

# Cheat Sheet

## Desarrollo Web



¿Necesitas ejemplos de código?

**CODERHOUSE**

¡Te damos la bienvenida!

Hemos recolectado una serie de atajos para guiar tus primeros pasos Desarrollo Web. Este recurso enriquecerá y optimizará tu tiempo de práctica.

## Atributo "value"

En este caso es el valor que se enviará al enviarse el formulario.  
Casillas de chequeo:

```
<form>  
  <div>Acepta términos y condiciones</div>  
  <input type="checkbox" name="acepta" value="1" />  
</form>
```

# FONT-FAMILY



Las tipografías Serif son aquellas que llevan remates, es decir, detalles adicionales en los bordes de las letras. El ejemplo por excelencia de Serif es la Times New Roman. Son muy usadas en los periódicos impresos, puesto que los detalles de las letras ayudan a seguir la lectura. Úsalas si quieres transmitir clasicismo, formalidad, precisión, tradición, delicadeza y/o refinamiento.



Por el contrario, las tipografías Sans Serif, como su nombre indica –sans es sin en francés–, carecen de estos detalles y también son denominadas de palo seco. Algunas de las más conocidas son la Arial o la Calibri. Se utilizan mucho en entornos digitales, puesto que los detalles son difíciles de plasmar en píxeles. Transmiten fuerza, modernidad, vanguardia, elegancia y actualidad, a los diseños y textos en los que se incluyen.

⇒ *Este contenido se vincula con la Clase 3*

# LINE-HEIGHT

## CSS

```
.interlineado {  
  line-height: 1.6;  
}
```

Se ve de la siguiente manera:

texto  
ejemplo

Valores posibles: none | <número> | <longitud> | <porcentaje>

⇒ Este contenido se vincula con la Clase 3

# PARALLAX

## CSS

```
.parallax {  
  
background-image: url("/img/maxresdefault.jpg");  
height: 600px;  
  
background-attachment: fixed;  
background-position: center;  
background-repeat: no-repeat;  
background-size: cover;  
  
margin-bottom: 50px;  
}
```

Se ve de la siguiente manera:

<https://codepen.io/mablen/pen/bZKPNB>

⇒ Este contenido se vincula con la Clase 3

# Menú con submenú

- El position se usa para hacer un menú que tenga un submenú emergente.
- La lista dentro de un list-item es absoluta.
- La sublista tiene display: none.

## Ejemplo



# HTML

```
<ul>
  <li><a href="">Item</a></li>
  <li><a href="">Item</a></li>
  <li><a href="">Item</a>
    <ul>
      <li><a href="">Subitem</a></li>
      <li><a href="">Subitem</a></li>
    </ul>
  </li>
  <li><a href="">Item</a></li>
</ul>
```

# CSS

```
ul {
  list-style: none;
  font-size: 0 /* truco por el uso de inline-block */
}
li {
  display: inline-block;
  width: 25%;
  position: relative;
  font-size: 14px
}
ul ul {
  position: absolute;
  display: none;
}
ul ul li {
  display: block;
}
ul li:hover ul {
  display: block;
}
```

## Flex-direction: row

Esta propiedad nos va a permitir especificar si queremos que los flex items se dispongan en filas o columnas.

```
.padre-flex {  
  display: flex;  
  flex-direction: row; /* predeterminado */  
}
```



## Flex-direction: row-reverse

Con el valor row-reverse (fila inversa) los flex items se apilan en una fila de derecha a izquierda.

```
.padre-flex {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



## Flex-direction: column

Con el valor column, los flex items se apilan en una columna de arriba hacia abajo.



```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
}
```

## Flex-direction: column-reverse

Con el valor column-reverse los flex items se apilan en una columna de abajo hacia arriba.



```
.padre-flex {  
  display: flex;  
  flex-direction: column-reverse;  
}
```

# Flex-wrap

Con flex-wrap vamos a poder especificar si queremos que los ítems puedan saltar a una nueva línea cuando el contenedor flexible se quede sin espacio.

## Flex-wrap: wrap-reverse



```
.padre-flex {  
  display: flex;  
  flex-wrap: wrap-reverse;  
}
```

## Flex-wrap: nowrap

```
.padre-flex {  
  display: flex;  
  flex-wrap: nowrap;  
}
```



## Flex-flow

Es la forma abreviada (shorthand) o rápida para las propiedades:

- flex-direction
- flex-wrap

Se pone primero la propiedad de flex-direction, y luego la de flex-wrap.

```
.padre-flex {  
  display: flex;  
  flex-flow: row nowrap;  
}
```



## Justify-content

Permite alinear los elementos, horizontal o verticalmente. Los siguientes ejemplos parten de la siguiente base:

```
.padre-flex {  
  display: flex;  
  flex-direction: row;  
}
```

## Justify-content: flex-end

Consiste en alinear los flex items (hijos) al lado derecho.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: flex-end;  
}
```



## Justify-content: center

Consiste en alinear los flex items (hijos) al centro.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: center;  
}
```



## Justify-content: flex-start

Consiste en alinear los flex items (hijos) al lado izquierdo.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: flex-start; /* predeterminado */  
}
```



## Justify-content: space-between

Es hacer que los flex items (hijos) tomen la misma distancia o espaciado entre ellos dentro del contenedor flexible, quedando el primer y último elemento pegados con los bordes del contenedor en el eje principal.



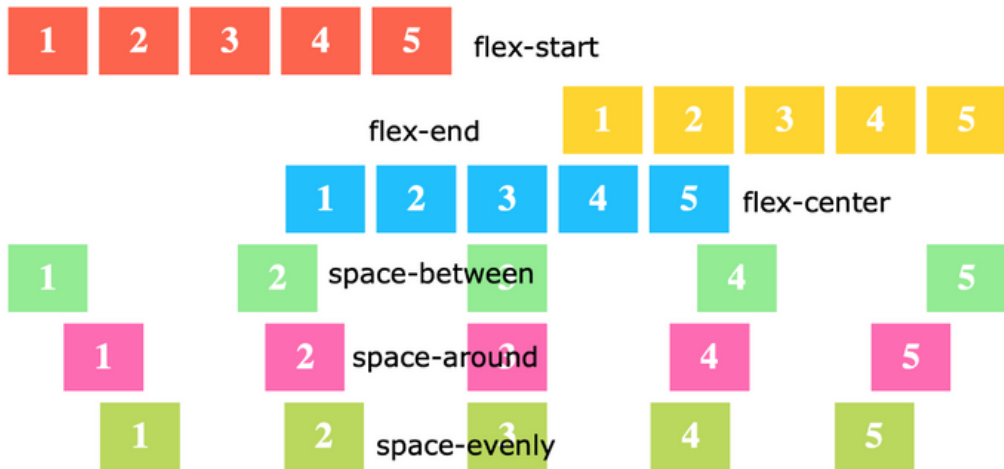
## Justify-content: space-around

Hace que el espacio entre los flex items (hijos) sea igual. No es lo mismo que `space-around`.

## Justify-content: space-around

Muestra los flex items (hijos) con el mismo espacio de separación entre sí. El espaciado entre los bordes lo toman del contenedor padre.





## Justify – content

Los siguientes ejemplos parten de la base del contenedor en "column", con altura definida:

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
}
```

## Justify-content: flex-start

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
  flex-start;  
  height: 300px;  
}
```



## Justify-content: flex-end

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-end;  
  height: 300px;  
}
```



## Justify-content: center

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  height: 300px;  
}
```



## Justify-content: space-between

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-between;  
  height: 300px;  
}
```



## Justify-content: space-around

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-around;  
  height: 300px;  
}
```



## Justify-content: space-evenly

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-evenly;  
  height: 300px;  
}
```



## Align-items

- Alinear los elementos verticales de forma horizontal. **flex-direction: column**
- Alinear los elementos horizontales de forma vertical. **flex-direction: row**



## Align-items: stretch

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: stretch;  
}
```



## Align-items: flex-start

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: flex-start; /* predeterminado */  
}
```



## Align-items: flex-end

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: flex-end; /* predeterminado */  
}
```



## Align-items: center

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: center; /* predeterminado */  
}
```

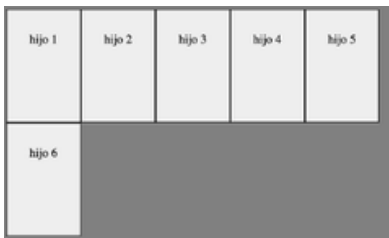


## Align-items: baseline

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: baseline; /* predeterminado */  
}
```



## Align-content: stretch



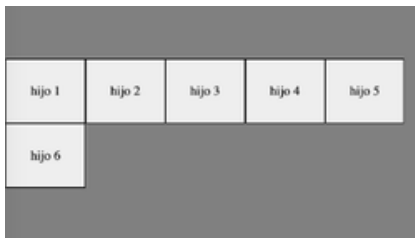
## Align-content: flex-start



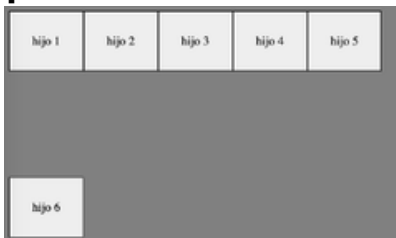
## Align-content: flex-end



## Align-content: center



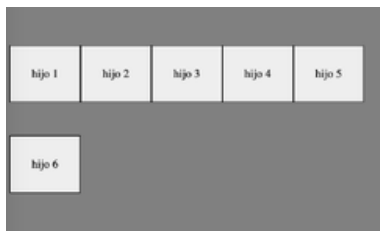
## Align-content: space-between



## Align-content: space-around



## Align-content: flex-end





# Posición de hijos (desde el padre)

Propiedad	Valores	Descripción
justify-items	start   end   center   stretch	Distribuye los elementos en el eje <b>horizontal</b>
align-items	start   end   center   stretch	Distribuye los elementos en el eje <b>vertical</b>

# Justify - Items y Align - Items



# Justify - Items: Center



# Justify - Items: Start



## Justify – Items: End

```
.padre {  
  justify-items: end;  
}
```



## Justify – Items: Center

```
.padre {  
  justify-items: center;  
}
```



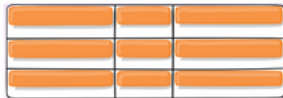
## Align – Items: Stretch

```
.padre {  
  align-items: stretch;  
  /* predeterminado */  
}
```



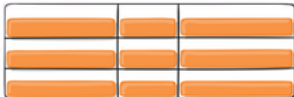
## Align – Items: Start

```
.padre {  
  align-items: start;  
}
```



## Align – Items: End

```
.padre {  
  align-items: end;  
}
```



## Align – Items: Center

```
.padre {  
  align-items: center;  
}
```



## Posición de Elementos

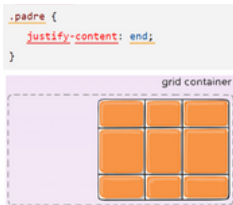
Propiedad	Valores	Afecta a
<u>justify-content</u>	<u>start</u>   <u>end</u>   <u>center</u>   <u>stretch</u>   <u>space-around</u>   <u>space-between</u>   <u>space-evenly</u>	Eje <b>horizontal</b>
<u>align-content</u>	<u>start</u>   <u>end</u>   <u>center</u>   <u>stretch</u>   <u>space-around</u>   <u>space-between</u>   <u>space-evenly</u>	Eje <b>vertical</b>

## Justify – Content: Start

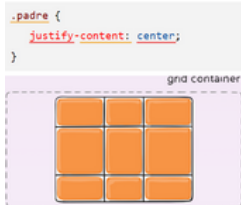
```
.padre {  
  justify-content: start;  
}
```



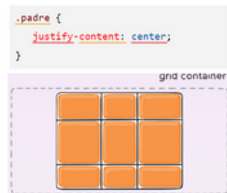
## Justify – Content: End



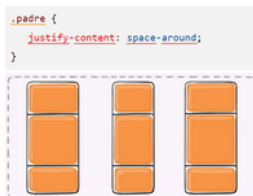
## Justify – Content: Center



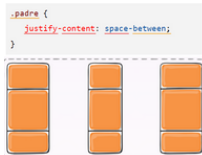
## Justify – Content: Stretch



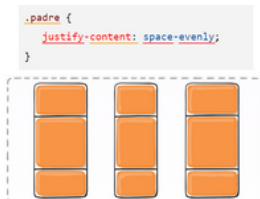
## Justify – Content: Space – Around



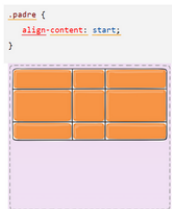
## Justify – Content: Space – Between



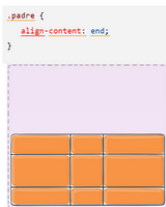
## Justify – Content: Space – Evenly



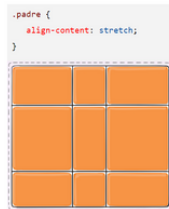
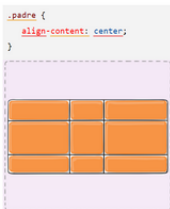
## Align – Content: Start



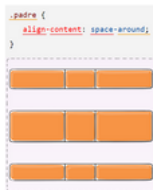
## Align – Content: End



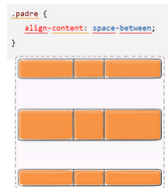
# Align – Content: Center    Align – Content: Stretch



## Align – Content: Space – Around



## Align – Content: Space – Between



## Align – Content: Space – Evenly



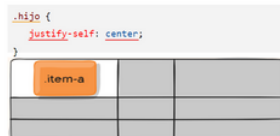
## Justify – Self: Start



## Justify – Self: End

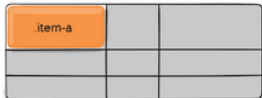


## Justify – Self: Center



## Justify – Self: Stretch

```
.hijo {  
  justify-self: stretch;  
}
```



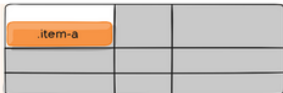
## Align – Self: Start

```
.hijo {  
  align-self: start;  
}
```



## Align – Self: End

```
.hijo {  
  align-self: end;  
}
```



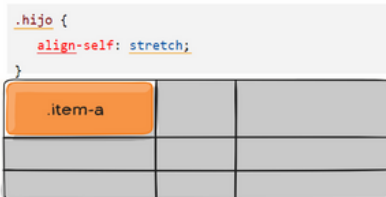
## Align – Self: Center

```
.hijo {  
  align-self: center;  
}
```





# Align – Self: Stretch



# ¡Éxitos!

**CODERHOUSE**