



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.

Explorando tus emociones.

Liga: <https://youtu.be/zmGogSbH5FQ>



# Interactividad y diseño multimedia

## 13. Layout avanzado y uso de frameworks





En esta sección exploramos las opciones que tenemos para posicionar objetos, los distintos tipos de pantallas y resoluciones que usan diferentes dispositivos, ver la diferencia entre tipos layouts, cómo utilizar grids para diseño y finalmente cómo usar Bootstrap como base de proyectos.



## Normal flow

Aparecen en una nueva línea cada uno, empujando cada nuevo elemento hacia abajo. Incluso si especificamos el ancho de las cajas y existe espacio para que dos

## Relative position

Mueve al elemento de la posición que tendría en flujo normal, cambiando a la parte superior, derecha, inferior o izquierda de donde habría sido colocado.

## Absolute position

Coloca al elemento en relación a su elemento contenedor. Se saca del flujo normal, lo que significa que no afecta a la posición de los elementos que lo rodean.

## Fixed position

Esta es una forma de posicionamiento absoluto que coloca al elemento en relación con la ventana del navegador. Los elementos con posicionamiento fijo no afectan la ubicación de los elementos de los alrededores.

## Float

Flotar un elemento permite colocarlo fuera del flujo normal y ubicarlo a la derecha o izquierda de la caja que lo contiene. El elemento flotante se convierte en un elemento tipo block que facilita el flujo de otro contenido.



## Fixed width layout

### VENTAJAS

- ✓ Los valores en píxeles son exactos para controlar el tamaño y la colocación de elementos.
- ✓ Existe mayor control sobre el aspecto y la posición de los elementos de la página que son layouts líquidos.

### DESVENTAJAS

- ✗ Si no se controla el ancho de las secciones de la página, el diseño puede ser muy diferente de lo previsto, mostrando espacios inesperados alrededor de ciertos elementos o artículos.
- ✗ Si el usuario tiene una pantalla muy grande, las líneas de texto pueden llegar a ser muy largas y el espacio entre elementos puede dejar grandes lagunas.



## Liquid layout

### VENTAJAS

- ✓ Las páginas se expanden en toda la ventana del navegador, de manera que no deja espacio alrededor de la página en una pantalla grande.
- ✓ Cuando se mira en una pantalla pequeña, la página se contrae para adaptarse a ella sin que el usuario tenga que desplazarse a los lados.

### DESVENTAJAS

- ✗ Si la pantalla del usuario tiene una resolución mucho más alta que nuestra pantalla de desarrollo, la página puede parecer más pequeña y el texto puede ser más difícil de leer.
- ✗ Si un usuario aumenta el tamaño de las fuentes, el texto podría salirse de los espacios asignados.



- Reflexiona sobre los diversos tipos de layouts y responde a lo siguiente.
  - a) ¿Para qué sirve un layout?
  - b) Realiza un ejemplo de un layout sencillo para una página web.





En este tema se te han brindado las herramientas para que practiques y experimentes con diversos layouts. Ya conoces cómo posicionar objetos para que tu layout esté diseñado correctamente y sea funcional.

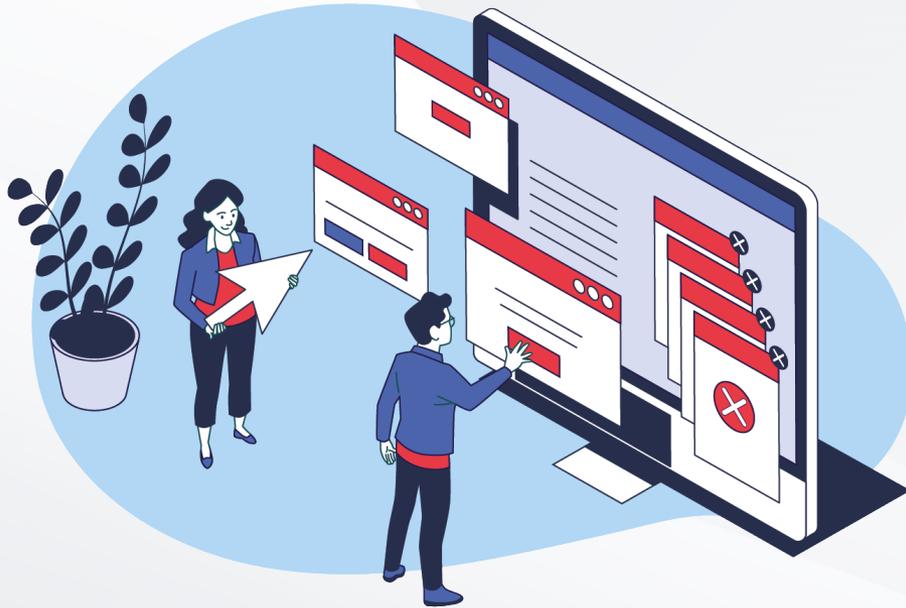
También aprendiste las ventajas y desventajas de los tipos de layout, fixed width y liquid.



# Interactividad y diseño multimedia

## 14. Básicos de Javascript





En este tema verás los aspectos básicos de Javascript, es probable que ya conozcas algo al respecto, pero no está de más volver a practicarlo para incorporarlo a tus páginas dinámicas.

Javascript ofrece efectos que no son posibles de otra manera, ya que se ejecuta dentro del navegador y tiene acceso directo a todos los elementos en un documento web.

Cuando buscas lanzar un pop-up al pasar el mouse sobre un elemento de una página web o cuando quieres que textos, colores o imágenes aparezcan en la página, así como agarrar un objeto y arrastrarlo a una nueva posición, debes recurrir a Javascript. Esta herramienta da esa funcionalidad extra en los sitios web.



## JAVASCRIPT

Lenguaje de script que se ejecuta en su totalidad dentro del navegador web.

01

Para llamarlo se coloca entre los tags `<script>` `</script>`.

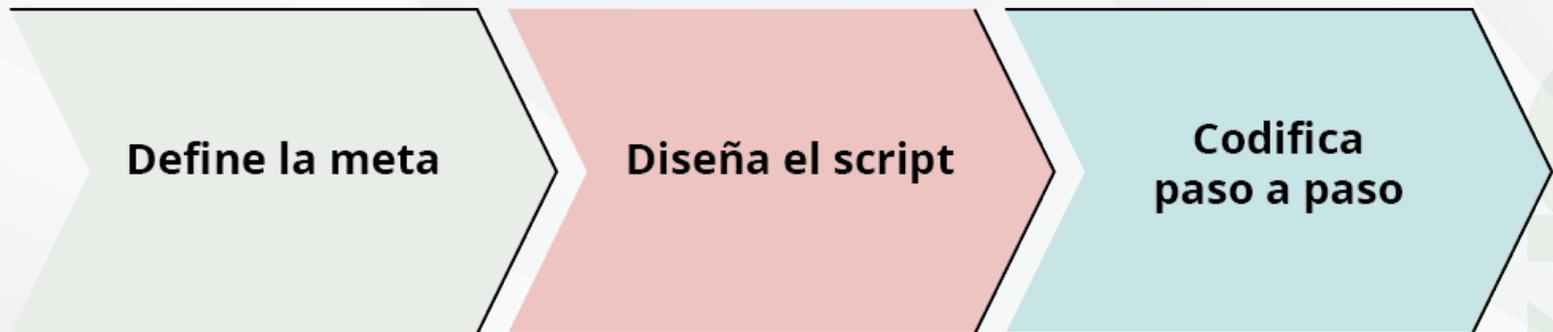
02

Permite crear sitios web más interactivos al acceder y modificar los elementos HTML.

03



## Pasos básicos para escribir un script



## Objetos y propiedades

---

En la programación a cada cosa que existe en el mundo se le considera un objeto.

### Ejemplo

La casa es un objeto.

## Propiedades

---

Son las características del objeto y tienen nombre y valor.

### Ejemplo

Las propiedades de la casa son las siguientes.

- a. Número de cuartos: 4
- b. Número de focos: 10



- Reflexiona y responde a lo siguiente.
  - a) ¿Qué es JavaScript?
  - b) ¿Qué hace el JavaScript?
  - c) ¿Cuál es el proceso para escribir un script?





En este tema viste para qué sirve Javascript, dónde puedes usarlo y cómo invocarlo desde una fuente externa. De la misma forma revisaste los elementos básicos de este lenguaje de programación que te será de gran utilidad para el desarrollo de tu sitio web dinámico.



# Interactividad y diseño multimedia

## 15. Javascript intermedio





Ya conoces los conceptos básicos de Javascript y con esa leve visión ahora puedes imaginar todo lo que es posible hacer con algunas técnicas más avanzadas. Javascript es un poderoso lenguaje que usa al browser como su esclavo, obligándolo a hacer el trabajo pesado, pero no te confíes en lo que has visto hasta ahora, con este lenguaje es posible escribir programas muy grandes y complejos.

Incluso programas poderosos como Unity permiten desarrollar juegos con su plataforma usando Javascript como lenguaje de programación.



## Uso de operadores y eventos

<b>Operadores aritméticos</b>	Estos son usados para realizar operaciones matemáticas. <code>+, -, *, /, %</code>
<b>Operadores de asignación</b>	Son utilizados para asignar valores a variables. Los que tienen un símbolo antes del = añaden, restan, dividen, el valor de la derecha al valor de la variable de la izquierda en lugar de reemplazar totalmente el valor de la izquierda. <code>=, +=</code>
<b>Operadores de comparación</b>	Son usados en la construcción de enunciados de comparación, por ejemplo, cuando usamos un <code>if</code> , de manera que podamos evaluar dos variables. <code>==, !=</code>
<b>Operadores lógicos</b>	Son usados para comparar varios enunciados entre ellos. <code>&amp;&amp;,   , !</code>
<b>Caracteres de escape</b>	Los caracteres de escape se usan para insertar caracteres especiales. <code>\b, \f, \n, \r</code>



## Eventos trigger

1 Primero, selecciona el nodo o nodos a los que quieras que responda Javascript. Por ejemplo, si quisieras que al activar un botón se dispare un evento, debes localizarlo en el DOM primero.

3 Finalmente, prepara el código que quieras que sea ejecutado cuando llegue el momento. Generalmente, es una función con un nombre específico o también puede ser una función anónima.



2 Debes señalar cuál evento en el nodo que indicas responderá a la interacción.

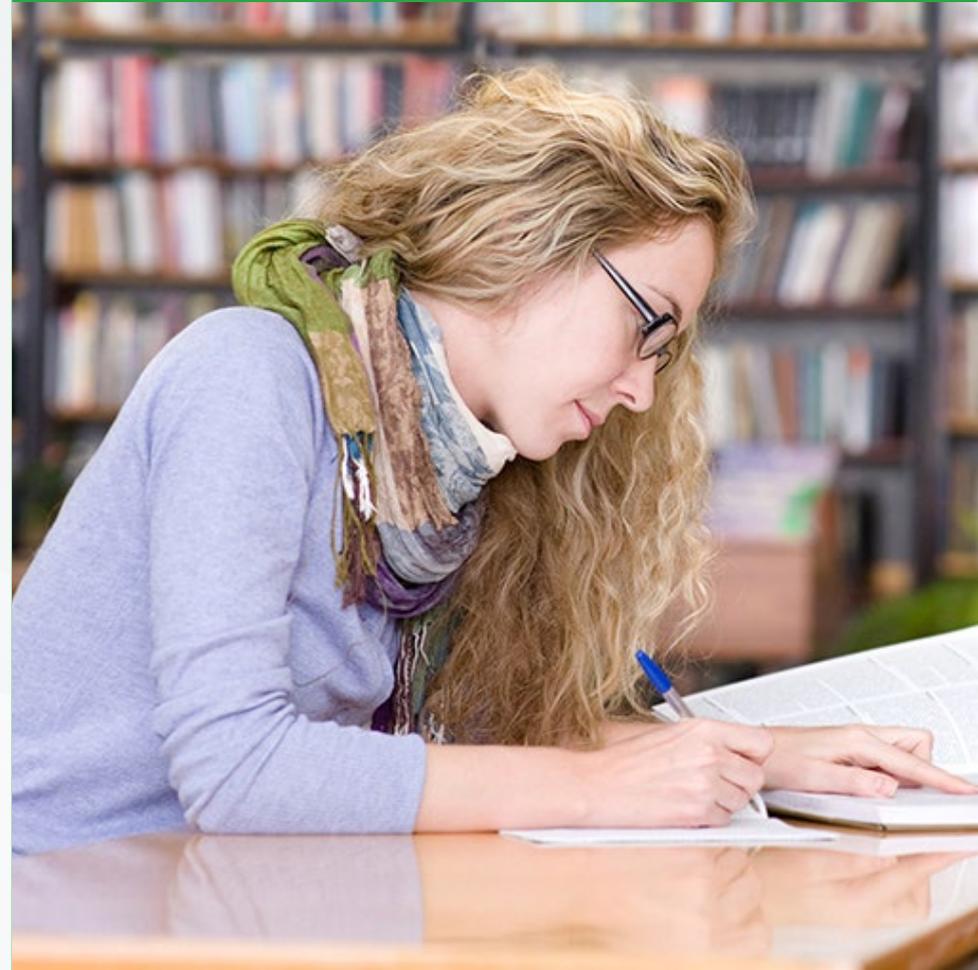


## Uso de números avanzado y Math

<b>isNaN()</b>	Verifica si el valor es un número.
<b>toFixed()</b>	Redondea el número a una cantidad específica de fracciones decimales. Regresa un string.
<b>toPrecision()</b>	Especifica la cantidad exacta de números que debe aparecer en la salida. Regresa un string.
<b>toExponential()</b>	Regresa un string que representa el número en notación exponencial.



- Reflexiona y responde lo siguiente.
  - a) ¿Qué es un operador en JavaScript?
  - b) ¿Para qué sirven?
  - c) Menciona dos ejemplos.





En este tema aprendiste lo necesario para darle la forma final a tu trabajo y con estos nuevos conocimientos ya puedes aplicar animaciones complejas o personalizar el website de tu proyecto final. Tus habilidades con CSS deben poder crear un layout limpio y fácil de navegar.

