

## CSS

- Cascade Style Sheets - Hojas de estilo en cascada.

### Concepto:

- Lenguaje de texto que se incrusta en las páginas web para definir su formato.
- Define como se van a mostrar los elementos HTML al ser visualizados en un navegador.

### Características y utilidad:

- Su lógica consiste en separar el contenido de la presentación al desarrollar páginas web.
- Al separar presentación de contenido, se puede hacer que HTML sea más semántico.
- Es obligatorio en HTML 5
- Simplifica el código HTML al eliminar las etiquetas de formato.
- Permite el mantenimiento de un sitio web más fácil.
- Un mismo contenido se puede visualizar correctamente en cualquier dispositivo.
- Facilita la homogeneidad del sitio.
- Vía a la accesibilidad.

### Versiones:

- CSS se crea en los 90 y con el tiempo se acaba estandarizando.
- Se habla de niveles más que de versiones.
- Niveles:
  - CCS1.
  - CCS2.
  - CSS3.

### Compatibilidad entre navegadores.

- Cada navegador usa un motor de renderizado diferente.
  - Gecko: Firefox
  - Webkit: Chrome, Safari, reconocido por Opera, Firefox, Edge...
- **Prefijo de navegador:**
  - Se usan para que un navegador reconozca nuevas propiedades, aunque no estén finalizadas.
  - Prefijos:
    - -webkit propiedades -----Chrome, Safari y Android.
    - -moz propiedades -----Firefox.
    - -o propiedades-----Opera.
    - -ms propiedades----- Microsoft Explorer.
    - -khtml propiedades-----Konkeror.
  - Para consultar el nivel de reconocimiento para cada navegador y si hay que usar, o no, un prefijo, se puede consultar una propiedad concreta en <https://caniuse.com/>

## INTEGRACION CSS EN HTML

### 1. Atributo style. (Estilos Inline).

- Atributo global, por lo que puede ser utilizado en casi todas las etiquetas HTML.

- Sintaxis:
  - <elemento HTML style =" propiedad 1: valor; propiedad 2: valor; ... ; propiedad N : valor;">
- Ejemplo:
  - <aside id="Secundaria" style="border: 1px solid black; margin: 3px; background-color: CadetBlue; padding: 2px; color: white; font-family: arial;">

## 2. Etiqueta <style>...</style> (Estilos Internos o Internal).

- Se incluye en la cabecera del documento, es decir dentro de la etiqueta <head>.
- Sintaxis:

```
<head>
  <style type="text/css">
    Selector {definiciones de estilo}
  </style>
</head>
```

- Ejemplo:

```
<head>
  <style>
    h1, p
    {
      font-family: arial, verdana, calibri;
      font-size: 50px;
      color: red;
    }
  </style>
</head>
```

## 3. Hojas de estilo externas. (Estilos Externos o External).

- Uso de la etiqueta <link> o </link>
- Se incluye en la cabecera del documento, es decir, dentro de la etiqueta <head>.
- Sin etiqueta de cierre.
- El código CSS se incluye en un archivo aparte (.css) del código HTML de la página web(.html).
- Un mismo archivo .css puede ser usado por distintos archivos .html.
- Una misma página web puede usar distintos archivos .css.
- Sintaxis:
  - <link rel = "hoja de estilo" type ="tipo MIME/formato de archivo" href = "ruta de acceso/archivo.extensión">
- Ejemplo:
  - <link rel = "stylesheet" type ="text/css" href ="css/estilos.css">
  - <link rel = "stylesheet" type ="text/css" href ="css/maquetacion.css">

## 4. CSS importada. (Estilos Importados o Imported).

- Se usa dentro de la etiqueta <style> que, a su vez, se incluye dentro de <head>.
- Se usa para importar archivos .css o de otro tipo, como tipografías externas.
- Si los archivos se cargan con @import, no se usa la etiqueta <link>.
- Se puede incluir una dirección externa o local.
- En un archivo de estilos externo, @import debe ser la primera instrucción.

- A su vez, un archivo con los estilos CSS puede incluir @import para llamar o cargar otros archivos de estilo.
- Cada instrucción @import debe terminar en punto y coma.
- Sintaxis:
  - Local:

```
<style>
    @import url(ruta de acceso/archivo.extensión);
</style>
```

- Externo:

```
<style>
    @import url(protocolo://nombre de dominio/ruta de
    acceso/archivo.extensión);
</style>
```

- Ejemplos:

```
<style>
    @import url(css/estilos.css);
    @import url(maquetacion.css);
    @import url(http://www.estilos.com/estilos/maquetacion.css);
</style>
```

## COMENTARIOS DE CSS.

- Para incluir comentario en CSS se utiliza la siguiente sintaxis:
  - /\* Texto del comentario \*/

## DECLARACIÓN DE ESTILOS EN CSS (páginas 88 a 107).

- Una declaración de estilos en CSS se compone de reglas:
- Sintaxis:
  - Selector/es {propiedad 1: valor; propiedad 2: valor; ...; propiedad N: valor;}
- Ejemplo:
  - div, header, footer {background-color: orange; border: 2px dotted blue;}
- **Selector.**
  - Indica sobre qué elemento HTML va a aplicarse el estilo.
  - Se pueden especificar varios selectores separados con comas para una misma declaración de estilo.
  - Su nombre es insensible a mayúsculas y minúsculas.
  - Caracteres no permitidos en nombre de un selector:
    - Caracteres acentuados.
    - Evitar espacios en blanco:
      - “poner\_negrita\_los\_parrafos” o “ponerNegritaLosParrafos” (Notación Camel Case), en lugar de “poner negrita” con espacio en blanco en medio.
    - Muchos caracteres especiales no están permitidos (¿,-“”).
    - No pueden empezar por una cifra.

## TIPOS DE SELECTORES.

### Selectores de tipo o elemento.

- Se refieren a las etiquetas HTML.
- Se escribe el mismo nombre de la etiqueta sin <>.
- Se pueden especificar varios separados con comas, para una misma declaración de estilo.
- Es el tipo de selector menos específico de todos.
- Sintaxis:
  - Selector de tipo1, Selector de tipo2, ... {propiedad 1: valor; propiedad 2: valor; ...; propiedad N: valor;}
- Ejemplo:
  - div {background-color: orange; border: 2px dotted blue;}
  - div, header, footer {background-color: orange; border: 2px dotted blue;}

### **Selectores de clase:**

- Se emplean para las etiquetas HTML que incorporan el atributo global class.
- Puede haber varias clases con el mismo nombre en el documento HTML.
- Sintaxis HTML:
  - <Etiqueta class = "valor1 valor2 ..." .....
- Ejemplos:
  - <div class = "colores" ...>
  - <header class = "colores margen" ...>
  - <footer class = "colores" ...>
- Sintaxis CSS:
  - Se especifica el nombre del selector con el nombre de la clase precedido por un punto (.).
  - .nombre de la clase {declaraciones de estilo}
- Ejemplos:
  - .colores {background-color: red;}
  - .margen {margin: 40px;}

### **Selectores de identificación:**

- Se emplean para las etiquetas HTML que incorporan el atributo global id.
- Sintaxis HTML:
  - <Etiqueta id = "valor" .....
- Ejemplos:
  - <div id = "colores" ...>
  - <header id = "margen" ...>
- Sintaxis CSS:
  - Se especifica el nombre del selector con el nombre de la clase precedido por una #.
  - #nombre del identificador {declaraciones de estilo}
- Ejemplos:
  - #colores {background-color: red;}
  - #margen {margin: 40px;}

### **Combinación de selectores.**

- Se pueden combinar selectores de clase con los de tipo e identificación.
- Se pueden usar en una misma declaración selectores independientes con selectores combinados.
- Ejemplos:

- Tipo y clase.
  - Selector de tipo.clase {estilos}
  - p.negrita{font-weight:bold}
  - p,.negrita{font-weight:bold}
  -
- Tipo e identificador.
  - Selector de tipo#identificador {estilos}
  - p#negrita{font-weight:bold}
- Clase e identificador.
  - Selector de clase#identificador {estilos}
  - .negrita#li1{font-weight:bold}
- Ejemplo selectores independientes y combinados.
  - div, h1#titulo1, p.negrita{estilos}

### Selectores jerárquicos.

- **Descendientes (A B)**
  - Se selecciona cualquier elemento de B que es descendiente del elemento A.
  - Sintaxis:
    - Selector A Selector B {estilos}
- **Descendientes directos o hijos (A > B)**
  - Se selecciona cualquier elemento B que es hijo de A, es decir, que está en el nivel inmediatamente inferior.
  - Sintaxis:
    - Selector A > Selector B {estilos}
- **Elementos contiguos o adyacentes o hermanos adyacentes (A + B)**
  - Se selecciona cualquier elemento B precedido inmediatamente por el elemento A.
  - Sintaxis:
    - Selector A + Selector B {estilos}
- **Elementos de un mismo nivel o hermanos (A~B)**
  - Virgulilla se muestra con ALT + 126.
  - Se selecciona cualquier elemento B que esté en el mismo nivel que el elemento A, aunque no sean adyacentes.
  - Sintaxis:
    - Selector A ~ Selector B {estilos}

### Selector Universal.

- Se usa el carácter Asterisco (\*).
- Aplica el estilo a todas las etiquetas de la página.
- Sintaxis:
  - \* {estilos}
- También se usa para resetear o inicializar los valores por defecto que incluyen los navegadores:
  - Ejemplo:
    - \*{margin:0; padding:0} Anula los márgenes internos y externos que aplican los navegadores por defecto a los documentos web.

### Selector de negación.

- Selector: not.
- Sirve para añadir excepciones.
- Indica qué elementos no se verán afectados por un estilo.
- Sintaxis:
  - :not(selector) {estilos}
  - elemento:not(selector){estilos}
  - Los elementos pueden ser etiquetas, identificadores o clases.
- Ejemplos:
  - div:not(#contenedor){estilos}
    - Todos los contenedores <div> excepto el identificado con el nombre contenedor.
  - \*:not(p) {estilos}
    - Todos los elementos menos los párrafos.
  - :not(p){estilos}
    - Todos los elementos menos los párrafos.
  - p:not(.segundo){estilos}
    - Todos los párrafos excepto los que tengan como nombre de clase segundo.
  - li:not(:last of type){estilos}
    - Todos los elementos de tipo <li> excepto el último de una lista.

### Selección por el usuario.

- Selector selection.
- Permite aplicar un formato a un texto seleccionado por el usuario.
- Cuatro puntos como separador son obligatorios.
- Formato temporal que desaparece al seleccionar otro texto.
- Los estilos o propiedades a las que se puede aplicar son:
  - color, background-color y text-shadow.
- Sintaxis:
  - elemento::selection {estilos}
- Ejemplos:
  - \*::selection {color:green}
  - p::selection {color:green}

### Pseudoclases y pseudoestilos, pseudoelementos o pseudoselectores.

- Se usan para ciertos elementos que no se pueden referenciar con etiqueta o atributo.
- Sintaxis:
  - Selector:pseudoclase{estilos}
  - Selector::pseudoestilo o pseudoelemento{estilos}
- **Tipos:**
- **Pseudoclases de Enlace.**
  - Son selectores de estado para enlace.
  - Para que el resultado sea correcto en apariencia hay que usarlos en el siguiente orden: link, visited, hover, active.
  - Estados:
    - **:link**
      - Permite cambiar la apariencia de un enlace que todavía no se ha seleccionado o visitado.

- Si no se incluye se usa por defecto.
- Sintaxis:
  - `a:link{estilos}`
- Ejemplo:
  - `a:link{color:#ae34e2; text-decoration:none}`
- **:visited.**
  - Permite cambiar la apariencia de un enlace que ha sido seleccionado o visitado.
  - Sintaxis:
    - `a:visited{estilos}`
  - Ejemplo:
    - `a:visited{color:orange; text-decoration:none}`
- **:hover.**
  - Permite definir la apariencia cuando se pasa o sobrevuela con el ratón por encima del enlace.
  - Sintaxis:
    - `a:hover{estilos}`
  - Ejemplo:
    - `a:hover{color:green; text-decoration:none}`
- **:active.**
  - Permite cambiar la apariencia de un enlace cuando se activa o pulsa para seleccionar o visitar.
  - Sintaxis:
    - `a:active{estilos}`
  - Ejemplo:
    - `a:active{color:blue; text-decoration:none}`
- **Pseudoclases de teclado o formulario.**
  - Se usan con controles de formularios que aceptan entradas de usuario o a los que se le quiere poner el foco con el teclado.
  - Tipos:
    - **:focus.**
      - Permite saltar de control en control de un formulario usando la tecla del tabulador, es decir, el teclado.
      - Sintaxis:
        - `elemento:focus{estilos}`
      - Ejemplo:
        - `a:focus{color:blue; text-decoration:none}`
    - **:checked.**
      - Afecta a botones de opción y casillas de verificación que hayan sido seleccionadas.
      - Sintaxis:
        - `input[type=elemento]:checked{estilos}`
      - Ejemplo:
        - `input[type=radio]:checked {color:blue; text-decoration:none}`
- **Pseudoestilos de párrafo.**
  - **::first-letter**

- Permite definir la apariencia de la primera letra de un párrafo.
  - Efecto letra capital.
  - Sintaxis:
    - Elemento::first-letter{estilos}
  - Ejemplos:
    - p::first-letter{margin-right: 20px; font-size: 40px; color:blue;}
    - #p1::first-letter{margin-right: 20px; font-size: 40px; color:blue;}
    - div p::first-letter{margin-right: 20px; font-size: 40px; color:blue;}
- **::first-line**
  - Permite definir la apariencia de la primera línea de un párrafo.
  - Sintaxis:
    - elemento:first-line{estilos}
  - Ejemplos:
    - p::first-line{margin-right: 20px; font-size: 40px; color:blue;}
    - #p1::first-line{margin-right: 20px; font-size: 40px; color:blue;}
    - div p::first-line{margin-right: 20px; font-size: 40px; color:blue;}
- **Pseudoelemento o pseudoselector de inserción.**
  - Permiten insertar un texto o una imagen delante o detrás de un elemento.
  - Asociados a la propiedad content, que permite especificar el contenido a insertar:
  - Tipos:
    - **::before.**
      - Permite insertar un contenido antes de un elemento.
      - El contenido puede ser un texto o imagen.
      - Sintaxis:
        - **Imagen**:
          - Elemento::before{content: url(ruta de acceso/archivo.extensión) }
        - **Texto**:
          - Elemento::before{content: "Texto" }
      - Ejemplo:
        - h1::before{content: url(estrellita.gif) }
        - h1::before{content: "Hola " }
      -
    - **::after.**
      - Permite insertar un contenido después de un elemento.
      - El contenido puede ser un texto o imagen.
      - Sintaxis:
        - **Imagen**:
          - Elemento::after{content: url(ruta de acceso/archivo.extensión) }
        - **Texto**:
          - Elemento::after{content: "Texto" }
      - Ejemplo:
        - h1::after{content: url(estrellita.gif) }
        - h1::after{content: " Hola" }
- **Pseudoclases de selección.**



- Se usan con cualquier tipo de elemento HTML, pero es muy común usarlas con las listas ordenadas y no ordenadas para acceder a alguno de sus elementos a través de su posición.
- Tipos:
  - **first-child.**
    - Selecciona el primer descendiente directo o hijo.
    - Entre el elemento y la pseudoclase debe haber un espacio en blanco.
    - Sintaxis:
      - elemento :first-child{estilos}
    - Ejemplo:
      - ul :first-child{color:green}
      - #div1 :first-child{font-family:comic sans ms}
  - **first-of-type.**
    - Selecciona el primer descendiente directo o hijo de un tipo determinado.
    - Entre el elemento y la pseudoclase no debe haber un espacio en blanco.
    - Sintaxis:
      - elemento:first-of-type{estilos}
    - Ejemplo:
      - ul li:first-of-type{color:green}
      - #div1 p:first-of-type{font-family:comic sans ms}
  - **last-child.**
    - Selecciona el último descendiente directo o hijo.
    - Entre el elemento y la pseudoclase debe haber un espacio en blanco.
    - Sintaxis:
      - elemento :last-child{estilos}
    - Ejemplo:
      - ul :last-child{color:green}
      - #div1 :last-child{font-family:comic sans ms}
  - **last-of-type.**
    - Selecciona el último descendiente directo o hijo de un tipo determinado.
    - Entre el elemento y la pseudoclase no debe haber un espacio en blanco.
    - Sintaxis:
      - elemento :last-of-type{estilos}
    - Ejemplo:
      - ul li:last-of-type{color:green}
      - #div1 p:last-of-type{font-family:comic sans ms}
  - **:nth-child(valor)**
    - Selecciona un elemento de nivel inferior o hijo, según un valor.
    - Entre el elemento y la pseudoclase debe haber un espacio en blanco.
    - Valores:
      - **Número:**
        - Selecciona el elemento especificado con el número.
      - **Odd:**
        - Selecciona los elementos impares.
      - **Even:**
        - Selecciona los elementos pares.
    - Sintaxis:

- elemento :nth-child(valor){estilos}
- Ejemplos:
  - ul :nth-child(4){color:green}
  - ul :nth-child(odd){color:green}
  - ul :nth-child(even){color:green}
- **:nth-of-type(valor)**
  - Selecciona un elemento del tipo especificado según un valor.
  - Entre el elemento y la pseudoclase no debe haber un espacio en blanco.
  - Valores:
    - **Número:**
      - Selecciona el elemento especificado con el número.
    - **Odd:**
      - Selecciona los elementos impares.
    - **Even:**
      - Selecciona los elementos pares.
  - Sintaxis:
    - elemento de nivel superior elemento hijo:nth-of-type(valor){estilos}
  - Ejemplos:
    - div p:nth-of-type(4){color:green}
    - div p:nth-of-type(odd){color:green}
    - div p:nth-of-type(even){color:green}
- **:nth-last-child(valor)**
  - Selecciona un elemento de nivel inferior o hijo empezando por el final, según un valor.
  - Entre el elemento y la pseudoclase debe haber un espacio en blanco.
  - Valores:
    - **Número**
      - Selecciona el elemento especificado con el número.
    - **Odd:**
      - Selecciona los elementos impares.
    - **Even:**
      - Selecciona los elementos pares.
  - Sintaxis:
    - elemento :nth-last-child(valor){estilos}
  - Ejemplos:
    - ul :nth-last-child(4){color:green}
    - ul :nth-last-child(odd){color:green}
    - ul :nth-last-child(even){color:green}
- **:nth-last-of-type(valor)**
  - Selecciona un elemento del tipo especificado según un valor empezando por el final.
  - Entre el elemento y la pseudoclase no debe haber un espacio en blanco.
  - Valores:
    - **Número:**
      - Selecciona el elemento especificado con el número.

- **Odd:**
      - Selecciona los elementos impares.
    - **Even:**
      - Selecciona los elementos pares.
  - Sintaxis:
    - elemento de nivel superior elemento hijo:nth-last-of-type(valor){estilos}
  - Ejemplos:
    - `div p:nth-last-of-type(4){color:green}`
    - `div p:nth-last-of-type(odd){color:green}`
    - `div p:nth-last-of-type(even){color:green}`
- **Pseudoclasas de atributos.**
    - Permiten aplicar estilos a aquellas etiquetas que tengan un determinado atributo.
    - Se utilizan los corchetes ([]) para especificar el atributo.
    - Tipos:
      - **[atributo]**
        - Aplica el estilo a todas las etiquetas que tengan el atributo especificado.
        - Sintaxis:
          - `elemento[atributo]{estilos}`
          - `*[atributo]{estilos}`
        - Ejemplos:
          - `p[title]{font-family:verdana}`
          - `*[title]{font-family:verdana}`
      - **[atributo="valor"]**
        - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado un determinado valor.
        - Sintaxis:
          - `elemento[atributo="valor"]{estilos}`
          - `*[atributo="valor"]{estilos}`
        - Ejemplos:
          - `a[title="titulo1"]{font-family:verdana}`
          - `*[title="hola"]{font-family:verdana}`
      - **[atributo ^= "valor"]**
        - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado valores que empiecen por los caracteres indicados.
        - Sintaxis:
          - `elemento[atributo^="valor"]{estilos}`
          - `*[atributo^="valor"]{estilos}`
        - Ejemplos:
          - `a[title^="tit"]{font-family:verdana}`
          - `*[title^="h"]{font-family:verdana}`
      - **[atributo \$="valor"]**
        - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado valores que terminen por los caracteres indicados.
        - Sintaxis:
          - `elemento[atributo$="valor"]{estilos}`

- `*[atributo$="valor"]{estilos}`
- Ejemplos:
  - `p[title$="os"]{font-family:verdana}`
  - `*[title$="45"]{font-family:verdana}`
- **[atributo \*= "valor"]**
  - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado y en cualquier posición valores con los caracteres indicados.
  - Sintaxis:
    - `elemento[atributo*="valor"]{estilos}`
    - `*[atributo*="valor"]{estilos}`
  - Ejemplos:
    - `p[title*="os"]{font-family:verdana}`
    - `*[title*="45"]{font-family:verdana}`
- **[atributo |= "valor"]**
  - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado una palabra exacta o una palabra seguida por un guion.
  - Sintaxis:
    - `elemento[atributo|= "valor"]{estilos}`
    - `*[atributo|= "valor"]{estilos}`
  - Ejemplos:
    - `<a href="#" id ="enlace-primero">Enlace 1</a>`
    - `<a href="#" id ="enlace">Enlace 2</a>`
    - `a[id|= "enlace"]{font-family:verdana}`
    - `*[title|= "45"]{font-family:verdana}`
- **[atributo ~="valor"]**
  - Aplica el estilo a todas las etiquetas que tengan en el atributo especificado una palabra exacta o una palabra seguida por un espacio en blanco.
  - Sintaxis:
    - `elemento[atributo~="valor"]{estilos}`
    - `*[atributo~="valor"]{estilos}`
  - Ejemplos:
    - `<a href="#" title ="a">Enlace 1</a>`
    - `<a href="#" title ="a b">Enlace 1</a>`
    - `<a href="#" title ="ab">Enlace 1</a>`
    - `a[title~="a"]{font-family:verdana}`
    - `*[title~="a"]{font-family:verdana}`

## ORDENAR CSS.

- Una hoja de estilos CSS puede tener miles de reglas.
- Tanto en su creación como en su mantenimiento futuro pueden aparecer problemas como:
  - **Reglas no utilizadas.**
    - No se usan.
    - Incrementan tamaño del fichero.
  - **Reglas duplicadas.**
    - Indican lo mismo o son contradictorias.
    - Confusión al modificar CSS.
  - **Propiedades heredadas.**

- Pasan desapercibidas.
  - No se tiene en cuenta al modificar un CSS.
- Solución a los problemas anteriores.
  - Definir un estilo propio teniendo en cuenta:
    - **Organizar reglas por orden alfabético.**
      - a, div, em, .importante, p, #principal, table.
    - **Organizar reglas por orden de aparición.**
      - html, body, div, h1, h2, #principal, p, .importante.
    - **Separar reglas según funciones que realizan.**
      - Por un lado, las que modifican bordes.
      - Por otro, las que modifican colores.
      - En otro archivo las que afectan a las fuentes.
      - Otros archivos para definir la maquetación.
      - Etc.
    - **Organizar las propiedades de una regla en orden alfabético.**
      - border, font-size, margin, padding, text-align.
    - **Organizar las propiedades de una regla del exterior del elemento al interior.**
      - margin, border, padding, text-align, font-size.
    - **Ver artículos en libros o Internet sobre metodologías para organizar estilos CSS como:**
      - OOCSS.
      - BEM.
      - SMACSS.
    - **Usar herramientas como:**
      - <https://www.cleancss.com/css-beautify/>
      - <https://csscompressor.com/>

## PROPIEDADES CSS.

### COLORES.

- El color se usa en varias propiedades como color, background-color, text-shadow, box-shadow, border, etc...
- **Notaciones:**
  - **Hexadecimal clásico:**
    - Uso los dígitos 0 a 9 y las letras a-f.
    - Sintaxis:
      - #rrggbb
    - Ejemplo:
      - #76a42e
  - **Hexadecimal abreviado:**
    - Uso los dígitos 0 a 9 y las letras a-f.
    - Sintaxis:
      - #rgb
    - Ejemplo:
      - #7a2 equivale a #77aa22
  - **RGB Decimal.**
    - 1 byte por color usando su valor decimal.
    - Sintaxis:

- rgb(color rojo, color verde, color azul)
  - rgb(0-255,0-255,0-255)
  - Ejemplo:
    - rgb(6,78,123)
- **RGB Porcentaje.**
  - 1 byte por color usando su valor porcentual.
  - Sintaxis:
    - rgb(color rojo, color verde, color azul)
    - rgb(0%-100%,0%-100%,0%-100%,)
  - Ejemplo:
    - rgb(6%,78%,23%)
- **RGB Canal Alfa.**
  - Permite incluir transparencia en el color.
  - 1 byte por color usando su valor decimal más un cuarto valor comprendido entre 0 y 1 con decimales para la transparencia:
    - 0 transparencia total.
    - 1 opacidad total.
  - Sintaxis:
    - rgba(color rojo, color verde, color azul, transparencia)
    - rgba(0-255,0-255,0-255,0.0 a 1.0)
  - Ejemplo:
    - rgba(6,78,123,0.75)
- **HSL.**
  - Modelo de color basado en la percepción humana.
  - Valores:
    - **h o hue:**
      - Tono o matiz.
    - **s:**
      - Saturación o cantidad de gris que tiene un color.
    - **l:**
      - Luminosidad o luminancia.
      - Cantidad de blanco o negro que tiene un color.
  - Sintaxis:
    - hsl(tono de color, saturación, luminosidad)
    - hsl(0-359,0%-100%,0%-100%)
  - Ejemplo:
    - hsl(210,56%,23%)
- **HSL Canal Alfa.**
  - Permite controlar el grado de transparencia de un color en el modelo hsl usando un cuarto valor comprendido entre 0 y 1 con decimales para la transparencia:
    - 0 transparencia total.
    - 1 opacidad total.
  - Sintaxis:
    - hsla(tono de color, saturación, luminosidad, transparencia)
    - hsla(0-359,0%-100%,0%-100%,0.0 a 1.0)
  - Ejemplo:

- hsla(210,56%,23%,0.40)
- **Nombre del color.**
  - Se escribe el nombre del color en inglés.
  - Hay 140 valores de colores seguros, es decir, colores que pueden mostrar en cualquier dispositivo.
  - Sintaxis:
    - propiedad: nombre del color.
  - Ejemplo:
    - color:green

## UNIDADES DE MEDIDA.

- Las unidades de medida en CSS pueden especificarse mediante 2 tipos de valores.
- **Valores absolutos:**
  - Su valor no depende de otro.
  - Medida con valor definido.
  - Permanecen constantes independientemente del dispositivo o software utilizado.
  - Poco flexibles porque no se adaptan fácilmente a distintos dispositivos.
  - Poco útiles de cara a la accesibilidad.
  - Sintaxis:
    - Número unidad de medida.
  - Ejemplos:
    - 20pt
    - 250px
    - 34mm
  - Unidades:
    - **Punto (pt)**:
      - 1 punto son 0,35 mm.
      - Valor entero.
    - **Pica (pc)**:
      - 1 pica son 12 puntos.
      - Valor real.
    - **Milímetro (mm)**:
      - 1 milímetro son 0,24 picas.
      - Valor entero.
    - **Centímetro (cm)**
      - 1 centímetro son 10 milímetro.
      - Valor real.
    - **Pulgadas (in)**
      - 1 pulgada son 2,54 centímetros.
      - Valor real.
    - **Píxeles (px)**
      - Picture element - Elemento de imagen.
      - Valor entero.
      - 16 px equivalen a 12 pt.
- **Valores relativos.**
  - Pueden variar según el dispositivo usado.

- No completamente definidas, ya que su valor depende de otro valor.
- Muy flexibles, se adaptan a todos los medios.
- Útiles en accesibilidad.
- Unidades:
  - **em.**
    - Unidad basada en el tamaño del tipo de fuente usado en la página o en el elemento padre.
    - Se multiplica el valor en em por el valor del tamaño de fuente.
    - Sintaxis:
      - número em
    - Ejemplo:
      - Si la fuente tiene 15pt de tamaño, otros elementos basados en ella tendrán, por ejemplo:
        - $1\text{em} * 15\text{pt} = 15\text{pt}.$
        - $0.5\text{em} * 15\text{pt} = 7,5\text{pt}.$
        - $4\text{em} * 15\text{pt} = 60\text{pt}.$
  - **rem.**
    - root em.
    - Unidad basada en el tamaño del tipo de fuente usado en el elemento raíz de la página <html>.
    - Unidad no escalable.
    - Se multiplica el valor en rem por el valor del tamaño de fuente.
    - Ejemplo:
      - Si la fuente tiene 15pt de tamaño en el elemento raíz, otros elementos basados en ella tendrán, por ejemplo:
        - $1\text{rem} * 15\text{pt} = 15\text{pt}.$
        - $0.5\text{rem} * 15\text{pt} = 7,5\text{pt}.$
        - $4\text{rem} * 15\text{pt} = 60\text{pt}.$
  - **ex.**
    - Basado en el tamaño de las fuentes.
    - Unidad relativa a la altura de la letra minúscula en el elemento seleccionado.
    - Es la mitad que unidad em.
      - $1\text{ex} = 0.5 \text{ em}$
    - Sintaxis:
      - número ex
    - Ejemplo:
      - Si la fuente tiene 15pt de tamaño, otros elementos basados en ella tendrán, por ejemplo:
        - 1ex equivaldría a 7,5pt.
        - 0.5ex equivaldría a 3,7pt.
        - 4ex equivaldría 30pt.
  - **Porcentaje (%).**
    - Se basa en el tamaño del elemento padre o contenedor.
    - Unidad escalable.
    - Valor entero.
    - Sintaxis:



- número %
- Ejemplo:
  - width: 50%
- **vw.**
  - Permite especificar la anchura visible del navegador.
  - viewport width.
  - Unidad no escalable.
  - **Viewport:**
    - Región visible de la página web en el navegador.
  - Sintaxis:
    - número vw
  - Ejemplo:
    - width: 1vw Equivale al 1% del ancho del navegador.
    - width: 100vw Equivale al 100% del ancho del navegador.
- **vh.**
  - Permite especificar la altura visible del navegador.
  - viewport height.
  - Unidad no escalable.
  - Sintaxis:
    - número vh
  - Ejemplo:
    - height: 1vh Equivale al 1% de la altura del navegador.
    - height: 100vh Equivale al 100% de la altura del navegador.

## FUENTES.

- Diseño que se aplica a todas las letras, números y símbolos.
- **Webfonts**
  - Son fuentes adaptadas para su uso en la Web.
  - Tipos más comunes:
    - **TTF.**
      - True Type Font.
      - Fuente de tipo real.
      - Formato más extendido.
    - **OTF.**
      - Open Type Font.
      - Evolución de TTF enriquecido tipográficamente.
    - **EOT.**
      - Embedded Open Type.
      - Formato propietario de Microsoft.
    - **WOFF.**
      - Web Open Font Format
      - Formato comprimido para la web.
- Los formatos para fuentes se aplican en CSS con la propiedad genérica font.
- Propiedades:
  - **font-family.**
    - Propiedad que permite aplicar distintos tipos de fuentes al texto.

- Fuentes con nombres compuestos se encierran entre comillas.
- Tipografías seguras:
  - Vienen instaladas en todos los dispositivos. Cualquier navegador las mostrará bien.
  - Están almacenadas en la carpeta Fonts de Windows.
- Ejemplos:
  - Arial, Helvética, Verdana, Georgia, Palatino, Times New Roman, Tahoma, ...
- Familias de fuentes:
  - Se identifican mediante un nombre de fuente genérico.
  - En vez de especificar una fuente es posible usar una familia de fuentes.
  - Tipos:
    - **Serif.**
      - Fuentes que tienen adornos en sus terminaciones.
      - Ejemplos:
        - Bookman R / Times New Roman R
    - **Sans-serif.**
      - Fuentes sin adornos en sus terminaciones.
      - Ejemplos:
        - Arial R / **Impact R**
    - **Cursive.**
      - Fuentes caligráficas.
      - Ejemplos:
        - Comic Sans ms R / *Impact R*
    - **Monospace.**
      - Todas las letras y espacios entre ellas con las mismas dimensiones.
      - Ejemplo:
        - Courier New R
    - **Fantasy.**
      - Fuentes de símbolo decorativas.
      - Ejemplos:
        - En Word se utilizan Webding y Windings.
        - Tfno. 333 ☎ 333
  - Fuentes genéricas de los navegadores:
    - **Serif:**
      - Times New Roman.
    - **Sans-serif:**
      - Arial.
    - **Cursive:**
      - Comic sans ms.
    - **Monospace:**
      - Courier New.
    - **Fantasy:**
      - Dependen de los navegadores que, si no las pueden representar, utilizan “Comic Sans MS” o “Impact”, o las genéricas serif y sans-serif.

- Sintaxis:
  - {font-family: fuente 1, fuente 2, ..., fuente N, genérica}
- Ejemplos:
  - {font-family: arial, "comic sans ms", "courier new", serif}

### Uso de fuentes que no están instaladas en el sistema.

#### Fuente externa importada.

- Uso de fuentes importadas mediante la directiva @import.
- Se usa dentro de la <style>, que a su vez se incluye dentro de <head>
- La fuente se descarga al mismo tiempo que se usa.
- Se pueden incluir direcciones externas o internas.
- Sintaxis:
  - **1º, importar la fuente:**

```
<style>
    @import url(protocolo://nombre de dominio/ruta de acceso/archivo de
    fuentes.extensión);
</style>
```
  - **2º, utilizar la fuente usando la propiedad font-family.**

```
{font-family:fuente}
```
- Ejemplo:

```
<style>
    @import url(http://www.estilos.com/estilos/roboto.ttf);
    p{font-family: roboto , sans-serif}
</style>
```

#### Fuente externa enlazada.

- La fuente se descarga al mismo tiempo que se usa.
- Se usa la etiqueta <link> para cargar una fuente externa.
- Sintaxis:
  - **1º, importar la fuente:**

```
<link href= "protocolo://nombre de dominio/ruta de acceso/archivo de
fuentes.extensión">
```
  - **2º, utilizar la fuente usando la propiedad font-family.**

```
{font-family:fuente}
```
- Ejemplo:

```
<link href ="http://www.estilos.com/estilos/roboto.ttf" rel ="stylesheet">
p{font-family: roboto, sans-serif}
```

### Fuentes descargadas o importadas.

- Se pueden importar fuentes e incluso usarlas sin estar instaladas en la carpeta Fonts de Windows, si no en cualquier otra carpeta, forme parte o no parte de un sitio web.
- La fuente se descarga al mismo tiempo que se usa.
- Se pueden usar varias fuentes.
- Las rutas de acceso pueden ser locales o externas, absolutas o relativas.
- Aconsejable usar fuentes tipo webfonts.
- Uso de la propiedad: @font-face
- Sintaxis:
  - **1º, cargar o importar la fuente:**

```
@font-face{ src: url (ruta de acceso/nombre.extensión);
font-family: "nombre de fuente real o inventado";}
```

- Pueden usarse, de manera opcional, rutas alternativas a otras carpetas donde se encuentren también las fuentes almacenadas)

```
@font-face{ src: url (ruta de acceso1/nombre.extensión);
url (ruta de acceso2/nombre.extensión);
font-family: "nombre de fuente real o inventado";}
```

- **2º, utilizar la fuente usando la propiedad font-family.**

```
Selector {font-family: nombre de la fuente creado en @font-face;}
```

- Ejemplo:

```
@font-face{ src: url (fuentes/arial.ttf);
font-family: "ar";}
p{ font-family:ar}
```

### Formato de Fuentes.

- Permite incluir las fuentes en todos los formatos posibles para que puedan utilizarlas los navegadores.
- Formatos:
  - woff, woff2, truetype, opentype y embedded-opentype
- Sintaxis:
  - Format ('Tipo de formato de fuente')
- Ejemplo:

```
@font-face{ src: url (fuentes/opensans.ttf) format ('truetype'),
url (opensans.woff) format ('woff'),
url(fonts1/opensans.eot) format('embedded-opentype'),
font-family: "opens";}
```

### Repositorios de fuentes.

- Google Fonts ----- fonts.google.com
- Fonts Squirrel -----www.fontsquirrel.com
- Da Font-----www.dafont.com

- My fonts-----[www.myfonts.com](http://www.myfonts.com)

## ASPECTO DE LAS FUENTES.

### font-style.

- Permite aplicar cursiva al texto.
- Sintaxis:
  - {font-style:valores}
- Valores:
  - **italic**:
    - Cursiva propia de la fuente utilizada.
  - **oblique**:
    - Inclinación a la derecha de los caracteres que aplica por defecto el navegador.
  - **normal**:
    - Devuelve el texto a la inclinación normal.

### font-variant.

- Muestra un texto en versales o versalitas.
- Un carácter versal es un carácter en mayúsculas con el tamaño de una minúscula.
- Sintaxis:
  - {font-variant:valores}
- Valores:
  - **small-caps**.
    - Aplicar versales.
  - **normal**.
    - Quitar versales.

### font-weight.

- Texto a negrita.
- Sintaxis:
  - {font-weight:valores}
- Valores:
  - **Normal**.
    - Para quitar el efecto negrita.
  - **Bold**.
    - Para aplicar negrita.
  - **Lighter**.
    - Disminuye negrita respecto del elemento de nivel superior (padre).
  - Valores numéricos:
    - Numero enteros de 100 a 900.
    - 500 = normal
    - 700 = bold
    - 900 = bolder

### font-size.

- Tamaño de fuente.

- Sintaxis:
  - {font-size: número unidad de medida}
- Tamaños prefdefinidos:
  - xx-small, x-small, small, medium, large, x-large, xx-large
  - smaller y larger.
    - tamaño de letra más pequeño o grande respecto del elemento padre.
- Ejemplos:
  - {font-size: 30pt}
  - {font-size: 130px}
  - {font-size: 2em}
  - {font-size: xx-large}

### **Agrupación de propiedades de fuente.**

- Uso propiedad font:
- Las propiedades que se pueden agrupar son font-style, font-weight , font-variant, font-size y font-family y sus valores hay que incluirlos en este orden.
- No es obligatorio incluir todos los valores.
- El cuarto valor se utiliza para el tamaño de las fuentes, pero también se podría incluir la distancia entre líneas junto con el tamaño:
  - valor tamaño/valor interlineado
  - 24pt/1.5
- Sintaxis:
  - {font: valor1 valor2 valor3 valor4 valor5}
- Ejemplo:
  - {Font: italic bold small-caps 24pt/1.5 arial, sans-serif}

### **font-stretch.**

- Permite variar la distancia entre caracteres.
- Funciona en fuentes con versiones condensadas o expandidas.
- Sintaxis:
  - {font- stretch: valores}
- Valores:
  - Ultra-condensed, extra-condensed, condensed, semi-condensed.
  - Normal.
  - Semi-expanded, expanded, extra-expanded, ultra-expanded.

### **color.**

- Aplica un color a las fuentes.
- Sintaxis:
  - {color:color}
- Ejemplos:
  - {color:#78e4a2}
  - {color:red}
  - {color: rgb(5,89,234)}

### **text-decoration.**

- Permite modificar la apariencia el texto mediante rayados.

- Se pueden usar varios valores en la misma propiedad separados por un espacio.
- Sintaxis:
  - { text-decoration:valor}
- Valores:
  - **underline:**
    - Subrayado.
  - **overline:**
    - Suprarayado.
  - **line-through:**
    - Tachado.
  - **none.**
    - Quitar rayado.
- Ejemplos:
  - {Text-decoration:underline }
  - {Text-decoration: underline overline }

#### **text-transform.**

- Permite cambiar el texto a mayúsculas, minúsculas o a tipo título, independiente a como aparezca en HTML.
- Sintaxis:
  - { text-transform:valores}
- Valores:
  - **capitalize:**
    - 1ª letra de cada palabra en mayúsculas.
  - **uppercase:**
    - Todas las letras a mayúsculas.
  - **lowercase:**
    - Todas las letras a minúsculas.
  - **none:**
    - Quitar las modificaciones.
- Ejemplo:
  - { Text-transform:uppercase}

#### **text-indent.**

- Permite indentar la primera línea de un párrafo para simular sangría de primera línea.
- Sintaxis:
  - { text-indent: número unidad de medida}
- Ejemplos:
  - { text-indent: 30px;}
  - { text-indent: 10%;}

#### **letter-spacing.**

- Permite cambiar la distancia entre caracteres.
- No validos valores relativos.
- Valor positivo aumenta espacio entre caracteres.

- Valor negativo disminuye espacio entre caracteres.
- Sintaxis:
  - { letter-spacing: número unidad de medida }
- Ejemplos:
  - { letter-spacing: 30px; }
  - { letter-spacing: -10px; }

#### **word-spacing.**

- Permite cambiar la distancia entre palabras.
- No validos valores relativos.
- Valor positivo aumenta espacio entre caracteres.
- Valor negativo disminuye espacio entre caracteres.
- Sintaxis:
  - { word-spacing: número unidad de medida }
- Ejemplos:
  - { word-spacing: 30px; }
  - { word-spacing: -10px; }

#### **text-align.**

- Alineación de texto en horizontal.
- Sintaxis:
  - { text-align: valores }
- Valores:
  - **left:**
    - Izquierda.
  - **right:**
    - Derecha.
  - **center:**
    - Centrado.
  - **justify:**
    - Justificado.
  - **none:**
    - Alineación por defecto a la izquierda.
- Ejemplos:
  - { Text-align: justify }

#### **vertical-align.**

- Alineación de texto en vertical.
- Sintaxis:
  - { vertical-align: valores }
- Valores:
  - **baseline:**
    - Alinea el texto recto a la línea base de escritura.
    - Valor por defecto.
  - **sub:**



- Formato subíndice.
- **super:**
  - Formato superíndice.
- **top:**
  - Alineación respecto de la parte superior del elemento padre.
- **middle:**
  - Alineación respecto de la parte media del elemento padre.
- **bottom:**
  - Alineación respecto de la parte inferior del elemento padre.
- **Unidades de medida:**
  - Alineación vertical respecto a la línea base de escritura usando valores absolutos o relativos.
  - Valor positivo alineación por encima de la línea base.
  - Valor negativo alineación por debajo de la línea base.
- Ejemplos:
  - {vertical-align:sub}
  - {vertical-align:top}
  - {vertical-align:5px}
  - {vertical-align: -10%}

## direction.

- Dirección de texto.
- Sintaxis:
  - { direction:valores}
- Valores:
  - **ltr**
    - Left to right.
    - Escritura de izquierda a derecha.
    - Valor por defecto.
  - **rtl**
    - right to left.
    - Escritura de derecha a izquierda.
    - Que el texto aparezca así escrito depende de si lo permite la lengua usada.
- **unicode-bidi.**
  - Propiedad que junto con la de dirección determina cómo manejar el texto bidireccional en un documento web.
  - Sintaxis:
    - unicode-bidi: valores;
  - Valores.
    - **normal:**
      - Opción por defecto, la dirección por defecto la determina el navegador.
    - **bidi-override;**
      - Anula el algoritmo que tiene el navegador para mostrar texto bidireccional.
      - Opción que fuerza el cambio de dirección cuando se usa la propiedad *direction:ltr*.
    - **inherit:**

- Dirección del texto heredada del elemento padre.

### **text-shadow.**

- Permite sombrear un texto.
- Valores en píxeles.
- Sintaxis:
  - { text-shadow: x y z color }
- Valores:
  - **x**
    - Margen de la sombra hacia la derecha si el valor es positivo o hacia la izquierda si es negativo.
  - **y**
    - Margen de la sombra hacia abajo si el valor es positivo o hacia arriba si es negativo.
  - **z**
    - Intensidad del degradado o desenfocado de la sombra.
    - Opcional.
    - Valor por defecto: 0
  - **color**:
    - Color de la sombra.
- Ejemplos:
  - { text-shadow: 5px 5px red }
  - { text-shadow: -34px 28px 10px blue }

### **line-height.**

- Modifica el interlineado de un párrafo.
- Sintaxis:
  - { line-height: valores }
- Valores:
  - **Número**:
    - Valor numérico que se multiplica por el tamaño de fuente existente.
  - **Unidad de medida**:
    - Se aplica una distancia entre líneas exacta en base al valor especificado.
    - Se pueden usar unidades absolutas o relativas
  - **normal**:
    - Interlineado por defecto.
- Ejemplos:
  - { line-height: 2.5 }
  - { line-height: 20px }
  - { line-height: 5% }

### **white-space.**

- Permite controlar los espacios vacíos.
- Sintaxis:
  - { white-space: valores }
- Valores:

- **pre:**
  - Preserva espacio múltiples y saltos de línea
- **nowrap:**
  - Impide saltos de línea automáticos y muestra le texto de forma horizontal sea cual sea el espacio disponible en el navegador.
  - Todo el texto en una línea horizontal.
- **normal:**
  - El navegador gestiona los espacios vacíos y los saltos de línea.
- Ejemplos:
  - { White-space: pre}
  - { White-space: nowrap}

### font-size-adjust.

- Permite variar la diferencia de tamaño entre mayúsculas y minúsculas.
- Sintaxis:
  - {font-size-adjust: valores}
- Valores:
  - **Número:**
    - Valor relativo al tamaño de fuente.
    - El número, multiplicado por el tamaño de la fuente actual, especifica el tamaño de las letras minúsculas.
    - Se usa cuando la fuente principal no está disponible y hay que usar otra que puede tener una relación de aspecto diferente.
    - Relación de aspecto:
      - Relación entre el tamaño de las letras minúsculas y el tamaño de la fuente
    - Para comparar 2 fuentes: <https://www.brunildo.org/test/xheight.pl>
  - **none:**
    - Valor por defecto.

### hyphens.

- Indica al navegador el método que debe seguir para dividir las palabras para mejorar la distribución del texto.
- Sintaxis:
  - {hyphens: valores}
- Valores:
  - **manual:**
    - Las palabras se cortan, o los guiones aparecen, donde el usuario considera oportuno, empleando para ello guiones suaves y/o fuertes.
    - El navegador añadirá los guiones suaves solo si son necesarios y siempre, si son guiones fuertes.
    - Guion suave.
      - El navegador los utiliza para separar palabras entre líneas.
      - Uso del carácter especial &shy; en el sitio donde se quiera usar el guion.
    - Guion fuerte.
      - Se utiliza para unir 2 o más palabras.

- Uso del carácter especial &hyphens; en el sitio donde se quiera usar el guion.
- **auto:**
  - El navegador se encarga de dividir las palabras donde considere oportuno si es necesario, sobre la base de la configuración idiomática que se tenga.
  - **Atributo lang.**
    - Se usa para establecer el idioma que, en el título, división, párrafo, etc., al que se le va a aplicar esta opción de hyphens.
    - Sintaxis:
      - lang = “código de idioma”
    - Ejemplos:
      - <div id=”div1” lang = “es”>...</div>
      - <p lang = “es”>...</p>
- **none:**
  - Las palabras no se cortan.
  - Puede provocar desbordamiento del texto respecto del contenedor donde éste, esté incluido.
- Ejemplos:
  - {hyphens:auto}
  - {hyphens:none}
  - {hyphens>manual}

## LISTAS.

### list-style-type

- Cambiar los símbolos de las listas.
- Sintaxis:
  - {list-style-type: valores}
- Valores:
  - Disc:
    - Circulo relleno.
  - Circle:
    - Circulo vacío.
  - Square:
    - Cuadrado.
  - Decimal:
    - Números correlativos.
  - Decimal-leading-zero:
    - Números precedidos por un cero (01 ,02, 03, ...)
  - Upper-roman:
    - Números romanos en mayúsculas (I, II, III, IV...).
  - Lower-roman:
    - Números romanos en minúsculas (i, ii, iii, iv...).
  - upper-alpha o upper-latin:
    - Letras mayúsculas.
  - lower-alpha o lower-latin:
    - Letras minúsculas.

- None:
  - Elimina el símbolo.
- Lower-greek:
  - Letras griegas en minúscula ( $\alpha, \beta, \chi \dots$ ).
- Armenian:
  - Símbolos armenios
- Georgian:
  - Símbolos georgianos.
- Ejemplos:
  - {list-style-type:upper-roman}
  - list-style-type:none}
  - {list-style-type:decimal}

### list-style-image

- Permite reemplazar los marcadores de las listas por símbolos gráficos.
- Se usan rutas absolutas o relativas, locales o externas.
- Sintaxis:
  - {List-style-image: url(ruta de acceso/archivo.extensión)}
- Ejemplos:
  - local:
    - {List-style-image: url(imagenes/icono.ico)}
  - Externa:
    - {List-style-image: url(http://www.imagenes.com/iconos/icono.ico)}

### list-style-position

- Desplaza la ubicación de los marcadores de viñeta o numeración a otra posición.
- Sintaxis:
  - {List-style-position: valores}
- Valores:
  - **outside:**
    - Valor por defecto.
    - Desplazamiento a la izquierda.
    - Todas las líneas de texto desplazadas respecto al marcador o número.
  - **inside:**
    - Desplazamiento a la derecha.
    - Primera línea desplazada y el resto alineadas respecto al marcador o número.
- Ejemplo:
  - {list-style-position: inside}

### Notación abreviada para listas:

- Los valores type, position e imagen se pueden incluir en una única declaración.
- Sintaxis:
  - {list-style: url(ruta de acceso/archivo.extensión) valor de position valor de type}
- Ejemplo:
  - {list-style: url(guitarra.png) inside upper-roman}

## TABLAS

- Propiedades para la estructura de la tabla no para el contenido.
- Propiedades:
  - **border-spacing:**
    - Permite especificar el espaciamiento entre celdas.
    - Sintaxis:
      - {border-spacing: número unidad de medida}
    - Ejemplo:
      - {border-spacing: 20px}
      - {border-spacing: 2%}
  - **border-collapse:**
    - Permite fusionar o no los bordes (tabla-celda) que HTML aplica a una tabla.
    - Sintaxis:
      - {border-collapse: valores}
    - Valores:
      - **collapse:**
        - Fusiona ambos bordes, aspecto borde único.
      - **separate:**
        - Se muestran los 2 bordes separados.
        - Valor por defecto.
    - Ejemplo:
      - {border-collapse: collapse}
  - **caption-side**
    - Permite cambiar la posición de la leyenda de una tabla.
    - Sintaxis:
      - {caption-side: valores}
    - Valores:
      - **top**
        - Arriba
      - **bottom**
        - Debajo de la tabla.
    - Ejemplo:
      - CSS:
        - {caption-side: bottom}
      - HTML:

```
<table >
  <caption>
    Texto de la leyenda
  </caption>
</table >
```
  - **empty-cells:**
    - Determina el comportamiento del navegador cuando se encuentre celdas vacías.
    - Elimina bordes y color de fondo de las celdas si éste ha sido aplicado con las etiquetas <tr> y <td>.
    - Sintaxis:
      - {empty-cells: valores}

- Valores:
  - **show:**
    - Muestra la celda vacía.
    - Bordes y fondo también son visibles.
  - **hide:**
    - No muestra la celda vacía.
    - Bordes y fondo no son visibles.
  - Ejemplo:
    - {empty-cells:hide}

## **COLUMNAS.**

### **column-count.**

- Determina el número de columnas.
- Sintaxis:
  - {column-count: número entero}
- Ejemplo:
  - {column-count: 3}

### **column-width.**

- Anchura de la columna.
- Sintaxis:
  - {column-width: número unidad de medida}
- Ejemplo:
  - {column-width: 50px}

### **column-gap.**

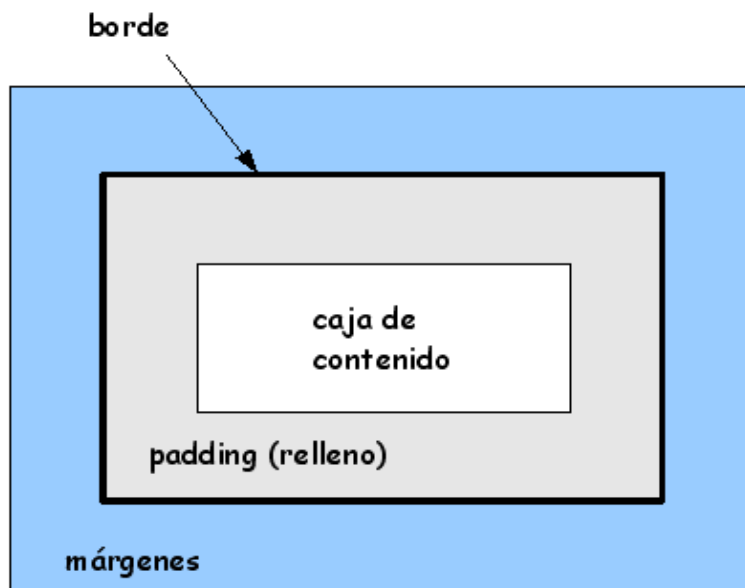
- Determina el espacio o padding entre 2 columnas.
- Sintaxis:
  - {column-gap: número unidad de medida}
- Ejemplo:
  - {column-gap: 10px}

### **column-rule.**

- Define un borde o línea entre columnas.
- Sintaxis:
  - {column-rule: grosor estilo color}
- Ejemplo:
  - {column-rule: 1px solid red}
- Variantes de la propiedad.
  - column-rule-color: color
  - column-rule-style: estilo
  - column-rule-width: grosor

## **MODELO DE CAJAS.**

- En CSS todos los elementos de una página se representan mediante cajas rectangulares.
- **Caja:**



### Modelo de caja ('box model')

- Zona rectangular que definida por:
  - Contenido.
  - Margen interior o padding.
  - Borde o border.
  - Margen externo o margin.
- Se crean automáticamente al insertar una etiqueta HTML para encerrar el contenido.
- Son invisibles mientras no se le dé un borde, un color de borde o un color de fondo.
- Las cajas se relacionan entre sí:
  - Apilándose.
  - Colocándose una al lado de otra.
  - Superponiéndose.

### Ajustes:

- Los 4 márgenes exteriores en cualquier sentido.
- Los 4 márgenes interiores en cualquier sentido.
- Los 4 bordes con un grosor, estilo y un color.
- Contenido en anchura y altura.

### Tipos de cajas:

#### En bloque (block).

- La caja fuerza un salto de línea al final del contenido.
- Respetan las propiedades width, height, márgenes y bordes.
- Elementos de bloque:



- address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul, dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.

### En línea (inline).

- Ocupan el espacio de su contenido.
- La caja no fuerza un alto de línea a no ser que varios elementos de línea acaben ocupando toda la línea.
- No respetan width, ni height, ya que su anchura y altura dependen del tamaño en píxeles de su contenido.
- No tienen márgenes externos inferior ni superior, sólo izquierdo y derecho.
- Elementos en línea:
  - a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

### Elementos híbridos.

- Elementos en línea o bloque colocados en una disposición distinta a la suya por defecto.
- Usar elementos híbridos es útil para maquetación.
- Los siguientes elementos pueden ser en línea y de bloque según las circunstancias:
  - button, del, iframe, ins, map, object, script.

### Flujo normal.

- Disposición de un elemento en bloque o en línea de forma predeterminada, si no se le ha dado otra con alguna instrucción.

### Propiedad display.

- Define el tipo de caja que se le aplica a un elemento.
- Sintaxis:
  - {display:valores}
- Valores:
  - **Block.**
    - Disposición en forma de bloque.
  - **Inline.**
    - Disposición en forma de línea.
  - **Inline-block.**
    - Para colocar elementos de bloque en línea.
  - **none.**
    - Oculta un elemento y el resto rellenan el hueco dejado por el elemento que se oculta con otro elemento.
  - **Otros:**
    - grid, flexbox, table, ...

### ANCHURA Y ALTURA DE LAS CAJAS.

- Permiten modificar las dimensiones del contenido.
- Propiedades:

- **width**
  - Fija la anchura del contenido.
  - Sintaxis:
    - {width: número unidad de medida o auto}
  - Unidades de medida relativas o absolutas.
  - Auto ajusta automáticamente las dimensiones al contenido.
- **height:**
  - Fija la altura del contenido.
  - Sintaxis:
    - {height: número unidad de medida o auto}
  - Unidades de medida relativas o absolutas.
  - Auto ajusta automáticamente las dimensiones al contenido.
- Otras propiedades para dimensiones:
  - max-width:
    - Ajuste de la anchura máxima.
  - min-width:
    - Ajuste de la anchura mínima.
  - max-height:
    - Ajuste de la altura máxima.
  - min-height:
    - Ajuste de la altura mínima.

## BORDES.

- Propiedad que permite establecer las características que tendrán los bordes de las cajas.
- Propiedades:
  - **border.**
    - Permite ajustar todas las características de un borde.
    - Sintaxis:
      - {border: grosor estilo color}
      - Las propiedades grosor y estilo son obligatorias.
      - Color por defecto el negro.
    - Valores:
      - Grosor:
        - Se pueden utilizar unidades de medida absolutas o relativas.
      - Estilos
        - Permite especificar la forma que tendrá el borde.
    - Tipos:
      - **solid**:
        - Trazo continuo simple.
      - **dashed**
        - Trazo discontinuo de guiones.
      - **dotted.**
        - Trazo discontinuo de puntos.
      - **double**
        - Trazo continuo doble.
      - **none.**

- Sin borde.
- **groove.**
  - Efecto 3D grabado en la página.
  - Opuesto a ridge.
- **ridge.**
  - Efecto 3D relieve que sobresale en la página.
  - Opuesto a Groove.
- **inset.**
  - Bordas entrantes, incrustados en la página.
  - Inverso a outset.
- **outset.**
  - Bordas salientes, resaltados en la página.
  - Inverso a inset.
- **hidden.**
  - Sin borde.
  - Bordas ocultos que influyen sobre el borde adyacente.
- **border-width.**
  - Permite ajustar el grosor de los bordes juntos o por separado.
  - El grosor no se muestra si no se define un estilo.
  - Unidades de medida absolutas o reactivas. Uso muy común de los píxeles.
  - Sintaxis:
    - {border-width: 1 valor o 2 valores o 3 valores o 4 valores.}
  - Valores:
    - 1 valor: 4 bordes mismo grosor (auto o valor numérico).
    - 2 valores: 1º valor para el grosor para bordes verticales (superior e inferior), 2º para los bordes horizontales (izquierdo y derecho).
    - 3 valores: 1º grosor borde superior, 2º número: grosores bordes izquierdo y derecho, 3º grosor borde inferior.
    - 4 valores: 1º borde superior, 2º borde derecho, 3º borde inferior y 4º borde izquierdo.
- **border-style.**
  - Permite ajustar el estilo de los bordes juntos o por separado.
  - El estilo no se muestra si no se define un grosor de borde.
  - Sintaxis:
    - {border-style: 1 valor o 2 valores o 3 valores o 4 valores.}
  - Valores:
    - 1 valor: 4 bordes mismo estilo.
    - 2 valores: 1º valor para bordes verticales (superior e inferior), 2º para los bordes horizontales (izquierdo y derecho).
    - 3 valores: 1º grosor borde superior, 2º número: estilo bordes izquierdo y derecho, 3º para borde inferior.
    - 4 valores: 1º borde superior, 2º borde derecho, 3º borde inferior y 4º borde izquierdo.
- **border-color.**
  - Permite ajustar el color de los bordes, o juntos, o por separado.

- Si no se especifica, el color por defecto es el negro.
- Sintaxis:
  - {border-color: 1 valor o 2 valores o 3 valores o 4 valores.}
- Valores:
  - 1 valor: 4 bordes mismo color.
  - 2 valores: 1º valor para bordes verticales (superior e inferior), 2º para los bordes horizontales (izquierdo y derecho).
  - 3 valores: 1º grosor borde superior, 2º número: colores bordes izquierdo y derecho, 3º para borde inferior.
  - 4 valores: 1º borde superior, 2º borde derecho, 3º borde inferior y 4º borde izquierdo.

### **Márgenes Externos:**

- Ajuste de los márgenes externos.
- Los que van desde el borde hacia otra caja o los límites de la ventana del navegador.
- Unidades de medida absolutas o relativas.
- Sintaxis:
  - {margin: 1 valor o 2 valores o 3 valores o 4 valores.}
- Valores:
  - **1 valor:** 4 márgenes iguales (auto o valor numérico).
  - **2 valores:** 1º valor márgenes verticales (superior e inferior), 2º valor márgenes horizontales (izquierdo y derecho).
  - **3 valores:** 1º margen superior, 2º número: márgenes izquierdo y derecho, 3º margen inferior.
  - **4 valores:** 1º margen superior, 2º margen derecho, 3º margen inferior y 4º margen izquierdo.
- Otras propiedades:
  - {margin-top: valor}
  - {margin-left: valor}
  - {margin-right: valor}
  - {margin-bottom: valor}
- Ejemplos:
  - <p style=" margin-left: 300px">
  - <p style=" margin: 45px">
  - <p style=" margin: auto">
  - <p style=" margin: 45px 30px">
  - <p style=" margin: 45px 30px 20px">
  - <p style=" margin: 45px 30px 20px 50px">

### **Márgenes Internos:**

- Ajuste de los márgenes internos.
- Los que van desde el contenido al borde.

- También se denominan relleno o espaciado.
- Unidades de medida absolutas o relativas.
- Sintaxis:
  - {padding: 1 valor o 2 valores o 3 valores o 4 valores.}
- Valores:
  - **1 valor**: 4 márgenes iguales (auto o valor numérico).
  - **2 valores**: 1º valor márgenes verticales (superior e inferior), 2º valor márgenes horizontales (izquierdo y derecho).
  - **3 valores**: 1º margen superior, 2º número: márgenes izquierdo y derecho, 3º margen inferior.
  - **4 valores**: 1º margen superior, 2º margen derecho, 3º margen inferior y 4º margen izquierdo
- Otras propiedades:
  - padding-top: valor
  - padding-left: valor
  - padding-right: valor
  - padding-bottom: valor
- Ejemplo:
  - <p style=" padding-left: 300px">
  - <p style=" padding: 45px>
  - <p style=" padding: 45px 30px>
  - <p style=" padding: 45px 30px 20px>
  - <p style=" padding: 45px 30px 20px 50px>

### Ajuste anchura-altura de las cajas.

- Se puede, o no, incluir las medidas de los márgenes y bordes en la medida del contenido.
- Sintaxis:
  - {box-sizing: valores}
- Valores:
  - **content-box**:
    - Al aplicar las medidas al contenido (width y height), las de padding y border añaden a la suma total.
    - El cálculo de la anchura y altura de la caja se realiza con los valores de las propiedades width y height, a los que se le añaden los valores de anchura del borde y el relleno (padding).
    - Valor por defecto.
  - **border-box**:
    - Al aplicar las medidas al contenido (width y height), ya están incluidas las medidas de padding y border en la suma total.
    - El cálculo de la anchura y altura de la caja se realiza solo con los valores de las propiedades width y height, no se tiene en cuenta la anchura del borde y el relleno (padding)
- Ejemplos:

- `p{width:500px;padding:10px;border:1px solid red; box-sizing: content-box;}->`  
500+10+10+1+1 = 522 content-box
- `p{width:500px;padding:10px;border:1px solid red; box-sizing: border-box;}-> }->`  
478+10+10+1+1 = 500 border-box

## COLOR DE FONDO.

- Permite aplicar un color de fondo un elemento.
- Sintaxis:
  - `{background-color:valores}`
- Valores:
  - **Hexadecimal clásico:**
    - Uso los dígitos 0 a 9 y las letras a-f.
    - Sintaxis:
      - `#rrggbb`
    - Ejemplo:
      - `#76a42e`
  - **Hexadecimal abreviado:**
    - Uso los dígitos 0 a 9 y las letras a-f.
    - Sintaxis:
      - `#rgb`
    - Ejemplo:
      - `#7a2` equivale a `#77aa22`
  - **RGB Decimal.**
    - 1 byte por color usando su valor decimal.
    - Sintaxis:
      - `rgb(color rojo, color verde, color azul)`
      - `rgb(0-255,0-255,0-255)`
    - Ejemplo:
      - `rgb(6,78,123)`
  - **RGB Porcentaje.**
    - 1 byte por color usando su valor porcentual.
    - Sintaxis:
      - `rgb(color rojo, color verde, color azul)`
      - `rgb(0%-100%,0%-100%,0%-100%,)`
    - Ejemplo:
      - `rgb(6%,78%,23%)`
  - **RGB Canal Alfa.**
    - Permite incluir transparencia en el color.
    - 1 byte por color usando su valor decimal más un cuarto valor comprendido entre 0 y 1 con decimales para la transparencia:
      - 0 transparencia total.
      - 1 opacidad total.
    - Sintaxis:
      - `rgb(color rojo, color verde, color azul, transparencia)`
      - `rgb(0-255,0-255,0-255,0.0 a 1.0)`
    - Ejemplo:

- rgb(6,78,123,0.75)
- **HSL.**
  - Modelo de color basado en la percepción humana.
  - Valores:
    - **h o hue:**
      - Tono o matiz.
    - **s:**
      - Saturación o cantidad de gris que tiene un color.
    - **l:**
      - Luminosidad o luminancia.
      - Cantidad de blanco o negro que tiene un color.
  - Sintaxis:
    - rgb(tono de color, saturación, luminosidad)
    - hsl(0-359,0%-100%,0%-100%)
  - Ejemplo:
    - hsl(210,56%,23%)
- **HSL Canal Alfa.**
  - Permite controlar el grado de transparencia de un color en el modelo hsl usando un cuarto valor comprendido entre 0 y 1 con decimales para la transparencia:
    - 0 transparencia total.
    - 1 opacidad total.
  - Sintaxis:
    - rgb(tono de color, saturación, luminosidad, transparencia)
    - hsl(0-359,0%-100%,0%-100%,0.0 a 1.0)
  - Ejemplo:
    - hsl(210,56%,23%,0.40)
- **Nombre del color.**
  - Se escribe el nombre del color en inglés.
  - Hay 140 valores de colores seguros, es decir, colores que pueden mostrar en cualquier dispositivo.
  - Sintaxis:
    - propiedad: nombre del color.
  - Ejemplo:
    - color:green

## OPACIDAD.

- Permite ajustar el grado de transparencia de un elemento.
- Sintaxis:
  - {opacity: valor}
- Valores:
  - Comprendido entre 0 y 1 con decimales, distintos grados de transparencia u opacidad.
  - Valor 1: opacidad completa.
  - Valor 0: opacidad nula.

## IMÁGENES DE FONDO.

- Para aplicar una imagen a un fondo.

- Se pueden usar rutas absolutas o relativas al archivo con la imagen.
- Sintaxis:
  - {background-image: url (ruta de acceso/archivo.extensión)}
- Ejemplos:
  - local:
    - {background-image: url(imagenes/playa.jpg)}
  - Externa:
    - {background-image: url(http://www.imagenes.com/imagenes/montaña.png)}

### Tamaño de la imagen de fondo:

- Sirve para definir el tamaño de la imagen de fondo.
- Sintaxis:
  - {background-size: valores}
- Valores:
  - **auto:**
    - Imagen ocupa el tamaño el real.
  - **cover:**
    - La imagen se redimensiona proporcionalmente y cubre toda la caja.
  - **tamaño x:** (anchura)
    - Valor numérico en unidades absolutas o relativas para definir dimensión horizontal
  - **tamaño y:** (altura)
    - Valor numérico en unidades absolutas o relativas para definir dimensión vertical.
    - Es opcional, si no se especifica se redimensiona proporcionalmente al eje x.
  - **contain:**
    - Escala la imagen haciendo que cubra el 100% de la anchura o la altura del tamaño de la ventana o contenedor, por lo que pueden quedar huecos que se rellenan con la propia imagen.

### Repetición de una imagen:

- Permite poner o no una imagen en mosaico.
- Sintaxis:
  - {background-repeat: valores}
- Valores:
  - Repeat:
    - La imagen se repite en mosaico.
    - Valor por defecto.
  - No-repeat
    - La imagen no se repite en mosaico.
  - Repeat-y
    - La imagen se repite en mosaico en vertical.
  - Repeat-x
    - La imagen se repite en mosaico en horizontal.

### Posición de una imagen:

- Permite posicionar con precisión una imagen de fondo a nivel pixel.
- No funciona si la imagen está repetida (background-repeat: no-repeat activado).
- No funciona si no hay imagen de fondo.



- Sintaxis:
  - {background-position: valor eje x valor eje y}
- Valores:
  - Valor eje x:
    - posición respecto del borde izquierdo.
  - Valor eje y:
    - posición respecto del borde superior.

#### Fijación del fondo:

- La imagen de fondo se fija y no se desplaza con el resto de la página.
- Sintaxis:
  - {background-attachment: valores}
- Valores:
  - **scroll.**
    - Imagen se mueve con el contenido de la página.
    - Valor por defecto.
  - **fixed.**
    - Imagen no se mueve con el contenido de la página.

#### EFFECTOS PARA IMÁGENES

##### FILTROS

- Aplican distintos efectos gráficos a las imágenes.
- Sintaxis:
  - **Efecto estático.**
    - Selector para la imagen { filter: tipo de efecto(valor);}
  - **Especo con sobrevuelo.**
    - Selector para la imagen:hover { filter: tipo de efecto(valor);}
- **TIPOS DE FILTROS.**
  - **Desenfoque (blur).**
    - Sintaxis:
      - filter:blur(valor numérico px);
      - valor:
        - Indica la intensidad del efecto de desenfoque medido en píxeles.
        - Se expresa con un número que puede incluir decimales.
        - A mayor valor, mayor efecto.
    - Ejemplos:
      - img:hover{filter:blur(1.5px);}
      - img{filter:blur(3px);}
  - **Escala de grises (grayscale).**
    - Sintaxis:
      - filter:grayscale(valor numérico %);
      - valor:
        - **0%.**
          - Imagen con color original.
        - **100%.**
          - Imagen sin color en escala de grises.
        - **Valores intermedios.**

- Distintas proporciones de color y grises.
- Ejemplos:
  - `img:hover{filter:grayscale(75%);}`
  - `img{filter:grayscale(0%);}`
- **Sepia (sepia).**
  - Sintaxis:
    - `filter:sepia(valor numérico %);`
    - valor:
      - **0%.**
        - Imagen con color original.
      - **100%.**
        - Imagen completamente en sepia.
      - **Valores intermedios.**
        - Distintas proporciones de sepia.
  - Ejemplos:
    - `img:hover{filter:sepia(75%);}`
    - `img{filter:sepia(0%);}`
- **Transparencia u opacidad (opacity).**
  - Sintaxis:
    - `filter:opacity(valor numérico);`
    - valor:
      - **0.**
        - Imagen transparente.
      - **1.**
        - Imagen opaca.
      - **Valores intermedios con decimales.**
        - Distintos grados de transparencia u opacidad.
  - Ejemplos:
    - `img:hover{filter:opacity(0.65);}`
    - `img:hover{filter:opacity(0);}`
    - `img{filter:opacity(1);}`
- **Saturación (saturate).**
  - Sintaxis:
    - `filter:saturate(valor numérico %);`
    - valor:
      - **100%.**
        - Saturación normal (color normal).
      - **> 100%.**
        - Sobresaturación (colores más intensos o vivos).
      - **< 100%.**
        - Desaturación (colores más apagados).
  - Ejemplos:
    - `img:hover{filter:saturate(175%);}`
    - `img:hover{filter:saturate(50%);}`
    - `img{filter:saturate(100%);}`
- **Rotación de color (hue-rotate).**
  - Sintaxis:
    - `filter:hue-rotate(valor numérico en grados);`
    - valor:

- **360 deg.**
    - Color original.
  - **Hacia 0 deg.**
    - Cambios de color según grado.
- Ejemplos:
  - `img:hover{filter:hue-rotate(180deg);}`
  - `img{filter:hue-rotate(222deg);}`
- **Inversión de colores (invert).**
  - Sintaxis:
    - `filter:invert(valor numérico %);`
    - valor:
      - **0%.**
        - Imagen con color original.
      - **100%.**
        - Imagen con colores invertidos o en negativo.
      - **Valores intermedios.**
        - Distintas proporciones de inversión.
  - Ejemplos:
    - `img:hover{filter:invert(100%);}`
    - `img{filter:invert(45%);}`
- **Brillo (brightness).**
  - Sintaxis:
    - `filter: brightness(valor numérico %);`
    - valor:
      - **0%.**
        - Imagen en negro.
      - **100%.**
        - Imagen con brillo original.
      - **Valores intermedios.**
        - Distintas proporciones de brillo.
  - Ejemplos:
    - `img:hover{filter: brightness(100%);}`
    - `img{filter: brightness(45%);}`
- **Contraste (contrast).**
  - Sintaxis:
    - `filter: contrast(valor numérico %);`
    - valor:
      - **0%.**
        - Imagen en gris.
      - **100%.**
        - Imagen con contraste original.
      - **Valores intermedios.**
        - Distintas proporciones de contraste.
  - Ejemplos:
    - `img:hover{filter: contrast(100%);}`
    - `img{filter: contrast(45%);}`
- **Sombra (drop-shadow).**
  - Sintaxis y efecto idéntico a **text-shadow** (sombra para textos) y **box-shadow** (sombra para cajas).

- Sintaxis:
  - `filter: drop-shadow (x y z color);`
  - valor:
    - **x**
      - Margen de la sombra hacia la derecha si el valor es positivo, o hacia la izquierda si es negativo.
    - **y**
      - Margen de la sombra hacia abajo si el valor es positivo, o hacia arriba si es negativo.
    - **z**
      - Intensidad del degradado o desenfocado de la sombra.
      - Opcional.
      - Valor por defecto: 0
    - **color:**
      - Color de la sombra.
- Ejemplos:
  - `img:hover{filter: drop-shadow (5px 5px 2px red);}`
  - `img:hover{filter: drop-shadow (10px 4px green);}`
  - `img{filter: drop-shadow (-34px 28px 10px blue);}`
- **Filtros múltiples.**
  - Se pueden aplicar varios filtros a la vez, pero usando una sola propiedad **filter** con varios valores separados por espacios en blanco, porque si se usan varios **filter** independientes, el último aplicado sustituirá todos los anteriores.
  - Sintaxis:
    - **Efecto estático.**
      - Selector para la imagen `{filter: tipo de efecto1(valor) tipo de efecto2(valor) tipo de efecto3(valor) ...;}`
    - **Especto con sobrevuelo.**
      - Selector para la imagen: `hover {filter: tipo de efecto1(valor) tipo de efecto2(valor) tipo de efecto3(valor) ...;}`
  - Ejemplos:
    - `img:hover {filter: saturate(80%) contrast(120%) brightness(56%);}`
    - `img {filter: sepia (80%) opacity(.75);}`

## SPRITES.

- Una página web puede necesitar cargar muchísimas imágenes, tanto para usarlas como fondo, como para usarlas como objetos o parte del contenido.
- Las imágenes como fondo se manejan con propiedades como *background-image*, *background-size*, *background-repeat* y *background-position*.
- **Concepto.**
  - Un *sprite* es un conjunto de imágenes en una sola imagen, de modo que, cargando una sola imagen, se tienen disponibles todas las que la conforman y van a usarse en el documento web sin necesidad de descargarlas del servidor cuando se necesiten. Así, se puede mejorar el tiempo de carga de una página web al reducir el número de peticiones a un servidor web.

- Cargar una imagen implica una petición al servidor, una respuesta de éste, la carga de la imagen en el navegador, etc. Aunque una imagen sea pequeña y pese poco, todas las peticiones que se realicen consumen tiempo y ralentizan la carga de una página.
- **Utilidad.**
  - Accediendo con CSS a la posición de cada imagen, se pueden visualizar todas las que forman el *sprite* como si fueran imágenes únicas e independientes y estuvieran guardadas en archivos diferentes.
  - Usando pseudoclases CSS como *hover* o scripts de JavaScript se pueden conseguir cambios de estado o de posición de la imagen.
  - También pueden crearse efectos de movimiento y animaciones usando JavaScript u otros lenguajes de programación.
- **Creación de un Sprite.**
  - Se pueden usar programas de edición imagen como Photoshop o Gimp, o crearlas en línea subiendo las imágenes a webs que permiten generar la imagen única con todas ellas.
    - <https://www.toptal.com/developers/css/sprite-generator/>

## AJUSTE DE OBJETOS.

### object-fit.

- Especifica como debe comportarse un objeto respecto de un contenedor.
- Se usa especialmente con imágenes, videos, etc., para cambiar su tamaño y que se ajusten a las dimensiones del contenedor donde están incluidos.
- Para que la imagen puede redimensionarse dentro del contenedor usando esta propiedad debe tener un tamaño aplicado con *width* y *height*, si no, tendrá su tamaño original y si es mayor que él, lo desbordará.
- Sintaxis:
  - object-fit: valores;
  - **valores.**
    - **fill.**
      - La imagen se redimensiona para ocupar todo el contenedor.
      - Si es necesario, se deforma o estira y no mantiene la proporción anchura - altura.
      - Valor predeterminado.
    - **contain.**
      - La imagen se redimensiona aumentando o reduciendo su tamaño para ajustarse al contenedor.
      - Mantiene su relación de aspecto.
    - **cover.**
      - La imagen mantiene su relación de aspecto para ajustarse al contenedor.
      - Si la relación de aspecto no coincide con la del contenedor, posiblemente la imagen se recorte para ajustarse.
    - **none.**
      - La imagen no cambia de tamaño.
    - **scale-down.**
      - El objeto se redimensiona a la versión más pequeña de **none** o **contain**.
- Ejemplos:

```
img{
width:300px;
height:200px;
object-fit:fill;
```

```

}
img{
  width:100%;
  height:100%;
  object-fit:cover;
}

```

## TRANSFORMACIONES.

- Aplican cambios de posición, tamaño, etc., a las imágenes.
- Sintaxis:
  - **Efecto estático.**
    - Selector para la imagen {transform: tipo de transformación(valor);}
  - **Especto con sobrevuelo.**
    - Selector para la imagen:hover {transform: tipo de transformación (valor);}
- **TIPOS DE TRANSFORMACIONES.**
  - **Escala o tamaño (scale).**
    - Sintaxis:
      - transform: scale(valor numérico);
      - valor:
        - **1.**
          - Imagen a tamaño original.
        - **> 1.**
          - Aumento de tamaño según múltiplo utilizado.
        - **< 1.**
          - Reducción de tamaño según múltiplo utilizado.
    - Ejemplos:
      - img:hover{transform: scale(1.3);}
      - img{transform: scale(0.5);}
  - **Zoom (scale).**
    - Para conseguir un efecto similar a un zoom se puede usar scale.
    - Para ello es necesario crear un contenedor con unas dimensiones iguales que la imagen y ocultar el aumento de éste cuando la imagen aumente, usando la propiedad overflow con el valor hidden.
    - Sintaxis:
      - **Para la imagen.**

```
div img:hover{transform: scale(valor numérico);}
```
      - **Para el contenedor de la imagen.**

```
div{width: misma anchura que imagen;
  height: misma altura que imagen;
  overflow: hidden;}
```
    - Ejemplos:
      - div img:hover{transform: scale(1.3);}
      - div{width: 500px;
 height: 400px;
 overflow: hidden;}
  - **Traducción (translate).**
    - Sintaxis:

- transform: **translate**(valor numérico X, valor numérico Y);
- valor:
  - **valor numérico X.**
    - Desplazamiento en horizontal de la imagen.
    - Valor positivo, desplazamiento hacia derecha.
    - Valor negativo, desplazamiento hacia izquierda.
    - Medida en píxeles.
  - **valor numérico Y.**
    - Desplazamiento en vertical de la imagen.
    - Valor positivo, desplazamiento hacia abajo.
    - Valor negativo, desplazamiento hacia arriba.
    - Medida en píxeles.
- Ejemplos:
  - `img:hover{transform: translate(50px, 150px);}`
  - `img{transform: translate (200px, 0px);}`

### **box-shadow.**

- Permite aplicar sombra a una caja.
- Valores en píxeles.
- Sintaxis:
  - `{box-shadow: x y z color}`
- Valores:
  - **x**
    - margen de la sombra hacia la derecha si el valor es positivo o hacia la izquierda si es negativo.
  - **y**
    - margen de la sombra hacia abajo si el valor es positivo o hacia arriba si es negativo.
  - **z**
    - Intensidad del degradado o desenfocado de la sombra.
    - Opcional.
    - Valor por defecto: 0
  - **Color**:
    - Color de la sombra.
- Ejemplos:
  - `{box-shadow: 5px 5px red}`
  - `{box-shadow: -34px 28px 10px blue}`

### **Degradados de color.**

- Mezcla de colores que se fusionan entre sí.
- Se pueden mezclar dos o más colores entre sí, o un color con blanco o negro.
- Se pueden usar las propiedades `background` o `background-image`.
- Tipos:
  - **Lineal**:
    - El gradiente de mezcla es lineal en vertical, horizontal o diagonal.
    - Sintaxis:
      - `background: linear-gradient (punto de partida, color inicial, color final)`
    - Valores punto de partida:

- to Top
- to Left.
- to Right
- to Bottom (por defecto).
- to “extremo a extremo”:
  - to right bottom.
- número grados expresado con deg:
  - 45deg.
- Si no se especifica, por defecto, el punto de partida es la parte superior y se produce el degradado hacia abajo.
- Ejemplos:
  - body {background: linear-gradient (to left, blue, white);}
  - body {background: linear-gradient (red, yellow);}
  - body {background: linear-gradient (to right bottom, blue, green);}
- **Punto de parada.**
  - Permiten mezclar varios colores especificando con un porcentaje donde aparece el color sólido.
  - Entre porcentajes se mezclan los colores especificados.
  - Sintaxis:
    - {background: linear-gradient(punto de partida, color 1 número %, color 2 número %,.....)}
  - Ejemplo:
    - p{background: linear-gradient(to left, yellow 20%, red 60%, blue 80%, green 90%)}
- **Radial o circular.**
  - Permite crear degradados circulares o elípticos.
  - Sintaxis:
    - {background: radial-gradient (posición de partida, color inicial, color final).
      - Con esta opción, puede ser necesario utilizar el prefijo de navegador correspondiente para visualizar correctamente el degradado. (-moz-, -webkit-, etc.).
    - {background: radial-gradient(forma at posición de partida, color inicial, color final)}
    - {background: radial-gradient(forma, color inicial, color final)}
    - {background: radial-gradient(forma at número % color inicial número % color final, color inicial, color final)}
    - {background: radial-gradient(forma at número px posición desde izquierda número px posición desde arriba, color inicial, color final)}
  - Punto de partida:
    - Se usa el punto central por defecto si no se especifica un punto de partida.
    - top, left, right, bottom, center (opción por defecto). Puede ser necesario el uso de prefijos de navegador para utilizarlos.
  - Formas:
    - Circle.
    - Ellipse.
  - Ejemplos:



- {background: radial-gradient (ellipse, #fdbb2d, #22c1c3)}
- {background: radial-gradient (circle at top left, #fdbb2d, #22c1c3)}
- background: -webkit-radial-gradient (top, violet, orange);
- {background: radial-gradient (circle at 20% 70%, red, blue)}
- background: radial-gradient(circle at 100px 500px, red, blue);
- **Puntos de parada.**
  - Permiten mezclar varios colores especificando, con la misma o distinta unidad de medida absoluta o relativa, dónde aparecerá el color sólido.
  - Entre porcentajes se mezclan los colores especificados.
  - Sintaxis:
    - {background: radial-gradient(punto de partida, color 1 número %, color 2 número %, ..., color N número %)}
    - Con esta opción, puede ser necesario utilizar el prefijo de navegador correspondiente para visualizar correctamente el degradado. (-moz-, -webkit-, etc.).
    - {background: radial-gradient (posición de partida, color 1, color 2, ..., color N)}.
    - Con esta opción, puede ser necesario utilizar el prefijo de navegador correspondiente para visualizar correctamente el degradado. (-moz-, -webkit-, etc.).
    - {background: radial-gradient(posición unidad de medida punto de partida, color 1 número %, color 2 número %, ..., color N número %)}
    - Con esta opción, puede ser necesario utilizar el prefijo de navegador correspondiente para visualizar correctamente el degradado. (-moz-, -webkit-, etc.).
    - 
    - {background: radial-gradient(color 1 número %, color 2 número %, ... , color N número % )
    - {background: radial-gradient(color 1 número unidad de medida, color 2 número unidad de medida, ... , color N número unidad de medida )
    - {background: radial-gradient (color 1, color 2, ..., color N)
- Ejemplos:
  - {background: -moz-radial-gradient (left, yellow 20%, red 60%, blue 80%, green 90%)}
  - {background: radial-gradient (yellow 20%, red 60%, blue 80%, green 90%)}
  - background-image: radial-gradient (violet 50px, pink 200px, maroon 300px);
  - background-image: radial-gradient (yellow 25mm, orange 200px, red 10em, navy 90%);
  - {background: radial-gradient (yellow, red, blue, green)}

**Desbordamiento de texto.**

- Determina qué debe hacer el navegador cuando un elemento es más grande que el elemento padre que lo contiene.
- Sintaxis:
  - {overflow: valores}
- Valores:
  - **hidden:**
    - La parte que se desborda se oculta, sin posibilidad de verla ni poder accederse a ella.
  - **scroll:**
    - La parte que se desborda se oculta, quedando accesible mediante barras de desplazamiento.
  - **visible:**
    - La parte que se desborda se muestra ignorándose las especificaciones del elemento padre que la contiene.
    - Opción por defecto.
  - **auto:**
    - Se deja la gestión al navegador.

### **Bordes redondeados.**

- Permite redondear los bordes de las esquinas de un elemento.
- Sintaxis:
  - {border-radius: valor}
    - Mismo valor en horizontal y vertical en las 4 esquinas.
  - {border-radius: valor x / valor y}
    - Distinto valor en horizontal y vertical en las 4 esquinas.
  - {border-radius: valor 1 valor 2 valor 3 valor 4}
    - Depende del número de valores se aplica a todas el mismo o distinto borde redondeado
  - {border-top-left-radius: valor x valor y}
  - {border-top-right-radius: valor x valor y}
  - {border-bottom-left-radius: valor x valor y}
  - {border-bottom-right-radius: valor x valor y}
- Listas de valores:
  - Se incluyen 2 listas separadas por una línea inclinada (/).
  - La primera lista con los valores para los radios horizontales.
  - La segunda lista con los valores para los radios verticales.
  - Valores:
    - 1º superior.
    - 2º derecho.
    - 3º inferior.
    - 4º izquierdo.
  - Sintaxis:
    - {border-radius: lista de valores eje horizontal / lista de valores eje vertical}
  - Ejemplo:
    - {border-radius: 50px 20px 5px 68px / 10px 10px 10px 10px}

## Flotación de las cajas.

- Diseñado para texto que flotase alrededor de una imagen.
- Un elemento flota en la dirección que se le indique dentro de su contenedor.
- Permite poner un elemento lo más a la izquierda o a la derecha dentro del elemento contenedor o padre.
- No puede aplicarse en caso de posicionamiento absoluto.
- Cuando un elemento flota deja el hueco que ocupaba y otro elemento puede ocuparlo.
- Sintaxis:
  - {float: valores}
- Valores:
  - **left:**
    - Alinea a la izquierda el elemento indicado empujando a la derecha los demás elementos.
  - **right:**
    - Alinea a la derecha el elemento indicado empujando a la izquierda los demás elementos.
  - **none:**
    - No se especifica nada, se deja la gestión al navegar.

## Prohibir flotación.

- Evita que un elemento flote o tenga a sus lados algún elemento flotante, pasándolo a la siguiente línea.
- Anula el efecto introducido por float.
- Sintaxis:
  - {clear: valores}
- Valores:
  - **left:**
    - Anula los elementos flotantes a la izquierda.
  - **right:**
    - Anula los elementos flotantes a la derecha.
  - **none:**
    - Anula los elementos flotantes.
  - **both:**
    - Anula los elementos flotantes a ambos lados.

## POSICIONAMIENTO DE LAS CAJAS.

### Estático

- Posicionamiento normal de un elemento según si es de bloque o línea, y según procede el navegador.
- Sintaxis:
  - {position:static}

### Relativo:

- Posicionamiento respecto a su posición original en el flujo de ejecución predeterminada.
- Posicionamiento del elemento respecto de su posición normal o estática.
- El elemento deja un hueco que no se ocupa por el resto de los elementos.
- Posicionamiento definido mediante **coordenadas**:

- **Eje x:**
  - Distancia respecto del borde izquierdo o derecho de la ventana del navegador o de un elemento padre.
- **Eje y:**
  - Distancia respecto del borde superior o inferior de la ventana del navegador o de un elemento padre.
- Atributos:
  - Top: número unidad de medida.
  - Bottom: número unidad de medida.
  - Left: número unidad de medida.
  - Right: número unidad de medida.
- Las medidas pueden ser relativas o absolutas.
- Sintaxis:
  - {position: relative; valores de posición}
- Ejemplo:
  - {position: relative;  
top: 50px;  
left: 100px  
}

#### **Absoluto:**

- Posicionamiento respecto a la esquina superior izquierda del navegador o del elemento padre.
- El elemento deja un hueco que es ocupado por el resto de los elementos.
- Posicionamiento definido mediante **coordenadas**:
  - **Eje x:**
    - Distancia respecto del borde izquierdo o derecho de la ventana del navegador o de un elemento padre.
  - **Eje y:**
    - Distancia respecto del borde superior o inferior de la ventana del navegador o de un elemento padre.
  - Atributos:
    - Top: número unidad de medida.
    - Bottom: número unidad de medida.
    - Left: número unidad de medida.
    - Right: número unidad de medida.
  - Las medidas pueden ser relativas o absolutas.
- Sintaxis:
  - {position: absolute; valores de posición}
- Ejemplo:
  - {position: absolute;  
top: 50px;  
left: 100px}

#### **Fijo**

- Posicionamiento respecto a la esquina superior izquierda del navegador.

- El elemento se mantiene fijo en una posición exacta independientemente de que la página se desplace haciendo scroll.
- El elemento deja un hueco que es ocupado por el resto de los elementos.
- Posicionamiento definido mediante **coordenadas**:
  - **Eje x:**
    - Distancia respecto del borde izquierdo o derecho de la ventana del navegador.
  - **Eje y:**
    - Distancia respecto del borde superior o inferior de la ventana del navegador.
  - Atributos:
    - Top: número unidad de medida.
    - Bottom: número unidad de medida.
    - Left: número unidad de medida.
    - Right: número unidad de medida.
  - Las medidas pueden ser relativas o absolutas.
- Sintaxis:
  - {position: fixed; valores de posición}
- Ejemplo:
  - {position: fixed;
  - top: 50px;
  - left:100px}

## Sticky

- La posición de un elemento pasa a ser de relativa a fija.
- Al desplazarse el elemento se mueve hasta que queda fijo en un punto.
- El elemento deja un hueco que es ocupado por el resto de los elementos.
- Posicionamiento definido mediante **coordenadas**:
  - **Eje x:**
    - Distancia respecto del borde izquierdo o derecho de la ventana del navegador o de un elemento padre.
  - **Eje y:**
    - Distancia respecto del borde superior o inferior de la ventana del navegador o de un elemento padre.
  - Atributos:
    - Top: número unidad de medida.
    - Bottom: número unidad de medida.
    - Left: número unidad de medida.
    - Right: número unidad de medida.
  - Las medidas pueden ser relativas o absolutas.
- Sintaxis:
  - {position: sticky; valores de posición}
- Ejemplo:
  - {position: sticky;
  - top: 50px;
  - left:100px}

## Visibilidad de las cajas.

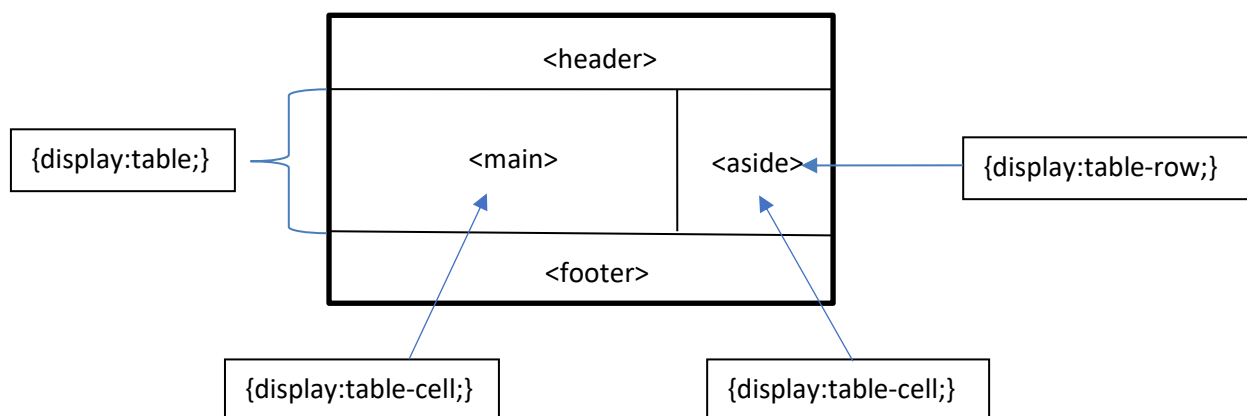
- Se puede definir si un elemento será o no visible.
- Sintaxis:
  - {Visibility: valores}
- Valores:
  - **Visible:**
    - Muestra el elemento.
    - Valor por defecto.
  - **Hidden:**
    - Oculta un elemento.
- La diferencia entre visibility: hidden y display: none es que el primero solo oculta, mientras que el segundo, rellena el hueco dejado por el elemento que se oculta con otro elemento.

## MAQUETACIÓN CON DISPLAY.

- Además de los clásicos valores para modificar la disposición del flujo normal de los elementos HTML (*block*, *inline* e *inline-block*) al usar la propiedad *display*, existen otros valores para componer o maquetar una página web.

## TABLAS.

- *display* permite maquetar contenidos parecidos a las tablas HTML, sin perder el significado del documento HTML al usar contenedores semánticos.
- Se pueden usar con contenedores <div>, <section>, <main>, <article>, etc.
- Los elementos se muestran como si fueran parte de una tabla, fila, columna o grupo de filas-columnas.
- La estructura de una página usando esta propiedad será la de una tabla, pero diseñada con CSS. Todas las columnas de la página tendrán las mismas dimensiones (altura o anchura) y el diseño no se deshará al redimensionar la ventana del navegador.



- Sintaxis:
  - {display: valores}
- Valores:
  - **table.**
    - Similar a <table>.
    - Asigna un comportamiento de tabla a los elementos que se le incluyan.
  - **table-row.**

- Similar a <tr>.
- Varios elementos se disponen en fila.
- Ejemplo:
  - `main, aside{display:table-row}`
- **table-column.**
  - Similar a <col>.
  - Varios elementos se disponen en columna.
- **table-cell.**
  - Similar a <th> y <td>.
  - Varios elementos se disponen en fila.
- **table-caption.**
  - Similar a <caption>.
  - Un título, por ejemplo, con <h1> a <h6> o párrafo se coloca como título de la tabla.
  - Puede incluir alineación o estilos para cambiar el aspecto como negritas, tipo y tamaño de fuente, etc.
  - Ejemplo:
    - `h1{display: table-caption; text align:center;}`
    - `p{ display: table-caption; text align:center;Font-size:20px; text-decoration:underline;}`
- **table-header-group.**
  - Similar a <thead>.
- **table-row-group.**
  - Similar a <tbody>.
- **table-footer-group.**
  - Similar a <tfoot>.
- **table-column-group.**
  - Similar a <colgroup>.
- **inline-table.**
  - El elemento es mostrado como una tabla de un nivel inline, una fila.

## FLEXBOX.

### Concepto.

- Método de diseño que permite que los elementos flexibles (ítems) de un contenedor, puedan alterar sus dimensiones y llenar el espacio disponible.
- Se usa con la propiedad *display*.

### Estructura.

- **Contenedor.**
  - Elemento de nivel superior (padre) que incluye en su interior una serie de elementos(ítems) flexibles.
  - Orientación del contenido:
    - **Eje principal:**
      - Orientación principal del contenido.
      - Puede ser vertical u horizontal.
      - Por defecto, horizontal (valor “row”).
    - **Eje secundario:**
      - Eje con orientación perpendicular respecto al principal

- Si el principal es horizontal, el secundario es vertical con valor “column”, y viceversa.
- **Ítem.**
  - Elementos (hijos) flexibles del contenedor.

#### Propiedades del contenedor o elemento padre de nivel superior.

- **display:flex.**
  - Convierte un contenedor en un contenedor flexible.
- **flex-direction.**
  - Establece como van a ubicarse los ítems dentro del contenedor.
  - Establece la orientación de éstos y su alineación.
  - Sintaxis:
    - {flex-direction:valores;}
  - Valores:
    - **row.**
      - Elementos en horizontal en el orden que cada uno tiene.
      - Alineación izquierda de los ítems.
    - **column.**
      - Elementos en vertical en su orden de creación.
      - Alineación superior de los ítems.
    - **row-reverse.**
      - Elementos en horizontal en orden inverso al de su creación.
      - Alineación derecha de los ítems.
    - **column-reverse.**
      - Elementos en vertical en orden inverso al de su creación.
      - Alineación inferior.
- **flex-wrap.**
  - Permite especificar el comportamiento del contenedor respecto a evitar que se desborden o no sus elementos contenidos.
  - En realidad, permite o no el salto de línea de los elementos hijos o ítems.
  - Si hay muchos, es posible que se modifiquen sus dimensiones para entrar en la línea que marca el eje.
  - Sintaxis:
    - {flex-wrap:valores}
  - Valores:
    - **nowrap.**
      - Los elementos aparecen en una sola línea, la del eje principal.
      - Para que entren todos en el contenedor puede ser necesario redimensionarlos, ya que, si no, podría producirse un desbordamiento.
      - Valor por defecto.
    - **wrap.**
      - Si los elementos no cogen en la línea, se producen uno o varios saltos de línea para irlos ubicando en los espacios disponibles.
      - Los elementos no se redimensionan.
    - **wrap-reverse.**
      - Igual que **wrap**, pero los elementos se ubican en orden inverso.
  - **Notación abreviada.**
    - Los valores de flex-wrap y flex-direction se pueden incluir en una sola declaración de estilo usando la propiedad flex-flow.



- Sintaxis:
    - {flex-flow: valor de dirección valor de desbordamiento}
  - Ejemplo:
    - {list-style: row wrap}
- **justify-content.**
  - Alineación respecto al eje principal.
  - Controla como se distribuyen los elementos o ítems a lo largo del contenedor.
  - Los elementos se distribuyen respecto al eje principal, horizontal si la dirección es “row” o vertical si es “column”.
  - Sintaxis:
    - {justify-content: valores;}
  - Valores:
    - **flex-start.**
      - Los elementos se colocan al principio del contenedor.
      - Valor por defecto.
    - **flex-end.**
      - Los elementos se colocan al final del contenedor:
        - Si el eje principal es *row* se colocan a la derecha.
        - Si el eje principal es *column* se colocan abajo.
    - **center.**
      - Elementos agrupados en el centro, en vertical u horizontal según eje principal.
    - **space-between.**
      - Los elementos se distribuyen ocupando todo el espacio disponible en el contenedor.
      - Los elementos no incluyen espacio al principio o al final, pero si entre ellos.
    - **space-around.**
      - Igual que space-between, pero incluyendo un espacio delante y detrás del primer y último elemento.
- **align-items.**
  - Alineación respecto al eje secundario o transversal.
  - Similar a **justify-content**.
  - Los elementos se distribuyen en vertical si la dirección es *row*, o en horizontal si es *column*.
  - Sintaxis:
    - {align-items: valores;}
  - Valores:
    - **flex-start.**
      - Los elementos se colocan arriba o a la izquierda dependiendo de flex-direction.
      - Valor por defecto.
    - **flex-end.**
      - Los elementos se colocan abajo o a la derecha del contenedor dependiendo de flex-direction.
    - **center.**
      - Elementos centrados según el eje secundario.
    - **stretch (estirar).**
      - Los elementos ocupan, en el eje secundario, todo el espacio permitido por el contenedor.

- Los elementos ocupan toda la altura o anchura dependiendo del valor flex-direction.
  - Valor por defecto.
  - **baseline.**
    - Elementos alineados según línea base de cada uno de ellos.
    - La línea base queda definida teniendo en cuenta los textos de los elementos.
- **row-gap y column-gap.**
  - Propiedades para especificar el espaciado entre ítems.
  - Son alternativas al uso de padding o margin en los elementos hijos.
  - Solo una de las dos propiedades tendrá efecto:
    - row-gap si flex-direction está establecida en *column*.
    - column-gap si flex-direction está establecida en *row*.
  - Se pueden usar ambas si la propiedad flex-wrap está establecida en *wrap*.
  - Sintaxis:
    - {row-gap: valor unidad de medida;}
    - {column-gap: valor unidad de medida;}
  - Valores:
    - Se especifica mediadas relativas o absolutas.
  - Ejemplo:
    - {row-gap: 10px; column-gap: 15px;}
  - **Notación abreviada.**
    - Los valores de row-gap y column-gap se pueden incluir en una sola declaración de estilo usando la propiedad gap.
    - Sintaxis:
      - {gap: valor espaciado horizontal valor espaciado vertical}
      - {gap: valor único para espaciado horizontal y vertical}
      -
    - Ejemplo:
      - {gap: 10px 15px;}
      - {gap: 10px;} // Mismo valor para espaciado horizontal y vertical.

### Propiedades de los ítems o hijos incluidos en el contenedor.

- **align-self.**
  - Alineación respecto al eje secundario o transversal de cada ítem o hijo de forma individual.
  - Esta propiedad debe aplicarse en cada ítem o hijo específico.
  - Reemplaza el valor de align-items.
  - Sintaxis:
    - {align-self: valores;}
  - Valores:
    - **Auto.**
      - Mantiene la alineación actual del elemento.
    - **flex-start.**
      - Alinea al elemento hacia el borde inicial del eje principal.
    - **flex-end.**
      - Alinea al elemento hacia el borde final del eje principal.
    - **Center.**
      - Centra el elemento en el eje principal.
    - **Baseline.**

- Alinea al elemento con la línea base de su contenido.
  - **Stretch.**
    - El elemento se estira para ocupar el espacio completo del contenedor en el eje principal.
    - Si es horizontal, estira su ancho; si es vertical, estira su largo.
- **flex-grow.**
  - Factor de crecimiento de los ítems.
  - Permite indicar el tamaño en que crecerá un ítem en relación con sus demás hermanos, siempre y cuando haya espacio disponible en el contenedor para crecer.
  - El aumento se hace a lo ancho o alto, según el eje principal definido.
  - Si no hay valor definido este es 0 (cero), por lo que el elemento no puede aumentar. Si su valor es 2, por ejemplo, aumentara el doble que otros que no lo tengan.
  - No funciona si los elementos tienen una anchura fija establecida con *width*.
  - Sintaxis:
    - {flex-grow: valor;}
  - Valores:
    - Se especifica con un valor entero.
    - El valor por defecto es 0, que indica que el elemento no puede aumentar.
  - Ejemplo:
    - {flex-grow: 1;}
- **flex-shrink**
  - Factor de decrecimiento de los ítems.
  - Permite indicar el tamaño en que decrecerá un ítem en relación con sus demás hermanos.
  - Opuesto a flex-grow, los elementos que tengan un valor numérico más grande serán más pequeños, mientras que los que tengan un valor numérico más pequeño serán más grandes.
  - La disminución se hace a lo ancho o alto, según el eje principal definido.
  - Funciona si los elementos tienen una anchura fija establecida con *width*.
  - Sintaxis:
    - {flex-shrink: valor;}
  - Valores:
    - Se especifica con un valor entero.
    - El valor por defecto es 1, que indica que el elemento no puede reducirse.
  - Ejemplo:
    - {flex-shrink: 2;}
- **flex-basis.**
  - Permite especificar el ancho o el alto del elemento según el eje principal definido.
  - Especifica que tamaño base tendrán los ítems antes de aplicarle la distribución de espacio.
  - Su valor por defecto es *auto*.
  - Sintaxis:
    - {flex-basis: valor unidad de medida;}
  - Valores:
    - Se especifica mediadas relativas o absolutas.
    - El valor por defecto es *auto*.
  - Ejemplo:
    - {flex-basis: 140px;}
  - **Notación abreviada.**
    - Los valores de flex-basis, flex-grow y flex-shrink se pueden incluir en una sola declaración de estilo usando la propiedad flex.
    - Sintaxis:

- {flex: valor de crecimiento valor de decrecimiento tamaño base}
  - Ejemplo:
    - {flex: 1 2 140px;}
- **order.**
  - Establece el orden de presentación de los ítems según un valor numérico entero.
  - Los ítems se van ubicando según su número, colocándose antes los ítems con valor menor o negativo y después, los que tienen valores más altos.
  - Por defecto, todos los ítems flex tienen como valor *order* implícito, 0 (cero), aunque no se especifique.
  - Si no se especifica ningún valor, el orden es el de creación del elemento.
    - Sintaxis:
      - {order: valor entero;}
    - Ejemplo:
      - {order:2}

## GRID LAYOUT.

- **Concepto.**
  - Es un sistema de maquetación en CSS basado en cuadrículas o rejillas de 2 dimensiones, esto lo diferencia de Flexbox, que se aplica sólo a una dimensión.
  - Es decir, se trabaja con filas y columnas para crear una cuadrícula, en la cual se colocan y distribuyen los distintos elementos.
- **Componentes.**
  - **Contenedor.**
    - El elemento padre contenedor que definirá la cuadrícula o rejilla.
  - **Ítem.**
    - Cada uno de los elementos hijos que contiene la rejilla.
  - **Celda (grid cell).**
    - Cada una de las intersecciones fila - columna de la cuadrícula.
  - **Área (grid area).**
    - Conjunto de celdas de la cuadrícula.
  - **Banda (grid track).**
    - Conjunto de celdas distribuidas horizontal o verticalmente en la cuadrícula.
  - **Línea (grid line).**
    - Separador horizontal o vertical de las celdas de la cuadrícula.

## PROPIEDADES PARA CONTENEDORES GRID.

- **Creación de un contenedor grid.**
  - Un contenedor grid se crea con la propiedad display.
  - Sintaxis:
    - display:valores;
    - valores:
      - **grid.**
        - Crea un contenedor grid con los elementos en bloque.
      - **inline-grid.**
        - Crea un contenedor grid con los elementos en línea.
  - Ejemplos:
    - .rejilla {display: inline-grid;}
    - #cuadrícula {display: grid;}

- **Definir filas y columnas.**

- **Columnas.**

- Para especificar la cantidad, así como el tamaño, de columnas que contendrá la cuadrícula se usa la propiedad `grid-template-columns`.
- La cantidad de columnas queda especificada por el número de medidas que se especifiquen, mientras que las dimensiones de cada una, por las propias medidas.
- Sintaxis:
  - `grid-template-columns: Medida Columna1 Medida Columna 2 ... Medida Columna N.`
- Ejemplos:
  - `grid-template-columns: 100px 300px` (2 columnas, una de 100 píxeles y la otra de 300).
  - `grid-template-columns: 30% 30% 30%` (3 columnas, cada una tiene una anchura del 30% respecto del contenedor padre).
  - `grid-template-columns: auto auto auto auto` (4 columnas, cada una ocupa el mismo espacio calculado automáticamente según el espacio disponible en el contenedor padre).
  - `grid-template-columns: 1fr 2fr 1fr` (3 columnas y unidad fracción restante, la segunda columna ocupa el doble que las otras 2).

- **Filas.**

- Para especificar la cantidad, así como el tamaño, de filas que contendrá la cuadrícula se usa la propiedad `grid-template-rows`.
- La cantidad de columnas queda especificada por el número de medidas que se especifiquen, mientras que las dimensiones de cada una, por las propias medidas.
- Sintaxis:
  - `grid-template-rows: Medida Columna1 Medida Columna 2 ... Medida Columna N.`
- Ejemplos:
  - `grid-template-rows: 100px 400px 100px` (3 filas, la primera y la última tienen una altura 100 píxeles y la segunda, 400).
  - `grid-template-rows: 45% 45%` (2 filas, cada una ocupa el 45% del contenedor padre).
  - `grid-template-rows: auto auto auto` (3 filas, cada una ocupa el mismo espacio calculado automáticamente según el espacio disponible en el contenedor padre).
  - `grid-template-rows: 1fr 2fr` (2 filas y unidad fracción restante, la segunda fila ocupa el doble que la primera).
- Ejemplo combinando columnas y filas:
  - `grid-template-columns: 1fr 2fr;`
  - `grid-template-rows: 200px 100px;`
    - La cuadrícula tendrá 4 celdas (2 columnas x 2 filas), siendo la segunda columna del doble de ancho que la primera y respecto a las filas, la primera será el doble de alta que la segunda.

- **Unidad de medida fr o fracción restante.**

- Unidad de medida nueva para los contenedores grid.
  - Se utilizan valores enteros como múltiplos para establecer la altura o anchura de las filas y columnas.
  - La suma de los valores establece la medida total dentro del contenedor grid, siendo los valores fracciones de ésta.

- En un contenedor inline-grid los repartos son iguales, pero las dimensiones son las relativas al contenido de las celdas.
- Sintaxis:
  - n<sup>º</sup>fr.
- Ejemplo:
  - grid-template-columns: 1fr 3fr 1fr (el tamaño total será 3, siendo el ancho de la segunda columna el triple que el ancho de la primera y última columna).
- **Filas y columnas repetidas.**
  - Para evitar incluir continuamente las mismas medidas para las diversas columnas y filas se puede usar la función repeat().
  - Sintaxis:
    - repeat(número de repeticiones, tamaño)
  - Ejemplos:
    - grid-template-columns: repeat(3, auto) equivale a:
      - grid-template-columns: auto auto auto
    - grid-template-columns: 100px repeat(2, 50px) 200px equivale a:
      - grid-template-columns: 100px 50px 50px 200px
    - grid-template-rows: repeat(3, 2fr 3fr) equivale a:
      - grid-template-rows: 2fr 3fr 2fr 3fr 2fr 3fr
- **Atajo grid-template.**
  - Se puede usar la propiedad grid-template para definir las filas y columnas en una sola instrucción.
    - Sintaxis:
      - grid-template: valores grid-template-rows / valores grid-template-columns;
    - Ejemplo:
      - grid-template: 150px 80px / auto auto auto; (La cuadrícula tendrá 3 columnas con la misma anchura calculada automáticamente según las dimensiones del contenedor padre, y 2 filas, la primera de 150 píxeles de alto y la segunda de 80).
- **Huecos.**
  - Las celdas de la cuadrícula están pegadas unas a otras, por lo que no hay separación entre ellas.
  - Para separarlas se puede aplicar un margen externo o usar la propiedad **gap**.
  - **column-gap.**
    - Permite especificar el tamaño del hueco o separación entre columnas.
    - Sintaxis:
      - column-gap: valor píxeles.
    - Ejemplo:
      - column-gap: 50px;
  - **row-gap.**
    - Permite especificar el tamaño del hueco o separación entre filas.
    - Sintaxis:
      - row-gap: valor píxeles.
    - Ejemplo:
      - row-gap: 20px;
  - **gap.**
    - Se puede utilizar gap como atajo para especificar a la vez la separación entre filas y columnas.
    - Sintaxis:

- gap: valor píxeles filas valor píxeles columnas.
- Ejemplos:
  - gap: 20px 50px equivale a:
    - row-gap: 20px;
    - column-gap: 50px;
  - gap: 100px equivale a:
    - row-gap: 100px;
    - column-gap: 100px;

## ALINEACIONES.

- **justify-content.**
  - Alinea las celdas dentro del contenedor padre según el eje horizontal (principal por defecto).
  - Debe existir espacio dentro del contenedor padre para poder visualizar la alineación.
  - Sintaxis:
    - justify-content: valores;
      - **valores:**
        - **start.**
          - Coloca las celdas al principio el contenedor padre.
          - Opción por defecto.
        - **end.**
          - Coloca las celdas al final el contenedor.
        - **center.**
          - Coloca las celdas centradas en el contenedor padre.
        - **space-between.**
          - Las celdas se distribuyen ocupando todo el espacio disponible en el contenedor.
          - Las celdas no incluyen espacio al principio o al final, pero si entre ellos.
        - **space-around.**
          - Igual que space-between, pero incluyendo un espacio delante y detrás de la primera y última celda.
        - **space-evenly.**
          - Igual que space-around, pero solapando los espacios, de modo que sean todos del mismo tamaño.
  - Ejemplos:
    - justify-content: center;
    - justify-content: space-around;
- **align-content.**
  - Alinea las celdas dentro del contenedor padre según el eje vertical (secundario por defecto).
  - Debe existir espacio dentro del contenedor padre para poder visualizar la alineación.
  - Sintaxis:
    - justify-content: valores;
      - **valores:**
        - **start.**
          - Coloca las celdas al principio el contenedor padre.
        - **end.**
          - Coloca las celdas al final el contenedor.

- **center.**
    - Coloca las celdas centradas en el contenedor padre.
  - **space-between.**
    - Las celdas se distribuyen ocupando todo el espacio disponible en el contenedor.
    - Las celdas no incluyen espacio al principio o al final, pero si entre ellos.
  - **space-around.**
    - Igual que space-between, pero incluyendo un espacio delante y detrás de la primera y última celda.
  - **space-evenly.**
    - Igual que space-around, pero solapando los espacios, de modo que sean todos del mismo tamaño.
- Ejemplos:
  - `justify-content: start;`
  - `justify-content: space-between;`
- **place-content.**
  - Atajo para las propiedades `justify-content` y `align-content`.
  - Sintaxis:
    - `place-content: valor para align-content valor para justify-content;`
  - Ejemplos:
    - `place-content: start space-between;` equivale a:
      - `align-content: start;`
      - `justify-content: space-between;`
    - `place-content: center;` equivale a:
      - `align-content: center;`
      - `justify-content: center;`
- **justify-items.**
  - Alinea el contenido de las celdas según el eje horizontal (principal por defecto).
  - Debe existir espacio dentro de las celdas para poder visualizar la alineación.
  - Sintaxis:
    - `justify-items: valores;`
      - **valores:**
        - **start.**
          - Coloca las celdas al principio el contenedor padre.
        - **end.**
          - Coloca las celdas al final el contenedor.
        - **center.**
          - Coloca las celdas centradas en el contenedor padre.
        - **stretch.**
          - Estira el contenido de la celda para que ocupe todo su espacio en horizontal.
  - Ejemplos:
    - `justify-items: start;`
    - `justify-items: stretch;`
  - **align-items.**
    - Alinea el contenido de las celdas según el eje vertical (secundario por defecto).
    - Debe existir espacio dentro de las celdas para poder visualizar la alineación.
    - Sintaxis:
      - `align-items: valores;`



- **valores:**
    - **start.**
      - Coloca las celdas al principio el contenedor padre.
    - **end.**
      - Coloca las celdas al final el contenedor.
    - **center.**
      - Coloca las celdas centradas en el contenedor padre.
    - **stretch.**
      - Estira el contenido de la celda para que ocupe todo su espacio en vertical.
  - Ejemplos:
    - align-items: end;
    - align-items: center;
- **place-items.**
  - Atajo para las propiedades justify-items y align-items.
  - Sintaxis:
    - place-items: valor para align-items valor para justify-items;
  - Ejemplos:
    - place-items: start end; equivale a:
      - align-items: start;
      - justify-items: end;
    - place-items: center; equivale a:
      - align-items: center;
      - justify-items: center;
- **justify-self.**
  - Propiedad que se aplica a ítems hijos, no al contenedor grid.
  - Alinea el contenido de una celda concreta en horizontal sobrescribiendo la alineación especificada con justify-items.
  - Tiene los mismos valores que justify-items: start, center, end y stretch.
- **align-self.**
  - Propiedad que se aplica a ítems hijos, no al contenedor grid
  - Alinea el contenido de una celda concreta en vertical sobrescribiendo la alineación especificada con align-items.
  - Tiene los mismos valores que align-items: start, center, end y stretch.
- **place-self.**
  - Propiedad que se aplica a ítems hijos, no al contenedor grid
  - Atajo para las propiedades justify-self y align-self.
  - Sintaxis:
    - place-self: valor para align-self valor para justify-self;
  - Ejemplos:
    - place-self: start end; equivale a:
      - align-self: start;
      - justify-self: end;
    - place-self: center; equivale a:
      - align-self: center;
      - justify-self: center;
- **order.**
  - Permite modificar le orden de los elementos dentro del contenedor grid.
  - El valor de orden por defecto de un elemento es 0.

- Un elemento con valor negativo o mayor que cero, se recolocará antes o después según dicho valor.
- Sintaxis:
  - order: 0 o un número entero positivo o negativo;
- Ejemplos:
  - order: -1;
  - order:2;

## ÁREAS.

- Zona de una cuadrícula compuesta por una o más celdas.
- A las áreas se les asigna un nombre para identificarlas.
- Las celdas se asignan a las áreas indicando el nombre del área en la cual se incluirán.
- **Propiedades:**
  - **grid-template-areas.**
    - Permite crear áreas asignando nombres a las filas.
    - Se especifican tantos nombres como celdas hay en la fila.
    - Todos los nombres de una misma fila se encierran entre comillas.
    - Si se utiliza el mismo nombre en una fila, el área ocupará todas las celdas.
    - Sintaxis:
      - grid-template-areas:
        - “nombre-área-1-fila-1 nombre-área-2-fila-2.....”
        - “nombre-área-1-fila-1 nombre-área-2-fila-2.....”
        - “nombre-área-1-fila-N nombre-área-2-fila-N”;
    - Ejemplo:
      - grid-template-areas:
        - “uno uno uno”
        - “dos dos tres”
        - “cuatro cuatro cuatro”;
      - Se han creado 4 áreas. Las áreas uno y cuatro ocupan por completo sus filas, mientras que las áreas dos y tres, ocupan 2 y una celda respectivamente.
      - Si se pone un punto (.) como nombre de área, se colocará una celda vacía en esa posición.
  - **grid-area.**
    - Permite asignar las celdas, y por tanto su contenido, a un área concreta.
    - Sintaxis:
      - grid-area: valores;
      - valores:
        - **nombre.**
          - Hace referencia al nombre del área definida con la propiedad **grid-template-areas** donde se va a ubicar la celda.
        - **auto.**
          - Coloca la celda en la siguiente área vacía disponible.
    - Ejemplo:
      - #division1: uno;
        - El contenido de la #division1 se ubicará en todas las celdas que conforman el área nombrada como uno.

## ÁREAS ANIDADAS.

- Subcuadrículas, subrejillas o subgrid.
- Una cuadrícula o rejilla puede estar incluida o anidada dentro de una o varias celdas de otra cuadrícula o rejilla.
- La disposición para esta nueva rejilla es también grid.
- **Propiedades.**
  - Se utilizan igualmente las propiedades grid-template-columns y/o grid-template-rows para definir el número de columnas y filas con el valor *subgrid*.
  - **subgrid.**
    - Permite que la subcuadrícula o cuadrícula anidada, adopte los mismos valores para filas y/o columnas de la cuadrícula donde está incluida.
    - Sintaxis:
      - grid-template-columns: subgrid;
      - grid-template-rows: subgrid;
    - Ejemplo:
      - #principal {display:grid; grid-template-columns: 20% 20%;}
      - #secundario {display:grid; grid-template-columns: subgrid;}
        - #secundario tendrá las mismas dimensiones y columnas que el contenedor #principal que lo incluye.

## CELDA IRREGULARES.

- Una cuadrícula puede tener una forma irregular en lugar de cuadrada o rectangular.
- Este aspecto lo adopta cuando las celdas expanden su tamaño para ocupar varios espacios y hacen que la cuadrícula adopte dicho aspecto irregular.
- La expansión de una celda puede ser horizontal, vertical o en ambas direcciones, ocupando el número de posiciones en la cuadrícula que se necesite.
- Las celdas que no se expanden, pueden verse desplazadas a otras posiciones para dejar paso a las irregulares.
- Las propiedades CSS que se utilizan, se aplican a los ítems hijos, no al contenedor padre que se convierte en grid.
- **Propiedades.**
  - **grid-column-start.**
    - Indica a partir de que columna se expande una celda en horizontal.
    - Sintaxis:
      - **grid-column-start: valores;**
      - **Valores:**
        - **auto.**
          - La celda no se expande, se queda igual.
          - Valor por defecto.
        - **número de línea.**
          - Número entero que indica a partir de que línea vertical empieza la celda a expandirse.
    - Ejemplo:
      - grid-column-start: 2;
        - Inicio en la línea 2, por lo que la celda que se expandirá será la segunda.
  - **grid-column-end.**
    - Indica en que columna terminará el ítem de la cuadrícula, o hasta qué punto se expande una celda en horizontal.
    - Sintaxis:

- **grid-column-end: valores;**
- **Valores:**
  - **auto.**
    - La celda no se expande, se queda igual.
    - Valor por defecto.
  - **número de línea.**
    - Número entero que indica hasta que línea vertical una celda o ítem debe expandirse o alargarse.
  - **span número de líneas.**
    - Indica el número líneas verticales que una celda debe expandirse o alargarse.
- Ejemplos:
  - grid-column-start: 1 y grid-column-end: 4;
    - La celda o el ítem se expande desde la línea 1 hasta la línea 4, es decir, ocupa 3 celdas en la cuadrícula.
  - grid-column-start: 1 y grid-column-end: span 3;
    - Equivalente a la anterior. La celda o el ítem se expande desde la línea 1 y se expande 3 líneas más, es decir, hasta la cuarta línea, por lo que ocupa 3 celdas en la cuadrícula.
- **grid-row-start.**
  - Indica a partir de que fila se expande una celda en vertical.
  - Sintaxis:
    - **grid-row-start: valores;**
    - **Valores:**
      - **auto.**
        - La celda no se expande, se queda igual.
        - Valor por defecto.
      - **número de línea.**
        - Número entero que indica a partir de que línea horizontal empieza la celda a expandirse.
  - Ejemplo:
    - grid-row-start: 1;
      - Inicio del alargamiento de la celda desde la línea 1, por lo que será la primera celda la que se expandirá.
- **grid-row-end.**
  - Indica en que fila terminará el ítem de la cuadrícula, o hasta qué punto se expande una celda en vertical.
  - Sintaxis:
    - **grid-row-end: valores;**
    - **Valores:**
      - **auto.**
        - La celda no se expande, se queda igual.
        - Valor por defecto.
      - **número de línea.**
        - Número entero que indica hasta que línea horizontal una celda o ítem debe expandirse o alargarse.
      - **span número de líneas.**
        - Indica el número líneas horizontales que una celda debe expandirse o alargarse.

- Ejemplos:
  - `grid-row-start: 1 y grid-row-end: 4;`
    - La celda o el ítem se expande desde la línea 1 hasta la línea 4, es decir, ocupa 3 celdas en la cuadrícula.
  - `grid-row-start: 1 y grid-row-end: span 3;`
    - Equivalente a la anterior. La celda o el ítem se expande desde la línea 1 y se expande 3 líneas más, es decir, hasta la cuarta línea, por lo que ocupa 3 celdas en la cuadrícula.
- **grid-column.**
  - Atajo para las propiedades **grid-column-start y grid-column-end.**
  - Si sólo se pone un valor equivale a ambos, es decir, la celda empieza y termina en la misma posición.
  - Sintaxis:
    - **grid-column: valor de inicio / valor final;**
  - Ejemplos:
    - `grid-column: 1 / 3;`
      - El ítem se expande desde la línea 1 a la 3 en horizontal, es decir, ocupa 2 celdas.
    - `grid-column: 1 / span 2;`
      - Equivalente al anterior.
    - `grid-column: 3`
      - La celda inicia su expansión en la línea 3 vertical y la termina también en la línea 3, por lo que se queda como esta o se mueve a otra posición desplazando a las celdas que tenga delante o detrás.
- **grid-row.**
  - Atajo para las propiedades **grid-row-start y grid-row-end.**
  - Si sólo se pone un valor equivale a ambos, es decir, la celda empieza y termina en la misma posición.
  - Sintaxis:
    - **grid-row: valor de inicio / valor final;**
  - Ejemplos:
    - `grid-row: 1 / 3;`
      - El ítem se expande desde la línea 1 a la 3 en vertical, es decir, ocupa 2 celdas.
    - `grid-row: 1 / span 2;`
      - Equivalente al anterior.
    - `grid-row: 3`
      - La celda inicia su expansión en la línea 3 horizontal y la termina también en la línea 3, por lo que se queda como esta o se mueve a otra posición desplazando a las celdas que tenga delante o detrás.

## CONSULTAS DE MEDIOS - MEDIA QUERIES.

### Concepto.

- Permiten a un diseño acoplarse con precisión a los distintos tipos de medios de difusión.
- Para ello, en las consultas de medios se usan determinados criterios o condiciones que establecen que estilo se aplicará según sean las características del medio de difusión.

### Diseño Web Adaptable o Responsivo - Responsive Web Design (RWD).

- Creación y diseño de sitios web que se adaptan automáticamente en función del soporte (ordenador, tableta, smartphone, etc.).
- RWD se basa en el módulo Media Queries.

### **Tamaños más comunes de dispositivos de salida.**

- 320px.
- 480px.
- 600px.
- 640px.
- 768px.
- 800px.
- 900px.
- 1024px.
- 1280px.
- 1920px.

### **CRITERIOS.**

- **width.**
  - Anchura de visualización.
  - Se usa para comprobar la anchura de la zona de visualización del navegador (viewport).
  - Admite variantes con prefijos min y max.
  - Sintaxis:
    - (width: número unidad de medida)
    - (min-width: número unidad de medida)
    - (max-width: número unidad de medida)
  - Ejemplo:
    - (width: 900px)
    - (min-width: 500px)
- **height.**
  - Altura de visualización.
  - Se usa para comprobar la altura de la zona de visualización del navegador (viewport).
  - Sintaxis:
    - (height: número unidad de medida)
    - (min- height: número unidad de medida)
    - (max- height: número unidad de medida)
  - Ejemplo:
    - (height: 680px)
    - (max-height: 1000px)
- **device-width.**
  - Anchura física de visualización.
  - Se usa para comprobar la anchura física de la pantalla de difusión.
  - Sintaxis:
    - (device-width: número unidad de medida)
  - Ejemplo:
    - (device-width: 1340px)
- **device-height.**
  - Altura física de visualización.
  - Se usa para comprobar la altura física de la pantalla de difusión.
  - Sintaxis:

- (device-height: número unidad de medida)
  - Ejemplo:
    - (device-height: 720px)
- **orientation.**
  - Útil para conocer a orientación de la pantalla de difusión.
  - Sintaxis:
    - (orientation: valores)
  - Valores:
    - **portrait.**
      - Orientación vertical.
    - **landscape.**
      - Orientación horizontal.
  - Ejemplo:
    - (orientation: portrait)
- **aspect-ratio.**
  - Relación de aspecto.
  - Permite comprobar la relación ente el ancho y el alto de la pantalla de difusión.
  - Sintaxis:
    - (aspect-ratio: valores)
  - Valores:
    - 4/3, 16/9, 185/100 o 91/50, 239/100.
  - Ejemplo:
    - (aspect-ratio: 16/9)
- **device-aspect-ratio.**
  - Relación de aspecto física.
  - Permite comprobar la relación ente el ancho y el alto del dispositivo de salida.
  - Sintaxis:
    - (device-aspect-ratio: valores)
  - Valores:
    - 4/3, 16/9, 185/100 o 91/50, 239/100.
  - Ejemplo:
    - (aspect-ratio: 16/9)
- **color.**
  - Permite comprobar si la pantalla usa color o no.
  - Admite variantes con prefijos min y max.
  - Sintaxis:
    - (color: valor)
    - (min-color: valor)
    - (max-color: valor)
  - Valor:
    - Número entero que representa el número de bits por componente de color (rojo, verde, azul) del dispositivo de salida.
    - Valor = 0, indica que el dispositivo no es en color.
  - Ejemplo:
    - (color)
    - (min-color: 8)
- **color-index.**
  - Permite comprobar si la pantalla usa un número determinado de colores.
  - Admite variantes con prefijos min y max.
  - Sintaxis:

- (color-index: valor)
  - (max-color-index: valor)
  - (min-color-index: valor)
- Valor:
  - Número entero que representa el número de colores en la tabla de colores del dispositivo de salida.
- Ejemplo:
  - (color-index:256)
  - (min-color-index: 256)
- **monochrome.**
  - Número de niveles de gris para dispositivos monocromáticos.
  - Admite variantes con prefijos min y max.
  - Sintaxis:
    - (monochrome: valor)
    - (max- monochrome: valor)
    - (min- monochrome: valor)
  - Valor:
    - Indica el número de bits por pixel en un dispositivo monocromático (escala de grises)
    - Si el valor es 0, el dispositivo no es monocromático.
  - Ejemplo:
    - (monochrome)
    - (min-monochrome: 8)
- **resolution**
  - Resolución de la pantalla de salida.
  - Admite variantes con prefijos min y max.
  - Sintaxis:
    - (resolution: valor)
    - (max-resolution: valor)
    - (min-resolution: valor)
  - Valor:
    - Se expresa en dpi o dpcm, puntos por pulgada o centímetro respectivamente.
  - Ejemplo:
    - (min-resolution: 300dpi)

## TIPOS DE MEDIOS DE COMUNICACIÓN.

- Describen la categoría general de un dispositivo.
- El tipo de soporte es opcional excepto cuando se utilizan los operadores lógicos **not** u **only**.
- Tipos:
  - **all.**
    - Apto para todos los dispositivos.
    - Valor por defecto.
  - **print.**
    - Destinado a material impreso y a visualizar documentos en pantalla en modo de vista previa de impresión.
  - **screen.**
    - Destinado a pantallas.
  - **speech.**
    - Destinado a los sintetizadores de voz.
- Tipos de medios de comunicación obsoletos.



- tty, tv, projection, handheld, braille, embossed y aural.

## Regla @media.

- Reglas de CSS que permiten aplicar estilos en los dispositivos que cumplan los criterios especificados como condición.
- **Reglas con condiciones simples.**

- Sintaxis:

- @media (condición) {Reglas CSS}
  - Si se cumple la condición se aplican los estilos.
- @media not (condición) {Reglas CSS}
  - Si se no cumple la condición, no se aplican los estilos o se aplican otros diferentes.

- Ejemplos:

```
@media (max-width: 1200px)
{
    body{background:blue;}
} // Si la pantalla tiene 1200 o menos píxeles de ancho el fondo aparece en azul.
```

```
@media not (max-width = "1200px")
{
    body{background:yellow;}
} // Si la pantalla tiene más de 1200 píxeles de ancho, el fondo aparece en amarillo.
```

- **Reglas con condiciones compuestas.**

- Para crear una condición compuesta se usan operadores lógicos que permiten encadenar varias condiciones simples.

- **Operadores lógicos:**

- **and.**
  - Y lógico.
  - Todos los criterios deben cumplirse.
- **or.**
  - O lógico.
  - Al menos un criterio debe cumplirse.
- **not.**
  - No lógico.
  - El criterio no debe cumplirse.
- **only.**
  - La consulta debe aplicarse únicamente sobre los criterios indicados.

- Sintaxis:

- @media [tipo de medios] operador lógico (condición 1) operador lógico (operador 2) operador lógico (operador N) ... {Reglas CSS}

- Ejemplos:

- @media (min-width: 700px) and (orientation:landscape) { #div1 {background:green}} // Si el dispositivo tiene como mínimo una anchura de pantalla de 700 píxeles y está orientado en horizontal, el fondo de la división #div1 aparecerá en color verde.
- @media screen and (min-width: 1024px) and (max-width: 1640px) {#div1 {background:green}} // El fondo verde se aplicará a la división #div1 solo en pantallas, que además tengan una resolución comprendida entre los 1024 y los 1640 píxeles.

- `@media not (all and (monochrome)) {#div1 {background:green}}` // El color verde no se aplica a medios de todo tipo que sean monocromo, o no sean en color.
- `@media only (screen and (color)) {#div1 {background:green}}` // Solo pantallas en color mostrarán la división #div1 en color verde.
- `@media all and (color) {#div1 {background:green}}` // Todos los dispositivos que soportan colores mostrarán la división #div1 en color verde.

### Inclusión Media Queries en página web.

- **Dentro de la etiqueta `<style>` en la misma página web. (Etiqueta Style).**

- Se especifican, con la etiqueta `<style>...</style>` dentro de `<head>`, las características y dimensiones que deben tener las distintas pantallas o medios de difusión, así como, las reglas CSS específicas a aplicar a cada una de ellas.
- Sintaxis:

```
<style>
    @media condición o criterio 1{
        Selectores {Propiedades CSS;}
    }
    @media condición o criterio 2 {
        Selectores {Propiedades CSS;}
    }
    @media condición o criterio N {
        Selectores {Propiedades CSS;}
    }
</style>
```

- Ejemplo:

```
<style>
    @media (max-width:480px) {

        #div1{
            width: 50%;
        }
    }
    @media screen and(max-width:900px) {

        #div1{
            Background:#8a72e1;
            width: 100%;
        }
    }
</style>
```

- **Enlace a hojas de estilo que incluyen las condiciones y los estilos para las diferentes pantallas. (Etiqueta Link).**

- Se especifica, con la etiqueta `<link>` dentro de `<head>`, el nombre y ubicación del archivo css con las características y dimensiones (condiciones), que deben tener las distintas pantallas o medios de difusión, así como, las reglas CSS específicas a aplicar a cada una de ellas.
- Las rutas de acceso pueden ser locales o externas, relativas o absolutas.
- Un mismo archivo .css puede ser usado por distintos archivos .html.

- Una misma página web puede usar distintos archivos .css.
- Sintaxis:
  - `<link rel = "hoja de estilo" type ="tipo MIME/formato de archivo" href = "ruta de acceso/archivo.extensión">`
- Ejemplos:
  - `<link rel = "stylesheet" type ="text/css" href ="css/estilosPantallas.css">`
  - `<link rel = "stylesheet" type ="text/css" href ="css/pantallasMóviles.css">`
- **Nota.**
  - Hay que tener en cuenta que, si se usan estilos en la cabecera (etiqueta `<style>`), o estilos en línea (atributo *style*), en la página web se sobrescribirán los de las hojas de estilos usados en las medias queries enlazadas con `<link>`.
- **Enlace a diferentes hojas de estilo que sólo incluyen los estilos para las diferentes pantallas. (Etiqueta Link Media).**
  - Se especifica, con la etiqueta `<link>` dentro de `<head>`, el nombre y ubicación de los archivos css que contienen sólo las reglas CSS específicas a aplicar a cada pantalla o medio de difusión.
  - Cada archivo css contiene las reglas para unas características de pantalla o medio de difusión determinado.
  - Puede haber archivos css con características comunes a todas o, a alguna, de las pantallas o medios de difusión.
  - Las condiciones, o características y dimensiones que deben tener las distintas pantallas o medios de difusión, se incluyen dentro de la etiqueta `<link>` usando el atributo *media*.
  - Se usan tantas etiquetas link como condiciones y archivos enlazados existan.
  - **Nota.**
    - Hay que tener en cuenta que, si se usan estilos en la cabecera (etiqueta `<style>`), o estilos en línea (atributo *style*), en la página web se sobrescribirán los de las hojas de estilos usados en las medias queries enlazadas con `<link>`.
  - Sintaxis:
    - Atributo media:
      - `media =" condiciones, criterios o características de medios"`
    - Enlace con <link>:
      - `<link rel = "hoja de estilo" type ="tipo MIME/formato de archivo" media =" condiciones, criterios o características de medios" href = "ruta de acceso/archivo.extensión">`
  - Ejemplos:
    - `<link rel="stylesheet" type="text/css" media="all" href="estilos.css"> // Carga hoja de estilos para todos los medios de difusión.`
    - `<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css"> // Carga hojas de estilos con tipos de fuentes solo para pantallas.`
    - `<link rel="stylesheet" type="text/css" media="print" href="serif.css"> // Carga hojas de estilos con tipos de fuentes solo para contenidos a imprimir.`
    - `<link rel="stylesheet" type="text/css" media = only screen and (min-width:1200px)" href="css/pantallasGrandes.css"> // Carga hojas de estilos para pantallas de más de 1200 píxeles de anchura.`
    - `<link rel="stylesheet" type="text/css" media="only screen and (min-width:501px) and (max-width:800px)" href="css/pantallasMedianas.css"> // Carga hoja de estilos para pantallas tipo tableta con dimensiones entre 501 y 800 píxeles.`

## PANTALLAS.

- **Tamaño:**
  - Medido en pulgadas.

- 1 pulgada = 2,54 cm, expresándose el valor de la siguiente forma: N°" (Ejemplo 6").
- El valor indica la longitud de la diagonal de la pantalla.
- Hasta 5 pulgadas → Móviles y Smartphones.
- Entre 5 y 7 pulgadas → Phablet. (Smartphone + Tablet).
- Mayor 7 pulgadas → Tablet.
- **Relación de aspecto:**
  - Relación entre la altura y anchura de la pantalla.
  - Ejemplos con valores típicos:
    - 16:9 – Pantalla panorámica.
    - 4:3 – Pantalla de las televisiones antiguas.
- **Resolución:**
  - Número de píxeles que tiene la pantalla.
  - A mayor número de píxeles mayor resolución y, por tanto, mayor calidad de imagen, pero también más consumo.
  - Un píxel es un elemento de imagen (Picture Element).
  - **Dot Pitch:**
    - Distancia entre puntos o píxeles adyacentes medida en milímetros.
    - Se utiliza para poder dar una idea de la definición de una pantalla.
    - Un "dot pitch" más pequeño implica normalmente una imagen más nítida, ya que significa que la pantalla contiene un número más grande de puntos o píxeles en un área dada.
  - Se define mediante 2 valores:
    - Número píxeles a lo ancho x número de píxeles a lo alto.
    - Ejemplo: 1920 x 1080 = 2.072.000 píxeles.
    - A mayor número de píxeles, mayor calidad de imagen.
  - **Resoluciones típicas.**
    - Móviles:
      - 320 y 480 píxeles.
    - iPads y Tabletas:
      - 481 y 768 píxeles:
    - Pantallas pequeñas, portátiles:
      - 769 y 1024 píxeles:
    - Ordenadores de escritorio, pantallas grandes:
      - 1.025 y 1.200 píxeles.
    - Pantallas extra grandes, TV:
      - 1.201 y más píxeles:
- **Densidad de píxeles.**
  - Número de píxeles o puntos de una pantalla por unidad de medida.
  - Unidades más utilizadas:
    - Píxeles por pulgada (PPI).
    - Puntos por pulgada (DPI).
  - En los monitores de ordenador se suele utilizar hasta 96 DPI, mientras que, en los móviles y dispositivos con pantallas de alta densidad, 300 DPI o más.
  - Por tanto, para hacer un diseño web adaptable a cualquier pantalla no sólo hay que tener en cuenta su tamaño, sino también, su densidad de píxeles.
  - **Resolución y densidad de píxeles.**
    - **Resoluciones estándar:**
      - HD (High Definition): 1280px x 720px.
      - FHD (Full HD): 1920px x 1080px.

- QHD/2K (Quad HD) → 2560px x 1440px.
- UHD/4K (Ultra HD): 3840px x 2160px.
- **Resoluciones plus:**
  - Extra de longitud en la dimensión más larga del teléfono, ya que la tendencia actual de los móviles es no tener marcos, aprovechándose así más espacio para pantalla.
  - Resoluciones:
    - HD+: 1440px x 720px.
    - FHD+: 2160px x 1080px.
    - QHD+/2k+: 2880px x 1440px.
    - UHD+/4k+: 4320px x 2160px.

## VIEWPORT.

- Hace referencia al tamaño de la ventana o área visible de una pantalla.
- En general, es el área que está disponible en la pantalla del navegador o área útil donde se mostrará la página web.
- Para facilitar el diseño web teniendo en cuenta las distintas densidades de píxeles de las pantallas se incluye en la cabecera del código HTML la etiqueta meta *viewport*.
- Ésta aplica automáticamente un zoom que facilita la lectura en dispositivos con pantallas de alta densidad.
- Sintaxis:

<head>

<meta name="viewport" content="anchura y escala o zoom">

</head>

- Valor de name.
  - viewport como valor obligatorio.
- Valores de content:
  - **width.**
    - Anchura virtual (emulada) de la pantalla o anchura del viewport.
    - Valores:
      - **device-width:**
        - Medida que hace referencia a la anchura de la pantalla del dispositivo.
        - Valor que permite que el viewport sea igual a la anchura real de la pantalla del dispositivo. Así, no se emulará una pantalla mayor de su tamaño real y se verán los píxeles reales.
        - La anchura (width) y la altura (height) hacen referencia a las dimensiones fijas del viewport inicial. Sin embargo, no se suelen indicar, siendo más habitual ajustar sólo la anchura con el valor "device-width".
      - Uso de valores con unidades de medida exactas.
  - **height.**
    - Altura virtual de la pantalla o anchura del viewport.
  - **Initial-scale.**
    - Escala o zoom inicial del documento o página web.
    - Valores:
      - Entre 0.1 y 10.0.
      - Valor típico es 1.0 (sin zoom).
    - Ejemplos:

- 0.1 Indica un 10% de zoom.
  - 1.0 indica 100% de zoom, tamaño real o sin zoom.
  - 2.0 indica 200% de zoom o el doble de tamaño.
- **minimum-scale.**
  - Escala mínima que se puede aplicar al documento.
- **maximum-scale.**
  - Escala máxima que se puede aplicar al documento.
- **user-scalable.**
  - Indica si se permite, o no, hacer zoom al usuario.
  - Valores:
    - **yes.**
      - Zoom permitido al usuario.
      - Valor por defecto.
    - **no.**
      - Zoom no permitido al usuario
  - Para no limitar al usuario el uso del zoom y darle la posibilidad de aumentar o reducir cualquier parte de la pantalla, no hay que definir ni maximum-scale, ni usar user-scalable con valor “no”.
- Ejemplo:
  - `<meta name="viewport" content="width=device-width, initial-scale=1.0">`