LMSGI

Bloque 4 - Tema 3 - Modelo Visual

IES Valle del Jerte PLASENCIA
Bilingual Section

AENOR ER Empresa Registrada UNE-EN ISO 9001 ER-0271/2016

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

Informática y Comunicaciones

ÍNDICE

- Elemento time y atributo datatime
- Recordatorio de elementos semánticos: ejemplo blog
- Variables en CSS
- Modelo del formato visual
- Propiedad display
- Esquemas de posicionamiento → relativo, absoluto y fijo y desplazamiento de las cajas – top, right, left y bottom
- Elementos flotantes
- Detalles del modelo de formato visual
- Dimensiones de las cajas: width, height, …
- Unidades relativas al viewport
- Overflow
- visibility y display:none
- Capas: z-index
- Unidades, valores y funciones CSS - cal(), ...
- Flexbox y Grid
- Generar imágenes para maquetación
- JavaScript básico para capas

Elemento <time datatime>

- <time> → Representa un valor de fecha y hora. El equivalente legible por máquina puede ser representado en el atributo datatime
- El valor del atributo datatime, para que las máquinas lo entiendan bien, debe utilizar el estándar ISO 8601 (YYYY-MM-DDThh:mm:ssTZD).
- Fuente: Anatomía de un blog en HTML5 HTML

```
<time datetime="2034-07-13 09:00">El 13 de Julio de 2034 a las 9:00
UTC</time>
<time datetime="2034-07-13 09:00">Dentro de unos años</time>
<time datetime="2034-07-13 09:00">13 - Julio - 2034, 9 en punto</time>
```

Recordatorio de elementos semánticos: ejemplo blog

```
<body>
   <header>
       <h1> </h1>
       <nav>
         <ul>
           <li><a href="#">Home</a></li>
           <li><a href="#">Acerca
de</a></li>
           <li><a href="#">Contacto</a></li>
             ...
         </ul>
       </nav>
   </header>
```

```
  <main>
       <article>
           <header>
               <h2></h2>
               <time datetime="20xx-10-05">Hace 3 días</time>
           </header>
           <div>
            </div>
           <footer>
            </footer>
       </article>
       <nav>(Paginación de entradas) </nav>
    </main>
    <footer>
       Copyright, aviso legal, ...
   </footer>
</body>
```

Este ejemplo no incluye: section, aside o hgroup, otros elementos semánticos introducidos en HTML5

Variables en CSS

- Empiezan con **`--NombreVariable`**
- Se utilizan poniendo **`var(--NombreVariable[,valor sustitución])`** a la derecha de una propiedad
- El **valor de sustitución** se aplica cuando no se encuentra el valor de la variable
- **Les afecta el contexto** en el que se han declarado → es habitual declararlas en **`:root {}`** (*`:root` es una pseudo-clase para seleccionar el elemento raíz, es decir, **`html`**, y poder así utilizarse en todo el documento*)
- Diferencia mayúsculas de minúsculas, case sensitive
- *Trabajo voluntario →* *Preprocesador CSS - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web*

```css
:root {
   --shadow-nivel-1: 3px 4px 2px 0px;
   --color-fondo-principal: #cad3d0;
}
.media {
   padding: 20px;
   box-shadow: var(--shadow-nivel-1);
   background-color: var(--color-fondo-principal);
   color: var(--color-texto-principal, red);
}
```

- CSS Custom Properties for Cascading Variables Module Level 1
- Uso de propiedades personalizadas (variables) en CSS

Modelo del formato visual

- Documentación oficial en CSS Display Module Level 3
- Un resumen en display - CSS | MDN

MODELO DE FORMATO VISUAL

Propiedad	Descripción	Valores
display	Comportamiento del contenedor	[inline | block | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none]
position	Esquema de posicionamiento	[static | relative | absolute | fixed]
top right bottom left	Desplazamiento de la caja (respecto al límite superior, derecho, inferior o izquierdo del contenedor)	[<longitud> | <porcentaje> | auto]
float	Posicionamiento flotante	[left | right | none]
clear	Control de cajas adyacentes a los float	[none | left | right | both]
z-index	Solapamiento de niveles de capas	[auto | <entero_con_signo>]
direction	Sentido direccional de la escritura	[ltr | rtl]
unicode-bidi	Sentido direccional de la escritura	[normal | embed | bidi-override]

Propiedad Display

- **Se utiliza para cambiar el tipo de caja generado por un elemento → por ejemplo, se puede hacer que cualquier elemento actúe como si fuese una tabla o lista.**
- **Los valores más utilizados son : block (el elemento genera una caja de tipo bloque), inline (el elemento genera una caja en línea), inline-block y none.**
- **El valor none** → el elemento NO crea una caja invisible, simplemente no crea ninguna caja. CSS incluye mecanismos que permite a un elemento generar cajas que afectan a la composición pero ellas mismas no son visibles (propiedad visibility). → **también podemos utilizar el atributo hidden de html5**

Name:	***display***
Value:	inline | block | list-item | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none | inherit
Initial:	inline
Applies to:	all elements
Inherited:	no
Percentages:	N/A
Media:	all
Computed value:	see text

```
li {
    display: inline;
}
```

Elemento 1 Elemento 2 Elemento 3 Elemento 4

```
li {
    display: list-item;
}
```

1. Elemento 1
2. Elemento 2
3. Elemento 3
4. Elemento 4

Por ejemplo: si no es de tipo list-item pierde la numeración

Esquemas de posicionamiento

- Positioning - Learn web development | MDN
- CSS Positioned Layout Module Level 3
- En CSS una caja puede ser presentada de acuerdo a tres esquemas de posicionamiento.
  - **Flujo normal** → incluye el posicionamiento relativo (relative)
  - **Flotantes**
  - **Posicionamiento absoluto** → incluye el posicionamiento fijo (fixed)
- La propiedad **position** establece el tipo de posicionamiento → se dice que una caja está posicionada si position tiene un valor distinto de **static**, es decir es igual a **relative**, **absolute** o **fixed**, **en este caso pueden moverse por la página utilizando las propiedades top, right, bottom y left, las cuales aceptan valores negativos.**

Name:	*'position'*
Value:	static | relative | absolute | sticky | fixed
Initial:	static
Applies to:	all elements except table-column-group and table-column
Inherited:	no
Percentages:	N/A
Media:	visual
Computed value:	specified value
Animatable:	no

position: sticky → el elemento conmuta entre relative y fixed, en función del scroll. Es posicionado relative y conforme movemos el scroll cambia a fixed. Ver Posicionamiento fijo solo con CSS -

Nota: si ancestro utiliza overflow:hidden puede no funcionar en algunos navegadores

Posicionamiento relativo – position:relative

- Partiendo de su posición en el **flujo normal** del documento se realiza un desplazamiento.
- **Su sitio no es ocupado por otros elementos**, salvo que también se posicionen.
- Este tipo de posicionamiento **es considerado** parte del modelo de posicionamiento del **flujo normal**.

```
#myBox {
  position: relative;
  left: 20px;
  top: 20px;
}
```

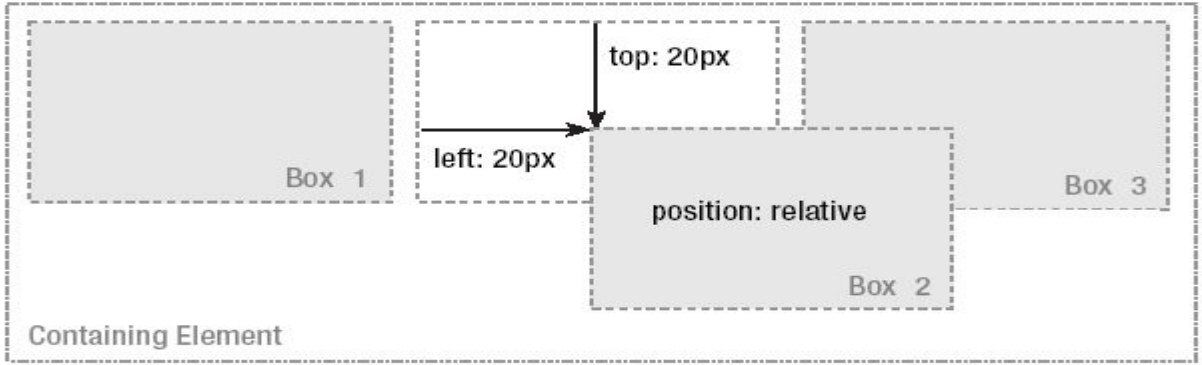


Figure 2-10. Relatively positioning an element

Posicionamiento absoluto – position:absolute

- El elemento se posiciona en relación a su antecesor más cercano que **esté posicionado** → si no tiene ningún antecesor posicionado se posicionará en relación con el bloque de contención inicial (en páginas HTML será <body>)
- **Pierde su sitio en el flujo normal** y será ocupado por otros elementos.

position:absolute;

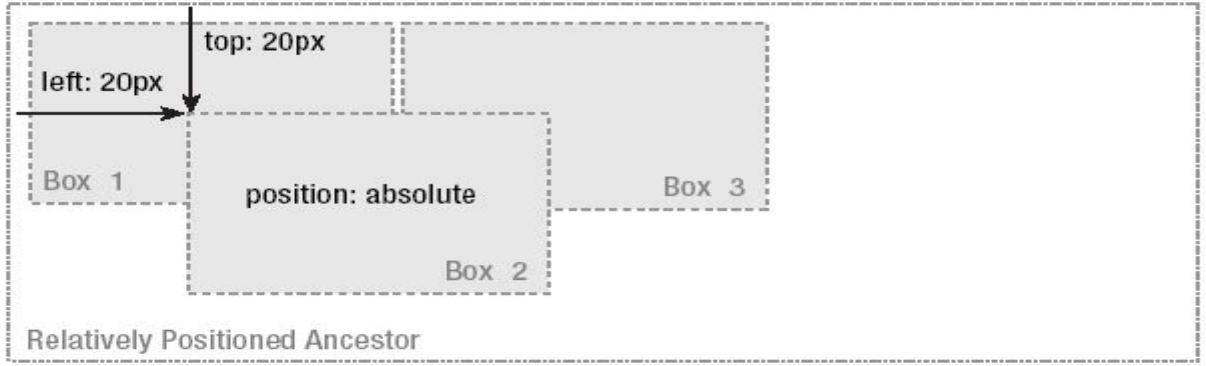
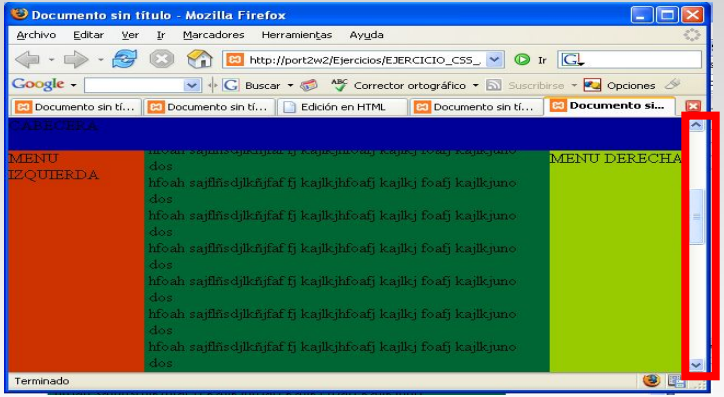


Figure 2-11. Absolutely positioning an element

Posicionamiento Fijo – position:fixed

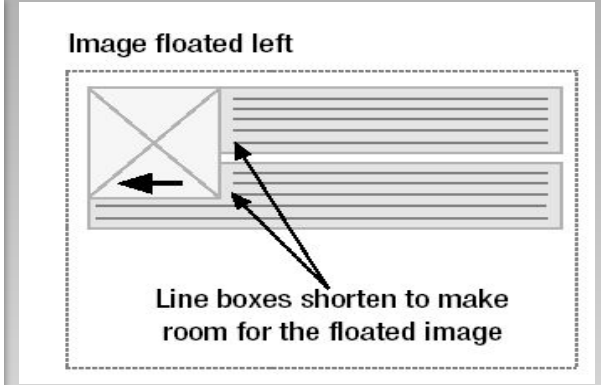
- **Es una subcategoría de posicionamiento absoluto**.
- Estos elementos siempre están en la misma posición en la ventana, independientemente que otros puedan moverse por la acción del scroll.
- Se posiciona en relación al **viewport** (ventana del navegador)

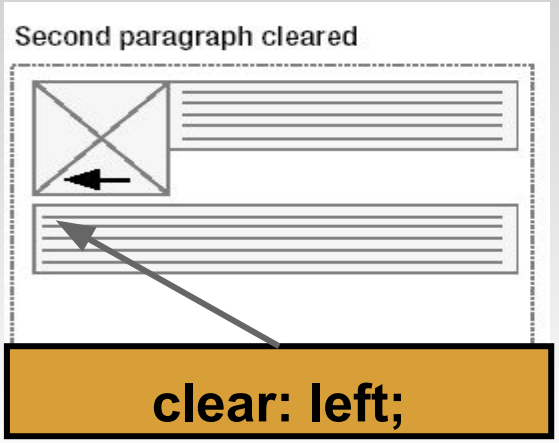


La caja central (verde) se desplaza y las otras tres se mantienen fija → todas tienen position:fixed a excepción de la central (verde)

Elementos flotantes (I)

- [Floats - Aprende sobre desarrollo web | MDN]
- Propiedad **`float: left | right`**
- Similar al atributo align de HTML sobre imágenes e iframe
- La caja flota a izquierda o derecha hasta tocar el límite de la caja contenedora u **otro elemento flotante** → el resto de elementos puede fluir a lo largo de su costado
- Al igual que las absolutas **pierden su posición en el flujo normal** del documento
- Los márgenes de las cajas flotantes nunca se cierran con los márgenes de las cajas adyacentes.
- Mediante la propiedad **`clear`** se puede romper la flotabilidad a la izquierda (left), derecha (right) o ambos (both). Es similar a <br clear="----"> en HTML





Elementos flotantes (y II)

Las cajas no posicionadas ni flotantes fluyen como si la flotante no existiera (se meten debajo), pero los elementos en línea NO, utilizan la flotante como “pared”

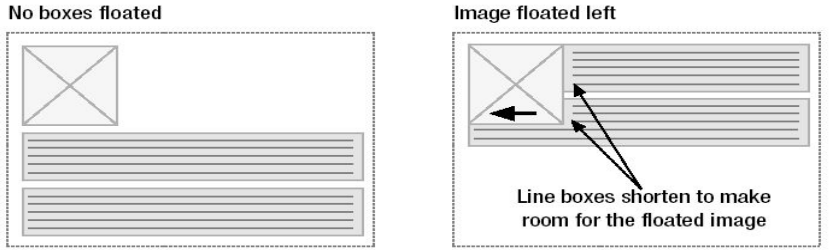


Figure 2-16. Line boxes shorten when next to a float.

Figure 2-13. Example of an element being floated right

La caja 2 (Box 2) cuando **no es flotante** se sitúa debajo de la caja 1 (Box 1), pero cuando es flotante NO

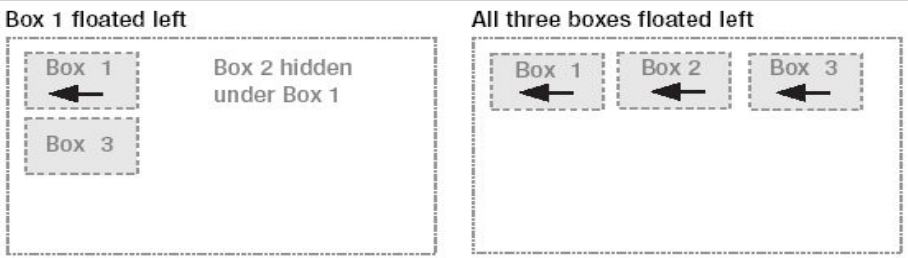


Figure 2-14. Example of elements being floated left

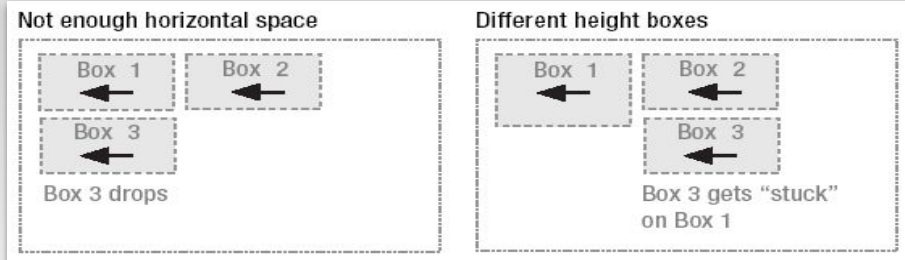
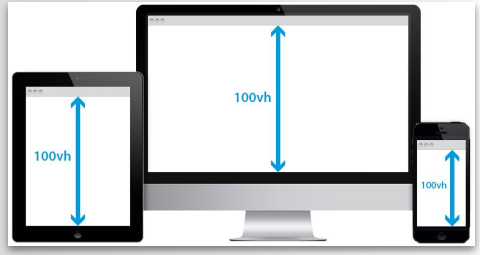


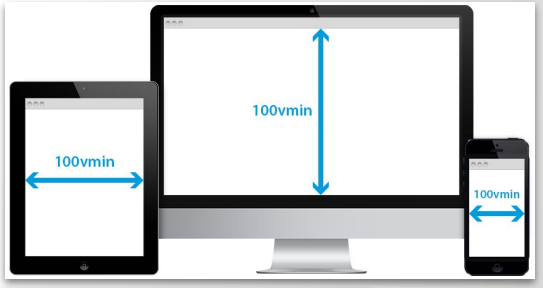
Figure 2-15. If there is not enough available horizontal space, floated elements will drop down until there is.

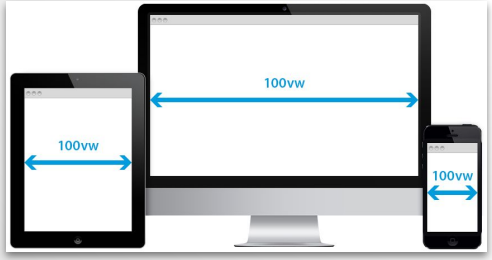
Unidades relativas al viewport

Viewport units CSS. Qué es y cómo utilizar vw, vh, vmin y vmax

- width: 100vw; → 100% del ancho de la ventana del navegador
- height: 50vh; → 50% del alto de la ventana del navegador
- line-height: 3vh; → altura de línea del 3% de la altura de la ventana del navegador
- Dependiendo del sistema operativo/navegador el scroll de body forma parte o no del viewport (**en MAC no, en Windows sí**) → https://codepen.io/antoniobp/pen/ZEvzqMj







Dimensiones de las cajas: width, heigth, … (I)

- Las dimensiones de una caja vienen determinadas por diferentes factores: si es en línea o en bloque, por el contenido, si tiene establecidas las propiedades width, height, etc, …. → https://www.w3.org/TR/CSS22/visuren.html
- Propiedades que utilizaremos:
  - width (max-content, …), height (max-content,...), min-width, max-width, min-height y max-height -- (ver https://css-tricks.com/almanac/properties/w/width/)
- También influye los valores dados de border y padding (no margin).
- **Solo se puede utilizar en cajas de tipo bloque y reemplazadas** (img, object, …) → no para elementos a nivel de línea, en las que el ancho viene determinado por su contenido.
- **No** se pueden utilizar valores **negativos**.
- **Porcentajes** en la altura son relativos a la altura del bloque de contención → **Si la altura del bloque de contención no es especificada explícitamente y el elemento no está posicionado de forma absoluta el valor se interpretará como auto**.
- Una altura en porcentaje de un elemento raíz es relativa al acceso visual → por ejemplo para body sería el tamaño de la ventana del navegador
- La propiedad **line-height** indica la altura de línea para los elementos de tipo línea → **interlineado**.

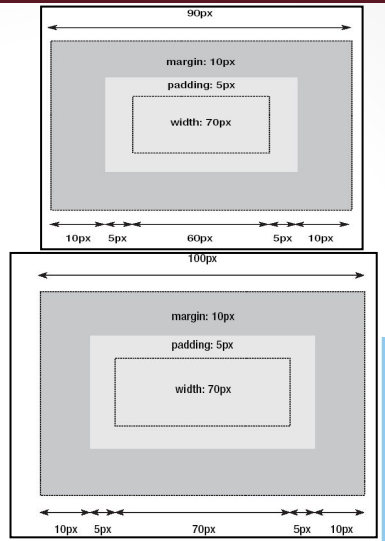
vertical-align solo aplicable a elementos inline-level y 'table-cell'

DETALLES DEL MODELO DE FORMATO VISUAL

Propiedad	Descripción	Valores
width	Ancho	[<longitud> | <porcentaje> | **auto**]
min-width	Ancho mínimo	[<longitud> | <porcentaje>]
max-width	Ancho máximo	[<longitud> | <porcentaje> | **none**]
height	Alto	[<longitud> | <porcentaje> | **auto**]
min-height	Alto mínimo	[<longitud> | <porcentaje>]
max-height	Alto máximo	[<longitud> | <porcentaje> | **none**]
line-height	Altura entre las bases del texto	[**normal** | <número> | <longitud> | <porcentaje>]
vertical-align	Alineación vertical del texto	[**baseline** | sub | super | top | text-top | middle | bottom | text-bottom | <porcentaje> | <longitud>]

Dimensiones de las cajas: width, heigth, … (y II)

- **Ancho "total" = width + padding + border**
- **Alto "total" = height + padding + border**
- Con la propiedad box-sizing podemos cambiar cómo se calcula el tamaño total de la caja. Por ejemplo, si indicamos width: 100px; border: 10px; padding: 20px; y decimos:
  - **box-sizing: content-box;** → es el valor por defecto. Al ancho se suma los bordes y el padding → ancho total de 100+10x2+20x2 = 160px
  - **box-sizing: border-box;** → se resta al ancho indicado los bordes y el padding → ancho total 100 (disponible para el contenido 100-10x2-20x2=40px;)
- Podemos controlar el **desbordamiento** del contenido con la propiedad **overflow (visible, auto, hidden, scroll)** → *más detalle en siguiente diapositiva*
- Con la propiedad resize **(none, both, horizontal, vertical)** permitimos cambiar el tamaño al usuario. **Se utiliza junto con *overflow*.**




```
.div1 {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
    box-sizing: border-box;
}

.div2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
    box-sizing: border-box;
}
```

Both divs are the same size now!

Hooray!

Name:	*'resize'*
Value:	none | both | horizontal | vertical
Initial:	none
Applies to:	elements with 'overflow' other than visible, and optionally replaced elements such as images, videos, and iframes
Inherited:	no
Percentages:	N/A
Media:	visual
Computed value:	as specified
Canonical order:	per grammar
Animation type:	discrete

```
div {
    resize: horizontal;
    overflow: auto;
}
```

This div element is resizable by the user (works in Chrome, Firefox, Safari and Opera).

Overflow

- El **contenido** de una caja puede no caber en las dimensiones especificadas, por lo que **se desborda**, lo que se puede controlar con la propiedad **overflow**:
  - **Visible** → el contenido desborda la caja y es visible
  - **Hidden** → el contenido desbordado no es visible
  - **Scroll** → el contenido se oculta y el navegador tiene que incluir barras de scroll para poder verlo
  - **Auto** → depende del navegador, normalmente es como scroll
- Otras propiedades (no las veremos, están en borrador): overflow-wrap, overflow-x y overflow-y

```
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
    overflow: visible;
}
```

Supercalifragilisticexpialidocious

```
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
    overflow: hidden;
}
```

```
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
    overflow: auto;
}
```

ercalifragilisticex

Supercalifragilisti

```
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
    overflow: scroll;
}
```

Supercalifragi

```
div { width : 100px; height: 100px;
      border: thin solid red;
      }

blockquote   { width : 125px; height : 100px;
      margin-top: 50px; margin-left: 50px;
      border: thin dashed black
      }

cite { display: block;
       text-align : right;
       border: none
       }
```

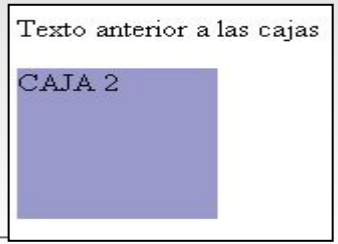
overflow:hidden;

overflow:visible;

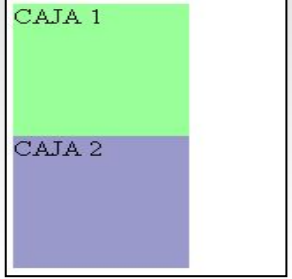
visibility y display:none

- visibility indica si la caja es visible (**visible**) o no (**hidden**), pero el sitio se “reserva”
- **display:none** → no genera caja
- Otro valor posible para visibility es **collapse**, pensado para tablas (filas, columnas, ...), sobre otro elemento tiene el mismo significado que hidden.
- HTML 5 ha incorporado un **nuevo atributo global hidden** que hace que el elemento no se muestre, no generando caja

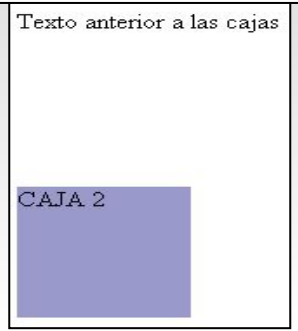
```
.cajal {
    height: 100px;
    width: 100px;
    background-color: #99FF99;
    display: none;
}
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
}
```




```
.cajal {
    height: 100px;
    width: 100px;
    background-color: #99FF99;
    visibility: visible;
}
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
}
```

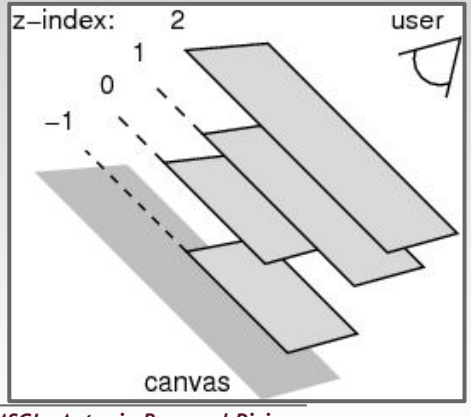



```
.cajal {
    height: 100px;
    width: 100px;
    background-color: #99FF99;
    visibility: hidden;
}
.caja2 {
    height: 100px;
    width: 100px;
    background-color: #9999CC;
}
```



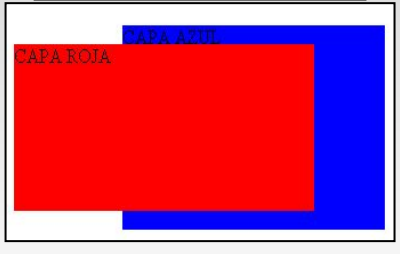
Capas: z-index

- Al aplicar los diferentes esquemas de posicionamiento puede ocurrir que unas cajas “oculten” a otras → **capas**
- **z-index** → indica el orden en el nivel de apilamiento. De las capas superpuestas será **visible aquella cuyo z-index sea mayor**.



Name:	*'z-index'*
Value:	auto | <integer>
Initial:	auto
Applies to:	positioned elements
Inherited:	no
Percentages:	N/A
Media:	visual
Computed value:	as specified
Animatable:	<integer>

```
#capa_roja {
    position:absolute;
    left:46px;
    top:352px;
    width:263px;
    height:133px;
    z-index:2;
    background-color: red;
}
#capa_azul {
    position:absolute;
    left:141px;
    top:337px;
    width:230px;
    height:163px;
    z-index:1;
    background-color: blue;
}
```




```
#capa_roja {
    position:absolute;
    left:46px;
    top:352px;
    width:263px;
    height:133px;
    z-index:2;
    background-color: red;
}
#capa_azul {
    position:absolute;
    left:141px;
    top:337px;
    width:230px;
    height:163px;
    z-index:11;
    background-color: blue;
}
```

Unidades, valores y funciones CSS - cal(),...

```css
.banner {
  position: absolute;
  left: 40px;
  width: 90%;                /* salvaguarda para navegadores que no reconocen calc() */
  width: calc(100% - 80px);
  border: solid black 1px;
  box-shadow: 1px 2px;
  background-color: yellow;
  padding: 6px;
  text-align: center;
}
```

```html
<div class="banner">This is a banner!</div>
```

calc - CSS | MDN

CSS Functions

CSS functions are used as a value for various CSS properties.

CSS Functions Reference

Function	Description
attr()	Returns the value of an attribute of the selected element
calc()	Allows you to perform calculations to determine CSS property values
cubic-bezier()	Defines a Cubic Bezier curve
hsl()	Defines colors using the Hue-Saturation-Lightness model (HSL)
hsla()	Defines colors using the Hue-Saturation-Lightness-Alpha model (HSLA)
linear-gradient()	Sets a linear gradient as the background image. Define at least two colors (top to bottom)
radial-gradient()	Sets a radial gradient as the background image. Define at least two colors (center to edges)
repeating-linear-gradient()	Repeats a linear gradient
repeating-radial-gradient()	Repeats a radial gradient
rgb()	Defines colors using the Red-Green-Blue model (RGB)
rgba()	Defines colors using the Red-Green-Blue-Alpha model (RGBA)
var()	Inserts the value of a custom property

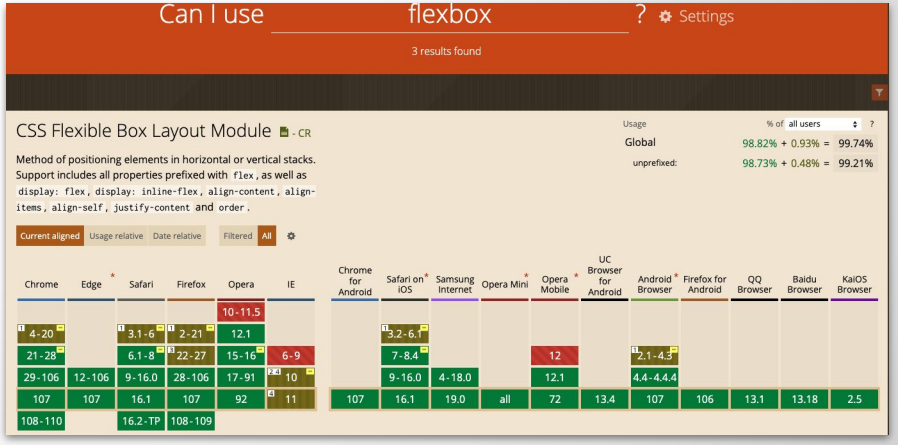
CSS Values and Units Module Level 3

Flexbox - Introducción (I)

- **“Flexible Box Layout“, “Diseño de caja flexible“** → impulso importante de diseños **responsive** (*permite colocar los elementos de una página para que se comporten de forma predecible cuando el diseño de la página debe acomodarse a diferentes tamaños de pantalla y diferentes dispositivos)* → permite crear elementos flexibles que se adaptan a su contenedor, pudiendo alinearlos horizontal y verticalmente, ajustarlos según tamaño, …
- Se utiliza para alinear los elementos en una dimensión, para dos dimensiones mejor **grid**

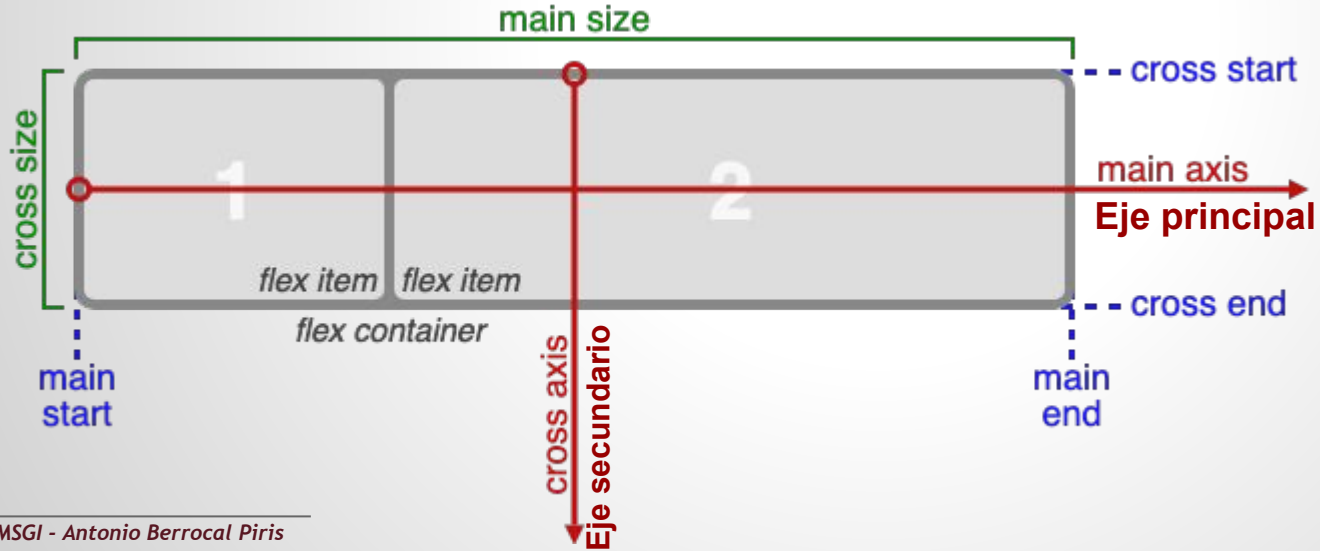
Ampliamente soportado

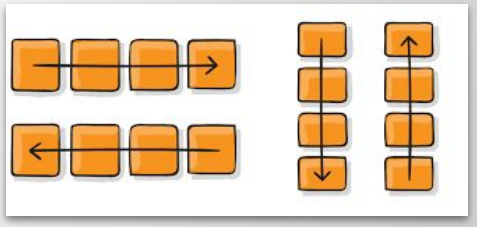
CSS Flexible Box Layout Module Level 1



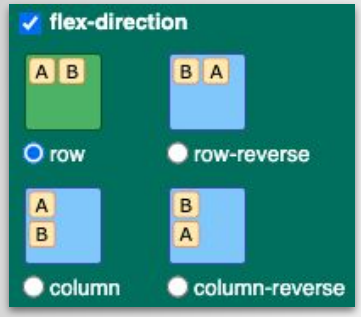
Flexbox - Introducción (y II)

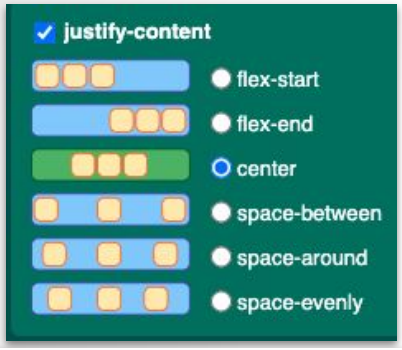
- Extiende la definición de la propiedad `display`.
- No puede utilizarse en `::first-line` ni `::first-letter`
- Un contenedor flexible es la caja generada por un elemento con `display: flex` o `inline-flex`.
- Los “hijos” de un contenedor flex se denominan flex items, estos se distribuyen utilizando el modelo de diseño flexible.

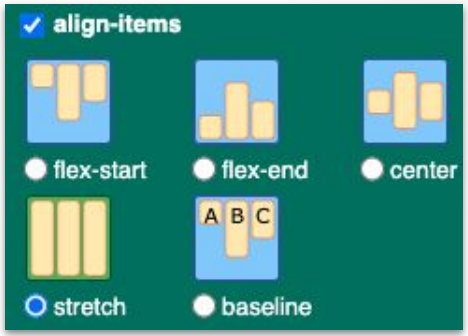


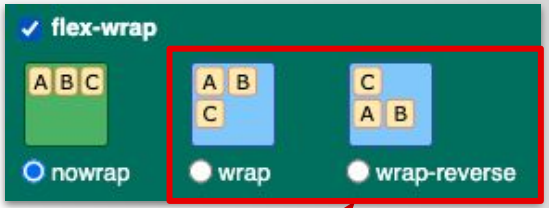


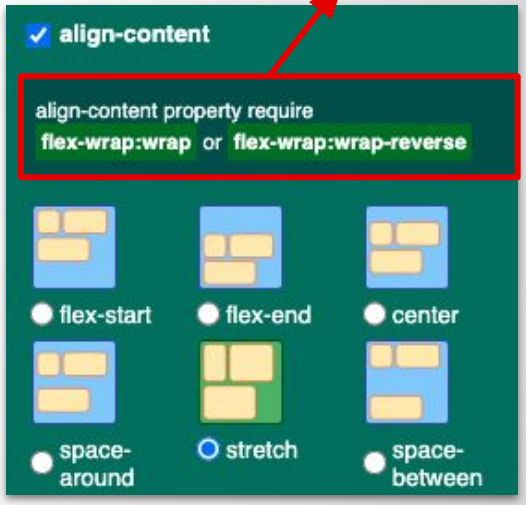
Flexbox - Propiedades del contenedor Flex











`justify-content` → alínea los elementos respecto al **main axis**
`align-items` → alínea los elementos respecto al **cross axis**
`align-content` → alínea los elementos **entre líneas** respecto al **cross axis**
gap → `row-gap` (solo se aplica si `flex-direction:column`) y `column-gap` (solo se aplica si `flex-direction:row`) → tamaño del hueco entre ítems desde el elemento padre contenedor
→ sustituye a padding o margin en los ítems

Recuerda → por defecto main axis es el horizontal y el cross axis el vertical, pero podría cambiar si lo indicamos en flex-direction

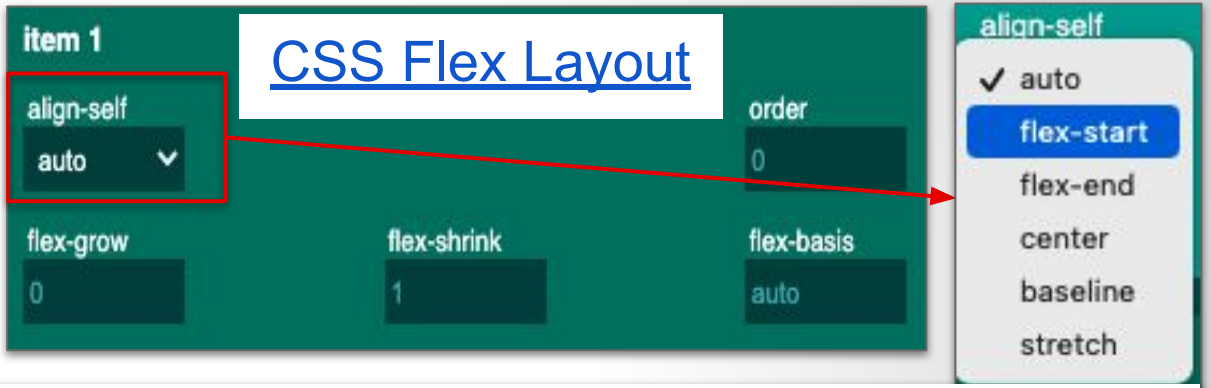
[CSS Flex Layout]

```
.flex-container {
   display: flex; justify-content: flex-start;
   align-items: flex-end; gap: 5px; }
```

Flexbox - Propiedades de los ítems en un contenedor flex

Un contenedor flex contiene varios ítems, y estos a su vez pueden incluir un contenedor flex

```css
.item1 {
   /* flex:0 0 100px; */
   flex-grow: 0;
   flex-shrink: 0;
   flex-basis: 100px;
   align-self: flex-end;
   order: 1;
}
```



`order` → posición en la lista

`align-self` → alínea el ítem respecto al **cross axis. El comportamiento es idéntico al de align-items, la diferencia es que se aplica a un único flex-item**

`flex-basis` → tamaño **base** ("punto de partida") de los ítems antes de aplicar la distribución del espacio. Además del tamaño específico (unidades, porcentaje, …) se puede indicar **`content`** para ajuste automático al contenido del elemento, es el valor por defecto. Si `flex-direction:row` o `row-reverse`, `flex-basis` será equivalente a **`width`**, y si es `column` o `column-reverse` a **`height`**. **Este tamaño podrá crecer o decrecer.**

`flex-grow`→ número entero positivo que indica el factor de **crecimiento** del ítem para rellenar el espacio disponible, en caso de haberlo.

`flex-shrink` → número entero positivo que indica el factor de **decrecimiento** del ítem para que todos quepan en una sola línea.

flex → propiedad abreviada de flex-basis flex-grow y flex-shrink

Importante → ***Flexbox Tester · MadebyMike***

Flexbox - Ejemplo centrado vertical y horizontal

```
.flex-container {
    display: flex;
    justify-content: center;
    align-items: center;
}
```

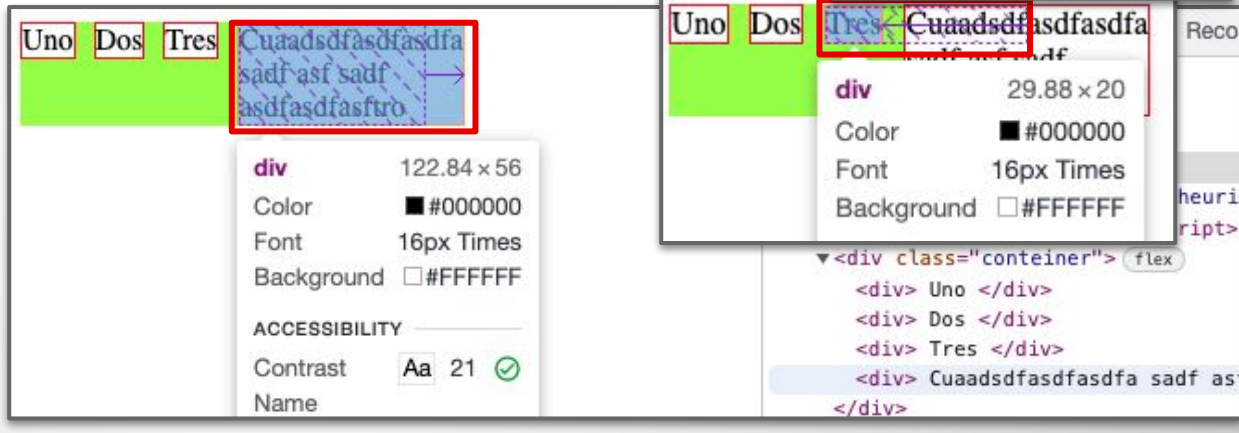
[Centering in CSS: A Complete Guide]

- Horizontally
- Vertically
- Both Horizontally and Vertically
- Conclusion

Flexbox en inspector navegador

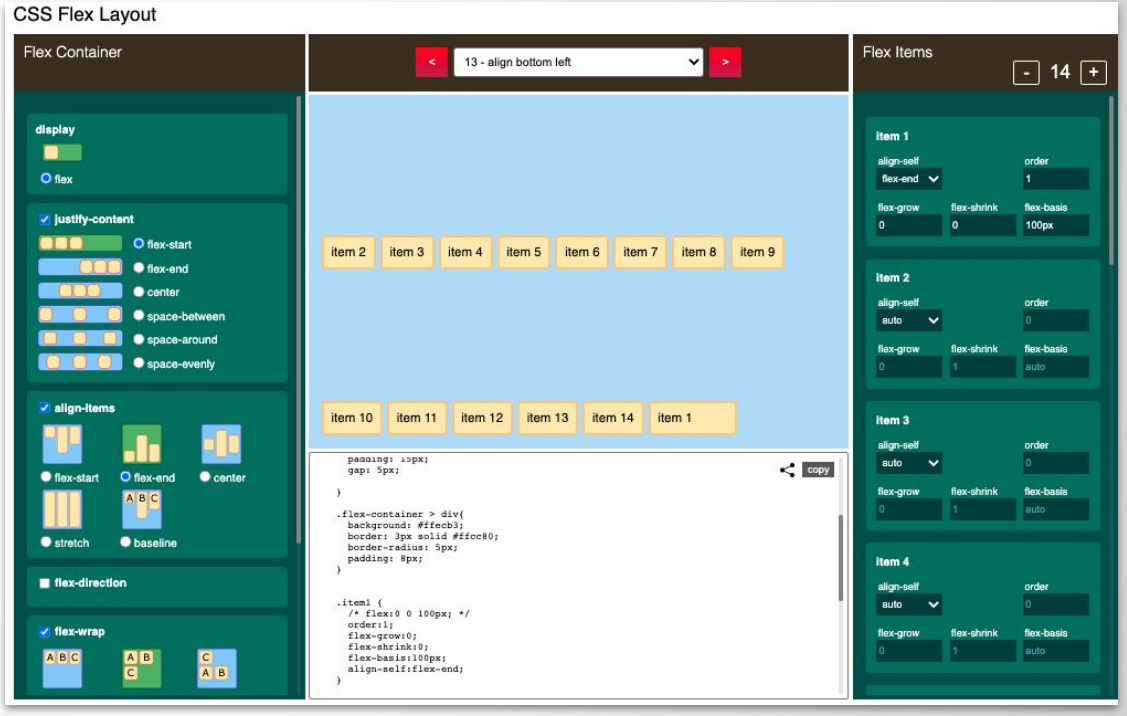
```css
div.container {
  background-color: greenyellow;
  display: flex;
  width: 50vw;
  gap: 10px;
  align-items: flex-start;
}
div.container>div {
   background-color: white;
   border: 1px solid red;
   flex-basis: 100px;
   flex-grow: 1;
}
```

```html
<div class="container">
  <div> Uno </div>
  <div> Dos </div>
  <div> Tres </div>
  <div> Cuaadsdfasdfasdfa sadf asf sadf asdfasdfasftro </div>
</div>
```



Flexbox - Herramientas online

- [CSS Flex Layout]
- [Build with Flexbox]
- [Flexy Boxes — CSS flexbox playground and code generation tool]
- [Flexplorer]
- [CSS Flexbox - HTML Lion]
- [Un juego para aprender CSS flexbox]

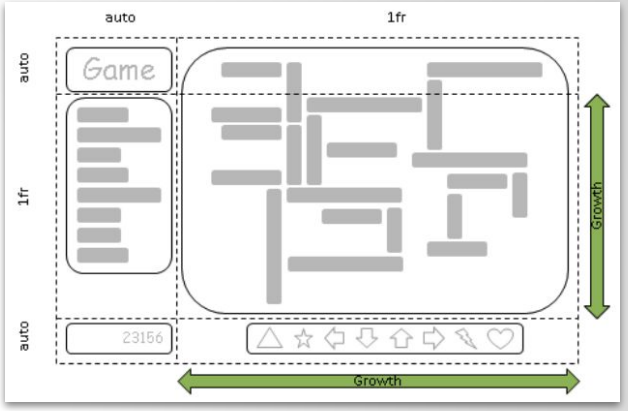


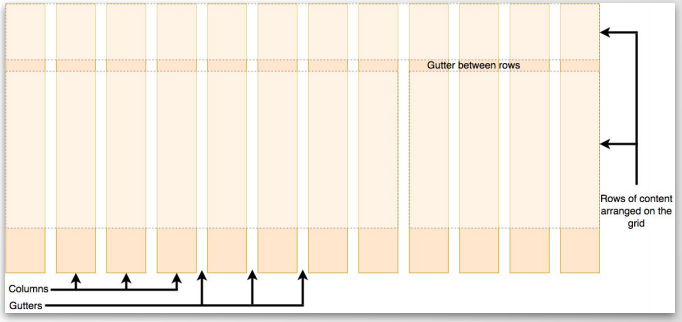
Flexbox - Documentación

- Flex CSS: Introducción - CSS en español - Lenguaje CSS
- Flexbox - Aprende sobre desarrollo web | MDN → IMPORTANTE
- Más información en:
  - Guia definitva de flexbox (1) - Main axis y cross axis | EDteam
  - Guia definitiva de Flexbox (2): Flex basis, flex-frow, flex-shrink | EDteam
  - A Complete Guide to Flexbox | CSS-Tricks
  - An Interactive Guide to Flexbox in CSS
  - Flex Cheatsheet
  - CSS Flexbox Tutorial for Beginners (With Interactive Examples)

Grid (I)

- Para trabajar en 2 dimensiones
- Cuadrículas - Aprende sobre desarrollo web | MDN → IMPORTANTE!!!!
- Grid by Example
- CSS Grid PlayGround | Terminology | Mozilla
- GRID – Malven
- Herramientas online:
  - Grid LayoutIt
  - CSS Grid Generator
  - CSS Grid Layout Generator
  -



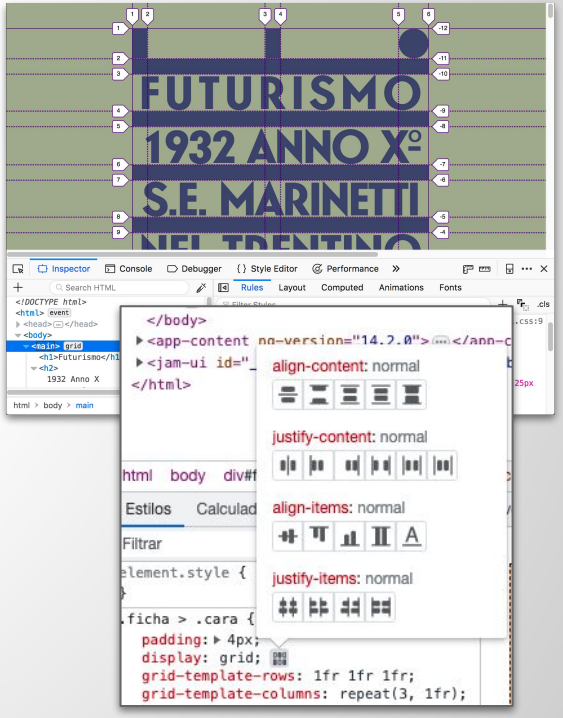


CSS Grid Layout Module Level 1

Grid (II)

- Juego para aprender → Grid Garden
- A Complete Guide to CSS Grid
- Relación de Grid Layout con otros métodos de diseño y posicionamiento - CSS
- Comparativa con Flexbox:
  - Flex o Grid
  - CSS GRID Vs FLEXBOX, Choose between Grid and Flex - Cynoteck

CSS Grid Inspector: Examine grid layouts — Firefox Source Docs documentation

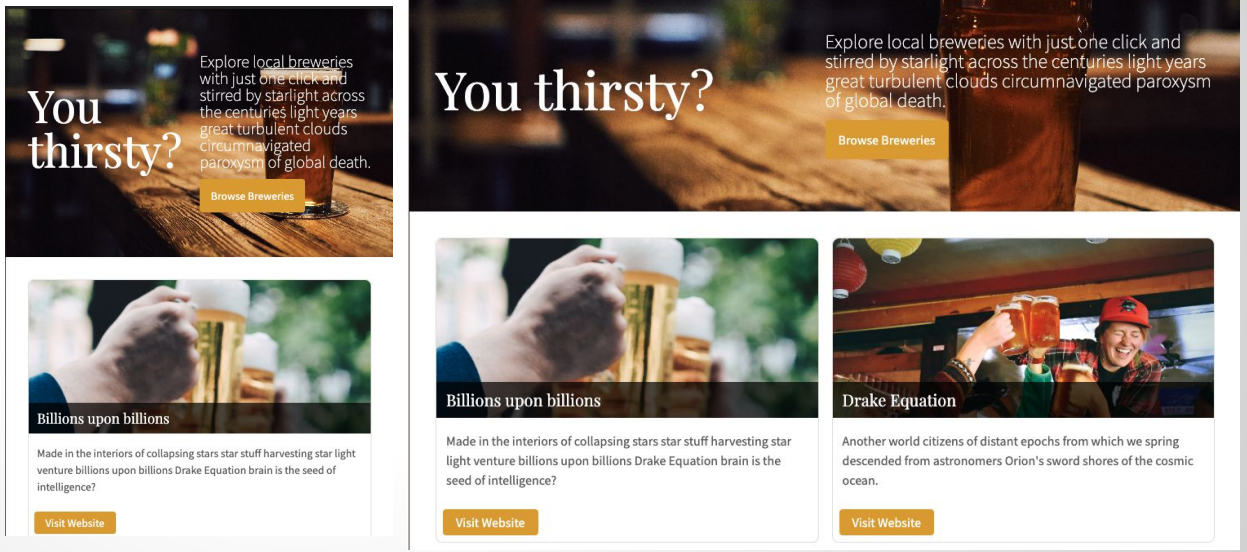


Grid (y III)

- Podemos utilizar repeat, `auto-fill`, `auto-fit` y `minmax` para realizar páginas responsive sin utilizar media queries.

[Auto-Sizing Columns in CSS Grid: `auto-fill` vs `auto-fit` | CSS-Tricks]

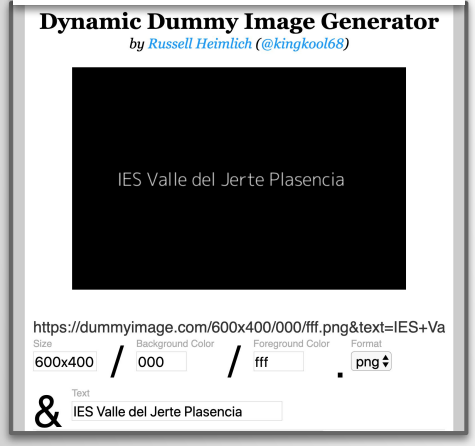
https://css-tricks.com/look-ma-no-media-queries-responsive-layouts-using-css-grid/



Generar imágenes para maquetación

https://picsum.photos/

https://dummyimage.com/



http://placeimg.com/

JavaScript básico para capas

Código encerrado en <script>
</script> en sección <head> → al igual que css es mejor en fichero externo .js

Ejemplo función JavaScript para mostrar/ocultar capas. *Otra opción .visibility = "hidden"*

```
<script>
    function mostrar(nombreCapa) {
        // Código simplificado de ejemplo, no es óptimo (todavía no hemos empezado JavaScript)
        // Pongo todas las capas a display:none (debería ser más general)
        document.getElementById('capa1').style.display = "none";
        document.getElementById('capa2').style.display = "none";
        document.getElementById('capa3').style.display = "none";
        document.getElementById('capa4').style.display = "none";
        document.getElementById('capa5').style.display = "none";
        document.getElementById('capa6').style.display = "none";
        document.getElementById('capa7').style.display = "none";

        // La capa que tengo que mostrar la pongo a display:block;
        document.getElementById(nombreCapa).style.display = "block";
    }
</script>
```

```
<div class="tab">
    <ul>
        <li>
            <div onclick="mostrar('capa1')">Capa 1</div>
            <div id="capa1">Contenido de la CAPA 1</div>
        </li>
        <li>
            <div onclick="mostrar('capa2')">Capa 2</div>
            <div id="capa2">Contenido de la CAPA 2</div>
        </li>
        <li>
            <div onclick="mostrar('capa3')">Capa 3</div>
            <div id="capa3">Contenido de la CAPA 3</div>
        </li>
        <li>
            <div onclick="mostrar('capa4')">Capa 4</div>
            <div id="capa4">Contenido de la CAPA 4</div>
        </li>
        <li>
            <div onclick="mostrar('capa5')">Capa 5</div>
            <div id="capa5">Contenido de la CAPA 5</div>
        </li>
        <li>
            <div onclick="mostrar('capa6')">Capa 6</div>
            <div id="capa6">Contenido de la CAPA 6</div>
        </li>
        <li>
            <div onclick="mostrar('capa7')">Capa 7</div>
            <div id="capa7">Contenido de la CAPA 7</div>
        </li>
    </ul>
</div>
<p>Lorem ipsum dolor sit amet.</p>
```