

# Autoria Web

Aula 08 - Layout CSS, arquitetura da informação  
e conteúdo para web

# Apresentação

---

Nesta aula, você aprenderá a desenhar o layout das páginas, tornando-as parecidas com aquelas que você viu no site ZENGarden. Aliás, com imaginação e criatividade você poderá criar coisas muito mais interessantes.

Lembre-se de que nesta aula você precisará dos arquivos (X)HTML e CSS desenvolvidos nas aulas anteriores para exercitar os assuntos estudados.



## **Vídeo 01** - Apresentação

## Objetivos

Ao final desta aula, você deverá ser capaz de:

- Especificar layouts CSS.
- Explicar os tipos de layout.
- Listar os passos para criação do layout.
- Especificar layouts para impressão.

# Layout CSS

Um layout CSS especifica a estrutura geral da aparência de um site. Nele define-se o posicionamento dos conteúdos, tais como slogan, menus, conteúdo principal, conteúdo secundário, dentre outros.

Para entender melhor, veja a **Figura 1**. Ela ilustra um exemplo de layout onde a página é dividida em cinco partes (ou containers): a primeira é nomeada de header; a segunda, mainnav; a terceira, contents; a quarta, menu; e a quinta, footer.

**Figura 01** - Exemplo de layout



Fonte: <http://www.maxdesign.com.au/> Acesso em: 05 out. 2015.

Antes de colocar a mão na massa, deve-se conhecer os tipos de layouts. Os layouts são classificados segundo dois critérios (SILVA, 2008):

1. **Quantidade de colunas** – pode ser uma, duas, três ou até mesmo mais colunas.
2. **Largura da página** – pode ser de largura fixa, líquido, elástico ou híbrido. Veja a seguir como cada um deles é definido:

- **Largura fixa** – a página e suas colunas possuem uma largura fixa, não importando o tamanho da janela do browser nem a resolução do monitor. Normalmente essa largura é definida pela propriedade width definida em pixel. Esse tipo de layout tem a vantagem de que todos os elementos estarão dispostos de uma forma previsível. Porém, se o usuário estiver usando uma janela menor (ou monitor com resolução menor) do que o necessário para ver todo o conteúdo, ele terá que ficar rolando a barra de navegação para direita e esquerda, ou se ele estiver usando uma janela muito grande (ou monitor com resolução maior do que o planejado) verá muito espaço em branco, como se não tivesse havido um bom planejamento no uso dos espaços. Por exemplo, o site da W3Schools <<http://www.w3schools.com/>> tem largura fixa.
- **Líquido (ou fluido)** – as larguras são definidas em porcentagem. Assim, o conteúdo se acomoda à largura da janela do browser (ou resolução do monitor), otimizando o aproveitamento do espaço. Um exemplo é o site da W3C <<http://www.w3.org/>> (porém, ele muda a forma de visualização a partir de uma certa largura da janela. Faça o teste!). Entretanto, a desvantagem desse tipo de layout é que se deve ter mais cuidado no posicionamento dos elementos para que quando estreitado um não sobreponha o outro. Além disso, para o usuário, o comprimento das linhas de texto pode ficar tão pequeno ou tão grande que a leitura se torna desagradável. Para resolver esses problemas, é possível estipular uma largura máxima e mínima para esses elementos, de forma a garantir certo conforto para o usuário. Dessa forma, o layout se torna líquido até um certo limite.
- **Elástico** – as larguras são definidas com base no tamanho das fontes, usando a unidade “em”. Isso fará com que a largura da página se comporte da mesma forma que o tamanho do texto, aumentando quando a letra aumenta e diminuindo quando a letra diminui. Esse tipo de layout é particularmente útil por questões de acessibilidade, para pessoas com necessidades especiais de visão. A desvantagem desse tipo de layout é que nem sempre se tem um bom aproveitamento do espaço horizontal ou a

possibilidade do aparecimento da barra de rolagem horizontal quando o tamanho das fontes cresce muito.

- **Híbrido** – a largura da página, colunas e demais elementos são definidos em “ems” e adicionalmente uma largura máxima é definida para eles em porcentagem. Ou definem-se as larguras em porcentagem e as larguras máximas e mínimas em “em”.



### Vídeo 02 - Layout

A seguir, são listados alguns passos básicos para você conseguir definir o layout de um site (Max Design e Layout CSS, 2012, extraído da internet):

1. Escolher e nomear os containers do layout.
2. Criar a marcação para esses containers.
3. Escolher o método de posicionamento para os containers.
4. Inserir e estilizar conteúdos.
5. Ligar o CSS.
6. CSS para impressão.

A seguir, é explicado como realizar cada um dos passos citados anteriormente.

## Passo 1 - Escolher e Nomear os Containers do Layout

---

Um container é uma caixa que geralmente agrupa vários elementos (X)HTML de forma a estruturar o conteúdo da página uniformemente. Por exemplo, na **Figura 1**, um dos containers é nomeado header e outro menu, ou seja, essas partes da página agrupam, respectivamente, o conteúdo referente ao cabeçalho do site, tal como o nome e slogan, e os itens pelos quais o usuário pode navegar, respectivamente.

Na web, há muitos layouts disponíveis nos quais você pode se inspirar para começar a criar os seus próprios. A **Figura 2** ilustra alguns exemplos.

**Figura 02** - Exemplos de layouts



Fonte: <http://www.mycelly.com/> Acesso em: 05 out. 2015.

Para definir os containers, você pode se basear em uma figura ou um esboço no papel que divida o conteúdo do site em grandes partes que serão representadas por áreas específicas no documento (X)HTML.

Essa divisão do conteúdo deve levar em conta a arquitetura da informação, mas por enquanto vamos focar apenas a especificação de layouts. Depois disso, ainda nesta aula, você aprenderá sobre arquitetura da informação.

Cada container deve ser nomeado. Os nomes dados aos containers devem ser associados à finalidade deles e não à aparência que eles devem ter. Então, para o exemplo da **Figura 1**, os nomes escolhidos foram:

- Container
- Header
- Mainnav

- Menu
- Contents
- Footer

## Passo 2 - Criar a Marcação para esses Containers

---

Os containers são marcados com a tag `<div>`. Como foi visto na **Aula 2** deste curso, a tag `<div>` serve para agrupar elementos e, juntamente com os atributos `id` e `class`, possibilita definir a estrutura de um documento.

Como todos os containers devem ser marcados com a mesma tag `<div>`, então é necessário nomeá-los diferentemente para que possamos aplicar diferentes estilos a cada um deles. Assim, temos duas opções: criar diferentes classes através do atributo `class`, ou criar diferentes identificadores através do atributo `id`.

Usualmente, cada container aparece uma única vez no documento, assim, se este for o caso, você deve usar o atributo `id` para nomear o container. Caso contrário, você pode nomeá-los usando o atributo `class`.

Para criar a estrutura definida na **Figura 1**, devemos iniciar a marcação dos elementos da seguinte maneira: primeiramente identificamos o container mais externo, no exemplo da **Figura 1** é o container de nome "container". Esse será o primeiro elemento `<div>` a ser definido no (X)HTML e dentro dele definiremos os demais containers identificados no passo anterior. Assim, a definição dos containers é estabelecida de acordo com o mostrado no (X)HTML a seguir.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2 "http://www.w3.org/TR/html4/strict.dtd">
3 <html>
4 <head>
5   <title>Exemplo de CSS Layout</title>
6 </head>
7 <body>
8 <div id="container">
9   <div id="header"></div>
10  <div id="mainnav"></div>
11  <div id="menu"></div>
12  <div id="contents"></div>
13  <div id="footer"></div>
14 </div>
15 </body>
16 </html>
```

Observe que para criar um container dentro do outro basta especificá-lo entre as tags <div> do container pai, ou seja, definir elementos <div> hierarquicamente.



### **Vídeo 03** - Estilo Aninhado

## Atividade 01

---

1. Explique com suas palavras quais os tipos de layout e as características de cada um.
2. Quais os passos básicos para definição de layouts?
3. O que são containers?
4. Explique com suas palavras como containers são especificados e cite um exemplo.



## Passo 3 - Escolher e Especificar o Método de Posicionamento para os Containers

---



### Vídeo 04 - Layout Cores

Analisando a estrutura escolhida para o layout, você deve decidir qual será o método de posicionamento de cada container. Por exemplo:

- O layout deve ter largura fixa ou variar de acordo com o tamanho da janela do browser?
- O layout será centralizado na página?
- Quais containers deverão ser fixos ou mantidos no fluxo normal do (X)HTML?
- Quais deles terão posicionamento absoluto?

Esta escolha não é fácil, e você certamente fará escolhas mais acertadas à medida que adquirir experiência.

Inicialmente, para começar um novo layout, defina o código CSS no próprio (X)HTML, na tag `<style>`. Isso evitará que você manipule mais de um arquivo na sua primeira página de layout. Este código CSS poderá ser posteriormente separado em seu arquivo próprio para poder ser aplicado à várias páginas de seu website.

O exemplo da **Figura 1** será utilizado para ilustrar a definição do posicionamento de cada container. Vamos começar?

Analisando a **Figura 1** pode-se perceber algumas características gerais de cada container. Vamos realçar cada um deles com uma cor de plano de fundo para termos ideias de como posicioná-los.

- **container** – abrange a tela inteira. Ele está posicionado a uma distância bem pequena do topo do browser, e como na figura não dá para perceber se ele está centralizado ou a uma certa margem esquerda e direita, podemos considerar que ele está centrado na página. Assim, podemos definir uma margem superior de 10px e o restante das margens deixarmos com valor auto, que fará com que esse container fique horizontalmente centralizado na página. Além disso, é bastante comum estabelecer um tamanho mínimo para esse container. Dessa forma, se a janela do browser for menor que o necessário para dispor os elementos, estes não serão sobrepostos uns sobre os outros para poder caber na janela. Com isso, podemos definir o seguinte CSS para esse container:

```
1 #container{
2   margin: 10px auto;
3   width: 650px;
4   background: #6E934A;
5 }
```

- **header** – está posicionado dentro de container, acima de todos os outros. Ele está a uma margem também pequena do topo do container e compreende toda a largura de seu pai. Assim, podemos definir o seguinte CSS para header:

```
1 #header {
2   background: #8074CB;
3   height: 50px;
4 }
```

- **mainnav** – está posicionado abaixo de header e também compreende toda a largura de container. Aplicando uma cor de plano de fundo para ele, tem-se:

```
1 #mainnav{
2   background: #4D4197;
3   height: 30px;
4 }
```

- **menu** – está posicionado à direita de contents. Como vimos em aulas anteriores, para posicionar um elemento do lado do outro se pode utilizar a propriedade float:right;. Nesse caso, usando essa propriedade, o próximo elemento, que é o container contents, ficará do lado esquerdo de menu. É necessário também estipular uma largura para menu. Dado que especificamos uma largura total de

650px para container, colocaremos uma largura de 200px para o menu, sobrando 450 para contents. Assim, o CSS para posicionar menu é:

```
1 #menu {  
2   background: #DBBD4B;  
3   width:200px;  
4   float:right;  
5   height: 500px;  
6 }
```

- **contents** – como o div#menu foi posicionado com a propriedade float:right, automaticamente posiciona-se contents do lado esquerdo dele. Assim, podemos aplicar somente uma cor de fundo para ver onde ele está posicionado.

```
1 #contents {  
2   background: #8074CB;  
3   height:500px;  
4 }
```

- **footer** – é posicionado abaixo de contents e menu. Abrange toda a largura de container.

```
1 #footer {  
2   background: #4D4197;  
3   height: 30px;  
4 }
```

---

Além dessas regras, foram definidas alturas para os containers. Com isso, tem-se uma ideia melhor do posicionamento deles. A **Figura 3**, a seguir, ilustra o resultado dessas regras CSS. Veja também o código relacionado.

**Figura 3** Aparência do XHTML descrita anteriormente

Com isso, está definida a estrutura geral do site.

Agora é necessário determinar qual a largura e altura real de cada um dos containers, bem como suas margens e bordas.

Lembre-se de que é importante, particularmente quando você está construindo um layout totalmente novo, que você teste em diversos navegadores sempre que terminar uma etapa, pois o comportamento do CSS nem sempre é o mesmo nos

diferentes browsers. Isso evita ter que retornar a etapas já concluídas quando um problema ocorrer.

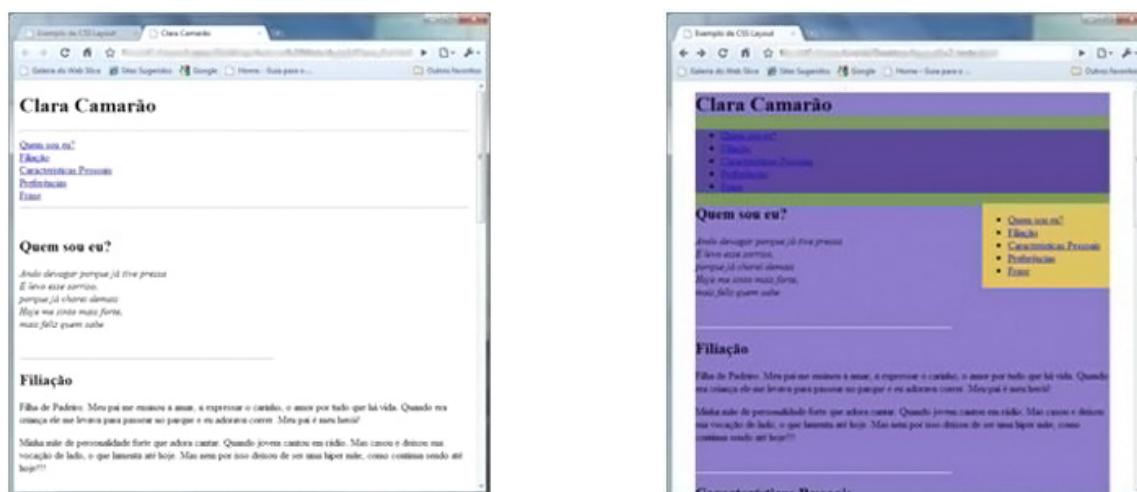
O comportamento de cada container pode variar quando os conteúdos forem inseridos neles. A técnica de colorir os containers permite que você trate e analise primeiramente o posicionamento de cada um deles, deixando as questões de conteúdos para a próxima etapa.

## Passo 4 - Inserir e Estilizar Conteúdos

Terminada a etapa de posicionamento, pode-se começar a inserção de conteúdo na página. Lembre-se de que o conteúdo deve ser marcado adequadamente de acordo com sua semântica, ou seja, usar as tags h1-h6 para títulos, p para textos, ul para listas e assim sucessivamente.

Inseridos os textos, o layout deverá ser testado novamente em diferentes navegadores.

A **Figura 4** (lado direito) ilustra o layout definido anteriormente com o conteúdo da página Clara Camarão, ilustrada na **Figura 4** (lado esquerdo), que você viu nas aulas sobre HTML (foram retiradas as propriedades referentes à altura dos containers para a adequação deles ao tamanho do conteúdo).



**Figura 4** - Ilustração do documento Clara Camarão com e sem o layout da Figura 2

Se você comparar com a primeira versão dessa página, irá perceber que embora inacabada, ela é mais atrativa que a página da **Figura 4** (lado esquerdo). E ainda falta estilizar cada um dos elementos que aparecem nela, tais como títulos, parágrafos, menus.

Observe também que o conteúdo do container contents fica ao redor do menu, e não como uma coluna no documento inteiro. Se você desejar criar colunas realmente separadas para contents e menu, será necessário aplicar as propriedades `float:left;` e `width:450px;` em contents, e também usar a propriedade `clear:both;` em footer.

Então, mãos à obra! Estilizando alguns elementos e inserindo aqueles menus definidos na aula passada nesse documento, pode-se ter o seguinte resultado:

**Figura 5** Site Clara Camarão estilizado

## Diretrizes para Melhorar a Organização do CSS

---

Para estilizar o documento, você deve escrever as regras de forma que você consiga facilmente modificá-las posteriormente.

Você pode fazer isso comentando o código CSS e juntando definições de elementos que têm a mesma aparência. Veja um exemplo na **Figura 6**.

Tente usar o mínimo de ids e classes possíveis e faça a estilização dos elementos usando seletores descendentes. Por exemplo, faça primeiro a estilização para o container como um todo; para listas e depois para itens de lista; para links e depois para cada um dos pseudoatributos do link, ou seja, primeiro os elementos mais gerais e depois os mais específicos.

**Figura 06** - Exemplo de organização do CSS

```
/* container header */
#header {}
/* listas do container header */
#header ul {}
#header ul li {}
...
/* links do container header */
#header a {}
#header li a:link {}
...
```



**Vídeo 05** - Multiplos Arquivos css

## Cuidados com o Tamanho da Fonte Usada



**Vídeo 06** - Layout Menu

Ao estilizar o texto do site, tenha cuidado ao definir o tamanho das fontes. Embora usar fontes pequenas seja muito atrativo, pois você conseguirá inserir mais conteúdo em menos espaço, isso pode resultar em deixar de atrair um determinado público devido ao tamanho da fonte usada. Assim, opte por uma das duas opções a seguir:

**1.** Não defina tamanho para fontes – dessa forma, a folha de estilo do browser ou do usuário determinará o tamanho das fontes.

**2.** Use unidades relativas para definir o tamanho das fontes – use porcentagem ou unidade em “[em](#) (A unidade de medida de comprimento 'em' referencia-se ao tamanho da fonte (letra) do seletor onde for declarada. Quando 'em' for declarada para a propriedade font-size, referencia-se ao tamanho da fonte (letra) do elemento pai. Quando 'em' for declarada para o elemento raiz do documento (p. ex: em

documentos html) referencia-se ao valor inicial (default) do tamanho de fonte (letra).” para definir o tamanho das fontes. Dessa forma, o tamanho da fonte poderá ser alterado no browser de acordo com a preferência do usuário, podendo se ajustar às necessidades de cada um. Tamanhos relativos para fontes são fáceis de implementar quando usamos um ou uma combinação dos três métodos a seguir descritos:

- Aplicar font-size aos containers – Os tamanhos de fontes podem ser definidos de forma global para cada container ou individualmente para os elementos neles contidos. Isso resulta em um alto grau de controle sobre os elementos no layout. Por exemplo: diferentes elementos <h1> podem ter diferentes tamanhos de fontes, dependendo do seu container. Assim, pode-se definir um tamanho diferente para cada container da seguinte maneira:

```
1 #header h1 {
2   font-size: 200%;
3 }
4 #content h1 {
5   font-size: 100%;
6 }
```

- Aplicar font-size ao body– Um tamanho de fonte relativo pode ser aplicado ao elemento body, fazendo com que todos os elementos usem aquele mesmo tamanho como base. No exemplo mostrado na **Figura 5**, definimos o tamanho da fonte em 80% no elemento body. Isso faz com que esse valor seja usado em todo o site, exceto nos elementos mais específicos em que for definido outro valor.

```
1 body {
2   font-size: 80%;
3 }
```

- Aplicar font-size a cada elemento individualmente – Um tamanho relativo de fonte pode ser aplicado diretamente nos elementos HTML. Por exemplo:

```
1 p {
2   font-size: 85%;
3 }
4 h1 {
5   font-size: 150%;
6 }
```

Contudo, ao aplicar essa última regra a elementos mais externos e também a elementos mais específicos, tem-se como resultado tamanhos de fontes proporcionais ao contexto. Por exemplo, definir font-size: 80% em body; como body está no contexto do browser, o resultado é 80% do tamanho definido pelo browser, e definir font-size: 200% em h1 do div header resulta num tamanho que é 160% do tamanho original do browser, ou seja, como o contexto é o elemento body, então tem-se 200% de 80% = 160%. Um exemplo muito comum são listas aninhadas uma dentro da outra. Se a lista for definida com um tamanho de 85%, obtém-se um tamanho que é 85% de 85% definido para a lista pai. Nesse caso, a lista filha será renderizada com uma fonte que será 72,25%.

```
1 ul {  
2   font-size: 85%;  
3 }
```

Caso se deseje ter o mesmo tamanho de fonte para ambas as listas, deve-se adicionar a seguinte regra:

```
1 ul ul {  
2   font-size: 100%;  
3 }
```

## Passo 5 - Ligar o CSS

---

Ao terminar a estilização e o teste em diferentes navegadores, transfira o CSS definido no próprio (X)HTML, no elemento head, para um arquivo CSS externo e ligue-o ao (X)HTML. Dessa forma, os vários arquivos de seu site poderão usar o mesmo CSS, sem que você tenha que replicá-los integralmente em cada arquivo. Você aprendeu a fazer essa ligação na Aula 4 desta disciplina.

Em resumo, para associar um CSS ao arquivo (X)HTML, você deve incluir a seguinte linha no elemento head de cada arquivo que deve ser estilizado dessa forma:

```
1 <link rel="stylesheet" href="nomeDoArquivo.css" media="screen">
```



## Passo 6 – Criar um CSS para Impressão

---

Geralmente é conveniente ter uma folha de estilo diferenciada para impressão. Pode-se, por exemplo, colocar todos os itens pretos e a cor de plano de fundo branca. Para isso, é necessário criar outro arquivo CSS e ligá-lo ao XHTML com a seguinte linha:

```
1 <link rel="stylesheet" href="print.css" media="print">
```

Observe que o atributo media é diferente do tipo definido para CSS que estiliza a apresentação no browser.

Para criar esse CSS, você pode fazer uma cópia do arquivo CSS já existente e modificar neste novo arquivo a aparência dos elementos já descritos. Antes disso, você deve analisar a página e identificar o que deve e o que não deve ser impresso.

Vamos ao nosso exemplo!

Com base no exemplo da **Figura 4**, foi definido o arquivo CSS a seguir. Os seguintes elementos foram modificados:

- Os elementos que não devem aparecer, inclusive containers, devem ser estilizados com a propriedade `display: none;`
- Todos os textos foram alterados para a cor preta.
- Todos os planos de fundo foram alterados para a cor branca.
- Os links foram alterados com a declaração `text-decoration: none;`
- Foram retiradas todas as declarações para tamanho de fonte, de forma que os tamanhos serão definidos pelo browser.
- As margens para a página foram definidas usando-se algo como: `body{margin: 2em; }.`

**Figura 7** Exemplo de arquivo impresso estilizado por um CSS

Por outro lado, caso deseje manter o mesmo layout para exibição na tela e impressão, é necessário simplesmente modificar o atributo media do elemento link para media="screen, print" ou media="all".



### **Vídeo 06** - CSS para Impressão

## Atividade 02

---

1. Qual a importância de se definir um CSS para impressão?
2. Explique como ligar os dois códigos CSS, um para layout na tela e outro para impressão, ao mesmo documento (X)HTML.
3. Seguindo os cinco primeiros passos definidos anteriormente, aplique o layout especificado na Figura 4, ao site sobre você, desenvolvido em aulas anteriores. Faça as adequações sobre cores, fontes e figuras ao seu gosto e estilo de ser.
4. Crie um layout para impressão do site que você estilizou na questão anterior.

## Conclusão

---

Até agora você aprendeu a escrever páginas web usando as linguagens de marcação HTML e XHTML e também viu como estilizá-las usando CSS.

Na próxima aula, você aprenderá que construir um site é muito mais do que apenas escrever seu código. É uma tarefa que demanda um bom planejamento, que inclui conhecimento do tema, escolha de layouts, conhecimento do público-alvo, entre outras atividades. Ou seja, você deve planejar bem seu site antes de escrever uma única linha de código sequer. Na verdade, escrever o código é a última coisa a ser feita.

## Resumo

---

Nesta aula, você aprendeu quais os tipos de layout, como defini-los e o que levar em consideração para criar layouts para impressão.

## Autoavaliação

---

1. Quais os tipos de layout e as vantagens e desvantagens de cada um deles?
2. Com suas palavras, diga por que usar medidas relativas na definição de tamanhos para fontes.
3. Siga os passos definidos para especificação de layouts e crie um layout líquido com largura mínima (diferente daquele mostrado no exemplo do site Clara Camarão) para o site que você criou para um colecionador nas aulas passadas.
4. Defina um CSS para impressão desse site.

## Referências

---

[VÁRIOS layouts para download]. Disponível em: <<http://www.mycelly.com/>>. Acesso em: 05 out. 2015.

SILVA, Maurício Samy. **Construindo sites com CSS e (X)HTML**. São Paulo: Editora novatec, 2008.

SITE DO MAUJOR. **LAYOUT CSS. passo a passo**. Disponível em: <<http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>>. Acesso em: 05 out. 2015.

WATRALL, Ethan; SIARTO, Jeff. **Use a cabeça: web design**. Rio de Janeiro: Editora Alta Books, 2009.