



# Media Queries



**Activen las cámaras los que puedan y  
pasemos asistencia**





# Inicio

{desafío}  
latam\_



*/\* Utilizar media queries para cambiar los estilos de un elemento. \*/*

*/\* Trabajar con múltiples breakpoints. \*/*

*/\* Definir múltiples breakpoints para variar la cantidad de columnas en una grilla diseñada con CSS grid. \*/*

*/\* Crear un layout responsivo con CSS grid. \*/*

# Objetivos

# Activación de conceptos

*Contesta la pregunta correctamente y gana un punto*

## Instrucciones:

- Se realizará una pregunta, el primero en escribir “YO” por el chat, dará su respuesta al resto de la clase.
- El docente validará la respuesta.
- En caso de que no sea correcta, dará la oportunidad a la segunda persona que dijo “Yo”.
- Cada estudiante podrá participar un máximo de 2 veces.
- Al final, el/la docente indicará el 1º, 2º y 3º lugar.
- Esta actividad no es calificada, es solo una dinámica para recordar los conceptos clave para abordar esta sesión.





Activación de conceptos



¿Con qué propiedad declaramos una grilla de CSS  
Grid en un elemento?



Activación de conceptos



¿Qué propiedades de CSS Grid definen los espacios entre filas y columnas?





Activación de conceptos



¿Cómo se escribe la unidad de medida que  
define una fracción?



Activación de conceptos



Primer lugar:

\_\_\_\_\_



Segundo lugar:

\_\_\_\_\_



Tercer lugar:

\_\_\_\_\_



# Desarrollo

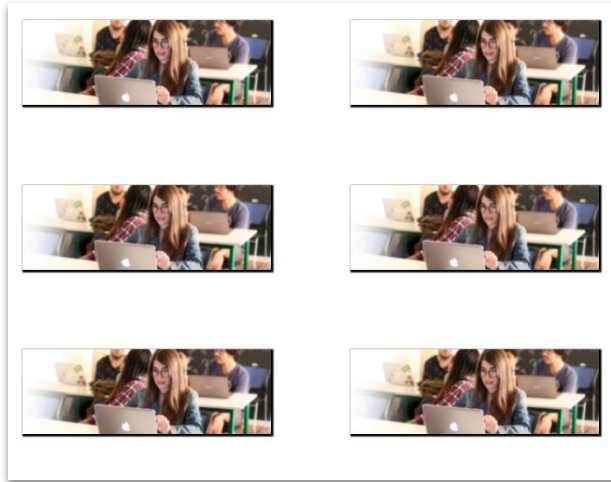
{desafío}  
latam\_



# Media Queries

*¿Qué es un media query?*

Es un bloque de CSS que se ejecuta solo si se cumple una condición, estas condiciones normalmente son anchos específicos del viewport; o sea podemos ejecutar código css si la pantalla es grande o pequeña.



Página vista  
en tablet

Página vista en notebook



# Media Queries

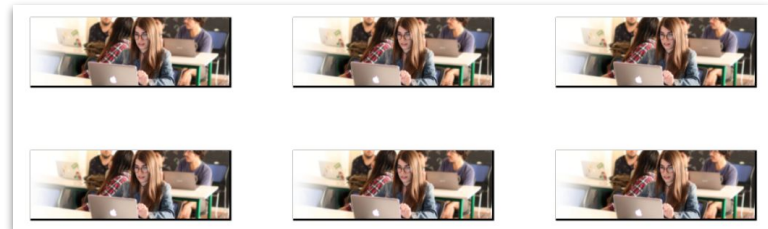
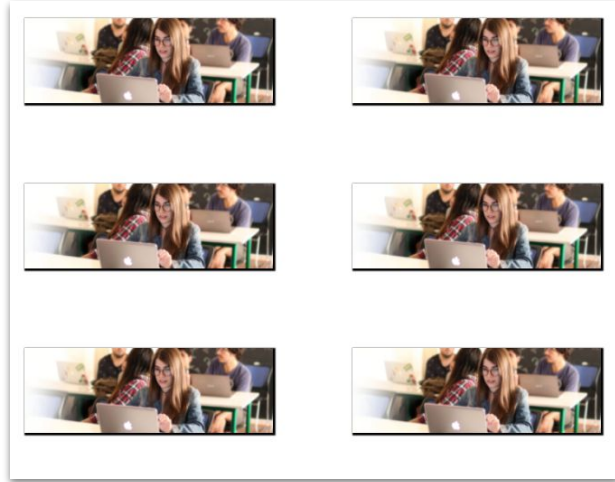
## ¿Qué es un breakpoint?

Para determinar en qué ancho o medida específica sucede el cambio se utilizan breakpoints; o puntos de quiebre.

Son los breakpoints los que determinan si estamos en una pantalla chica, mediana o grande, estos los definimos nosotros usualmente ocupando una unidad de medida en pixeles.

Ejemplo:

- Pantalla grande: sobre 768 pixeles
- Pantalla muy grande: sobre 1200 pixel

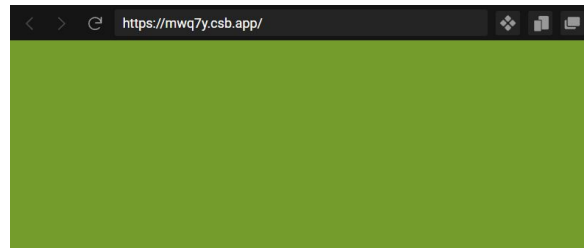
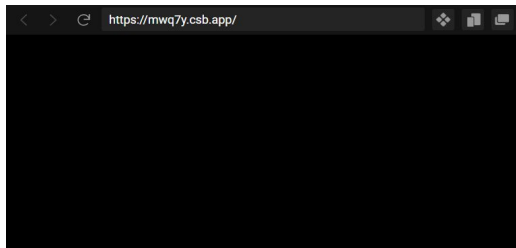


# Media Queries

## ¿Cómo se construye un breakpoint?

Ocupemos el siguiente código CSS en un archivo nuevo, y probemos cambiando el ancho de pantalla para visualizar cómo cambia el color de fondo.

```
@media (max-width: 576px) {  
  body {  
    background: black;  
  }  
}  
  
@media (min-width: 576px) {  
  body {  
    background: #749c2c;  
  }  
}
```



La declaración de un breakpoint es a través de un *@media* acompañado de parámetros que especifican qué tipo de *media* se va a ocupar y la condición que se debe cumplir.

# Media Queries

## Explicando el código

```
@media (max-width: 576px) {  
  body {  
    background: black;  
  }  
}  
  
@media (min-width: 576px) {  
  body {  
    background: #749c2c;  
  }  
}
```

La declaración básica de un media query es a través de un *@media* y la condición que se debe cumplir entre paréntesis.

Dentro del *@media* hay un bloque CSS que sólo se ejecutará si se cumple la condición.

*/\* Utilizar media queries para cambiar los estilos de un elemento. \*/* ✓

*/\* Trabajar con múltiples breakpoints. \*/*

*/\* Definir múltiples breakpoints para variar la cantidad de columnas en una grilla diseñada con CSS grid. \*/*

*/\* Crear un layout responsivo con CSS grid. \*/*

# Objetivos



## Ejercicio

Realicemos un ejercicio en donde cambiamos la tipografía de un párrafo utilizando un breakpoint. Sigue los siguientes pasos:

1. Crea un archivo nuevo con una base de HTML.
2. Crea una etiqueta de párrafo con un texto genérico (Lorem).
3. Utilizando CSS, crea un breakpoint para el tamaño extra small (<576px) que defina un font-family monospace
4. Crea un breakpoint para el tamaño small (>=576px) que defina un font-family: fantasy
5. Observa el cambio en el inspector de elementos.

## Ejercicio ¡Manos al teclado!



## Media Queries

### *¿Cuántos breakpoint debemos definir en nuestro sitio web?*

La respuesta siempre dependerá del objetivo de tu sitio web, lo importante es entender que no es necesario declarar todos los breakpoints si estos no aportan un valor importante a nuestro diseño.

Por ahora, podemos basarnos en frameworks como Bootstrap donde se definen los siguientes breakpoints:

Extra small	Small	Medium	Large	Extra large
<576px	≥576px	≥768px	≥992px	≥1200px

# Media Queries

## Construyendo múltiples breakpoints

A continuación revisemos el código de un ejercicio en donde - con diferentes breakpoint - le cambiamos el color de texto a una etiqueta h3.

```
h3 { color: crimson }

@media (min-width: 576px) {
  h3 { color: purple }
}

@media (min-width: 768px) {
  h3 { color: green }
}

@media (min-width: 992px) {
  h3 { color: chocolate }
}
```

Desafío Latam → Desafío Latam → Desafío Latam → Desafío Latam

{desafío}  
latam\_

## Ejercicio

Además de modificar el valor de una misma propiedad, también podemos cambiar o asignar otras propiedades del mismo elemento. En este ejercicio, nuestro objetivo será asignar color de fondo, de texto y bordes redondeados a un botón en diferentes breakpoint, sin embargo, asignaremos un display: none en un último quiebre para eliminarlo de la vista:

1. Crea un botón en el HTML con un texto dentro que diga enviar.
2. Define un color de fondo al botón.
3. Crea un breakpoint para la medida small que asigne bordes redondeados de 15px
4. Crea un breakpoint para la medida medium que asigne un color de texto al botón.
5. Crea un tercer y último breakpoint para eliminar el botón de la vista con un display: none

**{desafío}**  
latam\_



## Ejercicio

¡Manos al teclado!



*/\* Utilizar media queries para cambiar los estilos de un elemento. \*/* ✓

*/\* Trabajar con múltiples breakpoints. \*/* ✓

*/\* Definir múltiples breakpoints para variar la cantidad de columnas en una grilla diseñada con CSS grid. \*/*

*/\* Crear un layout responsivo con CSS grid. \*/*

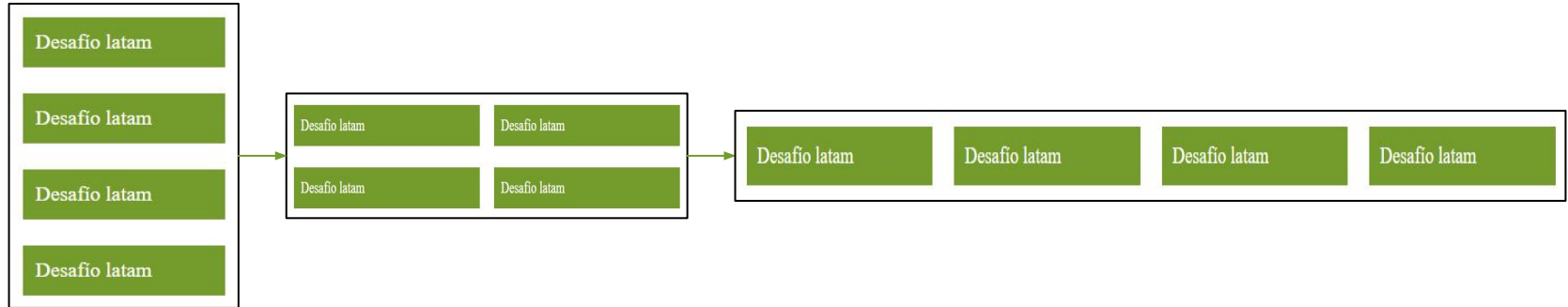
# Objetivos

# Media Queries

## *Las grillas y los media queries*

Los media queries son usados principalmente para los cambios de layout en una página. Con la ayuda de herramientas como CSS Grid y Flexbox podemos generar cambios importantes que ayuden y mejoren la experiencia de nuestros usuarios.

Para probarlo, crearemos una grilla que cambie la cantidad de columnas en función del viewport.



# Media Queries

## Las grillas y los media queries

Usualmente, cuando uno parte uno de estos ejercicios, implementa primero la versión mobile.

Creemos el html con una grilla de 1 columna, para esto utilizaremos propiedad grid-template-columns.

```
<div class="grid">
  <p>Desafío latam</p>
  <p>Desafío latam</p>
  <p>Desafío latam</p>
  <p>Desafío latam</p>
</div>
```

```
p {
  background: #749c2c;
  color: white;
  padding: 10px;
  margin: 10px;
}
.grid {
  display: grid;
  grid-template-columns: 1fr;
}
```

Desafío latam

Desafío latam

Desafío latam

Desafío latam

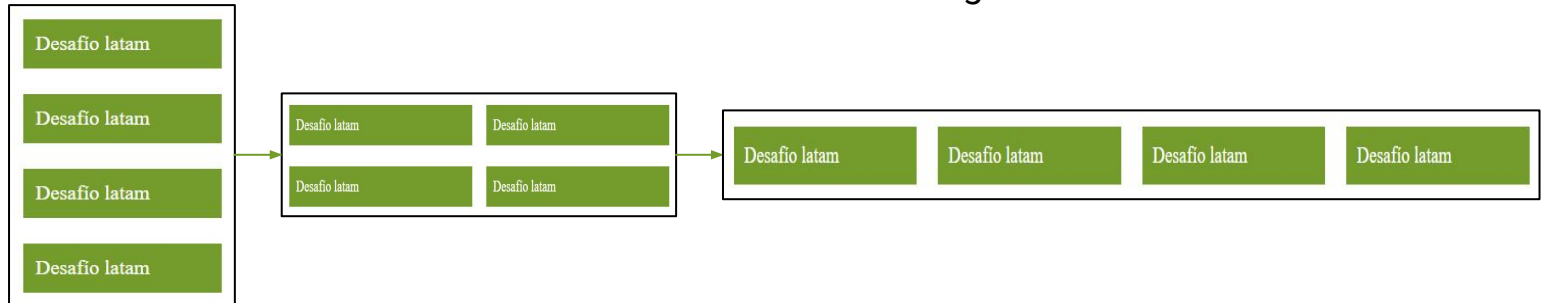
# Media Queries

## Las grillas y los media queries

El siguiente paso es agregar los medias queries y cambiar el número de columnas en función del ancho del viewport

```
@media (min-width: 576px) {  
  .grid { grid-template-columns: 1fr 1fr }  
}  
  
@media (min-width: 768px) {  
  .grid { grid-template-columns: 1fr 1fr 1fr 1fr }  
}
```

El resultado en las diferentes variantes de ancho sería entonces el siguiente:





## Ejercicio

Entre los usos más populares de CSS Grid se encuentra el diseño o maquetación de una galería de imágenes, y ahora que hemos aprendido a ocupar los breakpoints, pongamos a prueba nuestra nueva habilidad con un ejercicio en donde deberemos definir 3 medidas diferentes: Extra small, small, medium y large, para presentar una galería de imágenes de perritos obtendremos de [Place Dog](#).

1. Crea un div en el HTML que sirva como contenedor de una grilla de CSS.
2. Incluye 6 imágenes que muestren perritos de Place God.
3. Con CSS Grid, diseña la versión móvil de 1 sola columna.
4. Escribe 3 breakpoints para las versiones small, medium y large que varíe la cantidad de columnas.

## Ejercicio

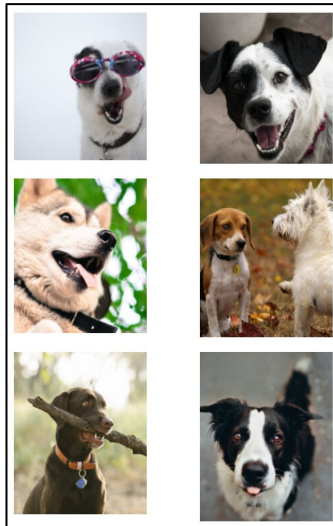
### ¡Manos al teclado!



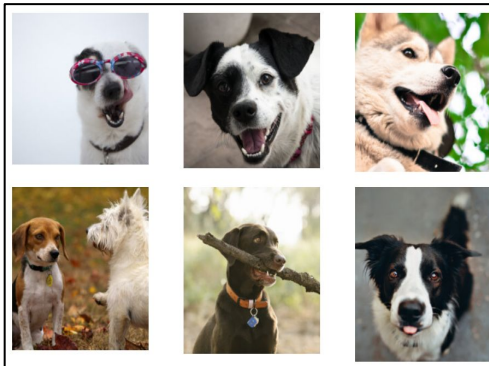
Extra Small



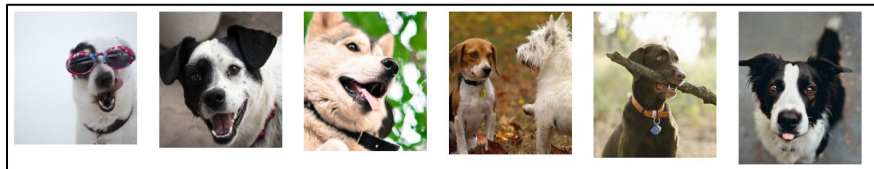
Small



Medium



Large



{desafío}  
latam\_

Ejercicio  
¡Manos al teclado!



*/\** Utilizar media queries para cambiar los estilos de un elemento. *\*/* ✓

*/\** Trabajar con múltiples breakpoints. *\*/* ✓

*/\** Definir múltiples breakpoints para variar la cantidad de columnas en una grilla diseñada con CSS grid. *\*/* ✓

*/\** Crear un layout responsivo con CSS grid. *\*/*

# Objetivos

# Media Queries

## *Implementado diferentes layouts*

Una práctica muy útil y fácil de ocupar para implementar diferentes layouts en nuestras páginas web, es mezclar los breakpoints con la propiedad grid-template-areas de CSS Grid.

Veamos un ejemplo en donde tenemos declarado en un HTML diferentes secciones y las vamos ubicando con un template definido de CSS GRID.

Utiliza el siguiente fragmento de código HTML para este ejemplo.

```
<div>
  <nav>Menú de navegación</nav>
  <main>Sección principal</main>
  <footer>Pie de página</footer>
</div>
```

# Media Queries

## *Diseñando diferentes layouts*

Ahora definamos los estilos base de cada sección, y la asignación de su nombre para *template area* que declararemos más adelante.

```
nav {  
  padding: 10px;  
  grid-area: menu;  
  background: #0037a1;  
  color: white;  
}  
  
main {  
  grid-area: seccion-principal;  
  padding: 10px;  
}  
  
footer {  
  grid-area: pie-de-pagina;  
  background: #d02a1e;  
  color: white;  
  padding: 10px;  
}
```

# Media Queries

## *Diseñando diferentes layouts*

Con este código tu página web debería verse de la siguiente manera:

Menú de navegación

Sección principal

Pie de página

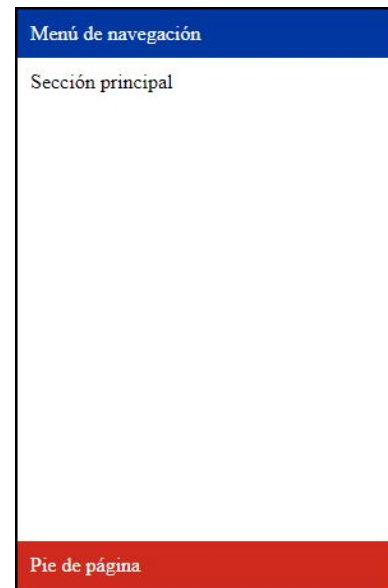
Ahora procedamos a definir la grilla con CSS GRID de nuestras secciones utilizando *grid-template-rows* y *grid-template-areas*:

```
div {  
  display: grid;  
  border: 1px solid black;  
  grid-template-rows: 1fr 10fr 1fr;  
  grid-template-areas:  
    "menu"  
    "seccion-principal"  
    "pie-de-pagina";  
}
```

# Media Queries

## *Diseñando diferentes layouts*

De esta manera ya tendríamos el template para la versión mobile, quedando de la siguiente manera:



# Media Queries

## *Diseñando diferentes layouts*

Ahora, debemos ocuparnos de escribir en el breakpoint que cambie el layout por una distribución más acorde a la versión Desktop. Esta tendrá 2 columnas en vez de 1 y cambiará la ubicación de los elementos para presentar una interfaz acorde a las tendencias de los últimos años donde encontramos menús de navegación verticales.

Utilizando el siguiente código:

```
@media (min-width: 992px) {  
  div{  
    grid-template-areas:  
      "menu seccion-principal"  
      "menu seccion-principal"  
      "menu pie-de-pagina";  
    grid-template-columns: 1fr 10fr;  
  }  
}
```



# Media Queries

## *Diseñando diferentes layouts*

Obtendremos el siguiente resultado:



`/* Utilizar media queries para cambiar los estilos de un elemento. */` ✓

`/* Trabajar con múltiples breakpoints. */` ✓

`/* Definir múltiples breakpoints para variar la cantidad de columnas en una grilla diseñada con CSS grid. */` ✓

`/* Crear un layout responsivo con CSS grid. */` ✓

# Objetivos



Cierre

{desafío}  
latam\_



¿Existe algún concepto que no  
hayas comprendido?

Reflexionemos

- Revisar la guía que trabajarán de forma autónoma.
- Revisar en conjunto el desafío.

¿Qué sigue?



*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam