

# Cascading Style Sheets

+estilos, +posicionamento, +efeitos

# Posicionando Tags

Arquivos de apoio:

posicao-div1.html

posicao-div2.html

posicao-div3.html

posicao-div4.html

exe-header.html

**<div>**

**<h1>, <h2>, ...**

**<p>**

**<span>**

**<a>**

**<ul>**

---

## O que há além do *float*? *posicao-div1.html*

`float: left;`

`float: right;`

Já vimos o comportamento flutuante dos elementos, mas há também um atributo mais poderoso: o *position*.

`position: static;`

Este valor default dos elementos *block* faz com que eles sejam posicionados da forma como foram declarados na estrutura do HTML.

Comportamento padrão dos elementos

Comportamento padrão dos elementos

## Posicionamento relativo (*posicao-div2.html*)

**position:** relative;

**left:** 20px;

**top:** 20px;

O valor **relative** faz com que o componente seja ajustado a partir de sua posição original na estrutura do HTML. Para efeito prático, ele precisa ser configurado em conjunto com as propriedades **top**, **right**, **bottom** e **left**.



## Posicionamento absoluto (*posicao-div3.html*)

O valor *absolute* faz com que o elemento se posicione com referência ao seu elemento pai (com *position* diferente de *static*). Caso não tenha nenhum elemento pai, ele se posicionará considerando o *body* da página. Esta configuração é muito utilizada para configurar botões fixos na *viewport* (área onde o HTML é renderizado).

```
div.elemento1 {  
    width: 300px;  
    height: 300px;  
    background-color:  
darkmagenta;  
    position: absolute;  
    top: 10px;  
    right: 0;  
}
```

```
div.elemento2 {  
    width: 100px;  
    height: 100px;  
    background-color: darkorange;  
    position: absolute;  
    left: 0;  
    bottom: 0;  
}
```

## Posicionamento absoluto (*posicao-div3.html*)

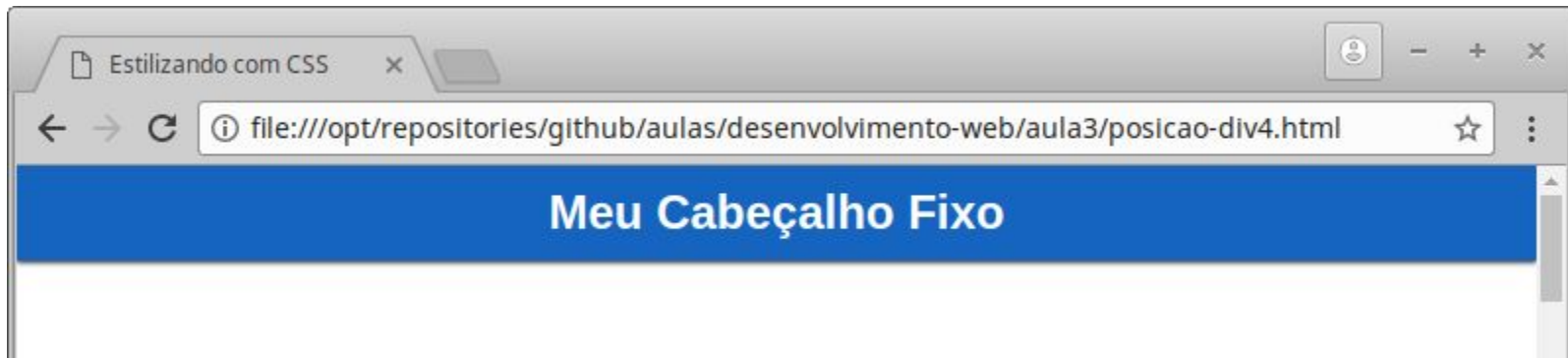
Note que ele é movido junto com o scroll da página, se houver.



## Posicionamento fixo (*posicao-div4.html*)

```
div.cabecalho {  
    width: 100%;  
    line-height: 48px;  
    /* Outras propriedades ... */  
    position: fixed;  
    top: 0;  
    left: 0;  
}
```

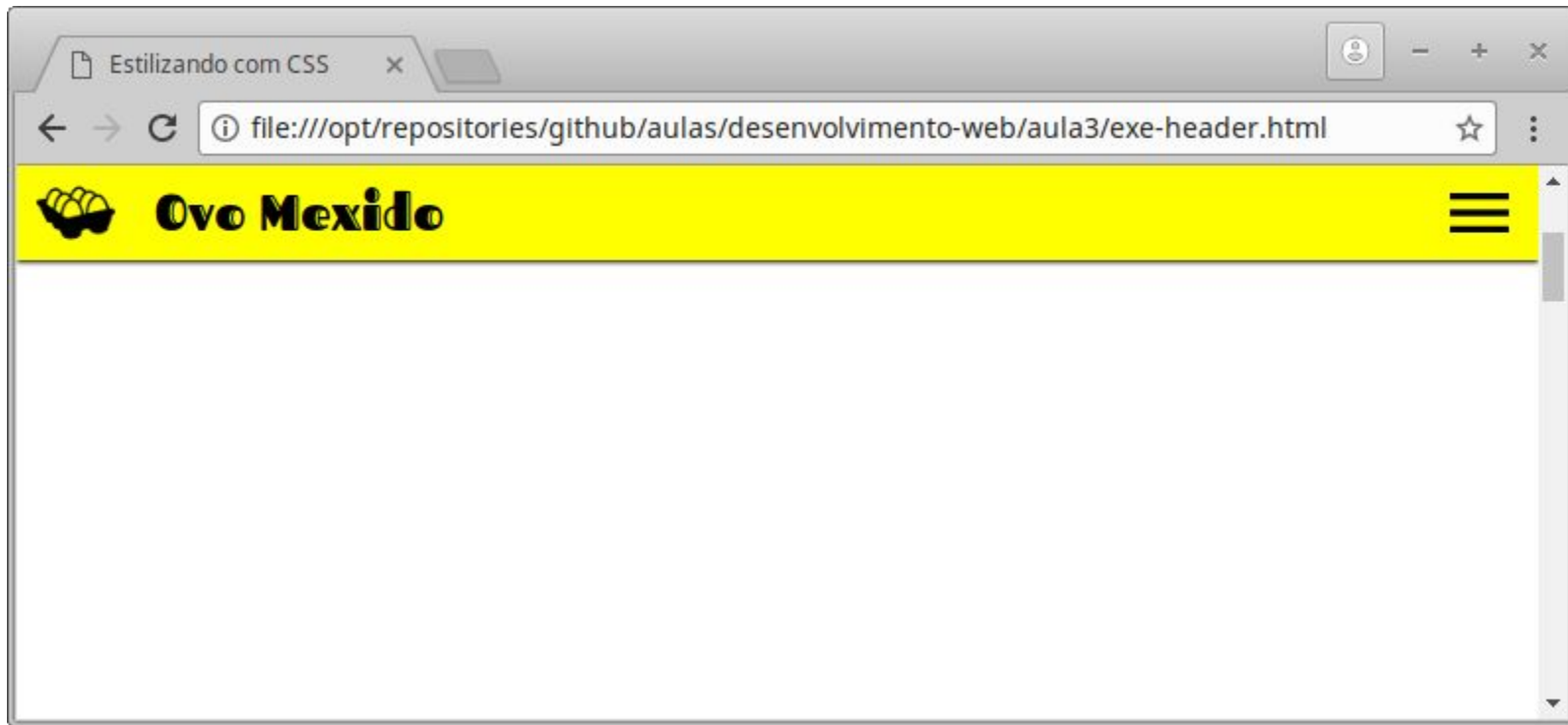
O valor *fixed* sempre será posicionado de acordo com a *viewport* (área onde o HTML é desenhado) do navegador. Ele **não** é movimentado com o scroll da página. Esta é uma ótima opção para criar cabeçalhos, rodapés e botões fixos na tela do usuário.



## Exercício em sala (*exe-header.html*)

Crie um menu fixo no topo da tela que tenha: efeito de sobreposição na parte inferior, logotipo, o nome do site e um ícone de menu à direita.

**Dica:** use *Google Fonts* e *Material Icons*.





# Imagens e background

Arquivos de apoio:

image1.html

image2.html

image3.html

image4.html

image5.html

**<div>**

**<h1>, <h2>, ...**

**<p>**

**<span>**

**<a>**

**<ul>**

---

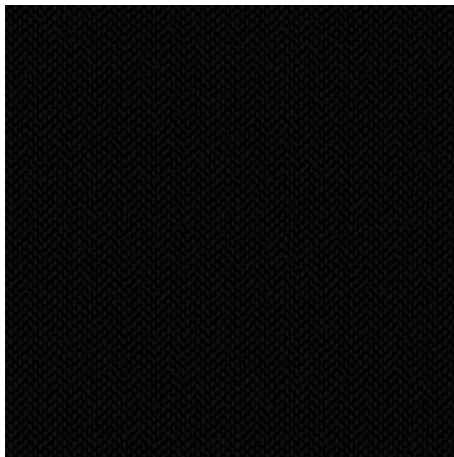
## Um pouco mais sobre imagens (*image1.html*)

Além da tag `<img>` podemos definir imagens de fundo para elementos através da propriedade *background-image*. Vamos montar um site igual a exemplo a seguir.



**background-image:** url(img/pattern-background.png)

Esta propriedade pode ser aplicada às tags HTML, inclusive ao *body*. Por default, ela vai repetir a imagem horizontalmente e verticalmente até preencher todo o conteúdo da tag. Isso é útil para aplicar *image patterns* que se encaixam de forma suave e criar um efeito de background bem interessante.



## Técnica para centralizar tags *block* (*image1.html*)

Uma simples técnica utilizada para centralizar tags do tipo *body* é definir uma margem com a palavra *auto*.

```
margin: 0 auto;
```

Na sintaxe acima, estamos definindo zero distanciamento entre o topo e a parte de baixo e uma borda automática da direita e esquerda. Isso pode ser utilizado para centralizar uma imagem no centro do seu elemento pai. **Atenção:** como `<img>` é uma tag *inline*, precisamos transformá-la em *block*.

```
display: block;
```

## Mais atributos do background (*image2.html*)

<code>background-image: url(img/smile.png);</code>	Exibe a imagem com o tamanho real do arquivo, repetindo, se necessário.
<code>background-size: 100px 100px;</code>	Utilizado para definir o comprimento e a altura da imagem que foi definida como background.
<code>background-repeat: repeat-x;</code>	Define se vai ou não repetir. Aceita: <i>repeat</i> , <i>repeat-x</i> , <i>repeat-y</i> e <i>no-repeat</i>
<code>background-position: 100px 100px;</code>	Define a posição da imagem de background a partir do topo esquerdo do componente pai. Valores negativos também são permitidos.

## Fixando background (*image3.html*)

**background-attachment:** *fixed*;

Outro efeito interessante é fixar a imagem de background no fundo da tela para que ela não role junto com o scroll da página. Esse estilo ganhou destaque com a utilização de frameworks CSS, como o *Bootstrap*, por exemplo. O valor default é *scroll*.

Recomenda-se utilizá-lo em conjunto o comando abaixo para evitar que a imagem se repita ao rolar o scroll.

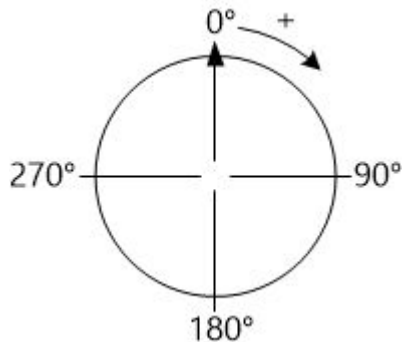
**background-repeat:** *no-repeat*;

## Gradiente (*image4.html*)

Antigamente (há 6 ou 7 anos atrás) era muito comum criar imagens com gradiente e aplicá-las ao fundo de componentes para estilizá-los. O CSS 3 trouxe a propriedade *linear-gradient* que pode ser criada dentro do código CSS e se comporta como se fosse uma imagem de fundo.

**background-image:** linear-gradient(180deg, #ef5350, #b71c1c);

to right to left to top to bottom	to top left to top right to bottom left to bottom right
--	--



## Dicas gerais de estilização com imagens

**Imagens no IE:** historicamente o Internet Explorer costuma ferir algumas especificações do W3C. Uma que acontece com frequência é a aplicação de um espaçamento “misterioso” entre imagens dispostas lado a lado. Para resolver, aplique a propriedade *display: block* com *padding*, *margin* e *border* zeradas.

**Tamanho das imagens:** apesar do CSS possuir sintaxe para redimensionar imagens, evite trabalhar com arquivos muito grandes (KB ou MB). Lembre-se que uma conexão 3G pode prejudicar o carregamento do seu site. Tenha em mente algo no máximo de 300KB. Se isso não for possível, existe uma técnica que envolve JavaScript que primeiramente carrega uma imagem pequena e, após o carregamento da imagem maior, substitui a imagem pequena pela imagem grande. Vide o arquivo *image5.html* ou acesse:

<http://linu.com.br/alunos/imagem/>

**Sprite de imagens:** faça uma busca pela Internet sobre esta técnica.



# Estilos Avançados

Arquivos de apoio:

avancado1.html

avancado2.html

avancado3.html

avancado4.html

avancado5.html

avancado6.html

exe-card.html

**<div>**

**<h1>, <h2>, ...**

**<p>**

**<span>**

**<a>**

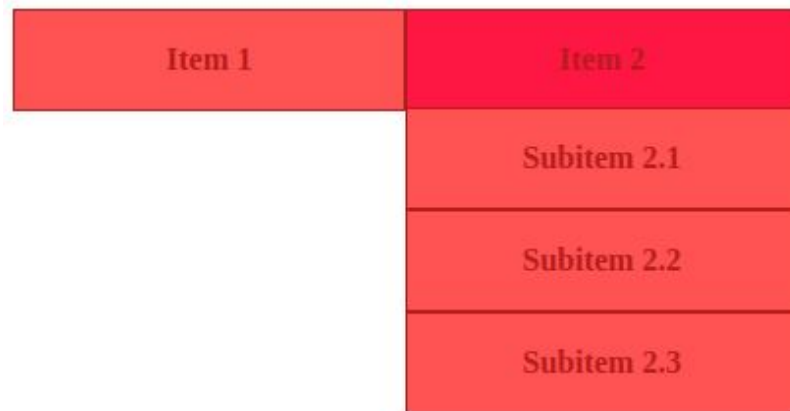
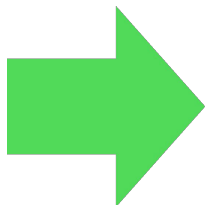
**<ul>**

---

## Mais atributos legais (*avancado1.html* e *avancado2.html*)

<code>text-decoration: none;</code>	Quando se utiliza links, o navegador automaticamente emprega um sublinhado sobre o texto. Para removê-lo, utilize o <i>text-decoration</i> como <i>none</i> .
<code>list-style: none;</code>	Usado para retirar os bullets de listas. Com isso e outros detalhes podemos transformar listas em menus de sistemas.

- Item 1
  - Subitem 1.1
  - Subitem 1.2
  - Subitem 1.3
- Item 2
  - Subitem 2.1
  - Subitem 2.2
  - Subitem 2.3



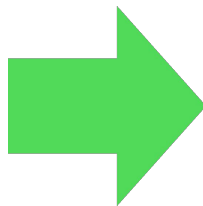
## Medidas (*avancado3.html*)

%	Define um percentual com base no componente pai.
px	Define em pixels. Muito fácil de trabalhar, pois a <i>viewport</i> dos browsers são definidas em pixels.
rem	Define um valor em relação ao tamanho da fonte do navegador (tag HTML). <b>1 rem = 16 pixels</b> , mais isso pode variar.
em	Semelhante ao <i>rem</i> , porém é relativo ao tamanho da fonte do elemento pai.

Outras medidas: <https://css-tricks.com/the-lengths-of-css/>






Mas afinal, qual a vantagem de se trabalhar com essas outras medidas?

Teste em Pixels  
Teste em Rem



Teste em Pixels  
Teste em Rem

## Diretivas específicas dos browsers (*avancado4.html*)

-webkit-	 Chrome
-moz-	 Mozilla Firefox
-ms-	 Internet Explorer
-o-	 Opera
-webkit-	 Safari

## Diretivas específicas dos browsers (*avancado4.html*)

Vamos testar o atributo *user-select* (que permite bloquear a seleção de um texto em uma tag). Veja a tabela de compatibilidade em:

[https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp)

<https://caniuse.com/#search=user-select>

Agora vamos testar os endereços abaixo no IE 10:

<http://linu.com.br/alunos/browser/errado.html>

<http://linu.com.br/alunos/browser/certo.html>

Para usar o IE 10, podemos utilizar sites como:

<https://ieonchrome.com/> (vamos usar esse agora...)

<https://www.browserling.com/> (pago)

<https://www.browserstack.com/> (pago)

<http://netrenderer.com/index.php> (apenas imagens)

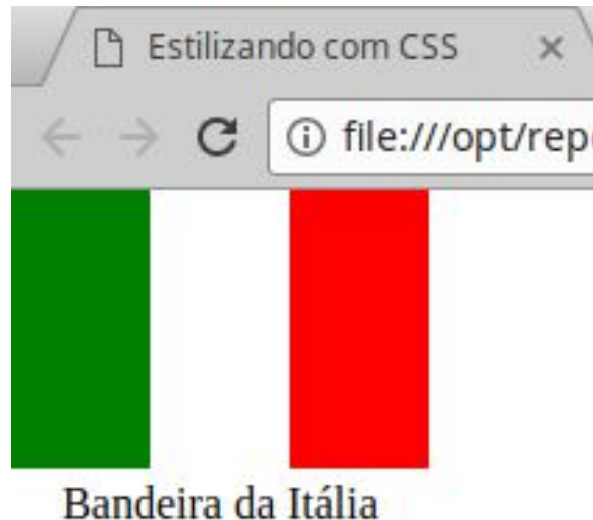
## Usando o atributo *clear* (*avancado5.html*)

Através da sintaxe *clear: both* nós podemos interromper o posicionamento flutuante de elementos. Por exemplo, imagine que todas as `<div>` de uma página fossem definidas como *float: left*. Para especificar que uma delas deve manter o comportamento padrão da estrutura, podemos ajustá-la com a propriedade *clear: both*.

*clear: left;* /\* Quebra float à esquerda \*/

*clear: right;* /\* Quebra float à direita \*/

*clear: both;* /\* Quebra float dos dois lados \*/



## Usando entidades HTML (*avancado6.html*)

Quer colocar símbolos especiais na página?

Use e abuse das HTML Entities.

[https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp)

& ¥ ® © € Ø ∕

# Exercício em Sala (*exe-card.html*)

Vamos montar um *card* sobre um fundo roxo, com imagens e entidades HTML sobrepostas. Use Google Icons, Entidades HTML, fundo gradiente. **Dica:** posicione o elemento de fora como *relative* e os internos como *absolute*.





Obrigado!