

2023/24

U.D.3. USO DE ESTILOS.

3.5 Guías prácticas.





ÍNDICE

1. Introducción	2
2. Estilos en botones CSS	2
2.1 Cómo crear botones de tipo <i>alert</i>	3
2.2 Cómo crear botones de tipo <i>outline</i>	4
2.3 Cómo crear botones que ocupen el 100% del contenedor	5
2.4 Cómo crear un paginado con CSS	7
2.5 Cómo crear migas de pan con CSS	8
3. Estilos, elementos y código para la composición de tu web.	10
3.1 Media queries para diseño web responsive	10
3.2 Lista convertida en menú horizontal	14
3.3 Contenedor con Parallax	14
3.4 Botón fijo en la parte inferior derecha	16
3.5 Formulario básico	17
3.6 Centrado vertical y horizontal de un contenedor	17
3.7 Contenedores mediante porcentajes	18
3.8 Contenedores mediante cajas flexibles	19
3.9 Regla !important	22
3.10 Ejemplo float/ inline-block /inline	22
3.11 Alinear imágenes con texto o botones	24
3.12 Contenedor fijo + contenedor variable o con porcentaje CSS	26
4. Tooltip en CSS	28
4.1 Pasos para crear tooltips en CSS	28
4.2 Crear un tooltip en diferentes posiciones	31
5. PRÁCTICAS CSS	33
5.1 EJEMPLO 1. Miniatura de producto	33
5.2 EJEMPLO 2. Web con menú, area hero con parallax, llamada a la acción, footer, botón de WhatsApp fijo, etc.	36



1. Introducción

En este documento, encontraréis ejemplos de código HTML y CSS para implementar, dependiendo de lo que queramos hacer en nuestra página web.

Además del código que podáis encontrar aquí, la W3schools pone a nuestra disposición el siguiente manual de casos prácticos:

<https://www.w3schools.com/howto/default.asp>

2. Estilos en botones CSS

Los botones CSS son **elementos interactivos** en una página web que se pueden estilizar utilizando estilos CSS para mejorar su apariencia y comportamiento.

A continuación, se muestra un ejemplo básico de cómo crear un botón con CSS:

HTML:

```
<button class="my-button">Haz clic aquí</button>
```

CSS:

```
.my-button {  
  padding: 10px 20px;  
  background-color: #4CAF50;  
  color: white;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}  
.my-button:hover {  
  background-color: #45a049;  
}  
.my-button:active {  
  background-color: #3e8e41;  
}
```



Haz clic aquí



En este ejemplo, hemos definido una clase **.my-button** que se aplica al botón en el HTML. Utilizamos propiedades CSS como **padding**, **background-color**, **color**, **border**, **border-radius** y **cursor** para estilizar el botón. También hemos añadido estilos para el estado **:hover** (cuando el cursor está sobre el botón) y **:active** (cuando el botón está siendo clicado).

Puedes personalizar los estilos y agregar más efectos según tus necesidades y preferencias. Los botones CSS ofrecen flexibilidad para crear diseños únicos y atractivos en tus páginas web.

2.1 Cómo crear botones de tipo *alert*

Veamos cómo crear botones de tipo *alert*, del mismo estilo que los botones de Bootstrap.

HTML:

```
<button class="btn primary">Primary</button>
<button class="btn secondary">Secondary</button>
<button class="btn success">Success</button>
<button class="btn warning">Warning</button>
<button class="btn danger">Danger</button>
```

CSS:

```
.btn {
  border: none;
  color: white;
  padding: 14px 28px;
  cursor: pointer;
  border-radius: 5px;
}
.primary {background-color: #007bff;}
.primary:hover {background: #0b7dda;}

.secondary {background-color: #e7e7e7; color: black;}
.secondary:hover {background: #ddd;}
```



```
.success {background-color: #04AA6D;}  
.success:hover {background-color: #46a049;}  
  
.warning {background-color: #ff9800;}  
.warning:hover {background: #e68a00;}  
  
.danger {background-color: #f44336;}  
.danger:hover {background: #da190b;}
```



2.2 Cómo crear botones de tipo *outline*

Veamos cómo crear botones de tipo *outline*, del mismo estilo que los botones de Bootstrap.

HTML:

```
<button class="btn primary">Primary</button>  
<button class="btn secondary">Secondary</button>  
<button class="btn success">Success</button>  
<button class="btn warning">Warning</button>  
<button class="btn danger">Danger</button>
```

CSS:

```
.btn {  
  border: none;  
  color: white;  
  padding: 14px 28px;  
  cursor: pointer;  
  border-radius: 5px;  
}
```



```
.primary {background-color: white; border: 2px solid #007bff;}  
.primary:hover {background: #0b7dda; color: white;}  
  
.secondary {background-color: white; border: 2px solid #e7e7e7;}  
.secondary:hover {background: #ddd; color: white;}  
  
.success {background-color: white; border: 2px solid #04AA6D;}  
.success:hover {background-color: #46a049; color: white;}  
  
.warning {background-color: white; border: 2px solid #ff9800;}  
.warning:hover {background: #e68a00; color: white;}  
  
.danger {background-color: white; border: 2px solid #f44336;}  
.danger:hover {background: #da190b; color: white;}
```



2.3 Cómo crear botones que ocupen el 100% del contenedor

Para que los botones ocupen el 100% de su contenedor podemos agregarles la siguiente clase `btn-block`.

HTML:

```
<button class="btn primary">Primary</button>  
<button class="btn secondary">Secondary</button>  
<button class="btn success">Success</button>  
<button class="btn warning">Warning</button>  
<button class="btn danger">Danger</button>
```



CSS:

```
.btn-block{
  width: 100%;
  display: block;
  margin-bottom: 5px;
}

.btn {
  border: none;
  color: white;
  padding: 14px 28px;
  cursor: pointer;
  border-radius: 5px;
}

.primary {background-color: #007bff;}
.primary:hover {background: #0b7dda;}

.secondary {background-color: #e7e7e7; color: black;}
.secondary:hover {background: #ddd;}

.success {background-color: #04AA6D;}
.success:hover {background-color: #46a049;}

.warning {background-color: #ff9800;}
.warning:hover {background: #e68a00;}

.danger {background-color: #f44336;}
.danger:hover {background: #da190b;}
```





2.4 Cómo crear un paginado con CSS

Crear un paginado en CSS implica utilizar estilos para diseñar los elementos que formarán parte de la paginación, como los botones de navegación y los números de página. A continuación, se muestra un ejemplo básico de cómo crear un paginado simple utilizando CSS:

HTML:

```
<ul class="pagination">
  <li><a href="#">«</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a class="active" href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">6</a></li>
  <li><a href="#">7</a></li>
  <li><a href="#">»</a></li>
</ul>
```

CSS:

```
ul.pagination {
  display: inline-block;
  padding: 0;
  margin: 0;
}
ul.pagination li {display: inline;}

ul.pagination li a {
  color: black;
  float: left;
  padding: 9px 18px;
  text-decoration: none;
}
```




```
ul.pagination li a.active {  
    background-color: #4CAF50;  
    color: white;  
}  
  
ul.pagination li a.active:hover {background-color: #397f3b;}  
  
ul.pagination li a:hover:not(.active) {background-color: #ddd;}
```



2.5 Cómo crear migas de pan con CSS

Para crear migas de pan (breadcrumbs) utilizando CSS, puedes guiarte del siguiente código:

HTML:

```
<ul class="breadcrumb">  
    <li><a href="#">Home</a></li>  
    <li><a href="#">Contact</a></li>  
    <li><a href="#">Blog</a></li>  
    <li><a class="active" href="#">Services</a></li>  
</ul>
```



CSS:

```
ul.breadcrumb {  
    padding: 8px 16px;  
    list-style: none;  
    background-color: #eee;  
}  
ul.breadcrumb li {display: inline;}  
ul.breadcrumb li+li:before {  
    padding: 8px;  
    color: black;  
    content: "/\00a0";  
}  
ul.breadcrumb a:link, ul.breadcrumb a:visited {  
    text-decoration: none;  
    color: black;  
}  
  
ul.breadcrumb li a.active {  
    color: #4CAF50;  
    font-weight: bold;  
}  
ul.breadcrumb li a.active:hover {  
    color: #397f3b;  
    font-weight: bold;  
}
```

Home / Contact / Blog / **Services**



3. Estilos, elementos y código para la composición de tu web.

3.1 Media queries para diseño web responsive

Código HTML:

```
<h1>Formación Profesional en Informática y Comunicaciones</h1>
<div id="ciclos">
  <h2>Ciclos formativos</h2>
  <ul>
    <li><strong>FPB </strong>Informática de Oficina </li>
    <li><strong>FPB </strong>Informática y Comunicaciones</li>
    <li><strong>GM </strong>Sistemas Microinformáticos y Redes</li>
    <li><strong>GS </strong>Administración de Sistemas Informáticos en
Red </li>
    <li><strong>GS </strong>Desarrollo de Aplicaciones Multiplataforma
</li>
    <li><strong>GS </strong>Desarrollo de Aplicaciones Web</li>
  </ul>
</div>
<div id="bloque2">
  <div id="cursos-avanzados">
    <h2>Cursos avanzados</h2>
    <ul>
      <li><strong>Curso de especialización:</strong> Inteligencia
Artificial y Big Data</li>
      <li><strong>Curso de especialización:</strong> Desarrollo de
videojuegos y realidad virtual</li>
      <li><strong>Curso de especialización:</strong> Ciberseguridad en
Entornos de las Tecnologías de la Información</li>
    </ul>
  </div>
</div>
```



```
<div id="como-estudiar">
  <h2>Cómo estudiar</h2>
  <ul>
    <li><strong>Modalidad presencial:</strong> se estudia en los
      centros educativos, con la asistencia regular a
      clase.</li>
    <li><strong>Modalidad a distancia:</strong> se estudia con
      comodidad desde cualquier lugar y los exámenes se
      realizan presencialmente.</li>
    <li><strong>Pruebas libres:</strong> vía alternativa para obtener
      las titulaciones sin necesidad de cursar los ciclos
      formativos.</li>
  </ul>
</div>
```

Código CSS:

```
body {
  font-family: Arial;
}
#cursos-avanzados li strong, #como-estudiar li strong {
  color: purple;
}
@media (min-width: 1201px) {
  #ciclos {
    float: left;
    width: 30%;
  }
  #bloque2 {
    float: left;
    width: 70%;
  }
}
```



```
#cursos-avanzados {
    float: left;
    width: 50%;
}
#como-estudiar {
    float: left;
    width: 50%;
}
}
@media (min-width: 601px) and (max-width: 1200px) {
    #como-estudiar li strong {
        text-decoration: underline;
    }
    #ciclos {
        float: left;
        width: 30%;
    }
    #bloque2 {
        float: left;
        width: 70%;
    }
    #ciclos ul {
        list-style-type: none;
        padding: 0;
    }
    #ciclos li + li {
        margin-top: 10px;
    }
    #ciclos li strong {
        display: block;
    }
}
```



```
@media (max-width: 600px) {  
  body {  
    font-family: "Times New Roman";  
  }  
  h1 {  
    font-size: 1.5em;  
  }  
  h2 {  
    font-size: 1.2em;  
    border-bottom: 1px solid #000;  
  }  
  ul {  
    list-style-type: none;  
    padding: 0;  
  }  
  li + li {  
    margin-top: 10px;  
  }  
  li strong {  
    display: block;  
  }  
}
```

Formación Profesional en Informática y Comunicaciones

Ciclos formativos

- FPB Informática de Oficina
- FPB Informática y Comunicaciones
- GM Sistemas Microinformáticos y Redes
- GS Administración de Sistemas Informáticos en Red
- GS Desarrollo de Aplicaciones Multiplataforma
- GS Desarrollo de Aplicaciones Web

Cursos avanzados

- **Curso de especialización:** Inteligencia Artificial y Big Data
- **Curso de especialización:** Desarrollo de videojuegos y realidad virtual
- **Curso de especialización:** Ciberseguridad en Entornos de las Tecnologías de la Información

Cómo estudiar

- **Modalidad presencial:** se estudia en los centros educativos, con la asistencia regular a clase.
- **Modalidad a distancia:** se estudia con comodidad desde cualquier lugar y los exámenes se realizan presencialmente.
- **Pruebas libres:** vía alternativa para obtener las titulaciones sin necesidad de cursar los ciclos formativos.



3.2 Lista convertida en menú horizontal

Para crear un menú horizontal, puedes tomar como referencia el código que hemos visto en el apartado de ‘Propiedades de las listas en CSS’.

3.3 Contenedor con Parallax

En este contenedor, el **logo se moverá sobre la imagen de fondo** si el usuario le da scroll.

Código HTML:

```
<p>Aquí mi menú, puedes ver <a href="https://www.menu.com/atributos-  
propiedades-listas-css/#85_Lista_convertida_en_botonera_horizontal"  
target="_blank">cómo hacer uno aquí</a></p>  
<div class="parallax">  
    
</div>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing 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  
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse  
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat  
non proident, sunt in culpa qui officia deserunt mollit anim id est  
laborum</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing 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  
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse  
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat  
non proident, sunt in culpa qui officia deserunt mollit anim id est  
laborum</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing 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  
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse  
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat  
non proident, sunt in culpa qui officia deserunt mollit anim id est  
laborum</p>
```



Código CSS:

```
.parallax {  
    background-image: url("https://www.imagenes.com/uploads/home-  
        historia.jpg");  
  
    height: 400px;  
    background-attachment: fixed; /* valor posible: scroll o fixed */  
    /*  
    scroll: la imagen de fondo se moverá dentro de la pantalla junto al  
        bloque que la contiene.  
    fixed: la imagen de fondo estará fija en la pantalla y no se moverá con  
        el bloque contenedor.  
    */  
    background-position: center;  
    background-repeat: no-repeat;  
    background-size: cover; /* valor posible: cover o contain*/  
    /*  
    cover: Escala la imagen al mayor tamaño posible sin estirla. Si las  
        proporciones de la imagen difieren de las del elemento, es  
        recortada vertical u horizontalmente para que no quede un  
        espacio vacío.  
    contain: Escala la imagen al mayor tamaño posible sin recortarla ni  
        estirla.  
    */  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}  
.parallax img{  
    max-width: 350px;  
}
```




Aquí mi menú, puedes ver [cómo hacer uno aquí](#)



Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.4 Botón fijo en la parte inferior derecha

Código HTML:

```
<a id="btn-fixed" href="#"></a>
```

Código CSS:

```
#btn-fixed {  
  /*display: block;*/  
  position: fixed;  
  bottom: 10px;  
  right: 10px;  
}
```





3.5 Formulario básico

Para crear un formulario básico, puedes tomar como referencia el código que hemos visto en el apartado de '**Estilos en formularios CSS**'.

3.6 Centrado vertical y horizontal de un contenedor

A continuación, se presenta un ejemplo comparativo con flexbox y con la propiedad **transform**.

Código HTML:

```
<div id="container"> <!--Centrado con transform-->
    
</div>
<div id="container-flex"> <!--Centrado con flexbox-->
    
</div>
```

Código CSS:

```
#container {
    height: 550px;
    background-color: black;
    position: relative;
}
#container img {
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%);
    -webkit-transform: translate(-50%, -50%);
    max-width: 80%;
}
```



```
#container-flex {  
  height: 550px;  
  background-color: grey;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}  
#container-flex img {  
  max-width: 80%;  
}
```



3.7 Contenedores mediante porcentajes

Código HTML:

```
<div class="img-container">  
    
</div>  
<div class="img-container">  
    
</div>  
<div class="img-container">  
    
</div>
```



Código CSS:

```
.img-container {  
  box-sizing: border-box;  
  -webkit-box-sizing: border-box;  
  float: left;  
  width: 33.33%;  
  padding: 5px;  
}  
.img-container img{  
  width: 100%;  
}
```



3.8 Contenedores mediante cajas flexibles

Código HTML:

```
<!doctype html>  
<html>  
  <head>  
    <script src="https://kit.fontawesome.com/509deda241.js"  
      crossorigin="anonymous"></script>  
  </head>  
<body>  
  <div class="info-box">  
    <h2>This is an info-box</h2>
```



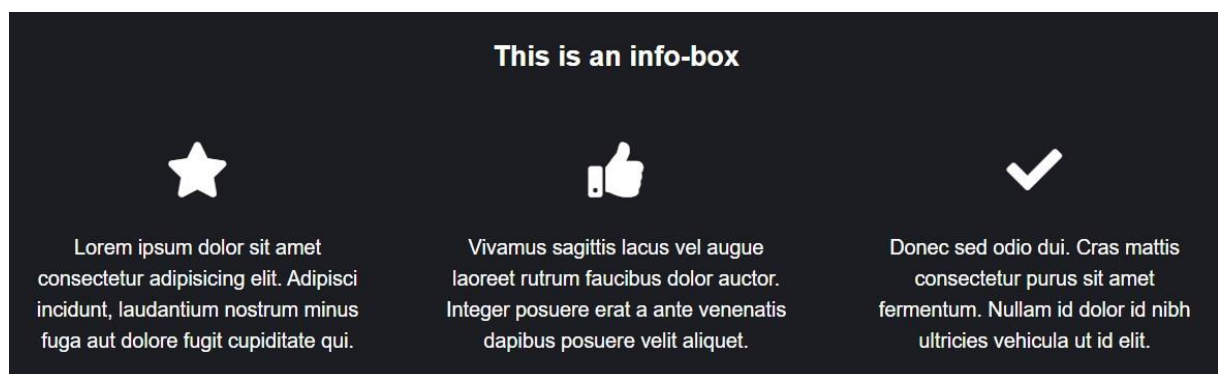
```
<div id="flex-p">
  <p class="big"><i class="fas fa-star"></i><br>Lorem ipsum dolor
    sit amet consectetur adipisicing elit. Adipisci incidunt,
    laudantium nostrum minus fuga aut dolore fugit cupiditate
    qui.</p>
  <p class="big"><i class="fas fa-thumbs-up"></i><br>Vivamus
    sagittis lacus vel augue laoreet rutrum faucibus dolor
    auctor. Integer posuere erat a ante venenatis dapibus
    posuere velit aliquet.</p>
  <p class="big"><i class="fas fa-check"></i><br>Donec sed odio dui.
    Cras mattis consectetur purus sit amet fermentum. Nullam
    id dolor id nibh ultricies vehicula ut id elit.</p>
</div>
</div>
</body>
```

Código CSS:

```
*{
  margin: 0px;
  padding: 0 px;
  font-family: "Roboto", Arial, Tahoma, sans-serif;
}
p.big{
  font-size: 1.1em;
  line-height: 1.5em;
}
.fas-star, .fas-thumbs-up, .fas-check{
  font-size: 2.7em;
  display: block;
  color: black;
  color: white;
  margin-bottom: 30px;
}
```



```
.info-box{
  background-color: #1C1D22;
  margin: 0px auto;
  padding: 110px 0 110px 0;
  color: white;
  text-align: center;
}
.info-box h2{
  margin-bottom: 30px;
}
#flex-p{
  justify-content: center;
  display: flex;
  flex-wrap: wrap;
  align-items: center;
}
#flex-p p{
  padding: 30px;
  max-width: 300px;
}
```





3.9 Regla !important

La palabra clave **!important** se utiliza para dar mayor prioridad al valor de una propiedad de forma que si se define la misma propiedad para un elemento no se pueda sobrescribir.

```
.a { background-color: red!important; }  
.a { background-color: blue; } /* los elementos con clase "a" tendrán el  
fondo de color rojo */
```

3.10 Ejemplo float/ inline-block /inline

Código HTML:

```
<div id="contenedor1">  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing  
    elit. Duis molestie turpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing  
    elit. Molestie urpis vitae ante.</p>  
</div>  
  
<div id="contenedor2">  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis  
    molestie turpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie  
    urpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie  
    urpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis  
    molestie turpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie  
    urpis vitae ante.</p>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie  
    urpis vitae ante.</p>  
</div>
```



```
<div id="contenedor3">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis
    molestie turpis vitae ante.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie
    urpis vitae ante.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie
    urpis vitae ante.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis
    molestie turpis vitae ante.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie
    urpis vitae ante.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie
    urpis vitae ante.</p>
</div>
<br>
<a href="https://www.teoria.com/posicion-comportamiento-contenedores-
  css/">Ver teoría</a>
```

Código CSS:

```
#contenedor1 p{
  float: left;
  width: 200px;
  margin: 10px;
}
#contenedor2{
  clear: both;
}
#contenedor2 p{
  display: inline-block;
  width: 200px;
  margin: 10px;
}
#contenedor3 p{
  display: inline;
  width: 200px;
  margin: 10px;
}
```




Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Duis molestie turpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Molestie urpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Duis molestie turpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Molestie urpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Molestie urpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Duis molestie turpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Molestie urpis vitae ante.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Molestie urpis vitae ante.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis molestie turpis vitae ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie urpis vitae ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis molestie turpis vitae ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie urpis vitae ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis molestie turpis vitae ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Molestie urpis vitae ante.

[Ver teoría](#)

3.11 Alinear imágenes con texto o botones

Código HTML:

```
<p>
  Uno</p>

<p>
  Uno</p>

<p>
  Uno</p>

<p>
  <button>Uno</button></p>

<p><button>Uno</button> Uno</p>

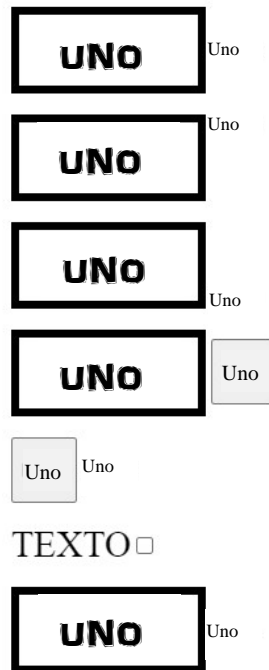
<p><label> TEXTO</label><input type="checkbox" /></p>

<div>
   <span
    class="alineadoConMargin"> Uno</span>
</div>
```



Código CSS:

```
img.alineadoTextoImagenCentro{
    vertical-align: middle;
    /* Ojo vertical-align: text-middle no existe*/
}
img.alineadoTextoImagenArriba{
    vertical-align: top;
}
img.alineadoTextoImagenAbajo{
    vertical-align: bottom;
}
button{
    height: 50px;
}
label, input{
    font-size: 28px;
    vertical-align: middle;
}
div img{
    float: left;
}
.alineadoConMargin{
    margin-top: 25px;
    float: left;
}
```



3.12 Contenedor fijo + contenedor variable o con porcentaje CSS

Realización de **cálculos con calc()** de CSS para conseguir un contenedor fijo y otro variable.

Código HTML:

```
<div id="div1">Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla  
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in  
culpa qui officia deserunt mollit anim id est laborum.</div>  
  
<div id="div2">Other text</div>  
  
<div id="div1">Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla  
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in  
culpa qui officia deserunt mollit anim id est laborum.</div>  
  
<div id="div2"><button>Button</button></div>
```



```
<div id="div1">Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in
    reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
    pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
    culpa qui officia deserunt mollit anim id est laborum.</div>

<div id="div2">
    </div>
```

Código CSS:

```
#div1 {
    float:left;
    width: calc(100% - 230px);
    background-color: yellow;
    padding: 15px;
    margin-bottom: 15px;
    clear: both;
    box-sizing: border-box;
}
#div2{
    float:left;
    width: 230px;
    padding: 15px;
    box-sizing: border-box;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Other text

Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Button

Lorem ipsum dolor sit amet, consectetur adipiscing 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 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

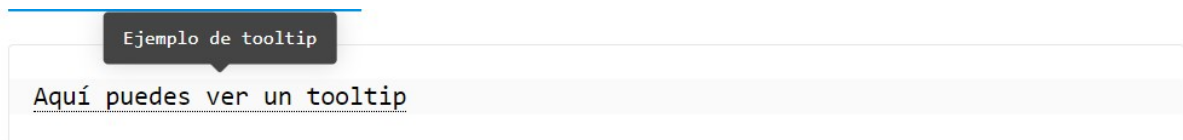
UNO



4. Tooltip en CSS.

Un **tooltip** es una **pequeña ventana emergente** que se muestra **cuando el usuario coloca el cursor sobre un elemento**. Proporciona **información adicional** o aclaratoria sobre el elemento al que está asociado, sin ocupar mucho espacio en la página. El tooltip suele aparecer cerca del elemento objetivo y desaparece cuando el cursor se mueve fuera de él.

El **contenido** de un tooltip puede ser **texto, imágenes u otros elementos HTML**. Se utiliza comúnmente para proporcionar descripciones breves, etiquetas o indicaciones sobre la función o el propósito de un elemento en particular, como enlaces, botones o imágenes. Los tooltips son una **forma efectiva de mejorar la usabilidad y la experiencia del usuario** al proporcionar información adicional de manera contextual y discreta. **Ejemplo:**



4.1 Pasos para crear tooltips en CSS

Puedes crear un *tooltip* utilizando **CSS y HTML** siguiendo estos pasos:

- 1) **Crea la estructura HTML** para el elemento que tendrá el *tooltip*. Por ejemplo, puedes usar un elemento `` dentro de un elemento con una clase específica.

```
<div class="tooltip-container">  
  <span class="tooltip-text">Este es un tooltip</span>  
  Mi elemento  
</div>
```

- 2) **Estiliza** el contenedor del *tooltip* y el texto del *tooltip* en tu archivo CSS. Puedes ajustar los estilos según tus preferencias.



```
.tooltip-container {  
    position: relative;  
    display: inline-block;  
}  
.tooltip-text {  
    visibility: hidden;  
    width: 120px;  
    background-color: #000;  
    color: #fff;  
    text-align: center;  
    border-radius: 6px;  
    padding: 5px;  
    position: absolute;  
    z-index: 1;  
    bottom: 125%; /* Posiciona el tooltip encima del elemento */  
    left: 50%;  
    transform: translateX(-50%);  
    opacity: 0;  
    transition: opacity 0.3s;  
}
```

- 3) Agrega una **regla CSS para mostrar el *tooltip*** cuando se pasa el cursor sobre el elemento. Puedes utilizar el selector **:hover** para lograrlo.

```
.tooltip-container:hover .tooltip-text {  
    visibility: visible;  
    opacity: 1;  
}
```



Con estos pasos, has creado un *tooltip* básico con CSS. Puedes personalizar los estilos y añadir efectos adicionales según tus necesidades. Además, también puedes utilizar JavaScript para crear *tooltips* más avanzados con funcionalidades adicionales.

A continuación, se muestra un **ejemplo**.

HTML:

```
<div class="tooltip-container">
  <span class="tooltip-text">Mensaje del tooltip</span>
  Este elemento tiene un tooltip
</div>
```

CSS:

```
.tooltip-container {
  position: relative;
  display: inline-block;
}
.tooltip-text {
  visibility: hidden;
  width: 120px;
  background-color: #000;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 5px;
  position: absolute;
  z-index: 1;
  top: 125%; /* Posiciona el tooltip bajo del elemento */
  left: 50%;
  transform: translateX(-50%);
  opacity: 0;
  transition: opacity 0.3s;
}
```



```
.tooltip-container:hover .tooltip-text {  
  visibility: visible;  
  opacity: 1;  
}
```

Este elemento tiene un tooltip

Mensaje del
tooltip

4.2 Crear un tooltip en diferentes posiciones

Para crear un **tooltip** que se muestre en diferentes posiciones, como encima, a la derecha, abajo y a la izquierda, puedes ajustar las propiedades de posicionamiento en CSS. Aquí tienes los pasos para lograrlo:

- 1) Asegúrate de tener la **estructura HTML** básica del *tooltip*, como se explicó anteriormente.
- 2) Agrega las **siguientes reglas CSS** para cada posición del *tooltip*:
 - a) Para el **tooltip encima** del elemento:

```
.tooltip-text {  
  bottom: 100%; /* Cambia el valor según sea necesario */  
  left: 50%;  
  transform: translateX(-50%);  
}
```

- b) Para el **tooltip a la derecha** del elemento:



```
.tooltip-text {  
    top: 50%;  
    left: 100%; /* Cambia el valor según sea necesario */  
    transform: translateY(-50%);  
}
```

c) Para el **tooltip abajo** del elemento:

```
.tooltip-text {  
    top: 100%; /* Cambia el valor según sea necesario */  
    left: 50%;  
    transform: translateX(-50%);  
}
```

d) Para el **tooltip a la izquierda** del elemento:

```
.tooltip-text {  
    top: 50%;  
    right: 100%; /* Cambia el valor según sea necesario */  
    transform: translateY(-50%);  
}
```

- 3) Asegúrate de ajustar las propiedades `width`, `background-color`, `color`, `border-radius`, `padding` y cualquier otro estilo que desees para que coincidan con el diseño de tu *tooltip*.

Con estos ajustes, el *tooltip* se posicionará correctamente en la dirección deseada según las reglas CSS proporcionadas.



5. PRÁCTICAS CSS

En este apartado se presentan algunos ejemplos de prácticas y ejercicios para comprobar tu aprendizaje y seguir trabajando en conseguir escribir un código CSS limpio, coherente y eficiente.

5.1 EJEMPLO 1. Miniatura de producto

En este apartado vamos a ver el código necesario para crear la siguiente **miniatura de producto**. Para ello se utilizarán métodos tradicionales (ni Bootstrap, ni Flexbox, ni CSS Grid).



Código HTML:

```
<main>
  <article>
    
    <div class="datos-productos">
      <div class="borrar">x</div>
      <div class="nombre">Zapatos de piel</div>
      <div class="precio">39,99 $</div>
      <div class="colores">
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
      </div>
    </div>
  </article>
</main>
```



Indicaciones para darle estilo:

- La etiqueta <article> flota a la izquierda, tiene un ancho de 350 px, un alto de 140px, el fondo es de color #F7F3F3 y tiene un margen de 10px.
- La etiqueta tiene un padding de 20px, un ancho de 100px y flota a la izquierda.
- La clase .datos-productos flota a la derecha, tiene un padding de 20 píxeles y un ancho de 190 píxeles. Al usar padding, no te olvides de la propiedad box-sizing: border-box; (quítala y ponla para ver cómo excede el contenido y al no caber se descuadra)
- La clase .borrar flota a la derecha, tiene un padding superior e inferior de 0 px y de 5px a derecha e izquierda, tiene un borde sólido de 1px de color negro, tiene un border-radius de 5 px, un margen superior de -10px y un margen izquierdo de 10 px.
- La clase .nombre flota a la derecha y está en negrita.
- La clase .precio flota a la derecha tiene un color de texto de #666, un tamaño de fuente de 14px y un margen superior e inferior de 5px, recuerda que los elementos anteriores son floats y que si no ponemos clear:both; quedarán pegados a los anteriores en vez de en el lugar que queremos (el elemento es movido hacia abajo para limpiar elementos flotantes).
- La clase .colores debe tener también clear:both por el mismo motivo que la clase precio.
- La etiqueta li debe tener un ancho de 20px y un alto de 20px, display:inline-block (para que tenga en cuenta los valores de width y height que se asignen. Prueba a modificarlo por display:inline y verás que con esa propiedad no tiene en cuenta los valores de width y height definidos), tiene un border-radius de 5px un margen izquierdo de 5px, fondo marrón y un borde sólido de 1px de color negro.



Código CSS:

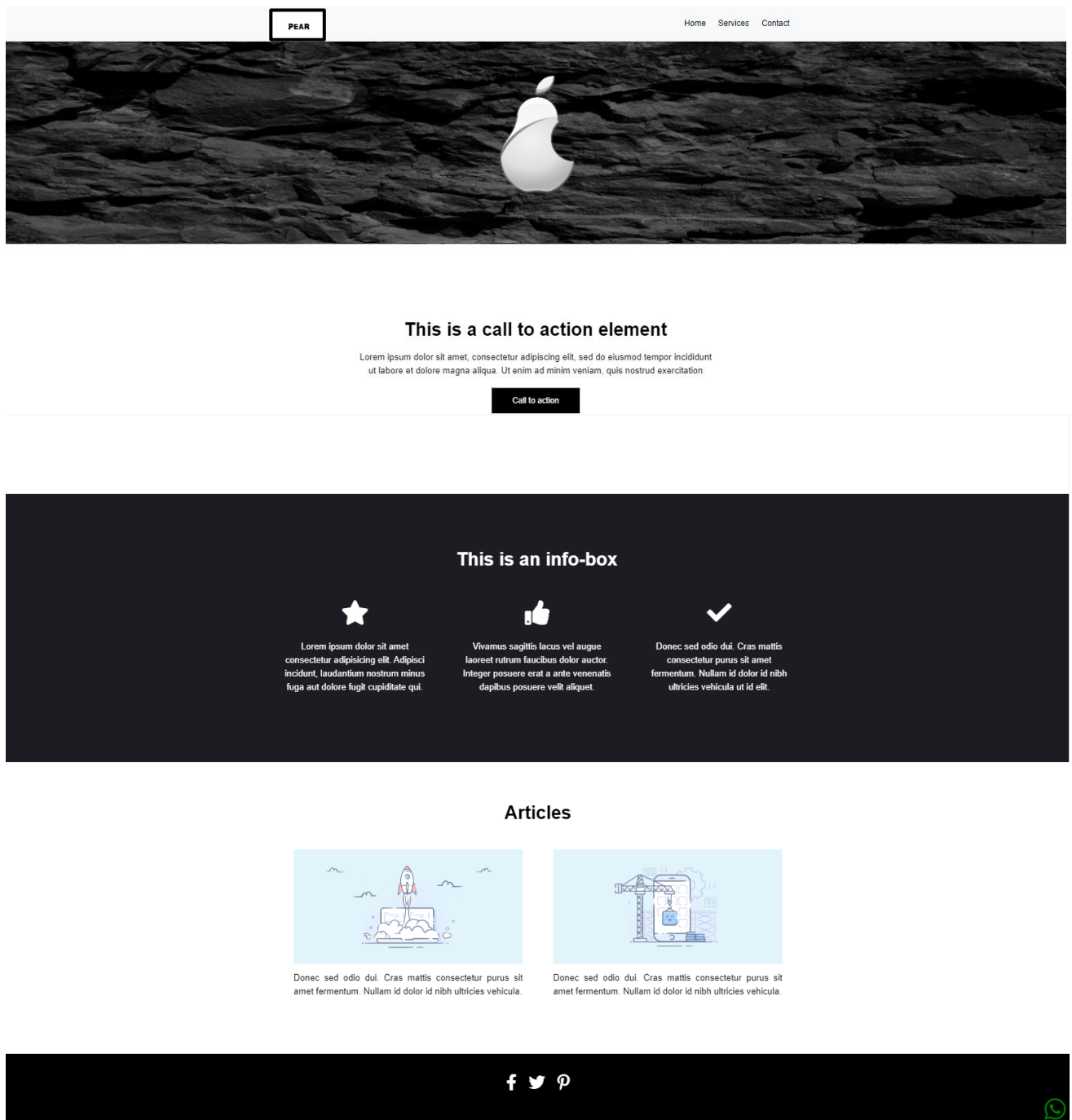
```
article{
    float: left;
    width: 350px;
    height: 140px;
    background-color: #F7F3F3;
    margin: 10px;
}
article img{
    padding: 20px;
    width: 100px;
    float: left;
}
.datos-productos{
    float: right;
    padding: 20px;
    max-width: 190px;
    box-sizing: border-box;
}
.borrar{
    float: right;
    padding: 0 5px 0 5px;
    border: 1px solid black;
    border-radius: 5px;
    margin-top: -10px;
    margin-left: 10px;
}
.nombre{
    float: right;
    font-weight: bold;
}
.precio{
    clear: both;
    float: right;
    color: #666;
    font-size: 90%;
    margin: 5px 0 5px 0;
}
.colores{
    clear: both;
}
.colores li{
    width: 20px;
    display: inline-block;
    height: 20px;
    border-radius: 12.5px;
    margin-left: 5px;
    background-color: burlywood;
    border: 1px solid black;
}
```



5.2 EJEMPLO 2. Web con menú, area hero con parallax, llamada a la acción, footer, botón de WhatsApp fijo, etc.

En ese ejemplo vamos a crear el código HTML y CSS necesario para crear la siguiente web. Incluye un efecto Parallax.

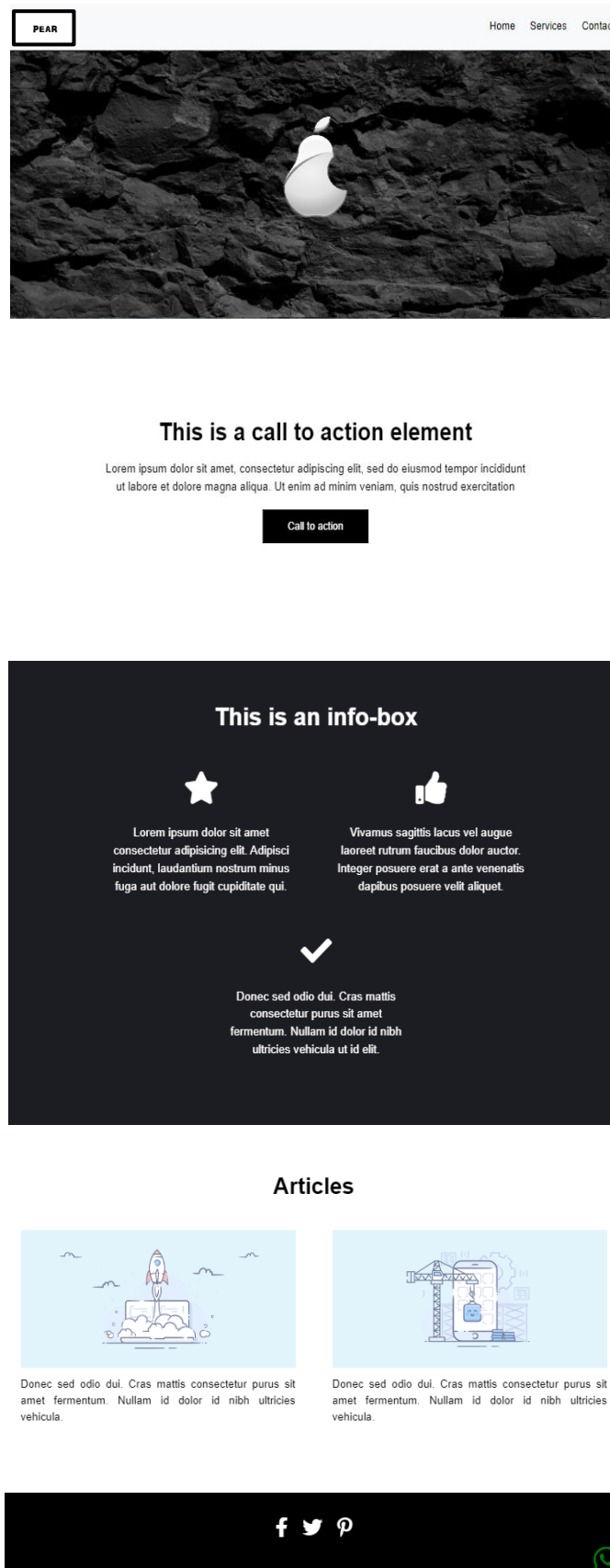
Ancho de pantalla superior a 700px:





Pantallas menores o iguales a 700px:

Todos los contenedores se ven a una columna.





Indicaciones:

- **Selector universal:** margen 0, padding 0, fuente de tipo Roboto
- **El header está compuesto del logotipo y del menú:**
 - **header:** va a estar fijo en la parte alta (position: fixed), para que el menú se posicione en lo más alto le pondremos top:0px. como el resto de elementos deben pasar por debajo del header cuando haces scroll, vamos a ponerle z-index con valor 1, así nos aseguramos que siempre esté encima, el ancho del header es del 100% y el alto de 70px, el color de fondo es #F7F9FA,
 - **nav:** mide como máximo 1024px y está centrado.
 - **logo (header img):** como queremos centrar el logotipo y tenemos su tamaño, podemos calcular fácilmente el padding que debe tener para que quede centrado: padding: 18px 10px.
 - **ul:** flotamos a la derecha el elemento ul.
 - **li:** le quitamos estilo a las listas, padding: 0px, le ponemos display:inline para que tenga en cuenta el ancho y el alto de cada lista.
 - **nav li a:** color negro, fuente 1em, quitar subrayado, como sabemos que el menú mide 70 píxeles se puede centrar poniendo line-height: 70px, como sabemos el tamaño del menú y del texto podemos poner: padding: 26px 10px;, cuando se pasa el puntero por encima cambia de color a #5E5D6E.
- **Vamos a hacer el contenedor del parallax:**

```
.parallax {  
  background-image: url("https://www.imagenes.com/uploads/pear-home-history.jpg");  
  height: 400px;  
  background-attachment: fixed;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-size: cover;  
  position: relative;  
}
```



```
.parallax img{  
  max-width: 400px;  
  position: absolute;  
  left: 50%;  
  top: 50%;  
  transform: translate(-50%, -50%);  
  -webkit-transform: translate(-50%, -50%);  
  overflow: hidden;  
}
```

Como el menú ahora está fijo, ha salido del flujo normal de la página y todos los contenedores que hay a continuación empiezan desde el principio (no se tiene en cuenta la posición del menú). Una solución sencilla es poner al body un padding-top de 70px. Otra solución sería añadir un contenedor que ocupe lo mismo que el menú sin position fixed.

➤ **Vamos a poner el botón de WhatsApp fijo:**

- **#btn-fixed:** le añadimos position:fixed, definimos su posición en la pantalla bottom: 10px y right: 10px
- **.fa-whatsapp:** color verde, fuente 3em.

➤ **Vamos a hacer el call to action:**

- **.call-to-action:** para que quede bonito el diseñador te dice que como máximo el texto que hay dentro ocupará 700px y que tendrá un margin: 150px auto y padding: 0 30px; el texto estará alineado en el centro.
- **.big:** fuente de 1.1em, interlineado de 1.5em.
- **.call-to-action h2, .call-to-action p, .call-to-action button:** margen inferior de 20px.
- **button:** padding superior e inferior de 15px y derecho e izquierdo de 40px, de esta forma es pulsable (se cumple mínimo 44px para que sea pulsable), fondo de color negro, fuente de 1em, color blanco, vamos a ponerle una transición (transition: background 0.5s linear;) esta propiedad necesita prefijo para navegadores (-webkit-transition: background 1s linear;), al pasar el cursor por encima se pone de color #4A4959.



➤ **Vamos a hacer el info-box con cajas flexibles:**

- **info-box:** color de fondo #1C1D22, centramos los textos, color del texto blanco, ponemos un padding para darle aire de 110px tanto inferior como superior.
- **h2:** vamos a conseguir algo de aire poniendo un margen inferior de 30px.
- **#flex-p:** vamos a usar flexbox para maquetar los contenedores: [justify-content: center, display: flex, flex-wrap: wrap.](#)
- **#flex-p p:** le damos un padding de 30 px para darles aire y le ponemos que como máximo ocupen 300px.

➤ **El resto de contenedores se hacen como cada uno quiera manteniendo el diseño lo máximo posible.**

Código HTML

```
<header>
  <nav>
    <a href="index.html"></a>

    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</header>
<div class="parallax">
  
  <h1>PEAR</h1>
</div>
<div class="call-to-action">
  <h2>This is a call to action element</h2>
  <p class="big">Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation</p>
  <button>Call to action</button>
</div>
```



```
<div class="info-box">
<h2>This is an info-box</h2>
  <div id="flex-p">
    <p class="big"><i class="fas fa-star"></i><br>Lorem ipsum dolor sit
    amet consectetur adipisicing elit. Adipisci incidunt, laudantium
    nostrum minus fuga aut dolore fugit cupiditate qui.</p>
    <p class="big"><i class="fas fa-thumbs-up"></i><br>Vivamus sagittis
    lacus vel augue laoreet rutrum faucibus dolor auctor. Integer
    posuere erat a ante venenatis dapibus posuere velit aliquet.</p>
    <p class="big"><i class="fas fa-check"></i><br>Donec sed odio dui. Cras
    mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh
    ultricies vehicula ut id elit.</p>
  </div>
</div>
<div class="description-box">
  <h2>Articles</h2>
  <div class="img-container">
    
    <p class="big">Donec sed odio dui. Cras mattis consectetur purus sit
    amet fermentum. Nullam id dolor id nibh ultricies vehicula.</p>
  </div>
  <div class="img-container">
    
    <p class="big">Donec sed odio dui. Cras mattis consectetur purus
    sit amet fermentum. Nullam id dolor id nibh ultricies vehicula.</p>
  </div>
</div>
<footer>
  <a href="#"><i class="fab fa-facebook-f"></i></a>
  <a href="#"><i class="fab fa-twitter"></i></a>
  <a href="#"><i class="fab fa-pinterest-p"></i></a>
</footer>
<a id="btn-fixed" href="#"><i class="fab fa-whatsapp"></i></a>
```

Código CSS

```
*{
  margin: 0px;
  padding: 0 px;
  font-family: "Roboto", Arial, Tahoma, sans-serif;
}
body { padding-top: 70px; }
h1{color:black; font-size:12px;float:right;font-weight: 10px; font-
  style:italic;}
h2{font-size:36px;}
```



```
/* Ejemplo navbar */
header{
    top: 0px;
    position: fixed;
    width: 100%;
    height: 70px;
    background-color: #F7F9FA;
    z-index: 1;
}
nav{
    max-width: 1024px;
    margin: 0 auto;
}
header img{
    padding: 18px 10px;
}
nav ul{
    float: right;
    list-style: none;
    text-align: center;
    vertical-align: top;
}
nav li{
    list-style-type: none;
    padding: 0px;
    display: inline;
}
nav li a{
    color: black;
    font-size: 1em;
    text-decoration: none;
    line-height: 70px;
    padding: 15px 10px;
}
nav li a:hover{
    color: #5E5D6E;
}
/*Ejemplo Parallax*/
.parallax {
    background-image: url("https://www.imagenes.com/uploads/pear-home-
        history.jpg");

    height: 400px;
    background-attachment: fixed;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}
```



```
.parallax img{
  max-width: 400px;
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  -webkit-transform: translate(-50%, -50%);
  overflow: hidden;
}
/* Ejemplo Whatsapp fijo*/
#btn-fixed {
  display: block;
  position: fixed;
  bottom: 10px;
  right: 10px;
}
/*Ejemplo Call to action*/
.call-to-action{
  max-width: 700px;
  margin: 150px auto;
  padding: 0 30px 0 30px;
  text-align: center;
}
p.big{
  font-size: 1.1em;
  line-height: 1.5em;
}
.call-to-action h2, .call-to-action p, .call-to-action button{
  margin-bottom: 20px;
}
button{
  padding: 15px 40px 15px 40px;
  background-color: black;
  font-size: 1em;
  color: white;
  -webkit-transition: background 1s linear;
  transition: background 0.5s linear;
}
button:hover{
  background-color: #4A4959;
}
.fa-whatsapp{
  color: green;
  font-size: 3em;
}
```



```
/*Ejemplo con cajas flexibles*/
.info-box{
  background-color: #1C1D22;
  margin: 0px auto;
  padding: 110px 0 110px 0;
  color: white;
  text-align: center;
}
.info-box h2{
  margin-bottom: 30px;
}
#flex-p{
  justify-content: center;
  display: flex;
  flex-wrap: wrap;
}
#flex-p p{
  padding: 30px;
  max-width: 300px;
}
.fa-star, .fa-thumbs-up, .fa-check{
  font-size: 2.7em;
  display: block;
  color: white;
  margin-bottom: 30px;
}

/*Ejemplo con floats y porcentajes*/
.description-box{
  max-width: 1024px;
  margin: 0 auto;
  margin-top: 80px;
}
.description-box h2{
  text-align: center;
}
.img-container {
  margin: 20px 0 80px 0;
  box-sizing: border-box;
  -webkit-box-sizing: border-box;
  float: left;
  width: 50%;
  padding: 30px;
}
```



```
.img-container img, .img-container p{
  width: 100%;
  padding-bottom: 10px;
}
.img-container p{
  text-align: justify;
}
footer{
  padding: 30px 0 30px 0;
  text-align: center;
  clear: both;
  background-color: black;
}
.fa-facebook-f, .fa-twitter, .fa-pinterest-p{
  font-size: 2em;
  display: block;
  padding: 10px;
  color: white;
  margin-bottom: 30px;
}

/*Ejemplo Media Queries*/
@media screen and (max-width: 700px) {
  .img-container {
    width: 100%;
    margin: 0;
  }
}
```