«Talento Tech»

Desarrollo Web 1

Clase 06

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase N° 6 | Conceptos básicos

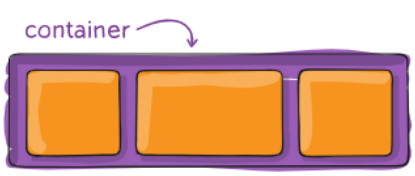
Temario:

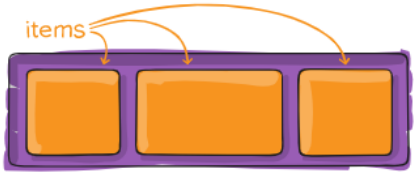
- Diseño Responsivo
- Mobile first
- Media query
- Flexbox
- Flexbox- centrados y alineados
- Flexbox Responsivo

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Flexbox

Hasta el momento, hemos estado agregando elementos de forma individual sin considerar su posición en el HTML. Sin embargo, con la introducción de flexbox, seremos capaces de organizar grupos de elementos de manera estratégica y crear diseños más complejos. Te invitamos a ver la siguiente presentación para aprender cómo lograrlo de manera efectiva.



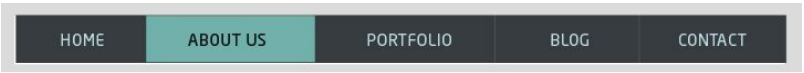


Flexbox es un método que permite crear diseños flexibles y responsivos en páginas web. Proporciona una forma eficiente y fácil de alinear, distribuir y organizar elementos en un contenedor, independientemente de su tamaño o estructura. En otras palabras, nos permite generar contenedores flexibles para poder ordenar los elementos uno al lado del otro, incluso los elementos de bloque.

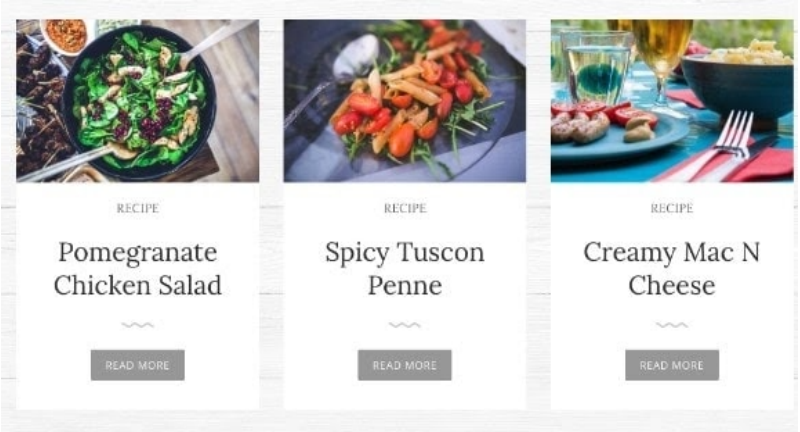
Buenos Aires aprende
Agencia de Habilidades para el Futuro

Ejemplos:

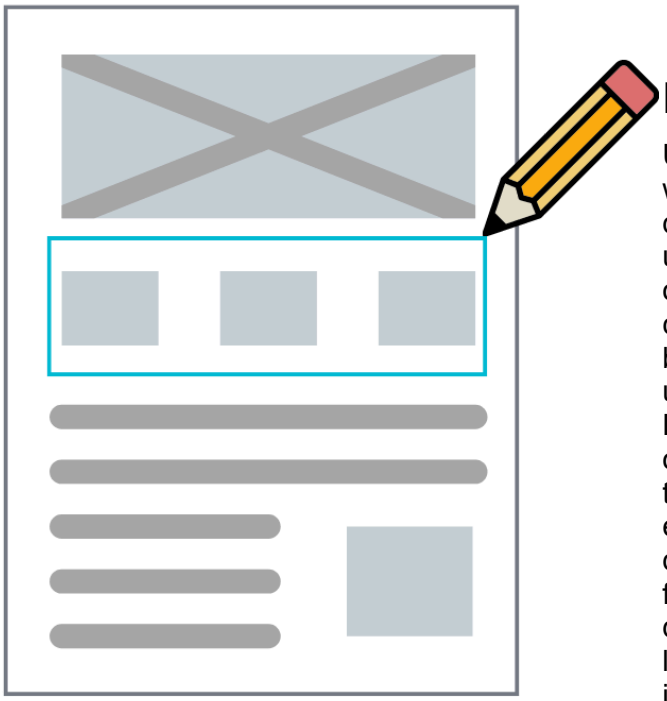
Veamos algunos ejemplos en sitios web donde se puede utilizar flexbox.



Algunos ejemplos pueden ser la barra de navegación donde hay enlaces unos al lado del otro, o cards (tarjetas). Pueden ser también galería de imágenes o layouts más complejos.



Buenos Aires aprende
Agencia de Habilidades para el Futuro



Layouts

Un layout, en el contexto del desarrollo web, se refiere a la estructura o disposición visual de los elementos en una página web. Es la forma en que se organizan y se distribuyen los componentes, como textos, imágenes, botones y otros elementos, para crear una interfaz coherente y atractiva. Determina la ubicación, el tamaño y el orden de los elementos en la página, y tiene un impacto significativo en la experiencia del usuario y en la facilidad de uso del sitio web. Un buen layout facilita la navegación, la comprensión de la información y la interacción con los elementos interactivos. Y lo más importante, marca el camino para los y las desarrolladoras que deben crear los códigos del sitio, cumpliendo una función de guía.

Display:Flex

Flexbox significa “caja flexible”. Quiere decir que lo que hace que los elementos se acomoden uno al lado del otro es su contenedor. Veamos un ejemplo.
Supongamos que tenemos el siguiente DIV y que contiene otros div más pequeños dentro.

El código es así:

HTML

```html
<div class="contenedor">
   <div class="caja uno"></div>
   <div class="caja dos"></div>
   <div class="caja tres"></div>
 </div>
```

CSS

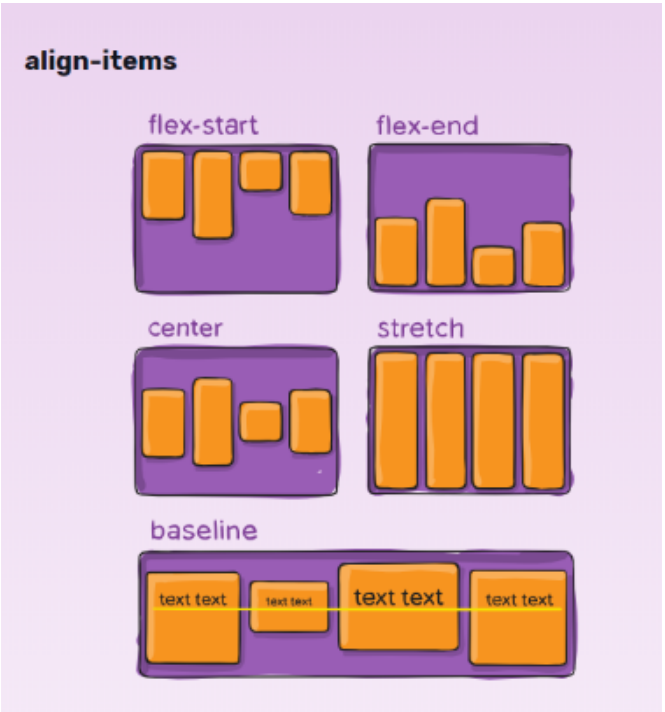
```css
.contenedor {
    border: 4px solid orangered ;
    margin: 0 auto;
    margin-top: 50px;
}
.caja {
    height: 50px;
    width: 50px;
    margin: 20px;
}
.uno{
    background-color: orange;
}
.dos{
    background-color: rgb(255, 102, 0);
}
.tres{
    background-color: rgb(255, 72, 0);
}
```

Para que las cajas se acomoden una al lado debemos aplicar la propiedad **display:flex**; en el contenedor. Esto afectará a los child o hijos directos del contenedor, no así a los nietos, es decir, a los elementos que se encuentren dentro de las cajitas naranjas. Esto posicionará los elementos de la siguiente manera sin importar si son elementos de línea o de bloque.

```css
.contenedor {
    border: 4px solid orangered ;
    margin: 0 auto;
    margin-top: 50px;
    display: flex;
}
```

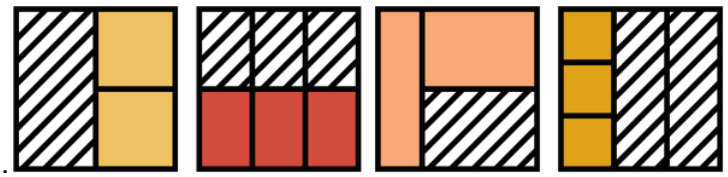
Con una sola línea de código podemos lograr este reordenamiento de elementos. Pero, no es sólo esto. Ahora queda identificar en nuestros layout qué elementos están dentro de contenedores como hicimos en la clase 2 cuando estudiamos contenedores DIV y SPAN, y aplicar display:flex; en el caso que sea necesario. Luego, alinear los elementos de manera vertical u horizontal con más propiedades para el contenedor flexible.

Alineación de elementos:



Ya aprendimos a hacer un contenedor flexible. Ahora, debemos alinear y centrar los elementos que contiene. Para ello existen propiedades para alinear los elementos de manera horizontal y vertical y que deben aplicarse al contenedor, no a los elementos que están adentro. Es al contenedor al que debemos pedirle cómo debe ordenar sus children.

Buenos Aires aprende
Agencia de Habilidades para el Futuro



justify-content: … ;

Justify content es una propiedad cuyos valores permite ordenar elementos de forma horizontal.

justify-content:start;

justify-content:end;

justify-content:center;

justify-content: space-evenly;

justify-content: space-around;

justify-content: space-between;

Buenos Aires aprende
Agencia de Habilidades para el Futuro

align-items: … ;

Align-items es una propiedad cuyos valores permite ordenar elementos de forma vertical.

align-items: start;

align-items: end;

align-items: center;

align-items: stretch;

align-items: baseline;

Self: Alineado individual

Además de acomodar los elementos de manera grupal, también podemos hacerlo de manera individual, en el caso de que se quiera que un elemento se ubique en un lugar diferente pero que siga formando parte del mismo contenedor. Esto se logra con **justify-self para el eje horizontal**, y **align-self para el eje vertical**, aplicándolo directamente sobre el elemento a acomodar y no en el contenedor.

En este caso el contenedor tiene **align-items:start**; pero el segundo contenedor tiene **align-self:end;** haciendo que se coloque de manera casi independiente en la base del contenedor y no en el inicio o start.

Media query

¿Alguna vez has abierto la misma página web en diferentes dispositivos como tablets o smartphones y has notado que se veía diferente?

Esto se debe a que el diseño se adapta al tamaño de pantalla del dispositivo. Esta adaptación es muy útil, ya que contribuye a la accesibilidad del sitio web y permite que los usuarios puedan visitarlo desde cualquier lugar.

Para lograr esta adaptación, se utilizan las media queries, que son consultas que el **CSS** realiza al navegador para determinar el tamaño de pantalla en el que se está visualizando la página. En función de este tamaño, el **CSS** aplica propiedades específicas para modificar los elementos según sea necesario. En la siguiente presentación, exploraremos más a fondo este concepto.

Responsive

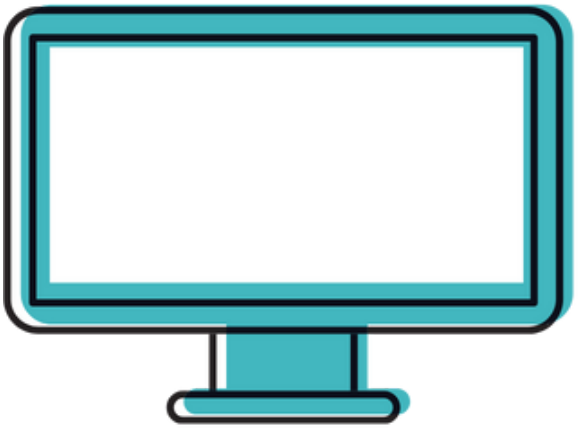
El término "responsive" o responsivo en español, se refiere a la capacidad de un sitio web para adaptarse y responder de manera óptima a diferentes tamaños de pantalla y dispositivos, brindando una experiencia de usuario adecuada en cada uno de ellos. Permite que el contenido y el diseño de un sitio web se ajusten automáticamente para adaptarse a dispositivos móviles, tabletas y pantallas de diferentes resoluciones.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

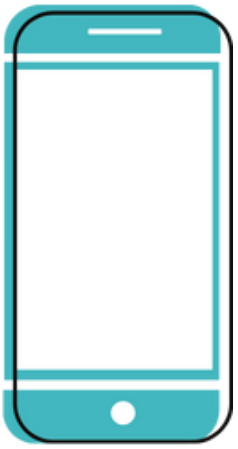
Cuando comenzamos a pensar en desarrollar una página web, **¿La imaginamos primero en computadora o en celular?**

Equipo Desktop

(Equipo Escritorio)

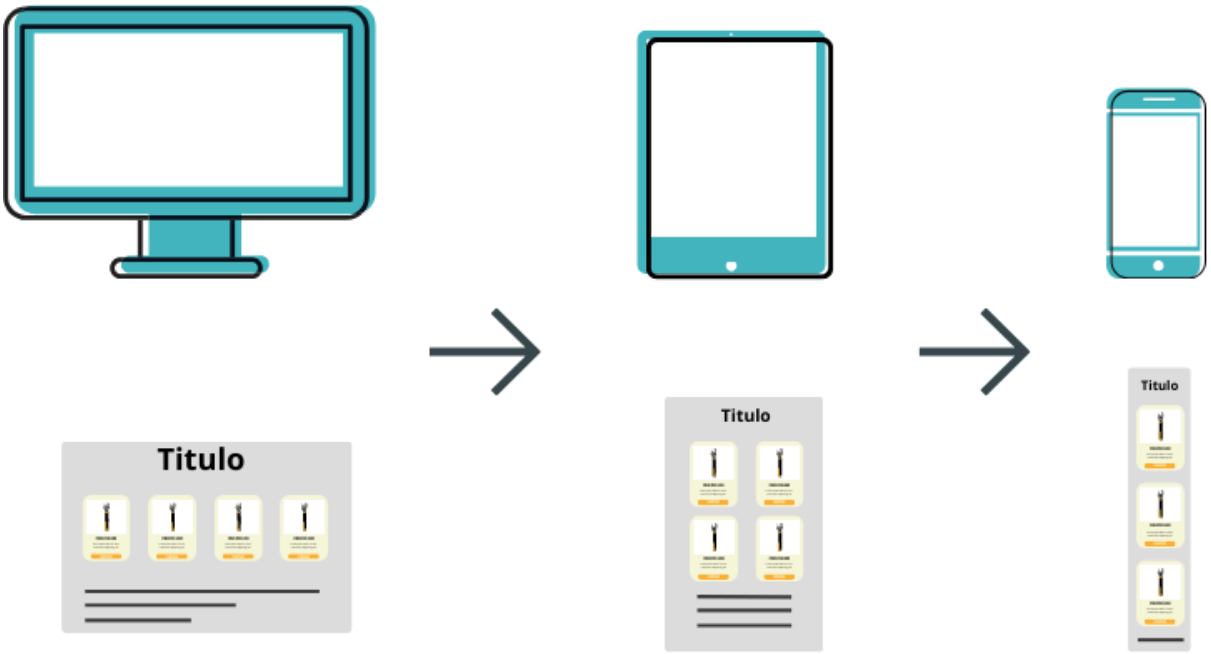


Equipo Mobile First
(Equipo Primero Mobil)

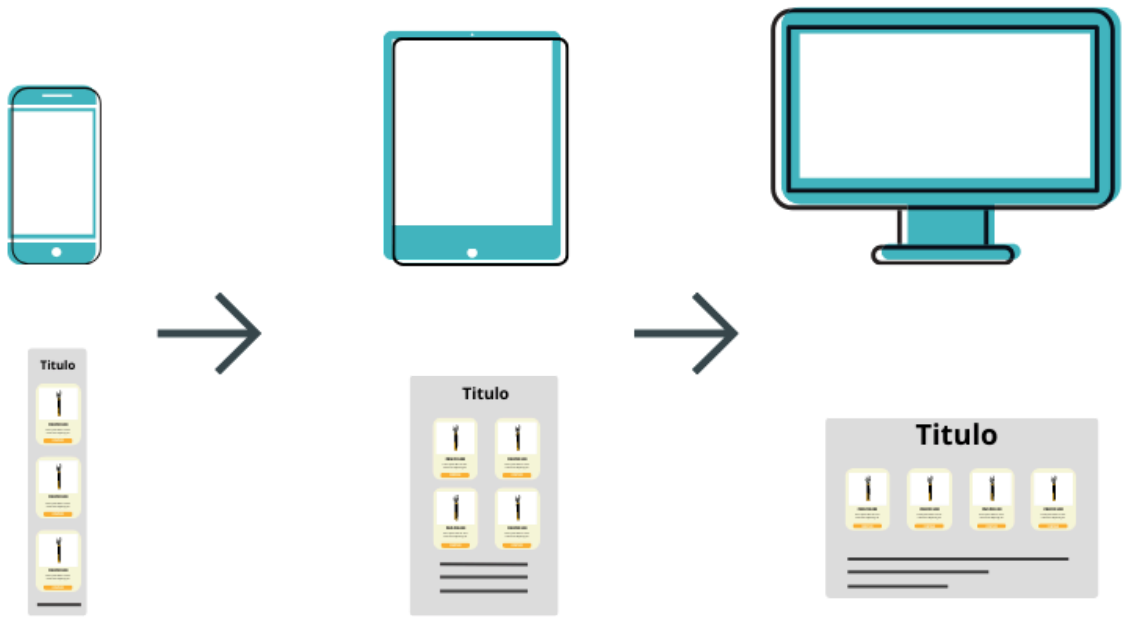


Buenos Aires aprende
Agencia de Habilidades para el Futuro

El equipo desktop primero piensa el desarrollo para computadoras y luego piensa cómo se va a adaptar a dispositivos más pequeños.



El equipo mobile primero piensa el desarrollo para dispositivos pequeños como celular y luego piensa cómo se va a adaptar a dispositivos más grandes.



Esta es una decisión que no debe tomarse a la ligera. Sea cual sea la mentalidad que elijamos para desarrollar el sitio, siempre vamos a tener que optimizarlo para todo tipo de dispositivos. Trabajar con mobile first, significa que el sitio web está originalmente optimizado para los dispositivos móviles y que la versión de escritorio es solo una adaptación de la versión móvil y lo mismo se aplica, aunque a la inversa, cuando pensamos en desktop first. Por eso, creo que lo mejor para decidir es considerar desde qué dispositivos tu sitio va a ser visto.

Media query

Las media queries (o consulta d medios en español) son una técnica de CSS que permite aplicar estilos diferentes según las características del dispositivo y la pantalla en la que se está visualizando un sitio web. Con las media queries, se pueden establecer condiciones y reglas CSS específicas para diferentes tamaños de pantalla, resoluciones, orientaciones o incluso características del dispositivo.
La sintaxis de la misma es:

@media (condición de tamaño de pantalla) {

Buenos Aires aprende
Agencia de Habilidades para el Futuro

```
  /* Estilos para pantallas con ancho máximo o mínimo de 000 px*/
}
```

Esta regla CSS puede ir escrita en cualquier parte del documento, pero es recomendable que siempre quede al final de todas las reglas CSS, ya que es lo último que se suele ajustar en el momento del desarrollo.

Veamos su estructura con ejemplos.

Esta media query se lee:
“Para todas las resoluciones menores o iguales a 768px, aplica las siguientes reglas”

Esta media query se lee:
“Para todas las resoluciones mayores o iguales a 768px, aplica las siguientes reglas”

Medidas estándar

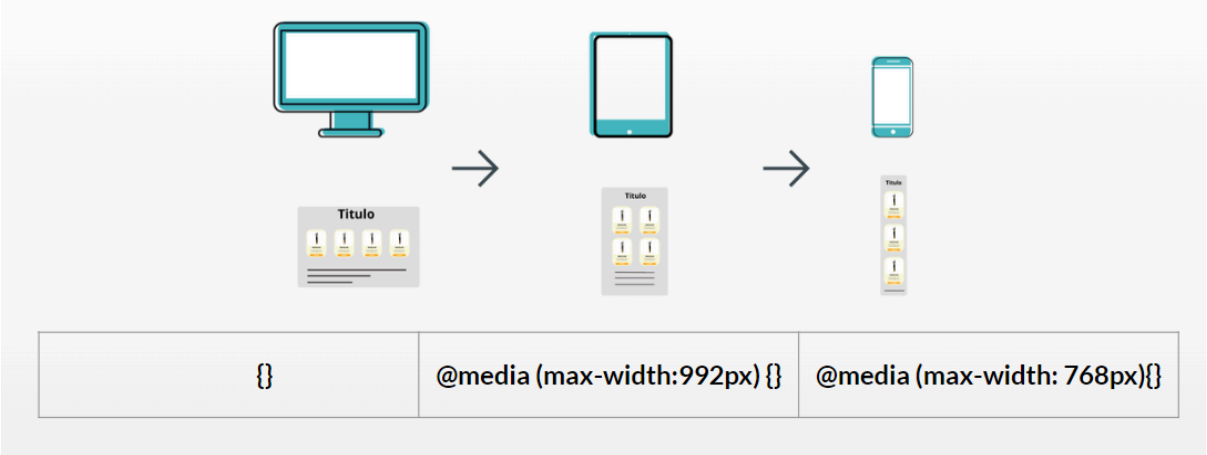
Si bien podemos usar muchas media queries, lo ideal es mantenernos dentro de un estándar, es por eso que te dejamos acá las medidas más populares que se usan al día de la fecha.

Dispositivos muy pequeños	Dispositivos pequeños	Dispositivos medianos	Dispositivos grandes	Dispositivos extra grandes
<576px	≥576px	≥768px	≥992px	≥1200px

Buenos Aires aprende
Agencia de Habilidades para el Futuro

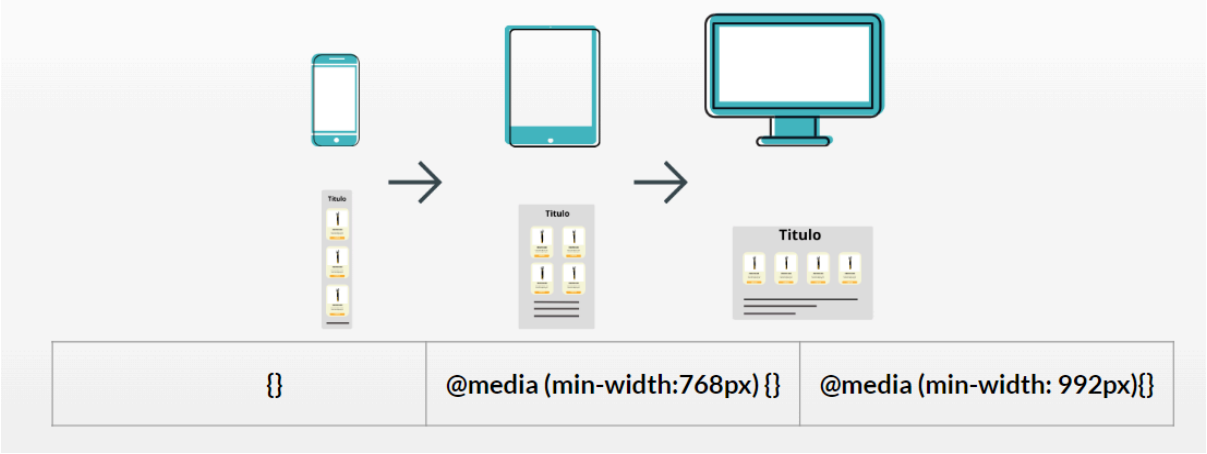
max-width

En este caso, somos equipo desktop, por lo tanto la media query a utilizar es con la condición max-width. La sentencia sería: asignar ciertas propiedades en dispositivos más pequeños que una cantidad determinada de px.



min-with

El equipo mobile first utilizará en su mayoría la media query con la condición min-with, y la sentencia sería: designar ciertas propiedades en dispositivos más grandes que tantos px.

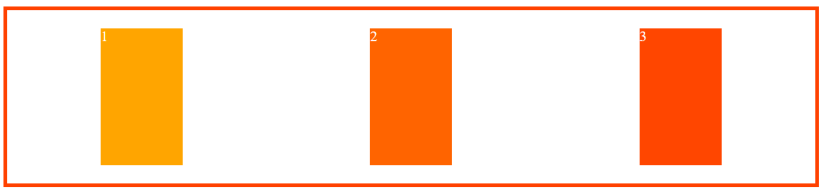


Buenos Aires aprende
Agencia de Habilidades para el Futuro

Si bien ambas media query se pueden combinar, es recomendable trabajar con una sola de estas estrategias. **Sólo se trabaja combinando ambas maneras en casos muy puntuales.**

Layout responsive

Ya aprendimos cómo acomodar los elementos que componen el sitio web uno al lado de los otros en un tamaño de pantalla de computadora.
También aprendimos sobre las media query, que son métodos para modificar propiedades del sitio web en diferentes tamaños de pantalla.
Ahora nos falta generar la acción de responsive, es decir, hacer que los elementos cambien su disposición según el tamaño de pantalla. Veamos cómo lograrlo con este ejemplo.
Supongamos que tenemos este contenedor con tres divs en su interior.



Buenos Aires aprende
Agencia de Habilidades para el Futuro

Para poder cambiar el tamaño de la pantalla debemos hacer clic derecho sobre el sitio web y seleccionar inspeccionar. (también podemos acceder al modo inspeccionar presionando la tecla f12) Luego, en la opción de visualización responsiva para poder cambiar el tamaño de la pantalla libremente.

Hasta el momento, su código es así.

HTML

```html
<div class="contenedor">
    <div class="caja uno">1</div>
    <div class="caja dos">2</div>
    <div class="caja tres">3</div>
</div>
```

CSS

```css
.contenedor {
  border: 4px solid orangered;
  margin: 0 auto;
  margin-top: 50px;
  display: flex;
  align-items: start;
  justify-content:space-around;
}
.caja {
  width: 90px;
  height: 150px;
  margin: 20px;
  display: flex;
  color: white;
}
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

```css
.uno {
  background-color: orange;
}
.dos {
  background-color: rgb(255, 102, 0);
}
.tres {
  background-color: rgb(255, 72, 0);
}
```

- Estructura HTML contenedor y cajas. Cada caja con sus clases
- CSS del contenedor y de las cajas
- CSS del color de cada caja

Aplicando media queries

Ahora, vamos a modificar las propiedades del contenedor y las disposiciones de las cajas cuando la pantalla mida menos de 768 px de ancho, y le asignaremos nuevas propiedades. Al final del código CSS entonces agregaremos el siguiente código. Veámoslo por partes.

```css
@media(max-width: 768px) {
  .contenedor{
    display: block;
    max-width: 300px;
  }
  .caja {
    margin: 0 auto;
    margin-top: 20px;
  }
}
```

- Declaramos la media query con el tamaño máximo de pantalla
- Le asignamos nuevas propiedades al contenedor:
-display:block; bloqueamos su flexibilidad
-Le damos un máximo de ancho de 300px
- Le asignamos nuevas propiedades a las cajas para centrarse y haya espacio entre ellas

Te invitamos a poner en práctica este ejercicio con un layout más completo:

Poné a prueba una disposición más compleja para los div.
¿Pasa algo con las imágenes? ¿Hacen falta más divs?

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Te recomendamos hacer uso de las media query en el tamaño de pantalla responsivo del inspector. Es decir, visualizar el sitio con un tamaño más pequeño y sobre esa vista realizar los cambios que creas necesarios.

Ahora sí, ¡a poner flexbox responsive a prueba!

Flexbox Playground

Preguntas de reflexión.

¿Por qué flexbox es más eficiente?
¿De qué manera puedo alinear elementos tanto de manera horizontal como vertical?
¿A qué nos referimos cuando hablamos de un sitio responsive?
¿Cuantas media query se recomiendan utilizar?

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Desafío N° 1:

¡Practica jugando con Flexbox froggy! 🐸

- El desafío para esta clase será completar los 24 niveles de este juego.

https://flexboxfroggy.com/#es


Buenos Aires
aprende
Buenos
Aires
Ciudad