

GUIA DE LABORATÓRIO 7.1

LAYOUTS COM FLOATS

(Beta)

OBJECTIVOS

- Explorar algumas alternativas para disposição de elementos, utilizando a propriedade float e propriedades acessórias
- Introduzir alguns elementos semânticos de HTML
- Rever ou aprender a utilizar estilos básicos para tipografia e aprender algumas propriedades e funções de CSS "moderno"

INSTRUÇÕES

PARTE I – ESTILOS BÁSICOS COM CSS

1. Crie o ficheiro lab71_parte1.html e a folha de estilos respectiva, lab71_parte1.css.

2. Crie o esqueleto básico da página:

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <title>Estilos Básicos com CSS</title>
  <link rel="stylesheet" href="lab71_parte1.css">
</head>
<body>
</body>
</html>
```

3. Vamos começar por estabelecer a estrutura da página. A nossa página terá as seguintes "áreas":

- . Cabeçalho

- . Conteúdo principal
- . Uma zona de navegação
- . Uma zona com conteúdos extra que poderão estar ou não relacionados com o conteúdo principal
- . Um rodapé para toda a página

Acrescente o seguinte conteúdo dentro do body:

```
<header></header>
<main></main>
<nav></nav>
<aside></aside>
<footer></footer>
```

Para já, utilizamos apenas elementos semânticos para seccionamento, e não recorremos a classes ou ids. Elementos **semânticos** transmitem significado ao conteúdo, facilitando a sua interpretação a quem lê o código, mas facilitando também a tarefa de motores de busca e a outras ferramentas que analisam a página, como é o caso dos próprios navegadores. Nesta página utilizamos os seguintes elementos:

header: introduz os conteúdos que se seguem; pode conter alguns títulos, um logo, uma barra de pesquisa, etc.

main: a zona principal, contendo os conteúdos principais, isto é, conteúdos relacionados com o tema da página, ou, no caso de um aplicação web, contendo a funcionalidade principal

nav: utilizado para menus, índices, mapas de navegação, etc.

aside: conteúdos extra, não directamente relacionados com os conteúdos em **main**.

footer: um rodapé para o elemento de seccionamento mais próximo (ou para a secção principal, como é o caso do body)

4. Vamos acrescentar algum conteúdo às diversas áreas da página. O header possui apenas o seguinte texto:

```
<header>Aqui está o cabeçalho!</header>
```

5. O main contém:

```
<main>
  <h1>Cá estamos, no <code>main</code>.</h1>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent
    non accumsan orci. Vestibulum at tincidunt enim, sit amet dapibus
    magna. Mauris tincidunt metus sed leo rhoncus, ac convallis dui
    porta. Phasellus consectetur dictum sem a gravida. Praesent
    rhoncus luctus massa in vulputate. Morbi quis nunc nisl. Nulla
    vel quam convallis, molestie eros id, mattis ante. Mauris
    condimentum et turpis eu dignissim. Nam fringilla pharetra
    tempor.
  </p>
</main>
```

6. Vem agora a nav:

```
<nav>
  <h4>Zona de navegação, apenas com texto.</h4>
  <p>Mauris condimentum et turpis eu dignissim. Nam fringilla pharetra
    tempor.</p>
</nav>
```

7. Acrescente agora os "tópicos" abordados na aside:

```
<aside>
  <h4>Uma <code>aside</code> com conteúdo vagamento relacionado.</h4>
  <p>
    Mutate velit esse molestie consequat, vel illum dolore eu feugiat
    nulla facilisis at vero eros et accumsan et iusto odio dignissim
    qui blandit
  </p>
</aside>
```

8. Finalmente, acrescente o conteúdo para o header:

```
<footer>E agora o rodapé</footer>
```

9. Antes de avançarmos, teste no navegador.

10. Vamos agora estilizar a nossa página, centrando a nossa atenção no texto e em aspectos tipográficos. Comece por acrescentar:

```
html {
  font-size: 62.5%;
  font-family: serif;
}
body {
  font-size: 1.8rem;
  line-height: 1.7;
  max-width: 40em;
  margin: auto;
  color: #4a4a4a;
  background-color: #f9f9f9;
  padding: 1.3rem;
}
```

*Relembrando, **em** e **rem** são duas unidades de medida proporcionais. Enquanto que **em** estabelece uma dimensão em função do tamanho de letra definido para o elemento (*), **rem** é calculado em função do tamanho de letra do elemento raiz (elemento **html**).*

*Note-se que o tamanho de letra estabelecido no elemento **html** é de 62,5%. Porquê? Por omissão, o tamanho de letra do texto é fixado em 16px pelos browsers, pelo que 62,5% de 16px dá 10px. A partir daqui sabemos que 0.1rem é 1px, 1.5rem é 15px e 1.8rem é 18px.*

() - Se utilizarmos **em** para definir o próprio tamanho de letra, então, neste caso, este é calculado em função do tamanho de letra do ascendente mais próximo.*

11. Para adaptarmos o tamanho de letra às dimensões do *browser*, acrescente:

```
@media (max-width: 684px) {
  body {
```

```
        font-size: 1.5rem;
    }
}
@media (max-width: 382px) {
    body {
        font-size: 1.4rem;
    }
}
```

12. Teste redimensionando o navegador.

13. Acrescente agora os seguintes estilos:

```
h1, h2, h3, h4, h5, h6 {
    font-family: Verdana, Geneva, sans-serif;
    font-weight: 700;
    line-height: 1.2;
    overflow-wrap: break-word;
    word-wrap: break-word;
    -ms-word-break: break-all;
    word-break: break-word;
    -ms-hyphens: auto;
    -moz-hyphens: auto;
    -webkit-hyphens: auto;
    hyphens: auto;
}
```

Em CSS podemos detalhar alguns aspectos relacionados com a configuração e aplicabilidade de uma folha de estilos utilizando o que se designa por "*@-rules*". Por exemplo, uma folha de estilos pode incluir outra via uma regra *@import*, podemos descrever uma animação com a regra *@keyframes*, e, como é aqui o caso, podemos descrever um conjunto de regras que só devem ser aplicadas se se verificarem determinadas características do dispositivo de exibição por via da regra *@media*.

Estas regras, também designadas de "*media queries*" (consultas sobre o dispositivo), são utilizadas para oferecer estilos mais adaptados ao dispositivo onde a página serão exibidas. São muito utilizadas para fazer regras à medida de dispositivos como telefones, tablets e computadores. As "*media queries*" estão no centro do desenho responsivo, uma filosofia de desenho de páginas que será abordada nos laboratórios seguintes.

Algumas propriedades e valores a destacar:

. *overflow-wrap*: indica se o browser pode quebrar linhas "dentro" de palavras de modo a evitar que uma linha demasiado longa saia para fora (ou, tentando traduzir *overflow*, "transborde/derrame para fora") da caixa que a contém; o valor *break-word* indica que a linha poder ser quebrada dentro da palavra

. *word-wrap*: renomeada para *overflow-wrap* desde CSS3; presente na regra para que esta seja compatível com browsers mais antigos

. *word-break*: como uma palavra deve ser quebrada em caso de mudança de linha; o valor *break-word* indica que deve ser quebrada em qualquer ponto, de acordo com o algoritmo de formatação de texto utilizado pelo browser.

. *hyphens*: se coloca hifens ou não na quebra de palavra

Consultar: <https://stackoverflow.com/questions/1795109/what-is-the-difference-between-word-break-break-all-versus-word-wrap-break>

~~14. Após o header, acrescente o seguinte parágrafo "temporário".~~

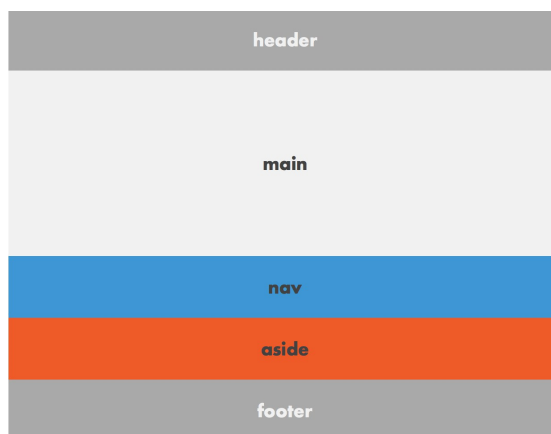
```
<p style="border: 1px solid; width: 100px; overflow-wrap: break-word;
hyphens:auto;">
    Bom dia inconstitucionalissimamente
</p>
```

~~15. Altere os valores das propriedades `word-break` e `overflow-wrap`.~~

~~16. Remova o parágrafo e avance para a parte seguinte~~

PARTE II – REDEFINIÇÕES E SOBREPOSIÇÕES

17. Vamos destacar as secções da nossa página de modo a ficarem semelhantes ao modelo em baixo:



NOTA: Nesta imagem e nas seguintes os tamanhos não obedecem a qualquer escala, e as respectivas proporções podem não corresponder às dimensões em página. Isto é especialmente notório no que à altura (height) diz respeito.

18. Grave `lab71_parte1.html` e `lab71_parte1.css` como `...parte2...`. Ajuste a referência para a folha de estilos em `lab71_parte2.html`.

19. Apague o conteúdo de `lab71_parte2.css` e depois acrescente:

```
@import url('lab71_parte1.css');
```

20. Modifique o título da página (`head` → `title`) de modo a ser igual ao desta parte do laboratório.

21. Vamos agora anular algumas das definições anteriores, sobrepondo as que se seguem. Após o `@import`, acrescente as seguinte propriedades a esta folha de estilos:

```
body {  
    margin: 0;  
    padding: 0;  
    background: #fff;  
    color: #4a4a4a;  
}  
  
header, footer {  
    font-size: large;
```

Vamos agora dispor (layout) as secções da página utilizando para tal CSS "clássico", ou seja, utilizando mecanismos do CSS que inicialmente não foram previstos para definir layouts mas que, em todo o caso, foram (e ainda são) abundantemente utilizados para tal. Acrescentámos algumas cores apenas para destacar as secções.

```
        text-align: center;
        padding: 0.3em 0;
        background-color: #795959;
        color: #f9f9f9;
    }
    nav {
        background: #eee;
    }
    main {
        background: #f9f9f9;
    }
    aside {
        background: #eee;
    }
}
```

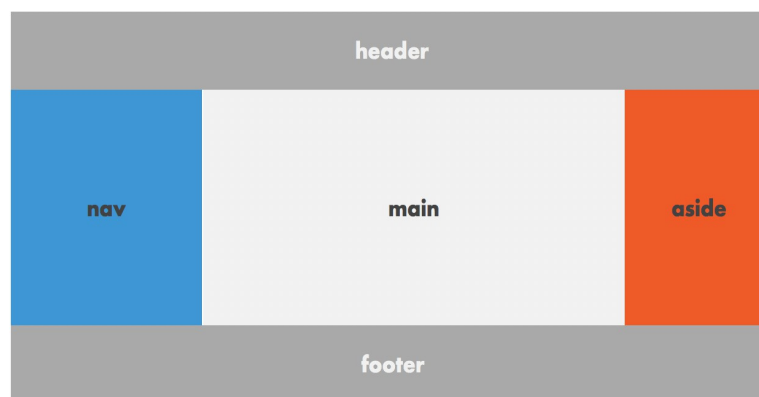
22. Após testar, acrescente a seguinte propriedade ao `body`:

```
max-width: inherit;
```

23. O que sucedeu?

PARTE III – FLOATS

24. Vamos organizar as secções entre *header* e *footer* em colunas. No final desta parte, o layout da nossa página vai evoluir para o modelo da figura seguinte:



25. Grave `lab71_parte2.html` e `lab71_parte2.css` como `...parte3...`. Ajuste a referência para a folha de estilos em `lab71_parte3.html`.

26. Apague o conteúdo de `lab71_parte3.css` e depois acrescente:

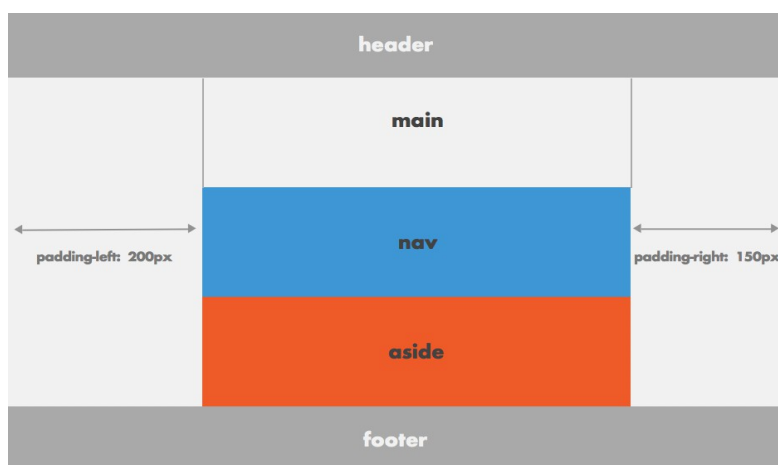
```
@import url('lab71_parte2.css');
```

27. Modifique o título da página de modo a ser igual ao desta parte do laboratório.

28. Comece por acrescentar as seguintes regras à folha de estilos:

```
body {  
    padding-left: 200px;  
    padding-right: 150px;  
    min-width: 240px;  
}  
header, footer {  
    margin-left: -200px;  
    margin-right: -150px;  
}
```

Agora a aparência do layout agora deve corresponder ao seguinte modelo:



29. Pretendemos que a *nav* ocupe a coluna da esquerda e a *aside* a coluna da direita. Vamos obter esse efeito com a propriedade *float*, utilizando margens negativas e posicionamento relativo. Para já, comece por acrescentar:

```
nav {  
    width: 200px;  
}  
aside {  
    width: 150px;  
}
```

30. Após testar, acrescente a seguinte regra:

```
main, nav, aside {  
    float: left;  
}
```

*A propriedade **float** foi concebida para misturar objectos, como imagens e tabelas, com texto, fazendo com que este texto circule à esquerda ou à direita desses objectos. Não foi pensada para ser utilizada para organizar os elementos numa página. Porém, é isso mesmo que tem sido feito, pelo menos até terem surgido alternativas mais sofisticadas para layout de elementos, (...)*

31. E agora acrescente também:

```
footer {  
    clear: both;  
}
```

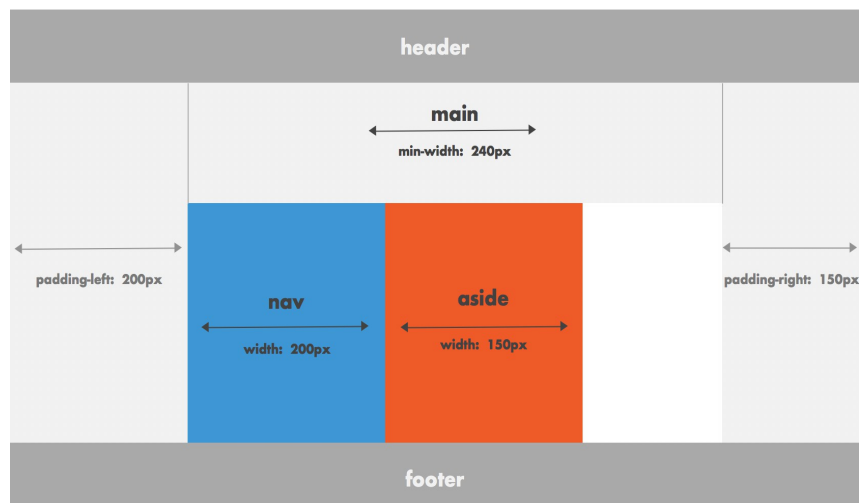
Esta regra deve ficar como sendo a última regra da folha de estilos.

(...) como o *flexbox* e o *grid* (ver labs seguintes). Como esta propriedade permite "flutuar" qualquer elemento, e não apenas imagens ou tabelas, desde cedo que foi aproveitada para dispor elementos de bloco em linhas e colunas.

Devido ao *float*, as secções *main*, *nav* e *aside* tendem a alinhar-se lado a lado, porém, como a coluna central - ie, o *main* - ocupa toda a área de conteúdos do *body*, a *nav* e a *aside* não têm espaço para se alinharem com o *main* e deslizam para a linha de baixo.

O *footer* não é afectado pelo *float* das secções anteriores devido à propriedade *clear*. Esta anula o efeito do *float*, evitando que o *footer* suba para ocupar o espaço livre à direita da *aside*.

32. A imagem seguinte esquematiza a estrutura da página:

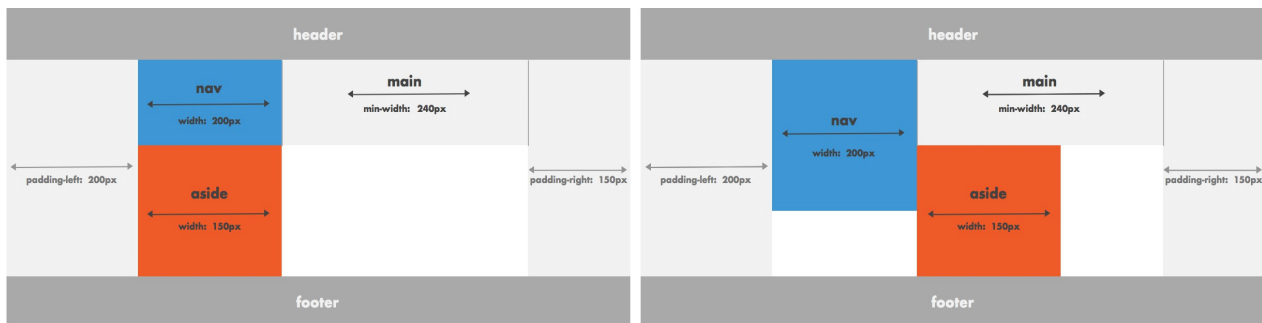


33. Vamos fazer com que a *nav* ocupe a coluna da esquerda. Acrescente a seguintes propriedade:

```
nav {  
    margin-left: -100%;  
}
```

Deverá ter-se deparado com um dos seguintes dois cenários:

De salientar que os 100% correspondem à dimensão da área de conteúdos do *body*, que é também a dimensão da coluna central, ou seja, do *main*. Com isto, a *nav* vai sobrepor-se ao *main*, isto é, vai ficar por cima do *main*. Se a altura da *nav* for inferior à do *main*, a *aside* fica encostada à esquerda, caso contrário permanece no mesmo local.

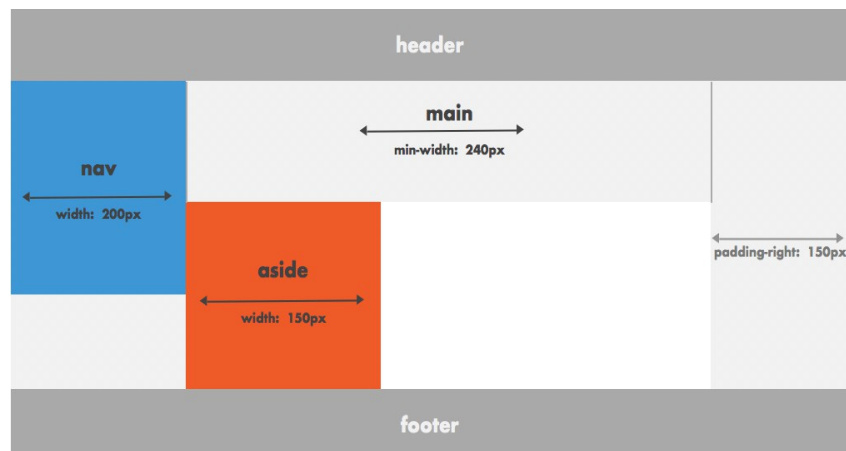


34. Para que a *nav* vá para o local esperado, vamos deslocá-la 200px para a esquerda com posicionamento relativo e a propriedade *right* (ou *left* e valor negativo):

```
nav {
  position: relative;
  right: 200px; /* ou left: -200px; */
}
```

Deverá então obter:

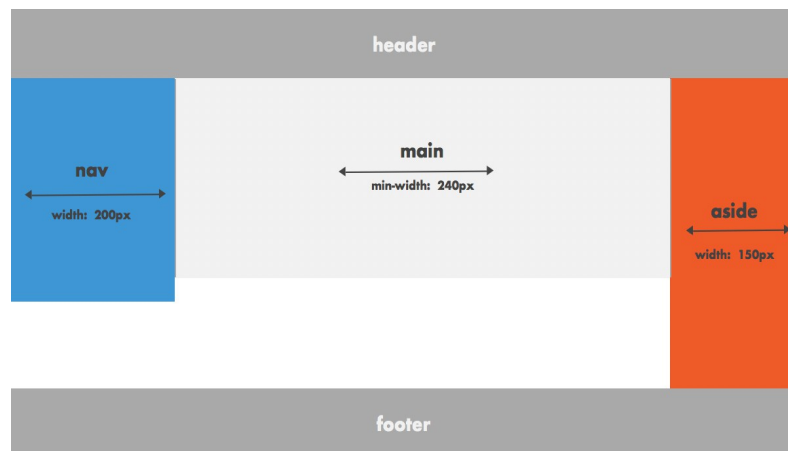
*Com posicionamento relativo, as propriedades **top**, **right**, **bottom** e **left** permitem especificar deslocamentos face à posição onde o elemento seria colocado com posicionamento estático. Neste caso damos um deslocamento à direita de 200px, precisamente a dimensão da coluna esquerda.*



35. E agora vamos "puxar" a *aside* para o destino pretendido com uma margem negativa:

```
aside {
  margin-right: -150px;
}
```

Por esta altura, à excepção da altura das três secções, devemos ter algo aproximado ao modelo da primeira imagem:



- 36.** Vamos agora afastar o texto das margens de cada secção. Começemos por dar 10px de *padding* à esquerda e 10px à direita à *nav*, ou seja, um total de 20px na horizontal, total esse que vamos subtrair ao *width* de modo a que a largura total da *nav* se mantenha. Procedemos da mesma forma para a *aside*:

```
nav {
    width: 180px;
    padding: 0 10px;
}

aside {
    width: 130px;
    padding: 0 10px;
}
```

- 37.** E agora para o *main* damos *padding* horizontal total de 40px:

```
main {
    padding: 0 20px;
}
```

De salientar que a área de conteúdos do main não tem largura definida (ou seja, a propriedade width não tem um valor definido). Ou seja, o main vai ocupar toda a área de conteúdos do body. E qual a largura da área de conteúdos do body? A largura da janela (html) menos padding-left + padding-right. Ou seja, a área de conteúdos do main adapta-se à largura que "sobra" após tirarmos o padding horizontal do body e o próprio padding horizontal do main. Daí, não precisarmos de ajustar o width do main após modificarmos acrescentarmos um padding de 40px.

De salientar, ainda, que se der 100% de width ao main, estes 100% referem-se à largura do body (excluindo o padding deste), mas incluindo o valor de padding horizontal que o main tem neste momento. Ou seja, o main iria "expandir" para abranger o padding. Neste caso teria que aumentar em valor absoluto, quer o deslocamento à direita da nav, quer a margem à direita da aside, ambos em 40px. Este valor corresponde ao total do padding horizontal do main.

PARTE IV – VARIÁVEIS PARA UMA FOLHA DE ESTILOS MAIS FÁCIL DE MANTER...

- 38.** Grave lab71_parte3.html e lab71_parte3.css como ...parte4... . Ajuste a referência para a folha de estilos em lab71_parte4.html.
- 39.** Ao contrário das partes anteriores, não apague o conteúdo de lab71_parte4.css. Ou seja, mantenha todas regras de lab71_parte3.css; em particular, lab71_parte4.css continua a importar a folha de estilos criada na parte 2.
- 40.** Antes de avançar, investigue as funções CSS *var* e *calc* consultando, por exemplo, as seguintes ligações:
- . <https://developer.mozilla.org/en-US/docs/Web/CSS/calc>
 - . https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables
 - . a propósito da função *var*, consulte <https://developer.mozilla.org/en-US/docs/Web/CSS/:root>
- 41.** Agora modifique a folha de estilos para a seguinte:

```
@import url('lab71_parte2.css');

:root {
  --left-col-width: 200px;
  --right-col-width: 150px;
  --main-horiz-padding: 40px;
  --left-col-horiz-padding: 20px;
  --right-col-horiz-padding: var(--left-col-horiz-padding);
}

body {
  padding-left: var(--left-col-width);
  padding-right: var(--right-col-width);
  min-width: 240px;
}

header, footer {
  margin-left: calc(0px - var(--left-col-width));
  margin-right: calc(0px - var(--right-col-width));
}

main, nav, aside {
  float: left;
}

main {
  padding: 0 calc(var(--main-horiz-padding) / 2);
```

```
}
nav {
  width: calc(var(--left-col-width) - var(--left-col-horiz-padding));
  margin-left: -100%;
  position: relative;
  right: var(--left-col-width);
  padding: 0 calc(var(--left-col-horiz-padding) / 2);
}
aside {
  width: calc(var(--right-col-width) - var(--right-col-horiz-padding));
  margin-right: calc(0px - var(--right-col-width));
  padding: 0 calc(var(--right-col-horiz-padding) / 2);
}
footer {
  clear: both;
}
```

- 42.** Analise e reflecta sobre estas alterações (e, talvez, medite um pouco sobre isto...)
- 43.** Altere as dimensões da coluna esquerda para 250px, da direita para 180px, do padding das colunas esquerda e direita para 30px, e teste.