

Layouts:

**¿Cómo hago que las cosas queden
posicionadas donde quiero?**

Desarrollo de Software - UTN FRSFCO

2024

Formas de organizar el layout de un documento

Vamos a ver alguns (hay muchas más):

- Flexbox
- CSS Grid
- Bootstrap Responsive Grid

Flexbox

Flexbox

- Incorporado en CSS
- Permite alinear y distribuir espacio entre los items de un elemento contenedor
- Es importante distinguir entre 2 elementos:
 - El padre (container): tiene la habilidad de alterar las dimensiones y orden de los hijos
 - Hijos (items): los elementos que conforman el documento

Ver [A Complete Guide to Flexbox](#)

Flexbox

Propiedades del elemento **padre**

```
.container {  
  display: flex; /* or inline-flex */  
}
```

Esta propiedad debe ir siempre para indicar que el contenedor usa un modo de display flex

Flexbox

Propiedades del elemento **padre**

```
.container {  
  display: flex;  
  flex-direction: row | row-reverse | column | column-reverse;  
  flex-wrap: nowrap | wrap | wrap-reverse;  
  justify-content: flex-start | flex-end | center | space-between ...;  
  align-items: stretch | flex-start | flex-end | center | baseline ...;  
  align-content: flex-start | flex-end | center ...;  
  gap: 10px;  
  gap: 10px 20px; /* row-gap column gap */  
  row-gap: 10px;  
  column-gap: 20px;  
}
```

Flexbox

Propiedades del elemento padre

- `flex-direction` : el eje (fila/columna en el que se distribuyen los hijos)
- `flex-wrap` : permite distribuir los items en múltiples filas/columnas
- `justify-content` : define la alineación a lo largo del eje principal
- `align-items` : define la alineación a lo largo del eje perpendicular al principal

Flexbox

Propiedades de los elementos hijos

```
.item {  
  order: 5;  
  flex-grow: 4;  
  flex-shrink: 3;  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```


Flexbox

Propiedades de los elementos **hijos**

- `flex-grow` : permite que el elemento se agrande si hace falta
- `flex-shrink` : permite que el elemento se achique si hace falta
- `align-self` : permite sobrescribir el alineamiento por defecto para éste elemento.

Flexbox

Veamos un ejemplo

CSS Grid

CSS Grid

- Permite disponer los elementos en una grilla o tabla
- Sólo se configuran las propiedades del elemento padre (y los hijos se acomodan)
- Se pueden hacer algunas cosas "raras" (?)
- No es responsive respecto a las columnas como la de Bootstrap (los elementos se estiran junto con la grilla)

CSS Grid

Ejemplo #1: grilla con 2 columnas fijas

```
.grid-container {  
  display: grid;  
  grid-template-columns: 100px 200px;  
}
```

Veamos un ejemplo

CSS Grid

Ejemplo #2: grilla con 3 columnas, una de tamaño automático

```
.grid-container {  
  display: grid;  
  grid-template-columns: 100px 200px auto;  
}
```

Veamos un ejemplo

CSS Grid

Ejemplo #3: creando varias columnas con repeat

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, auto);  
}
```

Veamos un ejemplo

CSS Grid

Ejemplo #4: layout con nombre de areas

```
.item1 { grid-area: header; }  
.item2 { grid-area: menu; }  
.item3 { grid-area: main; }  
.item4 { grid-area: right; }  
.item5 { grid-area: footer; }  
  
.grid-container {  
  display: grid;  
  grid-template-areas:  
    'header header header header header header'  
    'menu main main main right right'  
    'menu footer footer footer footer footer';  
}
```

Veamos un ejemplo

Bootstrap

Bootstrap

“ Powerful, extensible, and feature-packed frontend toolkit. ”

- Biblioteca muy popular
- Tiene MUUUCHAS funciones, sólo nos interesa el sistema de grilla responsive
- Muchas bibliotecas de componentes tienen una interfaz similar

Bootstrap's Responsive Grid System

- Basada en flexbox
- Divide la pantalla verticalmente en 12 columnas
- La suma de los anchos de una fila debe dar 12
- 12 porque es un número con muchos divisores (la idea viene de la imprenta?)
- Prácticamente todas las bibliotecas implementan un sistema similar a este
- Se ajusta a los cambios de tamaño automáticamente (responsive)

Bootstrap

Instalación

Seguir estos pasos

- Incluyo el `.css`
- Incluyo los scripts al final del body (no hacen falta para la grilla, sí para otros componentes)

Estructura básica

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Estructura básica

- Nuestro contenido va dentro de los elementos de clase `col`
- No meter nada en el medio sino no anda
- La división se puede hacer de manera automática (como en el ejemplo anterior) o...

Bootstrap

- Se puede especificar distintos tamaños de columna para distintos viewports

```
<div class="container">
  <div class="row">
    <div class="col-lg-6 col-sm-12">
      1 of 2
    </div>
    <div class="col-lg-6 col-sm-12">
      2 of 2
    </div>
  </div>
</div>
```

Ver [breakpoints](#)

Bootstrap

- Si les interesa curiosarse qué más se puede hacer, chequear la lista de componentes
<https://getbootstrap.com/docs/4.0/components/>
- En la documentación se muestra el uso con Javascript "puro", cuando usamos React no se usa así

References

- [A Complete Guide to Flexbox](#)
- [CSS Grid Layout Guide](#)
- [Bootstrap Grid System](#)