

# DISEÑO DE INTERFACES WEB

Docente: Luis Úbeda

Autor: Marc Salom



# UNIDAD 4: USO DE ESTILOS. AVANZADO

- Sección 1: Diseño web “responsive”
  - Media queries
  - Sistema de grid “responsive
- Sección 2: Flex CSS
  - Introducción
  - Elementos que participan
  - Propiedades
  - Notas adicionales



## S2: FLEX CSS

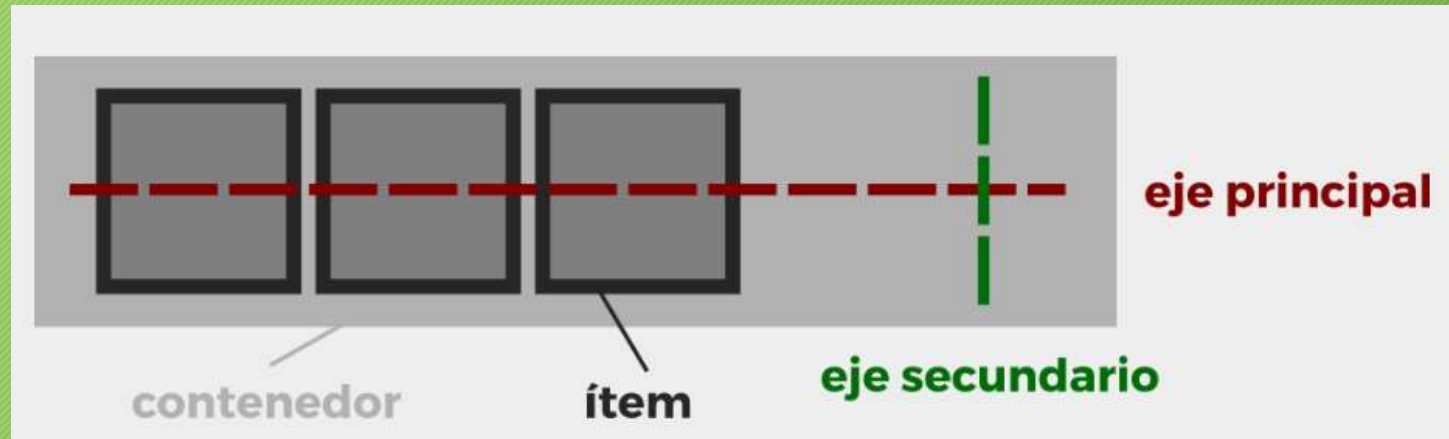
### 1. CSS Flex. Introducción

- En CSS, inicialmente se utilizaba el posicionamiento (static, relative, absolute...), los elementos en línea o en bloque (y derivados) o la propiedad float para realizar maquetaciones
- Usar estas técnicas en el diseño resulta bastante tosco que no encaja con los retos que tenemos en la actualidad: sistemas de escritorio, dispositivos móviles, múltiples resoluciones, etc...
- Flex (también llamado flexbox) es un sistema de elementos flexibles que llega con la idea de olvidar estos mecanismos y acostumbrarnos a una mecánica más potente, limpia y personalizable, en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños de una página web.



## S2: FLEX CSS

### 2. Elementos que participan en CSS Flex.



**Contenedor:** Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles. Observa que al contrario que muchas otras estructuras CSS, por norma general, en Flex establecemos las propiedades al elemento padre.

**Eje principal:** Los contenedores flexibles tendrán una orientación principal específica. Por defecto, el eje principal del contenedor flex es en horizontal (en fila).

**Eje secundario:** De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical (y viceversa).

**Ítem:** Cada uno de los hijos que tendrá el contenedor en su interior.



## S2: FLEX CSS

### 3. Propiedades

**display: flex** -> el contenedor se comporta como un bloque, ocupa todo el espacio horizontal disponible y los elementos ocupan

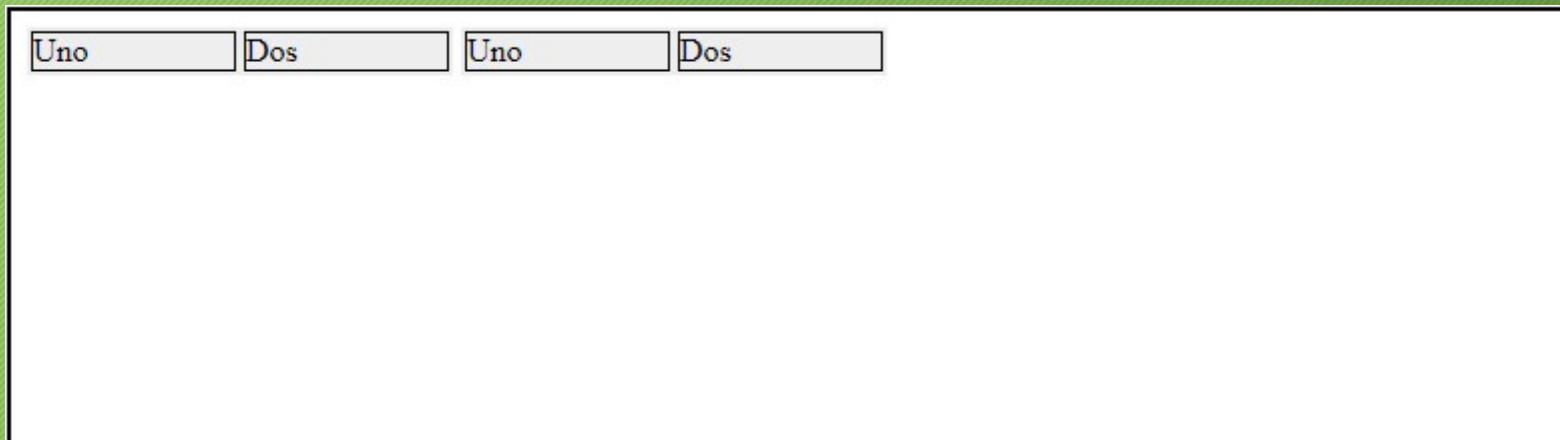
```
div { display: flex; margin-bottom: 5px; background-color: #eee; }  
  
p { width: 100px; border: black 1px solid; margin: 2px; }
```

Uno	Dos	Tres, con más texto que los otros elementos	Cuatro	
Uno	Dos	Tres, con más texto que los otros elementos	Cuatro	

## S2: FLEX CSS

**display: inline-flex** -> el contenedor se comporta como un elemento en línea. Ocupa sólo el espacio horizontal necesario.

```
div { display: inline-flex; margin-bottom: 5px; background-color: #eee; }  
  
p { width: 100px; border: black 1px solid; margin: 2px; }
```





## S2: FLEX CSS

**order: número** -> permite cambiar el orden de los elementos hijos que contienen la propiedad.

En el ejemplo, se ha dado la clase “primero” al cuarto párrafo y se muestra en primer lugar.

```
div { display: flex; }  
  
p { width: 100px; border: black 1px solid; margin: 2px; }  
  
p.primero { order: -1; }
```

Cuatro

Uno

Dos

Tres, con más  
texto que los  
otros  
elementos



## S2: FLEX CSS

**flex-basis** -> establece el tamaño inicial del elemento, antes de que se reparta en su caso el espacio libre.

Tiene como posibles valores:

- **content**: tamaño determinado por el contenido del elemento.
- **auto**: tamaño determinado por las propiedades width y height.
- **tamaño**: indica el tamaño inicial exacto. (en px, %, etc)

```
div { display: flex; }  
  
p { flex-basis: 120px; border: black 1px solid; margin: 2px; }
```

Uno	Dos	Tres, con más texto que los otros elementos	Cuatro
-----	-----	---	--------



## S2: FLEX CSS

**flex-grow** -> hace que los elementos crezcan hasta ocupar todo el espacio disponible en la dirección principal. Toma valores enteros que indican la proporción en que se reparte el espacio. Si tienen el mismo valor, se reparte a partes iguales.

```
div { display: flex; }  
  
p { flex-grow: 1; border: black 1px solid; margin: 2px; }  
  
p.mayor { flex-grow: 2; }
```

Uno

Dos

Tres (clase "mayor")

Cuatro



## S2: FLEX CSS

**flex-shrink** -> indica la proporción en que se estrechan los elementos cuando su contenido no cabe en el espacio disponible en la dirección principal. Si todos los elementos toman el mismo valor, el espacio se reparte a partes iguales.

```
div { display: flex; }  
  
p { flex-shrink: 1; border: black 1px solid; margin: 2px; }  
  
p.menor { flex-shrink: 2; }
```

Uno, con bastante texto, así que los elementos no van a caber en una sola línea

Dos, con bastante texto, así que los elementos no van a caber en una sola línea

Tres (clase "menor"), con una cantidad de texto similar al resto

Cuatro, con bastante texto, así que los elementos no van a caber en una sola línea



## S2: FLEX CSS

**flex** -> permite establecer simultáneamente las tres propiedades flex-grow, flex-shrink y flex-basis.

Admite 4 valores básicos pero nos centraremos en este:

- **número -> equivale a flex: número 1 0;**

El tamaño inicial de los elementos es proporcional al número indicado, y los elementos crecen de forma uniforme si hay espacio libre y se estrechan de forma uniforme si no hay espacio suficiente.

```
div { display: flex; }  
  
p { flex: 1; border: black 1px solid; margin: 2px; }
```

Uno, con bastante texto, así que los elementos no van a caber en una sola línea

Dos, con poco texto

Tres

Cuatro, con bastante texto, así que los elementos no van a caber en una sola línea



## S2: FLEX CSS

**justify-content** -> establece la forma en que se reparte el espacio libre disponible en la dirección principal.

Admite 4 valores: flex-start(ppio de la dirección principal), flex-end(final de la dirección principal), center (centrado), space-between(se reparte espacio entre los elementos), space-around(se reparte espacio entre elementos y extremos)

```
div { display: flex; justify-content: space-around; }  
  
p { border: black 1px solid; margin: 2px; }
```

Uno

Dos

Tres, con más texto que los otros elementos

Cuatro



## S2: FLEX CSS

**align-items** -> establece la forma en que se alinean los elementos en la dirección secundaria.

Admite 4 valores: flex-start(ppio de la dirección secundaria,arriba), flex-end(final de la dirección secundaria,abajo), center (centrado), stretch(elementos de mismo tamaño), baseline(los elementos se alinean según la primera línea de texto.)

```
div { display: flex; align-items: flex-start; }  
  
p { border: black 1px solid; margin: 2px; }
```

Uno, el más corto de todos	Dos, con más texto que el resto, para conseguir que los elementos no quepan en una sola línea y haya varios tamaños verticales distintos. No es fácil porque los navegadores equilibran los tamaños	Tres, más largo que el primero pero más corto que el segundo	Cuatro, con algo más de texto que el tercero, para ayudar a que los elementos no quepan en una sola línea
----------------------------	---	--	---



## S2: FLEX CSS

**align-content** -> establece la forma en que se reparte el espacio sobrante en la dirección secundaria.

Admite 4 valores: flex-start(ppio de la dirección secundaria,arriba), flex-end(final de la dirección secundaria,abajo), center (centrado), stretch(elementos aumentan su tamaño para ocupar todo el espacio disponible), space-between(el espacio disponible se reparte entre los elementos), space-around(espacio se reparte entre los extremos y los elementos).

```
div { display: flex; height: 10em;
      flex-wrap: wrap; align-content: space-around;
      background-color:#eee;
    }

p { width: 200px; border: black 1px solid; margin: 2px; }
```



Uno	Dos	Tres, con más texto que los otros elementos
Cuatro	Cinco	Seis



## S2: FLEX CSS

### 4. Notas adicionales

Para trabajar los ejercicios vistos en estas diapositivas, os recomiendo que preparéis en Visual Studio Code el siguiente fichero html que he usado en el primer ejemplo. De este, ya preparáis el resto.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="flex2.css">
</head>
<body>

  <div>
    <p>Uno</p>
    <p>Dos</p>
    <p>Tres, con más texto que los otros elementos<p>
    <p>Cuatro</p>
  </div>
  <div>
    <p>Uno</p>
    <p>Dos</p>
    <p>Tres, con más texto que los otros elementos<p>
    <p>Cuatro</p>
  </div>

</body>
</html>
```

**Enlace recomendado** para seguir estudiando Flex CSS y en el que me he basado:

<https://bartolomesintes.github.io/curso-cefire-html-css-2022-04/css/css-flexbox.html#flex>