**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.

**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;">

1. **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>

1. **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">

1. **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.

**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}

* **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 pesudoclase 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-oft-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}
* **Pseudoclases 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}º
    - \*[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://www.cleancss.com/css-beautify/)
      * [**https://csscompressor.com/**](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:
        + 1em \* 15pt = 15pt.
        + 0.5em \* 15pt = 7,5pt.
        + 4em \* 15pt = 60pt.
  + **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:
        + 1rem \* 15pt = 15pt.
        + 0.5rem \* 15pt = 7,5pt.
        + 4rem \* 15pt = 60pt.
  + **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.
      * 1ex = 0.5 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

**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 prefefinidos:
  + 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 identar 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%}

**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%}

**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 (...).
  + 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 usar 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:
  + {l**ist-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 muestras 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>

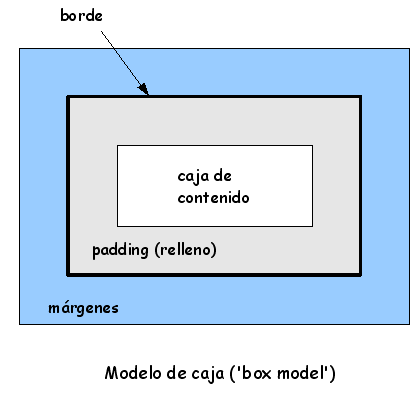
Texto de la leyenda

</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}

**MODELO DE CAJAS.**

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

**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.**
        + Bordes entrantes, incrustados en la página.
        + Inverso a outset.
      * **outset.**
        + Bordes salientes, resaltados en la página.
        + Inverso a inset.
      * **hidden.**
        + Sin borde.
        + Bordes 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**: 1ºvalor 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**: 1ºvalor 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 trasparencia 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.

**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.
      * Ente 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, quedado 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 ente visibility: hidden y display: none es que el primero solo oculta y 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.

**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-gapycolumn-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 finaldel eje principal.
    - **Center.**
      * Centrael 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}