



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# ➤ Aplicación de estilos y responsividad

## Manejando Hojas de Estilo

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :


- ✓ *La importancia de trabajar con el modelo de cajas*
- ✓ *incorporar estilos de fuente y de líneas*
- ✓ *Comprender la importancia del diseño flexbox*



# LEARNING PATHWAY

Nº2 . ¿Sobre qué temas trabajaremos?

2.

Start! 

FUNDAMENTOS DE  
DESARROLLO FRONT-END

Aplicación de estilos y  
responsividad

Actividad: Modelo responsivo

En esta lección, aprenderemos sobre el orden jerárquico de las aplicaciones web, como se aplica las reglas de CSS y el peso asociado a estas reglas, también aprenderemos a crear sitios responsivos para darle calidad y adaptación a cada dispositivo y tamaño de pantalla.

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



*Aprender sobre el uso del orden jerárquico en CSS*



*Comprender el uso de la responsividad*



*Utilizar el inspector para los estilos en el navegador*





# Rompehielo

1. Para abrir un candado es necesario de un código de 3 dígitos. Analiza las siguientes pistas y encuentra la contraseña correcta.

6 8 2

Un número es correcto  
y está en su sitio

6 1 4

Un número es correcto pero  
no está en su sitio

2 0 6

Dos números son correctos  
y no están en su sitio

7 1 8

Ninguno es correcto

8 7 0

Un número es correcto  
pero no está en su sitio



CONTRASEÑA:



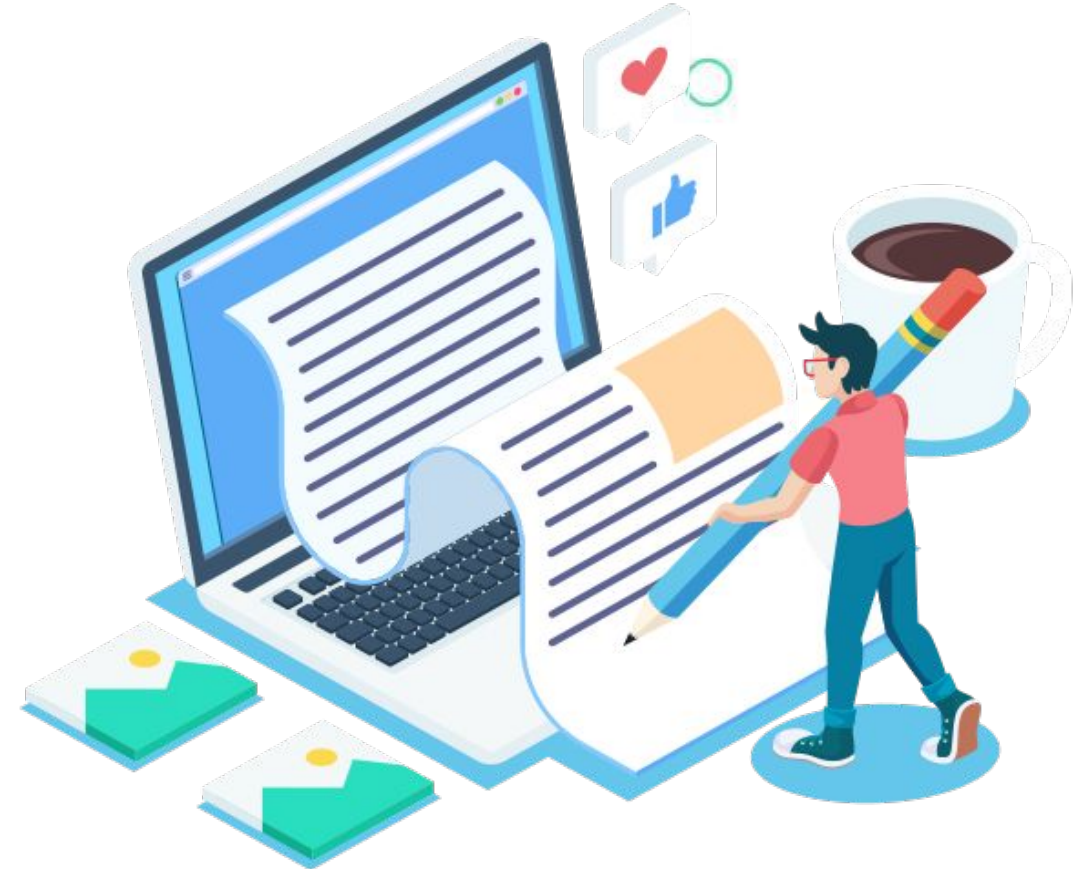
# › Orden jerárquico





# Orden jerárquico

El orden jerárquico de aplicación de las reglas CSS y el peso asociado a las reglas son conceptos importantes para comprender cómo se aplican y priorizan los estilos en una página web.





# Orden jerárquico



## Orden jerárquico de aplicación de las reglas CSS:

- Las reglas CSS se aplican siguiendo un orden jerárquico basado en la especificidad de los selectores y la cascada de estilos.
- Las reglas más específicas tienen prioridad sobre las reglas menos específicas.
- Si tienes reglas conflictivas con el mismo nivel de especificidad, la última regla definida en el archivo CSS prevalecerá y se aplicará.





# Orden jerárquico



## Peso asociado a las reglas CSS:

- Cada regla CSS tiene un peso asociado que determina su prioridad y cómo se aplicará en caso de conflictos.
- Los diferentes tipos de selectores tienen diferentes niveles de peso.
- En caso de que haya conflictos entre reglas con el mismo nivel de especificidad, la regla con un selector más pesado prevalecerá y se aplicará.



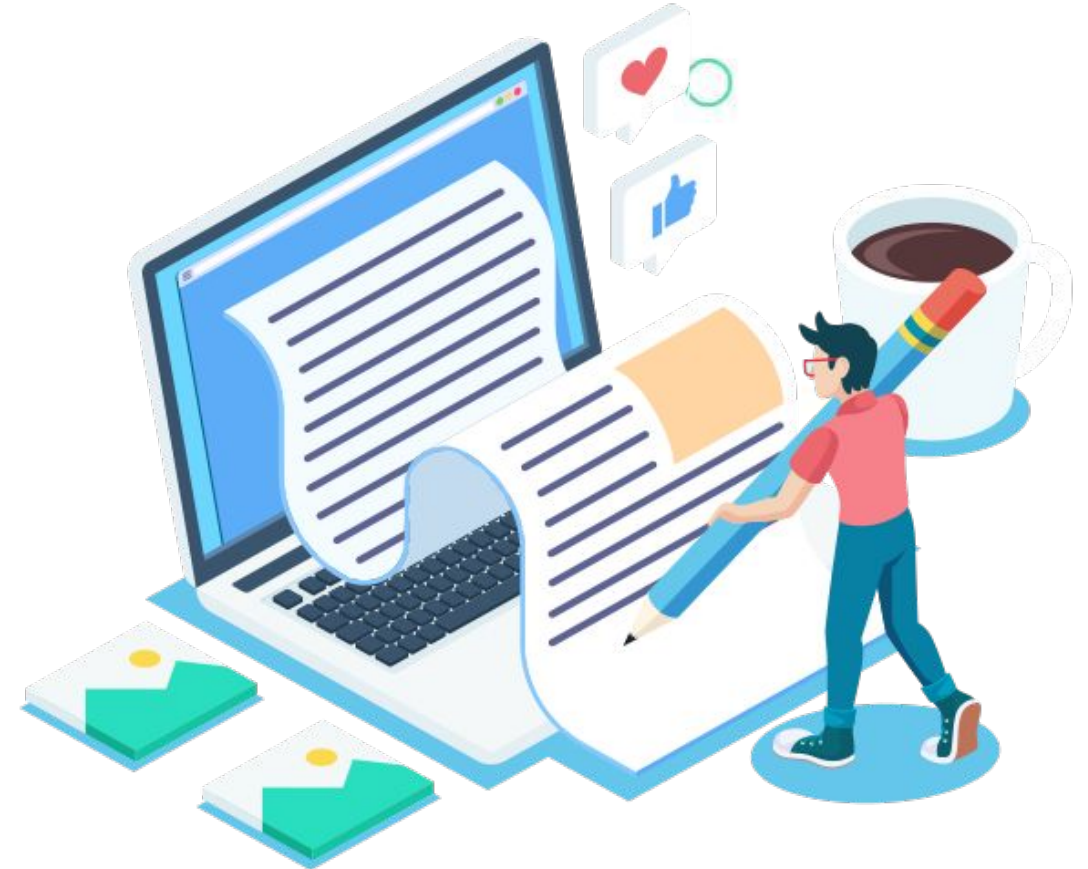


# Orden jerárquico



## **selectores ordenados de menor a mayor peso:**

- Selectores de tipo y pseudoelementos (p, div, ::before, ::after, etc.).
- Selectores de clase, atributo y pseudoclases (.clase, [atributo], :hover, etc.).
- Selectores de ID (#id).
- Selectores de estilo en línea (atributo style dentro del elemento HTML).





# Orden jerárquico

## **selectores ordenados de menor a mayor peso:**

- Selectores de tipo y pseudoelementos (p, div, ::before, ::after, etc.).
- Selectores de clase, atributo y pseudoclases (.clase, [atributo], :hover, etc.).
- Selectores de ID (#id).
- Selectores de estilo en línea (atributo style dentro del elemento HTML).



```
p {  
  color: red !important;  
}
```





# Orden jerárquico



## **Cascada (Cascading):**

La cascada se refiere al proceso de aplicar y combinar los estilos CSS a los elementos HTML.



La cascada sigue una serie de reglas para determinar qué estilo prevalece en caso de conflictos. Estas reglas se basan en el orden de aparición de las reglas, la especificidad de los selectores y el peso asociado a las reglas.

La cascada permite que los estilos se combinen y se apliquen de manera controlada y predecible, permitiendo la separación de la estructura HTML y la presentación visual de una página web.





# Orden jerárquico



## Especificidad (Specificity):

La especificidad es un concepto que determina qué estilo tiene prioridad cuando hay reglas conflictivas que se aplican a un mismo elemento.

Cada selector CSS tiene un nivel de especificidad que se calcula en función de los selectores utilizados en la regla.

En caso de conflictos, la regla con la especificidad más alta prevalecerá y se aplicará. La especificidad se calcula asignando un valor numérico a cada tipo de selector (ID, clase, tipo) y sumando estos valores para obtener la especificidad total.

```
/* Regla más específica */  
.clase1.clase2 {  
  color: blue;  
}  
  
/* Regla menos específica */  
p {  
  color: orange;  
}
```



# Orden jerárquico



## Herencia (Inheritance):

La herencia es el mecanismo por el cual los estilos se propagan desde un elemento padre a sus elementos hijos.



Algunas propiedades CSS son heredadas por defecto, lo que significa que los estilos aplicados a un elemento se transmiten automáticamente a sus elementos hijos, a menos que se especifique lo contrario.

Esto permite establecer estilos generales en elementos de nivel superior y afectar a todos sus descendientes sin tener que aplicar explícitamente los estilos en cada elemento hijo.








# Orden jerárquico

## Orden de Declaración:

Si dos reglas tienen la misma importancia y la misma especificidad, la regla que aparece más abajo en el archivo CSS prevalecerá. En otras palabras, la última regla declarada en el archivo CSS tomará precedencia.



```
p {  
  color: red;  
}  
  
p {  
  color: orange;  
}
```



# › Inspeccionando estilos



# Inspeccionando estilos

Las herramientas para desarrolladores en los navegadores web son una herramienta poderosa para inspeccionar y modificar los estilos de una página web en tiempo real.

Estas herramientas te permiten visualizar y analizar los estilos aplicados a los elementos, identificar conflictos, probar cambios y depurar problemas relacionados con el diseño y la apariencia.

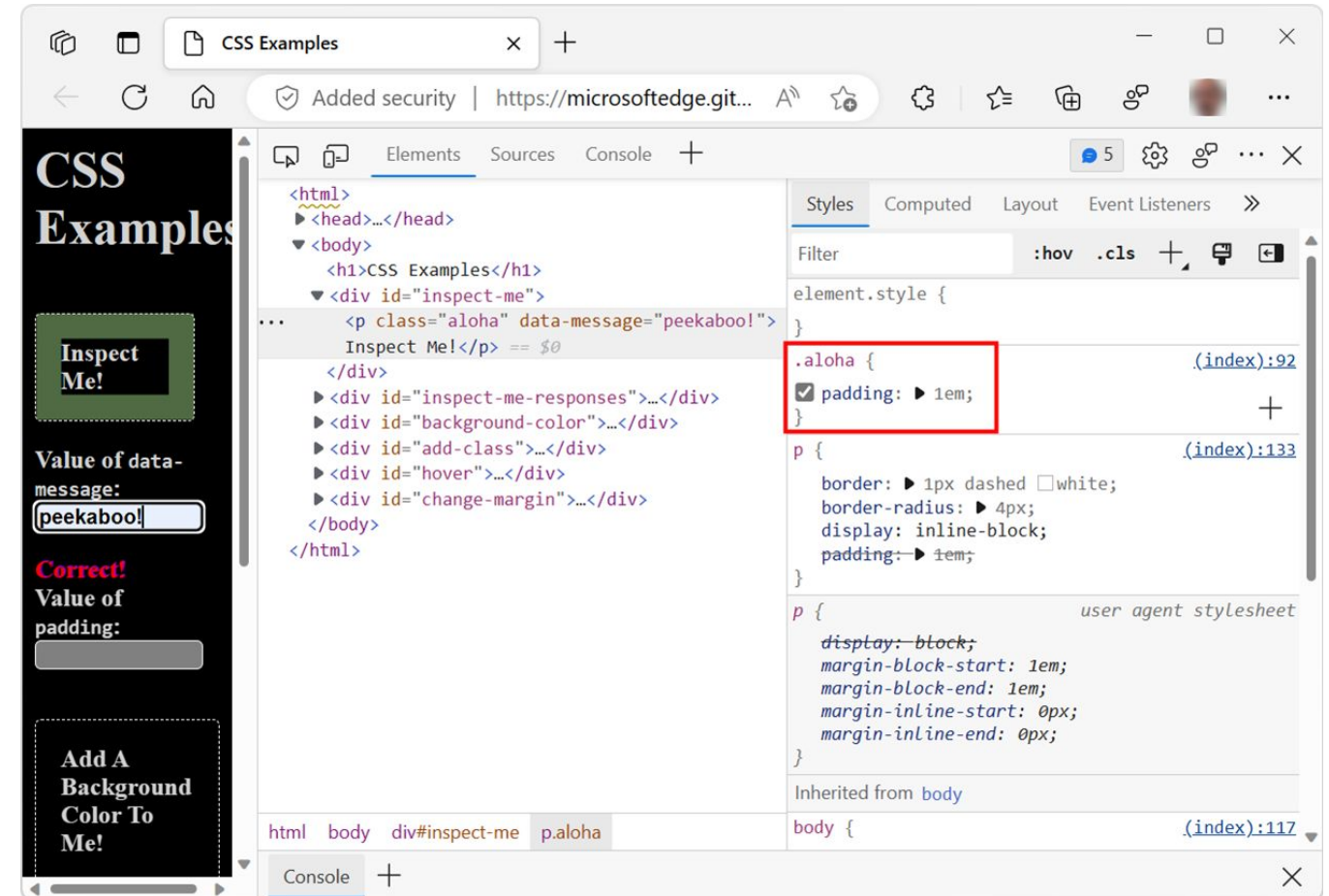




# Inspeccionando estilos

## Google Chrome:

- Abre Google Chrome y ve a la página web que deseas inspeccionar.
- Selecciona "Inspeccionar" en los 3 puntitos, o presiona Ctrl+Shift+I o Cmd+Option+I (en Mac) para abrir las DevTools directamente.
- En la ventana de DevTools hacer clic en la pestaña "Elements" (Elementos) para ver la estructura HTML de la página y los estilos aplicados.

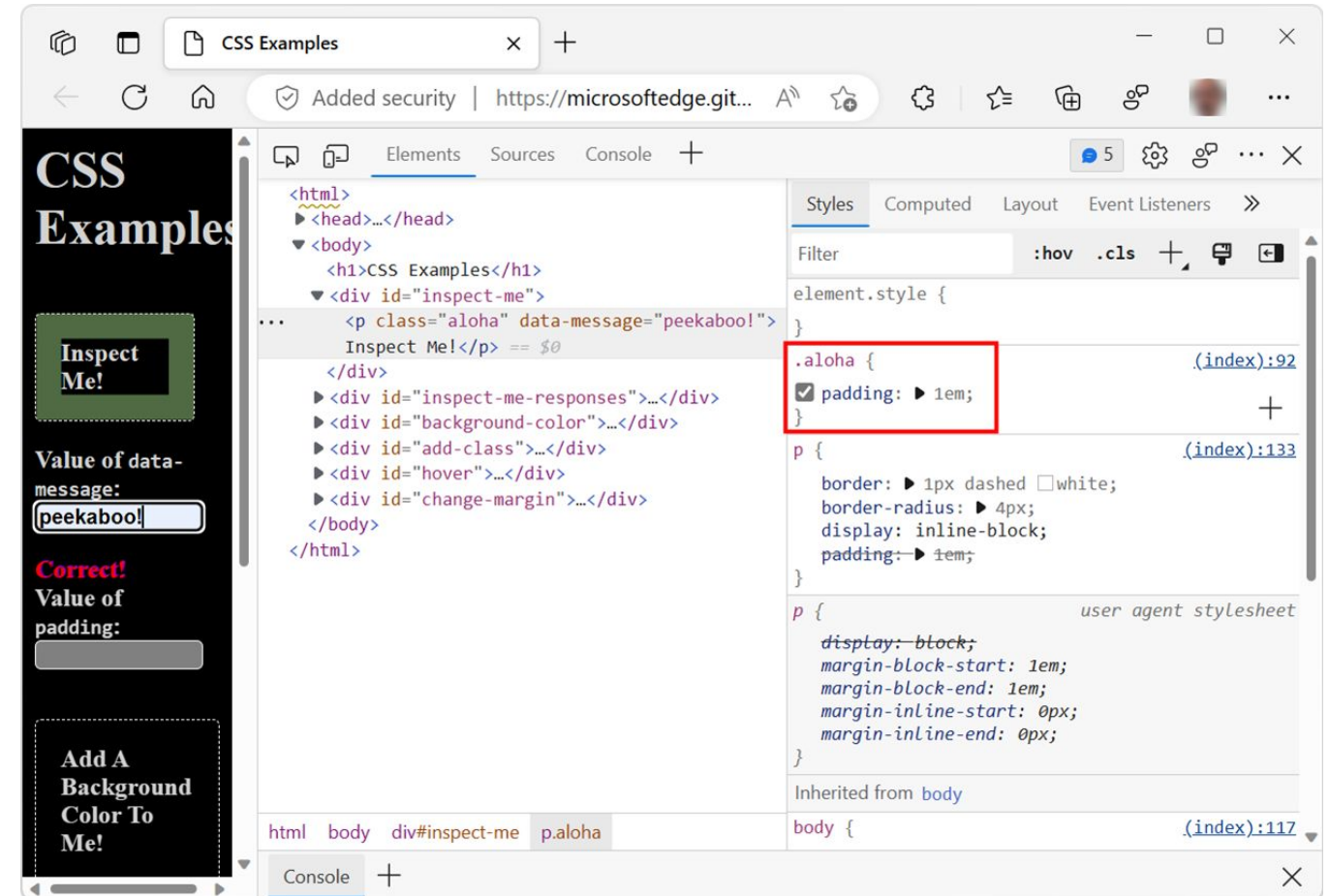




# Inspeccionando estilos

## Mozilla Firefox:

- Abre Mozilla Firefox y navega a la página web que deseas inspeccionar.
- Selecciona "Inspeccionar elemento" en los 3 puntitos, o presiona Ctrl+Shift+I o Cmd+Option+I (en Mac) para abrir las Herramientas de desarrollo.
- En la ventana de Herramientas de desarrollo, verás el "Inspector" (Inspector) para ver la estructura HTML de la página y los estilos CSS.

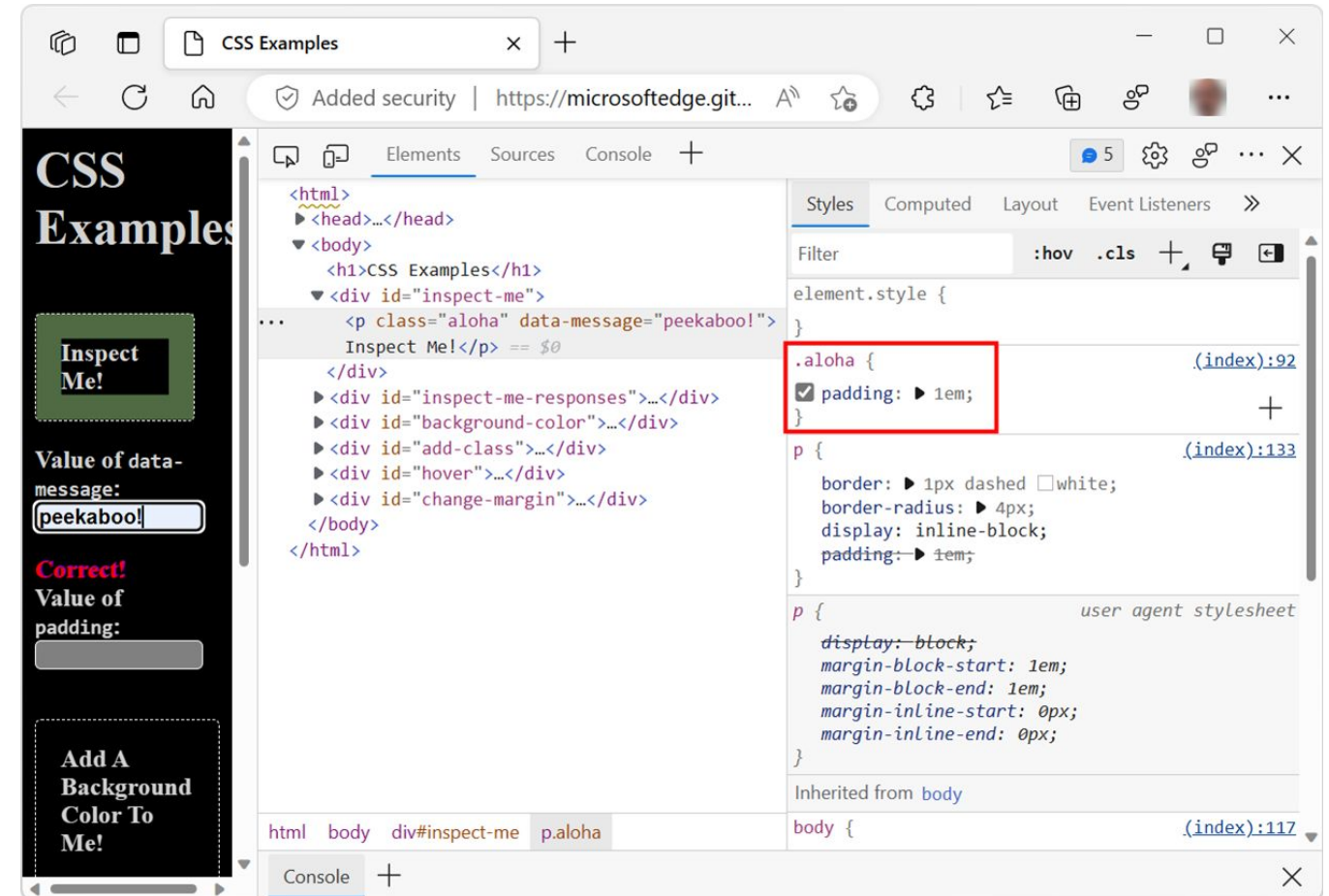




# Inspeccionando estilos

## Microsoft Edge:

- Abre Microsoft Edge y visita la página web que deseas inspeccionar.
- Selecciona "Inspeccionar elemento" los 3 puntitos o presiona Ctrl+Shift+I o Cmd+Option+I (en Mac) para abrir las Herramientas de desarrollo.
- En la ventana de Herramientas de desarrollo, verás la pestaña "Elements" (Elementos) para ver la estructura HTML de la página y los estilos CSS.



# ➤ Responsividad





# Responsividad

La responsividad se refiere a la capacidad de una página web para adaptarse y verse bien en diferentes dispositivos y tamaños de pantalla.

Con el aumento de la diversidad de dispositivos móviles y la proliferación de pantallas de diferentes resoluciones, es fundamental que los sitios web sean responsivos para brindar una experiencia de usuario óptima.

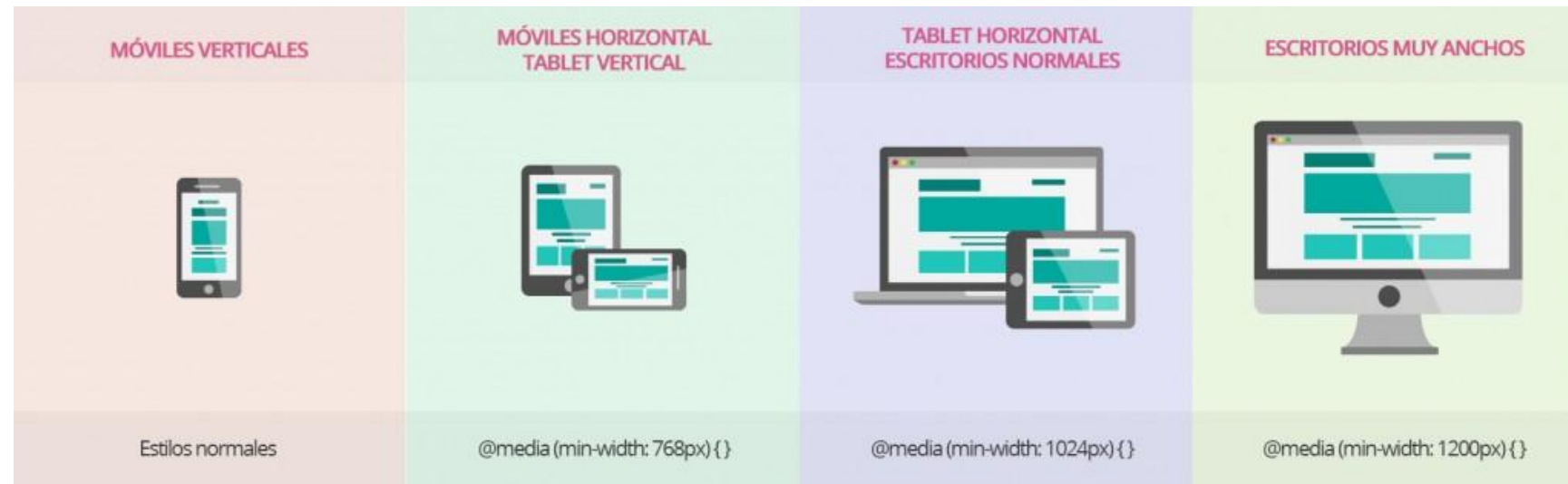
-







# Responsividad



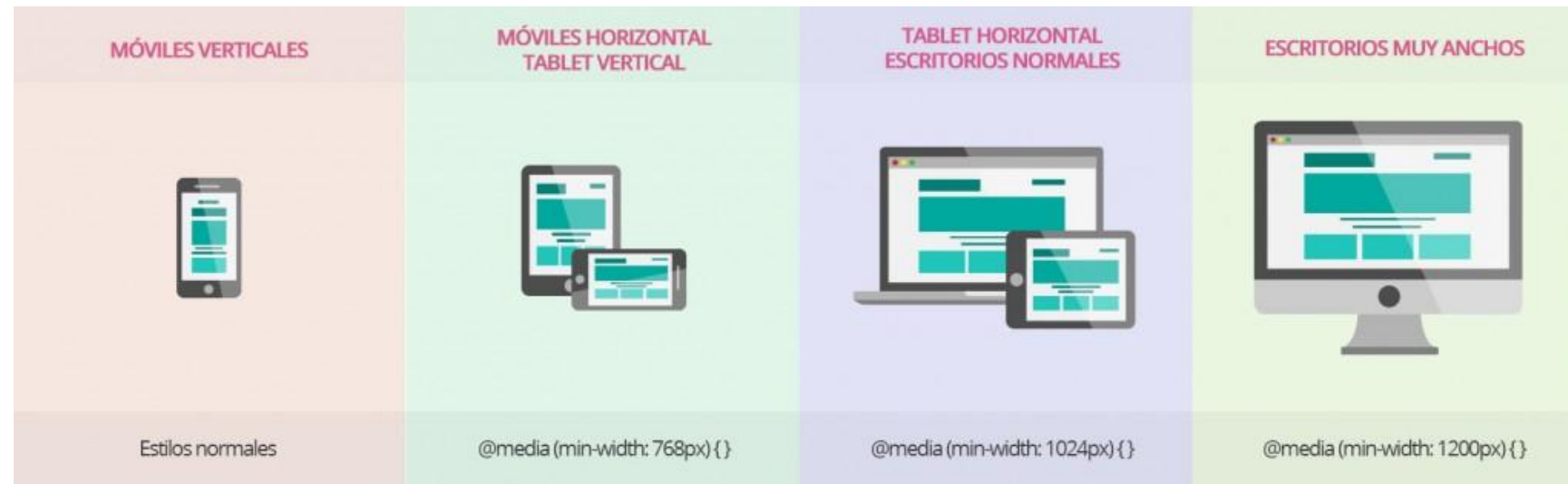
## El concepto de responsividad

La responsividad, también conocida como diseño web adaptable o responsive design en inglés, es el enfoque de diseño y desarrollo web que busca crear sitios web que se adapten y respondan a diferentes dispositivos y tamaños de pantalla.

El objetivo de la responsividad es brindar una experiencia de usuario óptima, independientemente del dispositivo que se esté utilizando para acceder al sitio web.



# Responsividad



## El concepto de responsividad

En un sitio web responsivo, los elementos y el diseño se ajustan automáticamente según el tamaño y las características del dispositivo. Esto incluye factores como el ancho de la pantalla, la resolución, la orientación (vertical u horizontal) y la capacidad táctil del dispositivo.

La responsividad se logra mediante el uso de técnicas como media queries, unidades de medida relativas (porcentajes, unidades de vista) y diseño fluido.



# Responsividad

## Diseño adaptable:

Los diseños responsivos se ajustan a diferentes tamaños y resoluciones de pantalla. Esto implica reorganizar los elementos, ajustar el tamaño de fuente y las imágenes, y ocultar o mostrar contenido según sea necesario.





# Responsividad

## Navegación intuitiva:

Los sitios web responsivos suelen tener una navegación simplificada y fácil de usar en dispositivos móviles, como menús desplegables o hamburguesas. Esto permite a los usuarios acceder fácilmente al contenido sin tener que hacer zoom o desplazarse demasiado.





# Responsividad

## Optimización del rendimiento:

Los sitios web responsivos también tienen en cuenta el rendimiento, optimizando el tamaño y la carga de los recursos para adaptarse a diferentes conexiones y velocidades de Internet. Esto ayuda a mejorar la velocidad de carga y la experiencia del usuario.







# Responsividad

## Pruebas multiplataforma:

Es fundamental realizar pruebas exhaustivas en diferentes dispositivos y tamaños de pantalla para asegurarse de que el sitio web se vea y funcione correctamente en todas las plataformas. Esto incluye dispositivos móviles, tablets y diferentes navegadores web.



# LIVE CODING

Ejemplo en vivo

## Prioridad de Reglas CSS:

*Imagina que estás trabajando en el desarrollo de un sitio web para una empresa de viajes. La empresa necesita una página de inicio que destaque diferentes destinos turísticos y ofertas especiales. Tu tarea es aplicar estilos CSS a la página de inicio, prestando especial atención al uso de reglas CSS y su prioridad.*

 **Tiempo: 15 minutos**



# LIVE CODING

Ejemplo en vivo

## Pasos:

- *En el archivo CSS (styles.css), define las siguientes reglas CSS:*
- *Crea una regla que establezca un fondo de color para el encabezado de la página (<header>).*
- *Crea una regla que aplique un color de texto y un tamaño de fuente a los títulos (<h1>) de los destinos turísticos.*
- *Crea una regla que cambie el color de texto y el tamaño de fuente de los párrafos (<p>) que describen los destinos.*
- *Define una regla para los elementos de oferta especial, como botones o divs (<div>), y cambia su estilo (por ejemplo, color de fondo o borde).*





# LIVE CODING

Ejemplo en vivo

## Pasos:

- *Añade una regla !important a una de las propiedades de estilo para resaltar su importancia.*
- *Asegúrate de aplicar las reglas CSS de manera que haya reglas que compitan por la misma propiedad en diferentes elementos.*
- *Comenta cada regla CSS en el archivo styles.css para indicar qué propiedad está afectando y por qué. Por ejemplo, puedes comentar que una regla tiene mayor especificidad o que se utiliza !important.*
- *Visualiza la página de inicio en un navegador y observa cómo se aplican las reglas CSS en función de su prioridad y el orden jerárquico.*
- *Experimenta con la eliminación o modificación de reglas para comprender cómo afecta al diseño de la página.*





# **Ejercicio**

# **Modelo responsivo**



# Modelo responsivo

Breve descripción

## Contexto: 🙌

Estás trabajando en el desarrollo de la página de inicio de un blog de viajes. El objetivo es crear un diseño responsive que se vea bien tanto en dispositivos móviles como en computadoras de escritorio.



## Consigna: 🖋️

Crear un diseño de página una wallet que se adapte a diferentes tamaños de pantalla, como dispositivos móviles y de escritorio.



Tiempo🕒: 10 minutos



# Modelo responsivo

Breve descripción



## Paso a paso:

- Define tres clases CSS para aplicar diferentes estilos a las tareas:
  - .task-red para aplicar un fondo rojo.
  - .task-bold para hacer que el texto sea más grueso.
  - .task-italic para aplicar estilo de texto en cursiva.
- Crea una lista de tareas (<ul>) en el archivo HTML y agrega varias tareas (<li>) a la lista. Asigna clases CSS a algunas de las tareas para aplicar diferentes estilos.
- Define reglas CSS que apliquen los estilos de las clases a las tareas correspondientes. Por ejemplo, utiliza la clase .task-red para establecer el fondo rojo y la clase .task-bold para hacer el texto más grueso.



# Modelo responsivo

Breve descripción



## Paso a paso:

- Añade un botón en cada tarea para que los usuarios puedan eliminarla.
  - Estiliza este botón utilizando CSS.
- A medida que trabajas en las reglas CSS, ten en cuenta la especificidad de las clases y cómo afecta la aplicación de los estilos. Por ejemplo, si una tarea tiene tanto la clase `.task-red` como `.task-bold`, ¿cuál estilo prevalece?
- Visualiza la aplicación en un navegador y verifica que los estilos se apliquen correctamente a las tareas. Observa cómo se comporta la especificidad cuando se aplican múltiples clases.
- Experimenta agregando más tareas con diferentes combinaciones de clases para comprender mejor cómo afecta la especificidad a la aplicación de estilos.

```
= "UTF-8">
  name="viewport" content="width=device-width,
">
ylesheet" href="styles.css">
Tareas</title>
```

```
de Tareas</h1>
```

```
class="task task-red">
span>Completar tarea 1</span>
button class="delete-button">Eliminar</button>
```

```
class="task task-bold task-italic">
span>Completar tarea 2</span>
button class="delete-button">Eliminar</button>
```

```
span>Completar tarea 3</span>
button class="delete-button">Eliminar</button>
```

```
/* Estilos generales */
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}

header {
  background-color: #333;
  color: white;
  text-align: center;
  padding: 10px;
}

main {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
}

/* Estilos para las tareas */
.task {
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ddd;
}

.task-red {
  background-color: #ff9999;
```

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la importancia del uso del orden jerárquico para aplicar estilos CSS**
- ✓ **Aprender la teoría sobre el uso de la responsividad y adaptación a distintos dispositivos**
- ✓ **utilizar el inspector para manipular y observar los estilos de CSS**





# #WorkingTime

Continuemos ejercitando

**¡Antes de cerrar la clase!** Te invitamos a: 🙌🙌🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Material 1: Lección 3: Aplicación de estilos y responsividad página: 25-30
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

