**Intro**

Los pies de página (o footers) también son elementos habituales en la mayoría de los sitios web. Se les representa con la etiqueta <footer> y por lo común contienen lo siguiente:

* Autoría e información del copyright.
* Información de contacto.
* Enlaces de navegación y a menudo un enlace "back to top" (regresar a la parte superior), y
* Documentos relacionados o fuentes.

Puedes tener varios elementos <footer> en tu página. Se puede definir el pie de página para todo el documento o por secciones dentro del mismo. Por ejemplo, si tienes un blog, puedes usar las etiquetas <footer> al final de cada publicación e incluso cerrar la página con una también.

**Un pie de página simple - El andamiaje en HTML**

Para empezar, vamos a construir un pie de página muy simple con un aviso de copyright y algunos enlaces de navegación:

Nota: Claro que puedes intentar realizar la actividad por tu cuenta primero y luego verificar la solución. Pero si quieres, también puedes programar conmigo.

Abre un bin nuevo y configura todo como acostumbras.

**CSS reset:**

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.css">

HTML

Copy

**Border-box switch:**

\* {

box-sizing: border-box;

}

CSS

Copy

Si ya terminaste, podemos continuar con el scaffold de HTML.

**El HTML**

Todo empieza con un elemento footer:

<body>

<footer>

</footer>

</body>

HTML

Copy

Ahora agregaré un div .wrapper porque ya sé que quiero utilizar el método de visualización de tabla para resolver el centrado vertical:

<body>

<footer>

<div class="wrapper">

</div>

</footer>

</body>

HTML

Copy

La información del copyright vendrá en un elemento small porque, semánticamente, cuenta como un *small print* (es decir, letra pequeña):

<body>

<footer>

<div class="wrapper">

<small>©2017 <strong>Awesome Company</strong>, All Rights Reserved</small>

</div>

</footer>

</body>

HTML

Copy

Y finalmente, agregaré un elemento nav con tres etiquetas de anclaje:

<body>

<footer>

<div class="wrapper">

<small>©2017 <strong>Awesome Company</strong>, All Rights Reserved</small>

<nav class="footer-nav">

<a href="#">Back to Top</a>

<a href="#">Terms of Use</a>

<a href="#">Privacy</a>

</nav>

</div>

</footer>

</body>

HTML

Copy

Genial, el escaffold de HTML ya está listo; ahora sólo tenemos que estilizarlo con CSS.

**CSS 1.**

Y como siempre, le daremos estilo al componente por pasos:

1. Posiciona el pie de página y su contenido.
2. Agrega estilos tipográficos básicos.
3. Termina con unos estados especiales.

Comencemos.

**Posicionar los elementos**

Para ver los cambios que hagamos, voy a agregar un fondo amarillo simple al pie de página:

footer {

background: yellow;

}

CSS

Copy

Los pies de página pueden ser tan delgados o gruesos como nosotros (o el contenido) queramos, pero en este caso particular, usemos uno de tamaño mediano:

footer {

background: yellow;

height: 8rem;

}

CSS

Copy

Centra horizontalmente el contenido:

footer {

background: yellow;

height: 8rem;

text-align: center;

}

CSS

Copy

Luego verticalmente:

footer {

background: yellow;

display: table;

width: 100vw;

height: 8rem;

text-align: center;

}

.wrapper {

display: table-cell;

vertical-align: middle;

}

CSS

Copy

¡Bien! Por último, cambia el fondo a un patrón sutil y agrega algo de relleno alrededor del contenido:

footer {

background: url('https://orange.codeberryschool.com/content/images/project-assets/simple-footer.png') repeat center;

display: table;

width: 100vw;

height: 8rem;

text-align: center;

padding: 1rem;

}

.wrapper {

display: table-cell;

vertical-align: middle;

}

CSS

Copy

**CSS 2.**

**Tipografía**

Primero, cambia el font-family y font-size de todo el pie de página:

footer {

background: url('http://i.imgur.com/f4jGhSF.png') repeat center;

display: table;

width: 100vw;

height: 8rem;

text-align: center;

padding: 1rem;

font-family: 'Arial', sans-serif;

font-size: .875rem;

}

CSS

Copy

Genial, excepto por el hecho de que en realidad no podemos ver el cambio porque el color negro predeterminado de la letra pequeña se pierde en el fondo. Es hora de cambiar los colores de nuestro texto (y eliminar la decoración predeterminada de los enlaces):

footer small {

color: #ccc;

}

.footer-nav a {

text-decoration: none;

color: #c74f78;

}

CSS

Copy

Las cosas empiezan a verse bien, pero un poco llenas. Algunos márgenes y espacios entre las letras resolverán eso:

footer small {

color: #ccc;

letter-spacing: .025rem;

margin-bottom: 1.5rem;

display: block;

}

.footer-nav a {

text-decoration: none;

color: #c74f78;

margin: 0 .5rem;

display: inline-block;

}

CSS

Copy

+1 consejo: Se necesitan los valores displays porque los elementos small y a son en línea por defecto, por lo tanto no pueden aceptar las propiedades del modelo de cajas como los márgenes.

**Toques finales**

El pie de página está casi listo, sólo quedan los estados especiales y algunos toques finales. Por ejemplo, los enlaces aún no tienen un hover state. Arreglemos eso:

.footer-nav a:hover {

color: #f26896;

transition: color .15s ease-in-out;

}

CSS

Copy

Y si observaste bien, habrás notado que el texto dentro de la etiqueta <strong> no está en negritas. La razón de esto es el CSS reset. Tenemos que restaurar manualmente la apariencia habitual del elemento. Copia esto debajo de tu selector general:

strong {

font-weight: 700;

}

CSS

Copy

Y eso es todo. El nombre de la compañía ya aparece en negrita (como queríamos) y ahora sí hemos terminado con nuestro pie de página. ¡Felicidades!

# PRACTICA 2 – PIE DE PAGINA

## La solución de la posición absoluta

Una manera de fijar el pie en la parte inferior de la página es aprovechando los beneficios de la propiedad position. Este método se compone de 4 pasos:

1. Envuelve todo tu contenido (incluido el pie) en un div.
2. Dirígete al wrapper, establécele una altura mínima de 100vh y configúralo a position: relative;.
3. Dirígete al pie de página, establece su posición a absolute y muévelo al final del wrapper.
4. Dale al wrapper un relleno inferior que iguale la altura del pie de página para que el contenido no se deslice debajo del pie.

Este es el HTML necesario:

<body>

<div class="wrapper">

<main class="content">

<span>content</span>

</main>

<footer class="footer">

<span>footer</span>

</footer>

</div>

</body>

HTML

Copy

CSS:

.wrapper {

min-height: 100vh;

position: relative;

padding-bottom: 50px; /\* Equals height of the footer and pushes away the content in the case they would touch. \*/

}

.footer {

position: absolute;

bottom: 0;

left: 0;

width: 100vw;

height: 50px;

}

CSS

Copy

Ten en cuenta de que debes declarar un ancho para el pie de página después de establecer su posición como absoluta, de lo contrario no rellenará el viewport.

Y puedes ver una demostración en directo aquí (hay algunos colores auxiliares para ayudarte a visualizar los elementos):

/\*==== HELPER STYLES ====\*/

\* {

    box-sizing: border-box;

  }

  body {

    background: #cfcfcf;

  }

  .content, .footer {

      padding: 10px;

      color: rgba(255, 255, 255, .9);

      font-family: monospace;

      font-size: 1.125rem;

  }

.content {

      background: #ae001d;

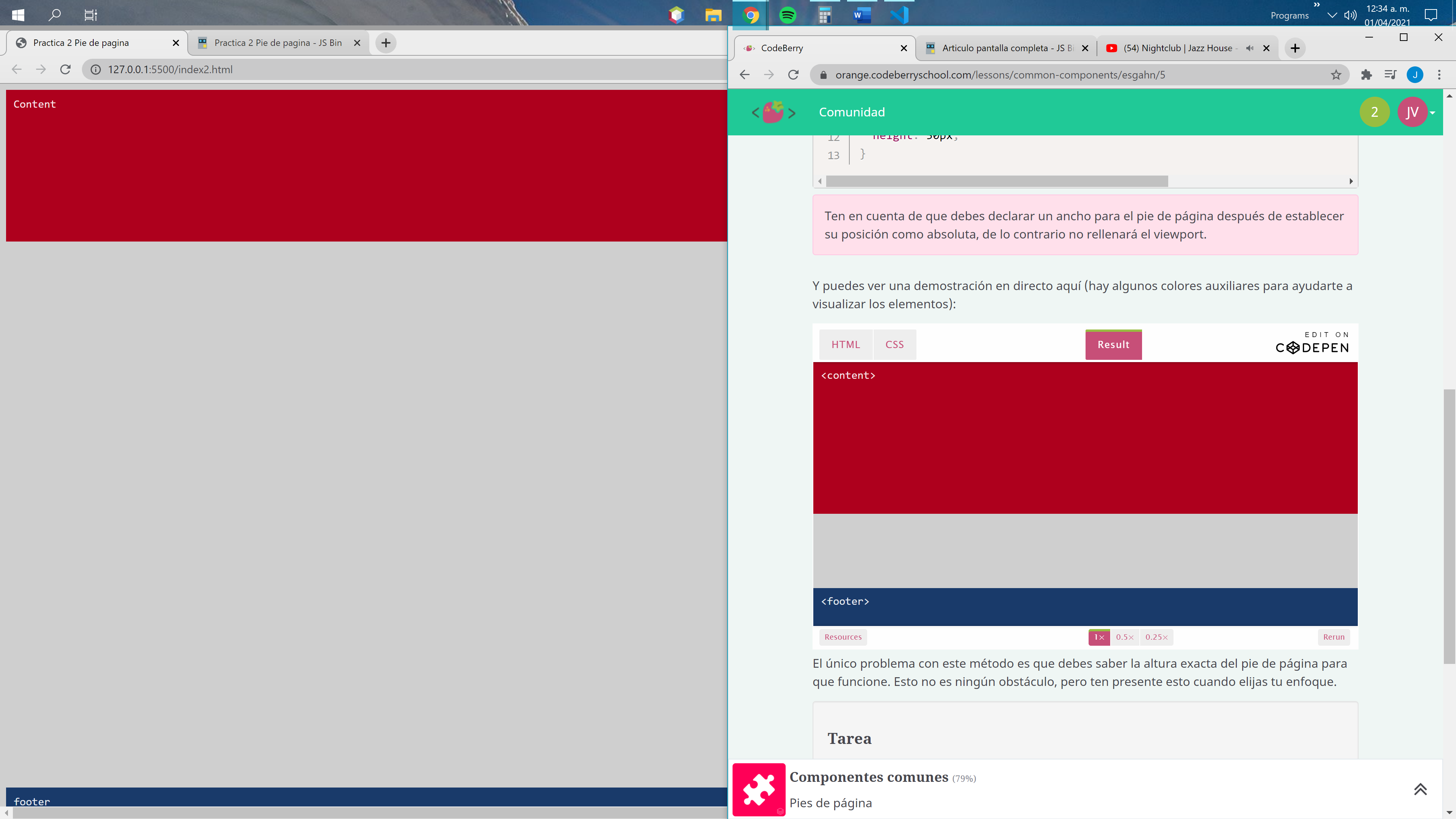
      height: 200px;

}

.footer {

    background: #193a6a;

}



El único problema con este método es que debes saber la altura exacta del pie de página para que funcione. Esto no es ningún obstáculo, pero ten presente esto cuando elijas tu enfoque.

## La solución calc()

La opción B utiliza la función incorporada calc(). Este método se basa en:

* Colocar todo en un contenedor, excepto el pie de página.
* Establecer la altura fija para el pie de página
* Calcular la altura del contenedor del contenido como "altura total del viewport - altura del pie".

Así es como se ve en un código. El HTML:

<body>

<main class="content">

<span>content</span>

</main>

<footer class="footer">

<span>footer</span>

</footer>

</body>

HTML

Copy

El CSS:

.content {

min-height: calc(100vh - 50px);

}

.footer {

height: 50px;

}

Y una demostración en directo con los colores auxiliares:

.content {

    min-height: calc(100vh - 50px);

}

.footer {

    height: 50px;

}

/\*==== HELPER STYLES ====\*/

\* {

    box-sizing: border-box;

  }

  body {

    background: #cfcfcf;

  }

  .content,

  .footer {

    padding: 10px;

    color: rgba(255,255,255, .9);

    font-family: monospace;

    font-size: 1.125rem;

  }

  .content {

    background: #ae001d;

    height: 200px;

  }

  .footer {

    background: #193a6a;

  }

El fundamento de esta solución es la función calc(). Puede calcular de manera dinámica la altura de .content al restar la altura del pie del viewport completo. De esta forma, el .content siempre mandará al pie de página al final del viewport.

Este método tiene la misma limitación que el anterior: sólo funciona si el pie de página tiene una altura fija. Una vez más, no es un obstáculo, pero debes tomarlo en cuenta.

## La solución Bootstrap

Para un ejercicio más completo, incluiré una tercera y última opción, que es una pequeña variación del método A (posicionamiento absoluto). Esta es la solución de ejemplo para el problema de pie de página fijo de Bootstrap framework, por cierto.

El HTML:

<body>

<main class="content">

<span>content</span>

</main>

<footer class="footer">

<span>footer</span>

</footer>

</body>

HTML

Copy

El CSS:

html {

min-height: 100%;

position: relative;

}

body {

margin-bottom: 50px;

}

.footer {

position: absolute;

width: 100%;

bottom: 0;

height: 50px;

}

CSS

Copy

Y una demostración en directo:

La ventaja de este método es que elimina la necesidad de una wrapper div mediante el uso de un elemento html como un contenedor posicionado. De lo contrario, sería lo mismo e igual a la opción A; esto también funciona con los pie de página de altura fija.