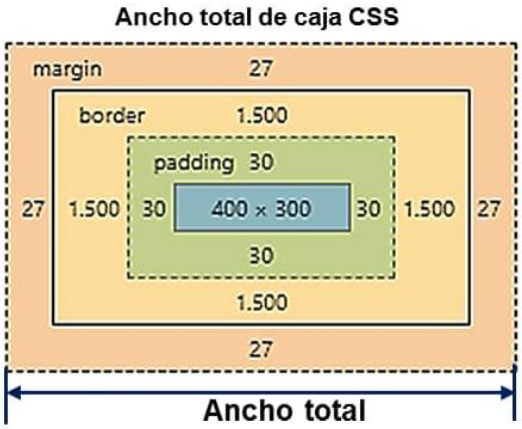
Modelo de caja

Los elementos de HTML están contenidos en cajas rectangulares. La etiqueta **body** es la caja principal dentro de la cual se van colocando las cajas de las etiquetas que se van creando, las cajas pueden estar unas dentro de otras, o unas al lado de otras, incluso se pueden superponer.

Podemos pensar en el modelo de caja como un borde rectangular (***border***) que rodea al contenido, hay un margen interior que define la propiedad ***padding*** y un margen exterior que define la propiedad **margin**.



Mas contenido acerca del modelo de cajas

https://desarrolladoresweb.org/css/modelo-de-cajas-css/

Los componentes del modelo de caja son los siguientes:

a) Contenido. En cada caso este contenido es de un tipo, puede ser un texto, una imagen, una lista, otra caja, ... El contenido depende de cada etiqueta, por ejemplo, la etiqueta **div** admite varios tipos de contenido (texto, imágenes, otras etiquetas, ...) mientras que una etiqueta **img** sólo admite una imagen como contenido.

b) Padding (margen interior). Espacio transparente entre el contenido y el borde. Puede ser de grosor cero. También podemos decir que el ***padding*** es el margen interior de la caja. En nuestro esquema el ***padding*** está representado por las dobles flechas azules.

c) Border (borde). Línea que rodea el ***padding***, y por lo tanto también al contenido. Puede ser de grosor cero y por lo tanto invisible. En nuestro esquema el borde está representado por la línea punteada roja.

d) Margin (margen exterior). Espacio transparente rectangular que rodea al borde y sirve de separación con otros elementos adyacentes. Puede ser de grosor cero. En nuestro esquema el **margin** está representado por las doble flechas verdes.

Y aunque no son estrictamente del modelo de caja, también intervienen:

e) Background-image (Imagen de fondo). Algunos elementos como los contendores div y los párrafos p pueden tener una imagen de fondo que se coloca por detrás del contenido, hasta el borde. En nuestro esquema no hay imagen de fondo. Más adelante veremos un esquema que sí la tiene.

f) Background-color (Color de fondo). Si se define un color de fondo, este se coloca por detrás de la imagen de fondo, hasta el borde. En nuestro esquema no hay color de fondo.

Valores por defecto

Es decir, el atributo **border** tiene inicialmente valor cero, es invisible. Y el rectángulo que rodea al contenido lo hemos dibujado para explicar el modelo, pero no existe como tal, no se puede dibujar.

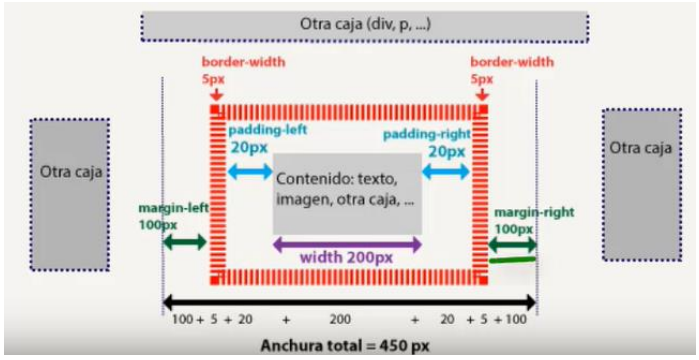
El padding suele no tener valor por defecto en la mayoría de etiquetas.

El padding es siempre transparente, y deja ver la imagen de fondo y el color de fondo de su propia caja.

El margin es siempre transparente, y deja ver la imagen de fondo y el color de fondo de la caja que lo contiene

Margin

El atributo margin define el espacio transparente entre el borde de la caja y los elementos adyacentes. Es el margen exterior de la caja. El valor cero indica que no existe margin. Un valor negativo hace que se "introduzca" en el elemento contiguo. El margin se compone de cuatro valores, uno para cada lado. Se pueden indicar de forma compacta o de forma explícita.



margin de forma explícita: margin superior con margin-top; margin derecho con margin-right; margin inferior con margin-bottom; margin izquierdo con margin-left.

- *margin de forma compacta:* en este caso hay cuatro variantes:

a) Con un sólo número. Se aplica el valor a las cuatro posiciones del margin. Por ejemplo, margin:15px, es lo mismo que margin-top: 15px; margin-right: 15px; margin-bottom: 15px; margin-left: 15px.

Con dos números: Por ejemplo, margin: 15px 25px, es lo mismo que margin-top: 15px; margin-right: 25px; margin-bottom: 15px; margin-left: 25px.

El valor del margin se puede indicar con una unidad absoluta (px, mm, ..) o con una unidad relativa (%, em, ...). Se puede dar el valor auto que indica que se cree un margen de forma automática

Para más información

https://www.aulaclic.es/html/secuencias/p10_5_margen.htm

Padding

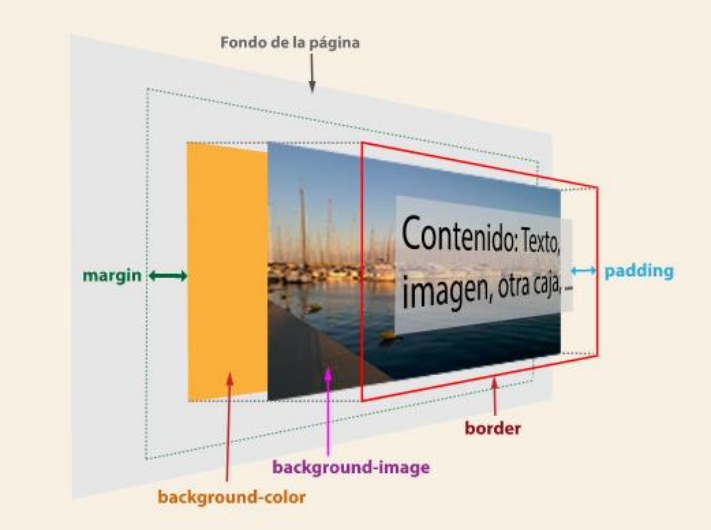
El atributo padding define el espacio transparente entre el contenido y el borde. Es el margen interior de la caja, el margen que queda dentro de la caja. Puede tener valores numéricos enteros positivos. El valor cero indica que no existe padding.

Con dos números. Se aplica el primer valor al padding superior e inferior; y el segundo valor al padding derecho e izquierdo.

Por ejemplo, padding: 15px 25px, es lo mismo que padding-top: 15px; padding-right: 25px; padding-bottom: 15px; padding-left: 25px.

Fondo

Dentro del modelo de cajas el fondo está formado por dos elementos opcionales, la imagen de fondo y el color de fondo. El contenido está delante de ambos. Por defecto, tanto la imagen como el color de fondo llegan hasta el border a continuación, tenemos el margin que separa esta caja de las adyacentes.



El contenido, la imagen de fondo y el color de fondo no pueden sobrepasar el border, que puede ser de color o transparente.

Como ya hemos visto entre el contenido y el border puede existir el padding. El padding es transparente.

Entre el border y la siguiente caja puede existir un margin que es siempre transparente.

Background-color

Este atributo opcional define el color de fondo, el valor debe ser un color escrito de una de las formas válidas que vimos anteriormente en el punto Colores

```
background-color: lightblue
```

Background-image

Este atributo opcional define la imagen de fondo, el valor debe ser una imagen, de la forma url(imagen).

```
background-image: url(graficos/cuadrado_mediano.png)
```

background-attachment

background-size

Bordes

Los bordes de un elemento de HTML se definen con tres atributos básicos: estilo, anchura y color.

border-style

El estilo del borde se refiere a la forma del borde y puede tomar estos valores: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset.

Ejemplos

- un valor: border-style: solid, se aplica el estilo a los cuatro bordes.
- dos valores: border-style: solid dotted
- tres valores: border-style: solid dotted double

border-width

- un valor: border-width: 15px

border-color

Se puede establecer un color para cada uno de los cuatro bordes (superior, derecho, inferior, izquierdo)

- un valor: border-color: royalblue

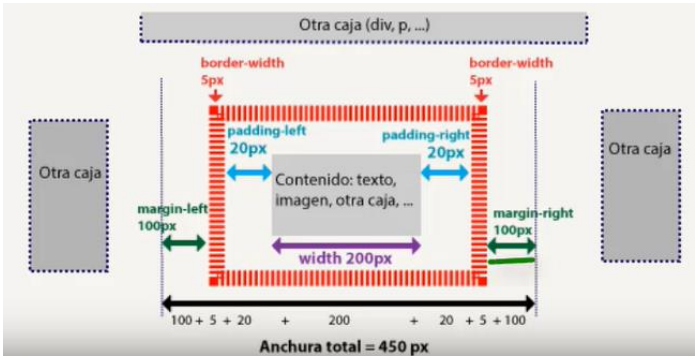
border-radius

- border-radius: 15px
- border-radius: 15px / 30px
- border-radius: 50% 50%
-

Cálculo de la anchura y altura

Como ya hemos visto los atributos width y height definen la anchura y altura de los elementos HTML

Vamos a explicar el caso de width con el siguiente esquema (el height sería similar):



El esquema corresponde a la definición de un contenedor (por ejemplo div) con estos atributos:

```
div { width: 200px; border-width: 5px; padding: 20px; margin: 100px }
```

Por defecto (sin definir el box-sizing) deberemos reservar un espacio en nuestro diseño de la página de 450 px., ya que al width del contenido hay que sumar el padding y el border, y luego añadir el margin. Es decir, 200 px, que ocupa el contenido, más 100 px, por cada lado del margin; otros 5 px. por cada lado del border; otros 20 px. por cada lado del padding.

Propiedad box-sizing

La propiedad **box-sizing** indica cómo se calcula el ancho y alto de la etiqueta correspondiente. Tiene dos valores: *content-box y border-box:*

- content-box, el ancho y el alto se calculan teniendo en cuenta sólo el contenido, sin incluir el border y el padding. Es el valor por defecto.

- border-box, el ancho y el alto se calculan teniendo en cuenta el contenido y también el border y el padding. Es decir, el border y el padding están incluidos en el ancho y alto.

```
div { box-size: border-box; width: 200px; border-width: 5px; padding:
20px; margin: 100px }
```

Desbordamiento. overflow

La propiedad overflow indica cómo se muestra el contenido cuando es mayor que las dimensiones de la caja que lo contiene. Puede tomar cuatro valores: visible, hidden, scroll, auto

1- **visible**, el contenido desborda la caja y se superpone con la caja siguiente. Es el valor por defecto
2- **hidden**, el contenido que desborda de la caja se oculta, no se muestra.
3- **scroll**, el contenido que desborda la caja no se muestra, pero se habilitan las barras de scroll para poder verlo (en los dispositivos táctiles se puede hacer scroll con el dedo).

Fuente

Font-family

La propiedad font-family indica la fuente a emplear. La fuente se indica como una lista de valores separados por comas. El navegador intentará utilizar la primera fuente y si no es posible, intentará con las siguientes.

```
font-family: 'Myriad Pro', Helvetica, 'sans-serif'. Lorem ipsum dolor sit amet, consectetur
adipisicing elit. eligendi porro.
```

Fuentes web

https://fonts.google.com/

```
<link href="https://fonts.googleapis.com/css?family=Aldrich" rel="stylesheet">
```

Importar la fuente en nuestra hoja de estilos desde la web de Google

```
<style> @import url('https://fonts.googleapis.com/css?family=Aldrich'); </style>
```

Font-style

La propiedad *font-style* permite indicar si la fuente es inclinada. Puede tener tres valores:

- **Normal**. no inclinada. Es el valor por defecto.

- **Oblique**, la forma de la letra no cambia, simplemente se inclina

- **Italic**, además de inclinarse, la forma de algunas letras cambia

Font-size

La propiedad font-size establece el tamaño de la fuente. El valor se puede indicar en varias unidades de medida, unidades absolutas (px, mm, ...) unidades relativas (em, %, ...) o con valores predeterminados (xx-small, x-small, small, medium, large, x-large y xx-large).

```
font-size: 1.2em. Lorem ipsum dolor sit amet, consectetur adipisicing elit. eligendi porro .
```

Font-weight

La propiedad font-weight establece el grosor de la fuente valores predeterminados (normal, bold, lighter y bolder)

```
font-weight: bold. Lorem ipsum dolor sit amet, consectetur adipisicing elit. eligendi porro .
```

Color del texto

caracteres que forman palabras, líneas y párrafos son las siguientes: color, text-align, vertical-align, text-decoration, text-transform, text-indent

```
color: #000000. Lorem ipsum dolor sit amet, consectetur adipisicing elit. eligendi porro .
```

Text-align

La propiedad text-align define la alineación horizontal de un texto. Se pueden aplicar los siguientes valores:

- **left**, texto alineado a la izquierda. text-align: left

right, texto alineado a la derecha.

- **center**, texto alineado al centro de su contenedor, dejando igual espacio a la derecha que a la izquierda en cada línea

justify, texto justificado, es decir, el texto se alinea a la derecha y a la izquierda en cada línea, para ello se introducen espacios en blanco entre las palabras. Con la propiedad text-justify

text-shadow

La propiedad text-shadow permite crear sombra en el texto

```
text-shadow: 3px 2px 4px grey;
```

Flexbox 1

Para el ejercicio partimos de lo siguientes elementos que tenemos creados ya para dar inicio y conocer las distintas referencias que se pueden usar con Flexbox

Página HTML

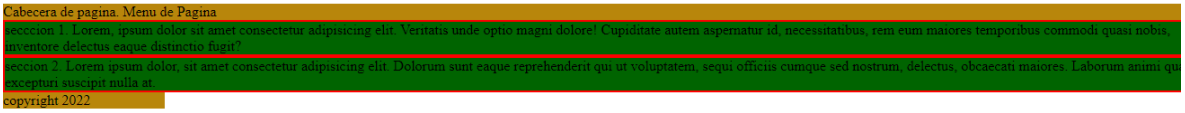
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../css/04_flex1.css">
    <title>Cajas Flexibles</title>
</head>
<body>
    <header>
        Cabecera de pagina.  Menu de Pagina
    </header>
    <main>
        <section>
            secccion 1. Lorem, ipsum dolor sit amet consectetur adipisicing
elit. Veritatis unde optio magni dolore! Cupiditate autem aspernatur id,
necessitatibus, rem eum maiores temporibus commodi quasi nobis, inventore
delectus eaque distinctio fugit?
        </section>
        <section>
            seccion 2. Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Dolorum sunt eaque reprehenderit qui ut voluptatem, sequi officiis
cumque sed nostrum, delectus, obcaecati maiores. Laborum animi quam
excepturi suscipit nulla at.
        </section>
    </main>
    <footer>
```

```
        copyright 2022
    </footer>
</body>
</html>
```

Archivo CSS

```
header{
    background-color: darkgoldenrod;
}

main{
    background-color: brown;
}

section{
    background-color:darkgreen ;
    border: 2px solid red;
}
footer{
    background-color: darkgoldenrod;
    width: 180px;
}
```

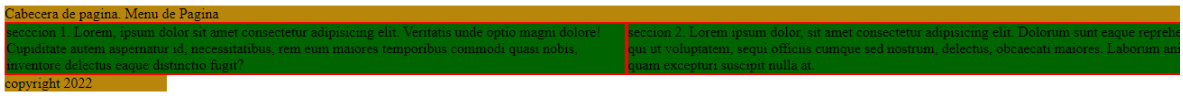
Vista en el navegador



1. Aumentamos en **main** por(al definir esta propiedad en el contenedor sus hijos se convierten en elementos flexibles)

```
display:flex;
```

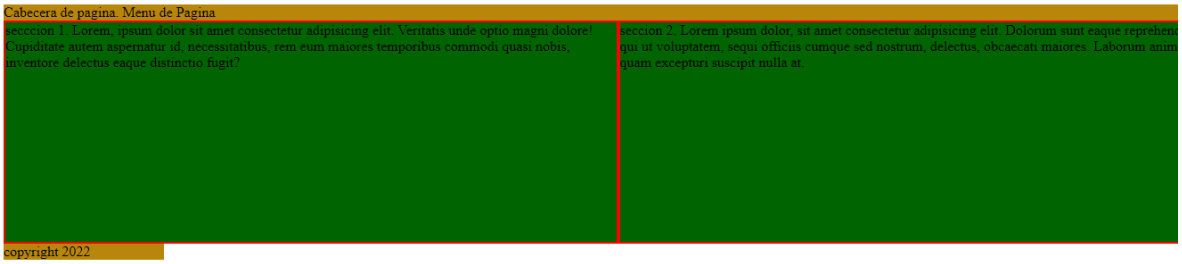
Vista en el navegador



Damos también un alto

```
height: 250px;
```

Vista en el navegador



Primera propiedad importante Display: flex;

Los elementos contenidos dentro del padre los elementos se hacen flexibles. Comprobamos que los elementos de main dejaron de ser de bloque para convertirse en flexibles

A **section** le aumentamos

```
width: 120px;
height: 60px;
```

Para posicionar los elementos en la dirección principal que es la horizontal disponemos de(Que es lo mismo que alinear a la izquierda)

```
justify-content: flex-start;
```

Alinear a la derecha	justify-content: flex-end;
Alinear de forma centrada	justify-content: center;
Alinear de forma centrada dejando espacio	justify-content: space-around;

2. Para alinear en la dirección secundaria disponemos de

```
align-items: flex-start;
```

Para alinear desde la dirección secundaria desde abajo	align-items: flex-end;
Para alinear en el centro	align-items: center;

3. Para establecer la dirección de alineación

```
flex-direction: row;
```

Para alinear por columna	flex-direction: column;

Estas serían las posiciones base de flex(por defecto)

```
justify-content: flex-start;
align-items: flex-start;
flex-direction: row;
```

Con la propiedad row observamos que se alinean los elementos por la vertical

```
justify-content: flex-end;
 align-items: flex-end;
 flex-direction: column;
```

SI decidimos dar un espacio entre los elementos y alinear

```
justify-content: space-around;
align-items: center;
flex-direction: column;
```

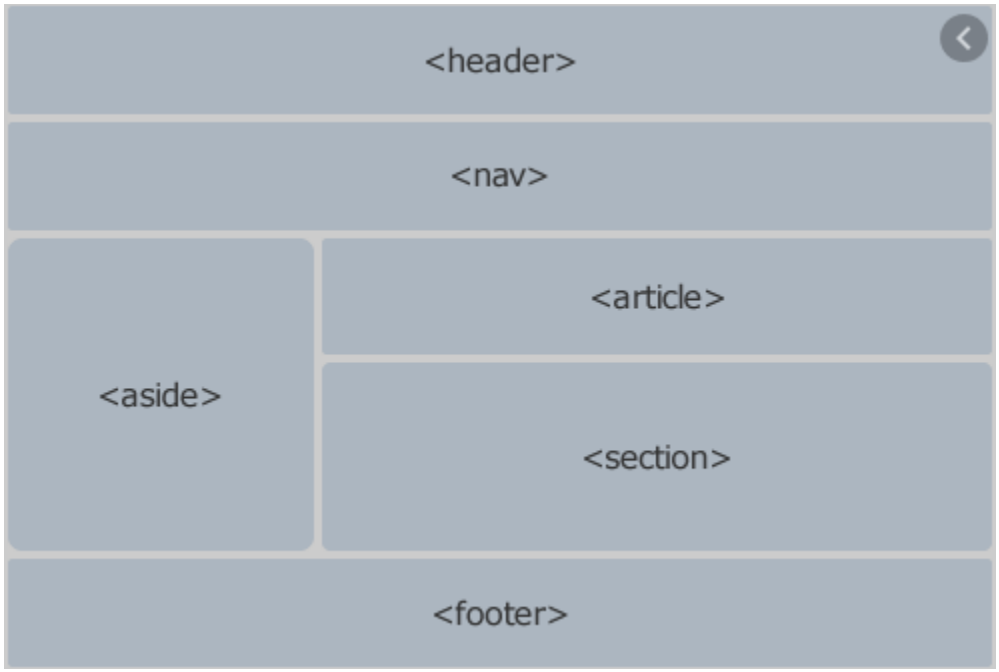
Dejamos nuevamente todas las posiciones por defecto y miramos las siguientes propiedades row-reverse

```
flex-direction: row-reverse;
```

Mas información Flexbox: Alineación

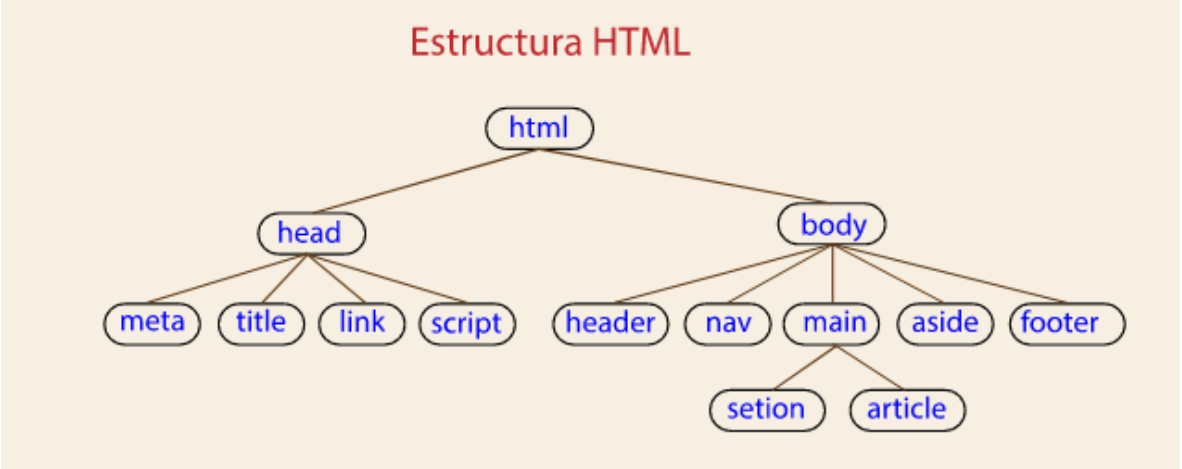
https://www.aulaclic.es/html/secuencias/p13_7_flex_2.htm

Estructura de una página web



En la cabecera tenemos etiquetas para definir los metadatos <meta> como el juego de caracteres (charset) y el **viewport**, a continuación el título <title> y enlaces a archivos de hojas de estilo <link> y Javascript <script>

Dentro del cuerpo <**body**> tenemos todo el contenido de la página web, por lo tanto, podemos incluir en él prácticamente todas las etiquetas de HTML, sin ninguna regla en cuanto al orden de las etiquetas, sin embargo, es conveniente dar una estructura lógica a la página mediante el uso adecuado de las etiquetas semánticas. Este tipo de etiquetas permiten definir cabeceras <**header**>, cuerpo principal <**main**>, secciones <**section**>, artículos <**article**>, contenido adicional <aside> y pies <**footer**>.



Diseño adaptativo con cajas flexibles

Utilizando la técnica **first Mobile**, una forma de mejora progresiva, es un enfoque de desarrollo y diseño web que se enfoca en la priorización del diseño y el desarrollo para dispositivos móviles

Unidad de medida viewport

En el concepto de la ventana viewport que se utiliza para optimizar el diseño en sitios adaptables, inicialmente el ancho del viewport se suele hacer coincidir con el ancho de la ventana. Por ejemplo, si se define el viewport con la etiqueta meta de la siguiente forma:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

En estas condiciones vw es casi equivalente a la unidad de porcentaje %, la diferencia es que el porcentaje se refiere al elemento contenedor, mientras que vw se refiere siempre al viewport. Es decir que si definimos un ancho de 100vw en un elemento que esté dentro de otro, siempre va a tener el ancho del viewport, independientemente del ancho del elemento que lo contiene. Algo que no ocurre si definimos el ancho como 100%, ya que en este caso se refiere al 100% del ancho del elemento que lo contiene.

***vh*(alto)**, el valor se calcula como un porcentaje de la altura del viewport.

vmin, igual al valor más pequeño entre vw y vh.

vmax, igual al valor más grande entre vw y vh.

Html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Diseño adaptativo de cajas con Flexbox</title>
    <link rel="stylesheet" href="../css/07_disenoweb.css">
</head>
<body>
    <div class="contenedor">
        <header>
            <div class="logo">
                <img src="../imagen/logo.png" width="150" alt="">
                <strong><a class="sena" href="#">REDMedia</a></strong>
            </div>

            <nav>
                <a href="#">Inicio</a>
                <a href="#">Blog</a>
                <a href="#">Registro</a>
```

```
                <a href="#">Login</a>
            </nav>
        </header>

   <section class="main">
      <article>
        <h2 class="titulo">Lorem ipsum dolor sit amet.</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo.</p>
     </article>

     <article>
       <h2 class="titulo">Lorem ipsum dolor sit amet.</h2>
       <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.</p>
     </article>

     <aside>
       <div class="widget">
         <div class="imagen">Imagen 1</div>
       </div>

       <div class="widget">
          <div class="imagen">Imagen 2</div>
       </div>
      </aside>
    </section>

        <footer>
            <div class="social">
                <a href="#">Autor</a>
                <a href="#">E-mail</a>
            </div>
        </footer>
    </div>

</body>
</html>
```

CSS

```css
header{
    background-color: lightyellow;
}

nav{
    background-color: burlywood;
}
main{
    background-color: beige;

}
aside{
    background-color: aquamarine;
}
```

Se establece los elementos base para todo el documento y el body

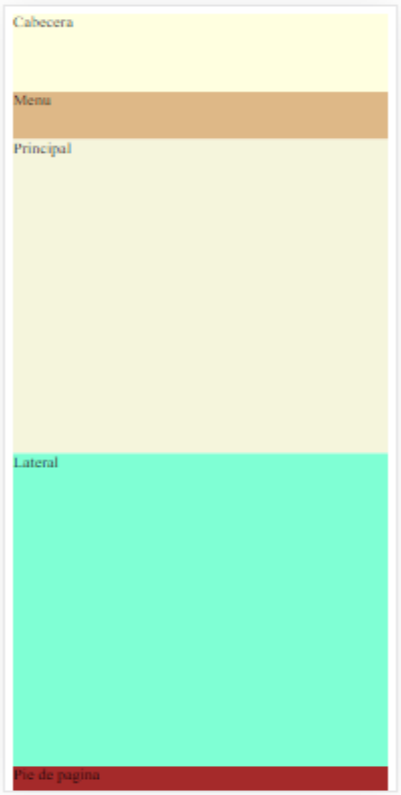
```css
*{
    margin: 0px;
    padding: 0px;
    /*se deben calcular las medidas de un elemento*/
    box-sizing: border-box;
}

body{
    background:#efefef;
    font: 14px arial;
}
```

Diseño establecido para pantallas móviles

Definimos una regla ***@media query*** para pantallas pequeñas

Para las pantallas pequeñas (<= 640)

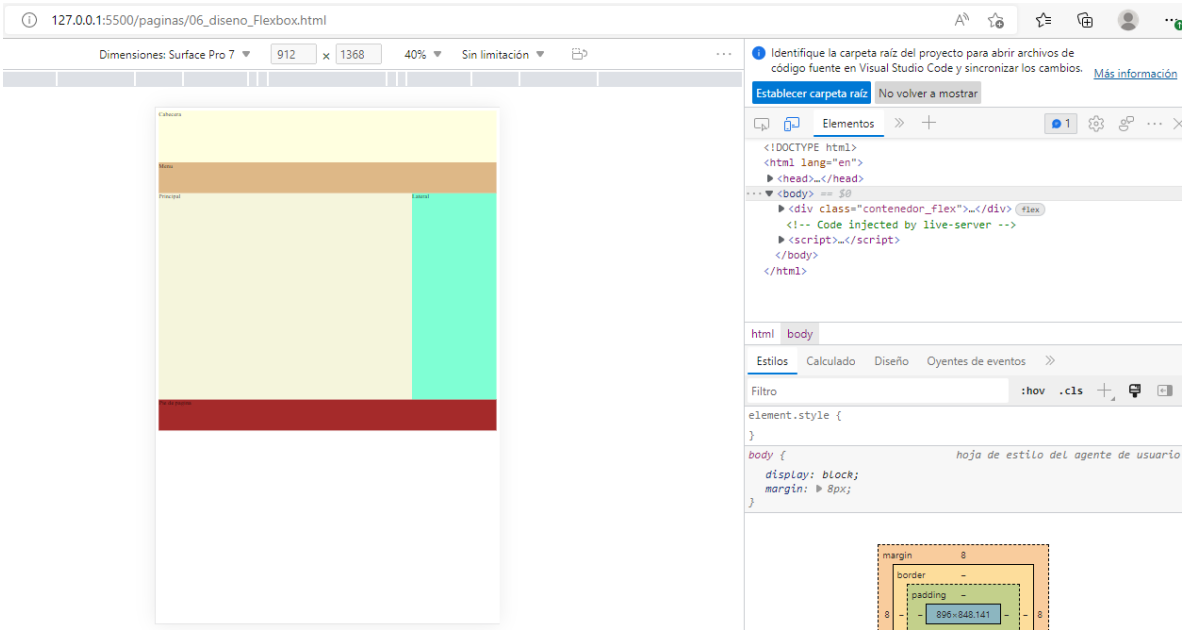
Definimos a contenedor flex que sea flexible para que ahora cada elemento sea un elemento flexible




```
/*Diseño sin cascada ancho menor a  640px*/
@media screen and (max-width:640px){
.contenedor{
    display: flex;
    flex-direction: column;
}
header{height: 10vh;}
nav{height: 10vh;}
.main{height: 80vh;}
footer{height: 10vh;
}
.main{ /*este tamaño es el total del div para
main y aside*/
    height: 80vh;
}
aside{
    height: 20vh;
}

}
```

Diseño para pantallas medianas

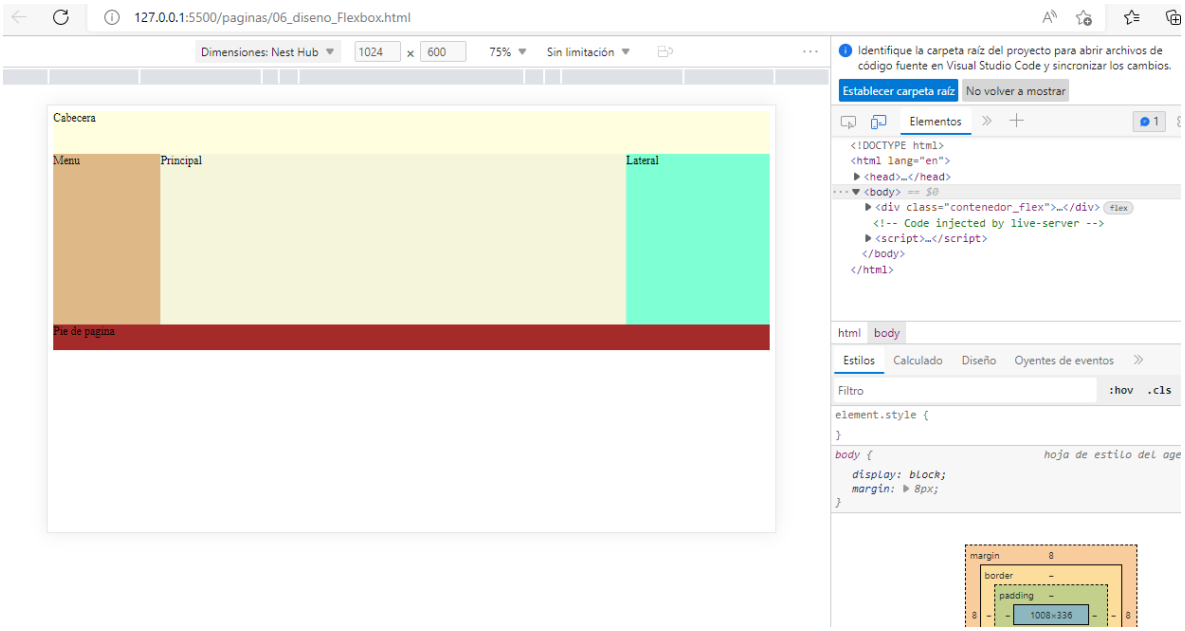
Para las pantallas medianas (>= 641 y <= 1023) aplicamos la propiedad flex-wrap: wrap;



```
/*ancho mayor o igual a 641px y menor a 1024px*/
@media screen and (min-width:641px) and (max-width:1023px){
    .contenedor{
        display: flex;
        flex-direction: column;
    }
    .main{
        display: flex;
        flex-direction: row;
        flex-wrap:nowrap;
    }
    header{height: 10vh;}
    nav{height: 10vh;}
    .main{height: 80vh;}
    footer{height: 10vh;
    }
    .main{ /*este tamaño es el total del div para main y aside*/
        height: 80vh;
        width: 70vw;
    }
    article{
        height: 80vh;
        width: 30vw;
    }
    }
```

Diseño para pantallas grandes

Para las pantallas grandes (>= 1024) aplicamos la propiedad flex-direction: row




```css
/*ancho mayor o igual a 1024px*/
@media screen and (min-width:1024px){
    .contenedor{
        display: flex;
        flex-direction: column;
    }
    .main{
        display: flex;
        flex-direction: row;
        /*nowrap para que no se desborde el contenido en
        pantallas grandes a otra fila*/
        flex-wrap:nowrap;
    }
    header{height: 10vh;}
    nav{height: 10vh;}
    .main{height: 80vh;}
    footer{height: 10vh;
    }
    .main{ /*este tamaño es el total del div para main y aside*/
        height: 80vh;
        width: 70vw;
    }
    article{
        height: 80vh;
        width: 30vw;
    }
    }
```

Distribuyendo contenido en nuestro diseño

En primera instancia ubicamos las propiedades para contenedor que tiene todos los elementos de contenido

```
/*Etiquetas para trabajar el contenido*/

.contenedor{
    background:#aadfc9;
    width:90%;
    max-width:1000px;
    margin:auto;
}
```

Preparo el ***header*** para recibir el contenido de logo e imagen

```
header{
    background-color: #5eddef;
    width: 100%;
    padding: 3px;
    display:flex;
    justify-content: space-between;
    align-items: center;
    flex-wrap: wrap;
}
```

```
.redmedia{
    color: #fff;
    width: 30%;
    font-size: 20px;
    text-decoration:none;
    line-height: 60px;
    padding: 5px;
    top: 5px;
}
```

```
.img_logo{
   width: 40px;
   height: 40px;
   padding-top: 6px;
   padding-left: 6px;
   vertical-align:top;

}
```

Menú de contenido

```
nav{
    border: 1px solid red;
    width: 50%;
    display:flex;
    flex-wrap: wrap;
    align-items: center; /*alineacion vertical*/
    justify-content: center; /*alineacion horizontal*/

}
```

```
.menu{
    background: #cceaf4;
    color: #114358;
    text-decoration: none;
    padding: 10px;
    flex-wrap:1;
    width: 100px;

}
```

```
a:hover{
    background: #5eddef;
}
```

Preparo el main

```
.main {
    border: 1px solid red;
    background:#fff;
    padding:20px;
    /*Flex 1(proporcion que puede crecer el elemento), 1(cuanto se encogeria
    70% seria el whidth*/
    flex:1 1 70%;

}
```

Sección de texto de la pagina

```
article {
    border: 1px solid red;
    margin-bottom: 5px;
```

```
    padding-bottom:5px;

}
```

```
aside {
    border: 1px solid red;
    background:#276e90;
    padding:5px;
    /*FLEX*/
    flex:1 1 30%; /*asigno el restante 30% del contenedor main*/
    /*flex:0 0 300px;*/

    /*manipular la informacion como hijos*/
    display: flex;
    flex-wrap:wrap;
    flex-direction:column;
    justify-content:flex-start;
}
```

Imagen de aside

```
.widget {
    background: #fff;
    height:150px;
    margin:5px;
}
```

Pie de pagina

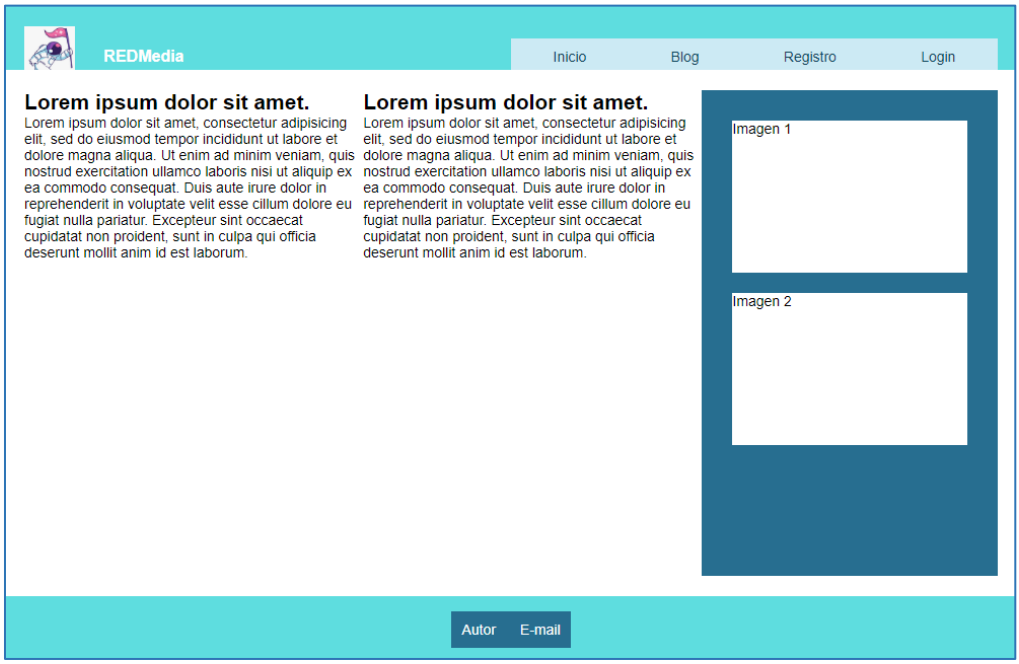
```
footer {
    border: 1px solid red;
    background:#5edddf;
    width: 100%;
    padding:15px;

}
```

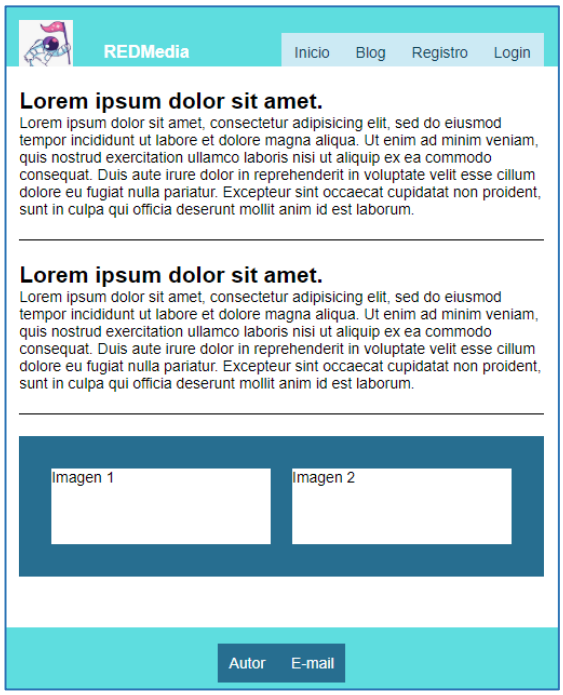
Contenido de footer

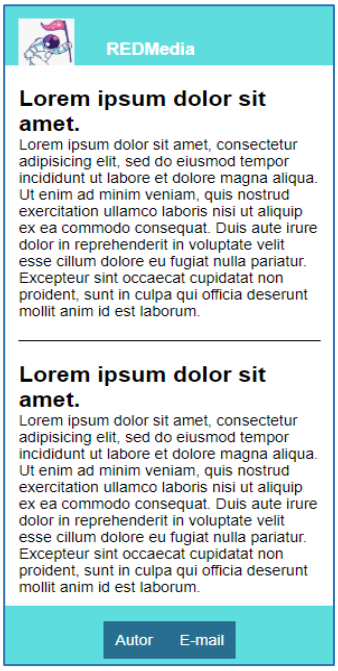
```
.social {
    background:#276e90;
    text-align: center;

}
```

```css
a {
    color:#fff;
    text-decoration: none;
    padding:10px;
    display: inline-block;
}
```

Diseño Final del Landingpage pantallas grande – mediana - pequeña







Unidades de medida viewport

https://www.yunbitsoftware.com/blog/2017/06/22/viewport-units-css-que-es/

Si en el diseño inicial necesitamos establecer un nuevo elemento de contenido este se va al final del diseño establecido

Bibliografía

Pasos iniciales en una página web

https://www.youtube.com/watch?v=UB1O30fR-EE

Recursos oficiales de HTML y CSS

https://www.w3schools.com/

Tutoriales para crear varios elementos web

http://www.falconmasters.com/category/tutoriales/

Pagina para referencias de los estilos CSS

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor(es)	Franco Reina	Instructor	CTPI	Julio 2022