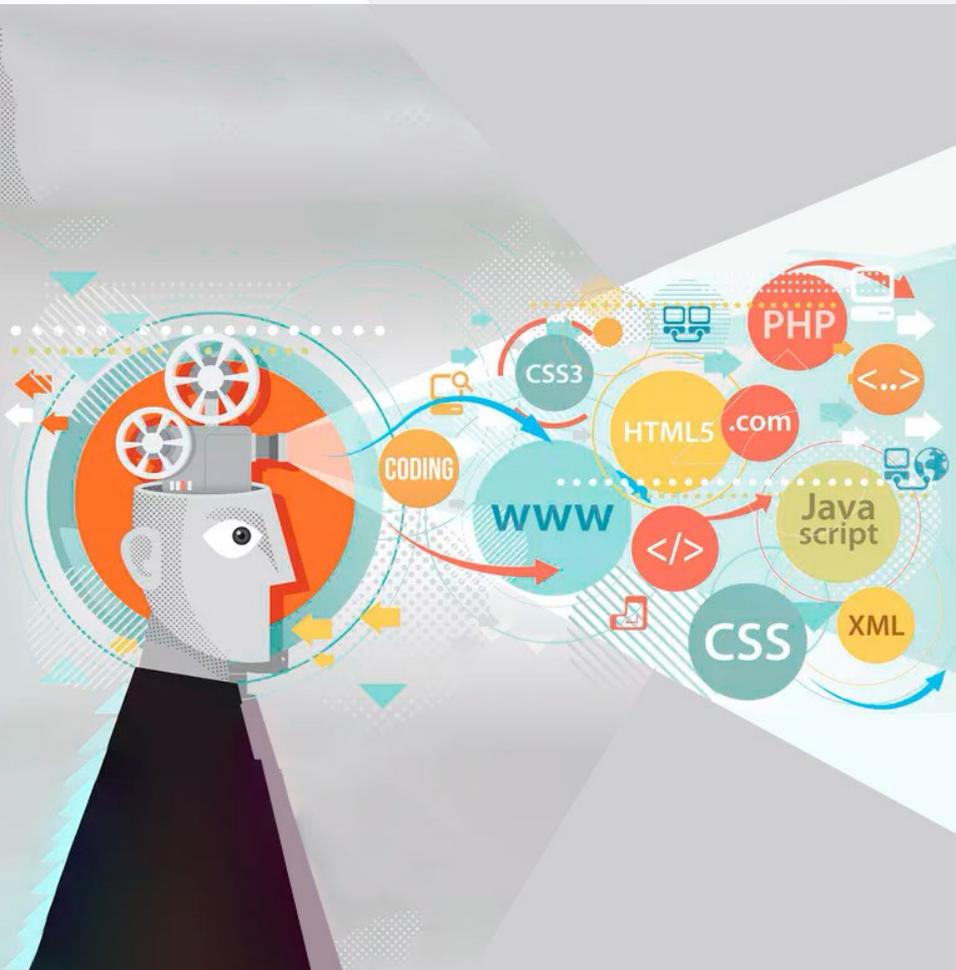
Ya sabes cómo crear una escena dibujada usando SVG incrustado en un documento HTML5. Ahora puedes examinar los elementos y atributos que forman un dibujo y puedes definir y reutilizar definiciones de contenido, agrupar y transformar elementos, y dibujar con formas, rutas y texto.



# Interactividad y diseño multimedia

## 11. Javascript avanzado





Aunque está bien leer páginas HTML estáticas de texto y aún mejor cuando se presentan con la ayuda de CSS, es mucho más divertido interactuar con las aplicaciones en nuestros navegadores como el correo electrónico, los calendarios, la banca, las compras, los dibujos, jugando juegos y edición de texto.

Todo eso es posible gracias a JavaScript, el lenguaje de programación de la web. JavaScript comenzó con líneas simples integradas en HTML, pero ahora se usa de formas mucho más sofisticadas.

Los desarrolladores aprovechan la naturaleza orientada a objetos del lenguaje para construir arquitecturas de código escalables compuestas de piezas reutilizables.



## Programación orientada a objetos

### Objetos JavaScript

---

- Si deseas dar más funcionalidad a un objeto existente con un nuevo método , usa el prototipo de objeto para extender el objeto.
- Una vez que un objeto se ha extendido a través de la propiedad del prototipo, todas las instancias del objeto dentro del alcance de la aplicación tienen esta funcionalidad.



## Writable

Si es verdadero, la propiedad puede ser cambiada; de otra forma no.



## Configurable

Si es verdadero, la propiedad se puede eliminar o cambiar; de otra forma no.



## Enumerable

Si es verdadero, la propiedad se puede iterar.



## Enumeración de las propiedades de un objeto

Supón que deseas ver qué propiedades tiene un objeto. Usa una variación especializada del ciclo for para iterar a través de las propiedades.

### Ejemplo

```
for (var prop in obj) {  
  alert(prop); // imprime el  
  nombre de la propiedad  
}
```

## Objetos y nombres únicos espaciados de JavaScript

Supón que deseas encapsular la funcionalidad de tu biblioteca, de tal forma que evite conflictos con otras bibliotecas.

Usa un objeto literal, lo que se llama un objeto único, para implementar la versión de JavaScript del espacio de nombres.



- Reflexiona y responde lo siguiente.
  - a) ¿Para qué sirve la programación orientada a objetos?
  - b) ¿Qué son los atributos?





En este tema aprendiste los fundamentos de la programación orientada a objetos, la cual, en el caso de JavaScript, está basada en prototipos.

Esto te permitirá comprender algunos temas avanzados de HTML5, los cuales se estudiarán más adelante.



# Interactividad y diseño multimedia

## 12. Interfases de usuario adaptables





Los diseñadores web siempre han tenido que lidiar con el diseño de varios tamaños de pantalla, desde computadoras portátiles de 760 píxeles hasta pantallas gigantes de pantalla ancha.

Afortunadamente, existe una manera que te permite construir un único sitio que se adapta a diferentes anchos de dispositivo; el diseño Web receptivo. Esta técnica utiliza varios trucos diferentes para hacer que una página cambie su diseño en función del ancho de la pantalla del navegador. En un teléfono inteligente, por ejemplo, puedes diseñar una página en una columna sencilla y fácil de leer para el ancho de la pantalla, mientras mantienes una distribución de múltiples columnas en la parte más ancha del monitor.



## Screen

---

Esta palabra clave es uno de los tipos de medios disponibles y se usa para indicar qué tipo de medios deben implementar las declaraciones adjuntas.

## Max-width

---

Esta es una de las muchas palabras clave condicionales que podemos encadenar para enumerar las características que deben estar presentes en un dispositivo para activar las declaraciones adjuntas.



## Min-width

Esta palabra clave se puede leer como "el ancho de la ventana es mínimo x", donde x es el valor asignado a la propiedad "min-width".

## Max-height y min-height

Estas palabras clave funcionan de la misma manera que \*-width, pero se aplican a la altura de la ventana del navegador.

## Min-device-width, max-device-width, min-device-height y max-device-height

Estas palabras clave identifican las dimensiones reales del dispositivo; así que si queremos apuntar solo a pantallas de más de 1900 x 1200 tenemos que escribir una regla como (min-device-width: 1900 px) y (min-device-height: 1200 px).

## Orientación

El valor de esta propiedad puede ser vertical u horizontal. Identifica la orientación actual del dispositivo.

## Content-box

En esta opción, el ancho se refiere solo al recuadro que rodea el contenido del elemento. Los anchos de relleno, margen y borde están excluidos.

## Padding-box

Esta opción es la misma que la anterior, pero este ancho de tiempo incluye el relleno. Los anchos de bordes y márgenes aún están excluidos.

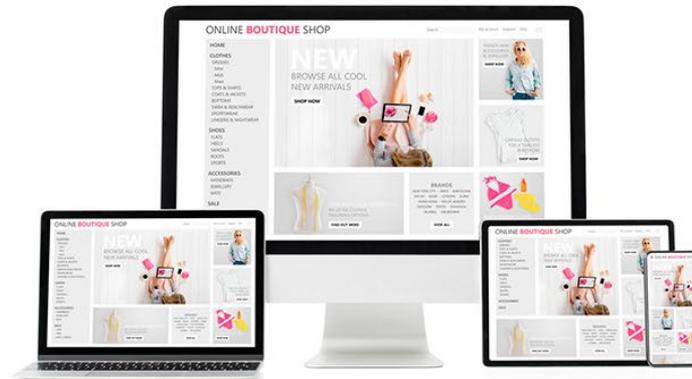
## Border-box

Esta opción es la misma que las dos anteriores, pero esta vez solo el ancho del margen está excluido.



- Reflexiona y responde lo siguiente.
  - a) Menciona una ventaja del uso responsivo para páginas web.
  - b) Menciona dos ejemplos de box-sizing.





En este tema aprendiste algo muy importante sobre el diseño web responsivo. También descubriste las diferencias entre la pantalla; bloque en línea y elementos flotantes.

Con este nuevo conocimiento puedes hacer que tu proyecto final pueda adaptarse a diferentes tipos de dispositivos.

