



CURSO DE HTML5, CSS3, JAVASCRIPT

Ministrado por:

- Bruno Teles dos Santos
- Matheus Vieira
- Allan Macedo

by NU△GE
IT's everywhere.

Parte 2

CSS3

CSS Histórico

CSS é a abreviatura para *Cascading Style Sheets* - Folha de Estilos em Cascata

CSS é uma linguagem para estilos que define o layout de documentos HTML. Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais.

HTML é usado para definir a estrutura do website. Contudo, o CSS proporciona mais opções e é mais preciso e sofisticado, é suportado por todos os navegadores atuais.



HTML é usado para estruturar conteúdos. CSS é usado para formatar conteúdos estruturados

HTML5
CSS3
JAVASCRIPT



CSS

Inserindo em uma página

Para que as regras tenham efeito na página, há três maneiras de inseri-las (veremos isso com mais detalhes na lição seguinte):

- Arquivo externo “.css” ligado à página HTML através da tag <link> (melhor método);
- Dentro da tag <head> de uma página, utilizando a tag “<style>”
- Dentro de um elemento, utilizando o atributo style (estilos em linha)

HTML5
CSS3
JAVASCRIPT



CSS TAG Style

Com este método, aplicamos estilos apenas na página onde o elemento está inserido. Para isso, utiliza-se a tag style.

Exemplo:

```
<style>
  p {
    color: red;
  }
</style>
```

No exemplo acima, os estilos definidos aplicarão a cor vermelha para todos os parágrafos do documento HTML. Todos os elementos style devem ficar dentro do elemento head de uma página.



Os estilos em uma tag style tem precedência sobre os estilos em um arquivo externo.

HTML5
CSS3
JAVASCRIPT



CSS

Arquivo externo

Este é o método que apresenta maior versatilidade. Um arquivo externo CSS pode ser ligado a quantas páginas desejarmos, desta forma deixando a manutenção de um site muito mais fácil (apenas um arquivo CSS pode controlar o visual de um site inteiro). Para este método, utilizamos o elemento link, da seguinte forma:

```
<link href="css/arquivo.css" rel="stylesheet">
```

HTML5
CSS3
JAVASCRIPT



CSS

Arquivo externo

A tag link é uma tag que auto-fecha, assim como br e meta. O atributo href indica o endereço do arquivo CSS (hiper-referência), neste exemplo um arquivo chamado "arquivo.css" dentro de uma pasta "css". O atributo rel determina a relação deste "link" com a página, aqui sendo stylesheet ou folha de estilos. Se estivéssemos utilizando a sintaxe XHTML, também é necessário o atributo type com o valor text/css.

HTML5
CSS3
JAVASCRIPT



CSS

Atributo style

Este é o método que deve ser menos utilizado, por ir contra a divisão de uma página em 3 camadas, como vimos anteriormente. Utilizando o atributo style, podemos aplicar estilos a um elemento específico.

Exemplo:

```
<p style="color: red;">Texto</p>
```

No exemplo acima, aplicamos a cor vermelha apenas para um parágrafo escolhido. Este método é chamado de "*inline styles*" (ou estilos em linha) por serem localizados junto com o código HTML. Ele tem precedência de aplicação sobre todos os outros. Seu uso **não é recomendado, pois mistura o código CSS com o HTML** e dificulta a manutenção do site, além de ir contra a ideia de separação da página em camadas. Seu uso esporádico deve ser feito com cuidado e bom senso.

HTML5
CSS3
JAVASCRIPT



CSS

Classes e IDs

Para possibilitar uma organização melhor da página e da seleção de elementos por CSS, podemos utilizar classes e/ou ids para identificar tags.

Uma id, como o nome diz, é uma identificação única: só pode ser utilizada uma vez no documento inteiro. Normalmente é utilizada para identificar elementos estruturais da página.

Uma classe é reutilizável: pode se repetir na página e também combinar-se com outras (podemos pôr mais de uma classe em um elemento).

É possível fazer um site apenas com ids ou apenas classes, com uma combinação das duas, ou sem nenhuma das duas

HTML5
CSS3
JAVASCRIPT



CSS

Classes e IDs

Por exemplo, vamos imaginar um site com três blocos de conteúdo que aparecem em todas as páginas do site:

```
<div>
```

Aqui vai o cabeçalho do site

```
</div>
```

```
<div>
```

Aqui vai o conteúdo do site

```
</div>
```

```
<div>
```

Aqui vai o rodapé do site

```
</div>
```

HTML5

CSS3

JAVASCRIPT



CSS

Classes e IDs

Como estes elementos sempre aparecem nas páginas e aparecem apenas uma vez em cada página, podemos utilizar uma id para cada um, através do atributo id, da seguinte forma:

```
<div id="cabecalho">
```

Aqui vai o cabeçalho do site

```
</div>
```

```
<div id="conteudo">
```

Aqui vai o conteúdo do site

```
</div>
```

```
<div id="rodape">
```

Aqui vai o rodapé do site

```
</div>
```

HTML5

CSS3

JAVASCRIPT



CSS

Classes e IDs

Quanto a classes, podemos imaginar um site onde temos vários parágrafos de texto:

```
<p>  
...  
</p>  
<p>  
...  
</p>  
<p>  
...  
</p>
```

HTML5
CSS3
JAVASCRIPT



CSS

Classes e IDs

Se utilizarmos o seletor p para aplicar estilos, estes serão aplicados a todos os parágrafos. Se quisermos, por exemplo, destacar um ou mais parágrafos, podemos criar uma classe com o nome "destaque" e colocá-la apenas nos parágrafos que desejamos, através do atributo class como no código abaixo:

```
<p class="destaque">  
...  
</p>  
<p>  
...  
</p>  
<p class="destaque">  
...  
</p>
```

HTML5
CSS3
JAVASCRIPT



CSS

Classes e IDs

Desta forma, quando selecionarmos a classe "**destaque**" no CSS, estaremos aplicando os estilos apenas nos parágrafos onde a classe está aplicada. Outra vantagem é que podemos aplicar a classe "**destaque**" para outros tipos de elementos também, como links, entre outros. Classes são mais versáteis que ids e por isso muitos desenvolvedores utilizam apenas classes nos seus sites.

HTML5
CSS3
JAVASCRIPT



CSS Seletores

Muito do poder da linguagem CSS está nos seletores. Com os seletores, conseguimos escolher qualquer elemento (ou elementos) na página para aplicar estilos.

Vejamos primeiro os principais tipos de seletores:

- **Seletor de tipo:** este é o tipo de seletor que utilizamos nos exemplos até agora. Com este seletor, selecionamos todas as tags de um mesmo tipo. Por exemplo, se digitamos `a` estaremos selecionando todas as tags `a` (links) da página e poderemos aplicar estilos a elas. Útil para estilos gerais, mas para maior especificidade utilizamos outros seletores.

HTML5
CSS3
JAVASCRIPT



CSS

Seletores

- **Seletor descendente:** com este seletor, podemos escolher um ou mais elementos que estão dentro de outro, ou seja, que são descendentes do elemento principal. **Exemplo:** `p strong`. Com isso, selecionamos apenas tags strong que estão dentro de parágrafos. Podemos selecionar com ainda mais especificidade, escrevendo mais elementos, como: `div p strong a`. Neste exemplo, selecionamos links que estão dentro de tags strong que estão dentro de parágrafos que estão dentro de tags div.

HTML5
CSS3
JAVASCRIPT



CSS

Seletores

- **Seletor de classe:** seleciona elementos com uma classe específica aplicada. Exemplo: `.destaque` seleciona todos os elementos com a classe "destaque".
- **Seletor de id:** seleciona o elemento com a id especificada. Exemplo: `#cabecalho` irá selecionar o elemento com a id "cabecalho". Cada id é única e não pode ser repetida no mesmo documento.

HTML5
CSS3
JAVASCRIPT



CSS

Combinando Seletores

Podemos também combinar os seletores que aprendemos acima, para conseguir diferentes elementos e partes mais específicas de nossos sites. Alguns exemplos:

- **p.destaque** seleciona apenas os parágrafos que possuem a classe "destaque".
- **div#cabecalho h1** seleciona tags h1 que estejam dentro da div com a id "cabecalho".
- Separando itens por vírgulas, como **p.destaque, h1, a.saiba-mais** seleciona todos os respectivos elementos para as regras. Muito útil para diminuir a repetição de regras no arquivo CSS.

HTML5
CSS3
JAVASCRIPT



CSS

Pseudo-classes

Pseudo-classes são tipos de classes especiais que não são definidas pelo desenvolvedor (já são pré-definidas). A sua maior utilização é com links (tags a) para definir estilos diversos para quando o link é novo, visitado, etc. O seletor de pseudo-classe é escrito com o nome do elemento + dois pontos + nome da pseudo-classe.

Para começar, vamos nos familiarizar com as pseudo-classes de links:

- Com o seletor **a:link**, estilizamos apenas os links não-visitados, ou seja, links no seu estado normal.
- Com o seletor **a:visited**, estilizamos apenas links visitados, ou seja, que já foram clicados.

HTML5
CSS3
JAVASCRIPT



CSS

Pseudo-classes

- Com o seletor **a:hover**, estilizamos links quando o mouse está em cima do mesmo. Com esta pseudo-classe podemos fazer diversos efeitos interessantes. Esta pseudo-classe pode ser aplicada a qualquer elemento, não apenas links, o que a torna ainda mais útil.
- Com o seletor **a:focus**, estilizamos links quando os selecionamos com o teclado, através da tecla Tab. Esta pseudo-classe é útil para estilizar links para pessoas que possuem habilidade limitada e não conseguem utilizar o mouse, por exemplo.
- Com o seletor **a:active**, estilizamos um link quando o mouse está sendo clicado ou pressionamos Enter, ativando o link.

HTML5
CSS3
JAVASCRIPT



CSS

Pseudo-classes

[Link normal \(:link\)](#)

[Link visitado \(:visited\)](#)

[Link com o cursor em cima \(:hover\)](#)

[Link focado com o teclado \(:focus\)](#)

[Link ativado \(com o clique ou tecla enter\) \(:active\)](#)

HTML5
CSS3
JAVASCRIPT



CSS

Cascata (Herança)

A cascata é outra das características importantes das CSS. Tão importante, que está até no nome: Cascading Style Sheets. O que "Cascading" quer dizer é que a maioria das propriedades que aplicamos a um elemento, irá passar aos seus "herdeiros", ou seja, os elementos que estão contidos naquele.

Exemplo:

```
<!-- HTML -->
<div class="texto">
  <p>
    Texto exemplo <strong>com um destaque</strong>.
  </p>
</div>
```

```
/* CSS */
.texto {
  color: #F00;
}
```

Texto exemplo com um destaque.

HTML5
CSS3
JAVASCRIPT



CSS

Cascata (Herança)

Uma utilização comum da cascata é definir uma fonte para a página inteira.

Exemplo:

```
body {  
  font-family: Arial, sans-serif;  
}
```

Com esta regra, todos os elementos que estão dentro de body (ou seja, todo o conteúdo do site) irá usar a fonte Arial ou, se não estiver disponível, a fonte sem serifa padrão do sistema operacional

HTML5
CSS3
JAVASCRIPT



CSS

Especificidade e precedência

As CSS também obedecem algumas regras de precedência. Quando várias regras se aplicam a um mesmo elemento, o navegador precisa decidir qual delas tem precedência. Vamos criar uma div:

```
<div id="sidebar">  
  <p class="descricao">  
    Texto exemplo.  
  </p>  
</div>
```

Vamos supor que temos duas regras aplicadas à div:

HTML5
CSS3
JAVASCRIPT



CSS

Especificidade e precedência

Vamos supor que temos duas regras aplicadas à div:

```
#sidebar {  
  color: #F00;  
}
```

```
div#sidebar p {  
  color: #F0F;  
}
```

Qual delas será aplicada? Se você pensou a "mais específica", acertou.

HTML5
CSS3
JAVASCRIPT



CSS

Especificidade e precedência

O seletor `div.sidebar p` tem precedência. Por que ele é mais específico? Para termos uma ideia melhor de especificidade, há algumas regras gerais:

- Regras em tags **style** tem precedência sobre regras em arquivos externos CSS;
- Regras em linha (atributo **style**) tem precedência sobre regras em tags **style** ou arquivos externos CSS;
- Seletores de elemento (por exemplo, **p**) tem a menor precedência (podemos dizer que, numa escala, este tipo de seletor vale "1");
- Seletores de classe (por exemplo, **.destaque**) tem mais precedência (vale "10");
- Seletores de id (por exemplo, **#conteudo**) tem maior precedência (vale "100").

HTML5
CSS3
JAVASCRIPT



CSS

Especificidade e precedência

O seletor `div.sidebar p` tem precedência. Por que ele é mais específico? Para termos uma ideia melhor de especificidade, há algumas regras gerais:

- Regras em tags **style** tem precedência sobre regras em arquivos externos CSS;
- Regras em linha (atributo **style**) tem precedência sobre regras em tags **style** ou arquivos externos CSS;
- Seletores de elemento (por exemplo, **p**) tem a menor precedência (podemos dizer que, numa escala, este tipo de seletor vale "1");
- Seletores de classe (por exemplo, **.destaque**) tem mais precedência (vale "10");
- Seletores de id (por exemplo, **#conteudo**) tem maior precedência (vale "100").

HTML5
CSS3
JAVASCRIPT



CSS

Margens, bordas e box model

Uma página é feita de blocos. Estes blocos são "empilhados" de cima para baixo, de acordo com a ordem do código: o que está no início do código fica em cima e os elementos subsequentes vão se acomodando em baixo uns dos outros e através de CSS podemos mudar esta ordem.

Cada elemento pode ter uma margem interna (*padding*), margem externa (*margin*) e uma borda (*border*). A forma como as margens e bordas se comportam junto com os elementos constitui o que é chamado de **box model**.

HTML5
CSS3
JAVASCRIPT



CSS Margens

Exemplo1:

```
div.margem {  
  background: #900;  
  color: #FFF;  
}
```

```
<div class="margem">  
  div sem margem interna  
</div>
```

Exemplo2:

```
div.margem2 {  
  background: #900;  
  color: #FFF;  
  padding: 10px;  
}
```

```
<div class="margem2">  
  div sem margem interna  
</div>
```



No exemplo 1 nada aparece pois não existe valor e não existem uma margem definida.
A declaração padding: 10px coloca margens internas de 10 pixels em cada um dos lados da div.

HTML5
CSS3
JAVASCRIPT



CSS Margens

Exemplo3:

```
div.margem3 {  
  background: #900;  
  color: #FFF;  
  padding: 10px;  
  margin: 10px;  
}
```

```
<div class="margem3">  
  div com margem interna e externa  
</div>
```

Margens externas foram aplicadas a todos os lados da div, assim como as internas.

HTML5
CSS3
JAVASCRIPT



CSS

Margens

Caso queiramos inserir margens em apenas alguns dos lados, podemos utilizar as propriedades `padding-top`, `padding-right`, `padding-bottom` e `padding-left`; para as margens externas, `margin-top`, `margin-right`, `margin-bottom` e `margin-left`.

Exemplo4:

```
div.margem4 {  
  background: #900;  
  color: #FFF;  
  padding-left: 10px;  
  padding-right: 10px;  
  margin-bottom: 10px;  
}
```

```
<div class="margem4">  
  div com margem interna e externa  
</div>
```

HTML5
CSS3
JAVASCRIPT



CSS

Margens

Ao invés de digitar uma propriedade para cada lado, podemos utilizar a forma abreviada, ou shorthand, das propriedades margin e padding. Funciona da seguinte forma:

- Para aplicar um valor para todos os lados, apenas digitamos um valor, como padding: 10px;
- Para aplicar um valor para cima/baixo e um para esquerda/direita, escrevemos dois valores: padding: 5px 10px;. O primeiro será aplicado para cima/baixo e o segundo para esquerda/direita.
- Para aplicar um valor para cada lado, utilizamos a ordem cima/direita/baixo/esquerda (sentido horário partindo de cima), desta forma: margin: 5px 10px 15px 30px. O primeiro valor será aplicado para cima, o segundo para direita, o terceiro para baixo e o quarto para esquerda.

HTML5
CSS3
JAVASCRIPT



CSS

Margens

Podemos também utilizar outras unidades além de pixels, como %, em (unidade relativa ao tamanho da fonte no elemento), pt (pontos, mais utilizado para impressos), cm (idem), entre outras.

HTML5
CSS3
JAVASCRIPT



CSS Borda

Com a propriedade border, podemos adicionar bordas ao redor de qualquer elemento. Primeiro, vamos conhecer os componentes desta propriedade. No exemplo abaixo:

Exemplo1:

```
.borda1 {  
    border-top-width: 3px;  
    border-top-style: solid;  
    border-top-color: #000;  
}  
  
<div class="borda1">  
    texto com borda  
</div>
```

Podemos resumir as três declarações acima em apenas uma (*shorthand*), como o exemplo abaixo:

```
.borda1 {  
    border-top: 3px solid #000;  
}
```

HTML5
CSS3
JAVASCRIPT



CSS Borda

O mesmo vale para **border-right**, **border-bottom** e **border-left** e suas **sub-propriedades**, como **border-left-color**, **border-bottom-style** ou **border-right-width**.

Para controlar as bordas em um elemento inteiro, podemos utilizar as propriedades **border-width**, **border-style** e **border-color**:

Exemplo2:

```
p.borda2 {  
  /* a ordem dos lados é a mesma que com margens */  
  border-width: 2px 5px 3px 6px; /* aqui, cada lado possui uma borda de largura diferente */  
  border-style: dashed dotted; /* estilos de borda diferentes para cima-baixo e esquerda-direita */  
  border-color: #000 #090 #900 #009; /* uma cor de borda para cada lado */  
  padding: 20px;  
}
```

HTML5
CSS3
JAVASCRIPT



CSS Borda

Finalmente, podemos utilizar a propriedade border para definir as bordas uniformemente no elemento:

Exemplo3:

```
p.borda3 {  
    border: 1px solid #000;  
}
```

HTML5
CSS3
JAVASCRIPT



CSS

Box Model

Por padrão, um elemento como `p` ou `div` irá ocupar todo o espaço horizontal disponível. Podemos dizer, em propriedades CSS, que temos `width: auto; height: auto;`, ou seja, que a largura e a altura são determinadas automaticamente, sendo que a largura por padrão será 100% na maioria dos casos e a altura será determinada pela quantidade de conteúdo dentro do elemento.

Um bloco como um `p` pode ter até quatro áreas distintas na sua composição. Da mais externa para a mais interna: margem externa, borda, margem interna e conteúdo.

HTML5
CSS3
JAVASCRIPT



CSS

Box Model - Exemplo

```
.margem-externa1 {  
  background: #099;  
  padding: 20px;  
}
```

```
.exemplo-boxmodel1 {  
  border: 3px solid #900;  
  padding: 20px;  
  background: #CCC;  
  margin: 0;  
}
```

```
.conteudo1 {  
  display: block;  
  background: #FFC;  
}
```

```
<div class="margem-externa1">  
  <p class="exemplo-boxmodel1">  
    <span class="conteudo1">  
      Neste exemplo, a área amarela representa o conteúdo. A parte cinza  
      representa a margem interna do parágrafo, a linha vermelha a borda e a parte  
      azul a margem externa.  
    </span>  
  </p>  
</div>
```

Neste exemplo, a área amarela representa o conteúdo. A parte cinza representa a margem interna do parágrafo, a linha vermelha a borda e a parte azul a margem externa.

HTML5
CSS3
JAVASCRIPT



CSS

Elementos bloco

Todos os elementos HTML podem ser separados em dois grupos, em termos de comportamento visual CSS: elementos bloco (block) e elementos em linha (inline). Ser bloco ou em linha muda o comportamento visual do elemento, além de outras características. Apesar de podermos mudar o modo de exibição de cada elemento com CSS, cada tag por padrão pertence a um ou outro grupo.

Elementos bloco ocupam todo o espaço horizontal disponível e iniciam uma nova linha no documento. Novos elementos irão começar na próxima linha livre.

HTML5
CSS3
JAVASCRIPT



CSS

Elementos bloco - Exemplos

- **div** (division, ou divisão, tag bloco genérica)
- **h1** até **h6** (títulos)
- **p** (parágrafo)
- **blockquote** (citação em bloco)
- **ul** (lista não ordenada)
- **ol** (lista ordenada)
- **form** (para formulário)

HTML5
CSS3
JAVASCRIPT



CSS

Elementos em linha

Elementos em linha ocupam apenas o espaço necessário e não iniciam uma nova linha. São chamados elementos em linha justamente por aparecer na mesma linha que outros elementos, caso seja possível.

HTML5
CSS3
JAVASCRIPT



CSS

Elementos em linha - Exemplos

- **span** (tag em linha genérica)
- **strong** (destaca importância)
- **em** (ênfase)
- **a** (âncora, usada para links)
- **img** (imagem)

HTML5
CSS3
JAVASCRIPT



CSS

Aninhamentos de Elementos

Uma página é feita de blocos empilhados. Dito isso, muitas vezes iremos inserir um elemento dentro de outro para conseguirmos construir um layout adequado e com a hierarquia de informações que desejamos. Inserir um elemento dentro de outro chama-se aninhamento (ou nesting) e em documentos HTML há algumas regras principais a serem notadas:

- Elementos bloco podem conter qualquer elemento em linha;
- Elementos bloco podem conter outros elementos bloco, observadas algumas restrições (por exemplo, não podemos colocar uma div dentro de um parágrafo mas o contrário é válido). Geralmente segue-se o bom-senso nestes casos;

HTML5
CSS3
JAVASCRIPT



CSS

Aninhamentos de Elementos

- Elementos em linha nunca poderão conter elementos bloco (ex: colocar um parágrafo dentro de um link ou uma div dentro de uma span). A HTML5 apresenta algumas mudanças no caso do que links (tags a) podem englobar, exceto isso esta regra é geral;
- Elementos em linha podem conter outros elementos em linha (por exemplo, colocar span dentro de um link ou o contrário).

HTML5
CSS3
JAVASCRIPT



CSS

Aplicação de propriedades CSS

Além das diferenças já apresentadas, elementos bloco ou em linha também diferem nas propriedades CSS que podem ser aplicadas com confiabilidade. Elementos em linha podem ter os seguintes tipos de propriedades modificadas:

- Cores de texto e de fundo;
- Propriedades de texto, como font-family, font-size e outras, como text-decoration;
- Bordas.

Propriedades como width, height, margin e padding podem surtir efeito no elemento em questão mas geralmente não irão afetar os outros elementos. Ou seja, para fins de layout e posicionamento, quando se usam muitas definições de dimensões e margens, usar elementos em linha não é recomendado.

HTML5
CSS3
JAVASCRIPT



CSS

Aplicação de propriedades CSS

Elementos bloco, além de todas as propriedades dos elementos em linha, também podem ter modificadas:

- Largura (width) e altura (height);
- Margens internas (padding) e externas (margin).

Quando queremos usar tags como a para compor um menu, na grande parte das vezes a transformamos em elemento bloco. Para isso, usa-se a importante propriedade display CSS.

HTML5
CSS3
JAVASCRIPT



CSS Display

A propriedade display serve para mudarmos o comportamento visual de um elemento, como mudar uma tag a (em linha) para bloco ou uma tag p para elemento em linha. Ela aceita vários valores, dos quais os principais são:

- **block**: elemento aparece como se fosse bloco;
- **inline**: elemento aparece como se fosse em linha;
- **none**: elemento não aparece, útil quando utilizado junto com javascript para fazê-lo aparecer com algum efeito visual, entre outras aplicações.



Mesmo que mudarmos visualmente como um elemento aparece na página, as regras de aninhamento ainda são as mesmas.

HTML5
CSS3
JAVASCRIPT



CSS

Display - Exemplo

```
.link-normal {  
  background: #00F;  
  color: #FFF;  
  margin: 10px;  
  padding: 20px;  
  border: 1px solid #000;  
}
```

```
.link-bloco {  
  display: block;  
  background: #00F;  
  color: #FFF;  
  margin: 10px;  
  padding: 10px;  
  border: 1px solid #000;  
}
```

```
<p>  
  <a class="link-normal" href="#">  
    Link Normal  
  </a>  
  texto exemplo  
  <a class="link-bloco" href="#">  
    Link Bloco  
  </a>  
</p>
```

HTML5

CSS3

JAVASCRIPT



CSS Background

A propriedade background permite inserir cor ou imagem no fundo de um elemento. Alguns exemplos:

- **Background: ovalordacor:** altera a cor do fundo do elemento, esse valor pode ser definido de diversas formas: rgb, hexadecimal...;

```
div{  
    background: #990000;  
}
```

HTML5
CSS3
JAVASCRIPT



CSS

Background - Exemplo

- **Background: url("enderecodaimagem.jpg"):** altera o fundo de um elemento inserindo uma imagem

```
div{  
    background: url("html5.jpg");  
}
```

- **Background: ovalodacor url("enderecodaimagem.jpg"):** essa opção permite que seja preenchido com uma cor sempre que a imagem atingir o seu tamanho máximo.

```
div{  
    background: #990000 url("html5.jpg");  
}
```

HTML5
CSS3
JAVASCRIPT



CSS

Background – Repetir a imagem de fundo

- **Background: url("endereço da imagem.jpg") repeat-x:** altera o fundo de um elemento inserindo uma imagem, caso atinja o tamanho máximo da imagem, essa imagem se repete até que se conclua o preenchimento total do elemento.

```
div{  
    background: url("html5.jpg") repeat;  
}
```

HTML5
CSS3
JAVASCRIPT



Por default o background se repete, até concluir o preenchimento total do elemento, para que isso não ocorra utilize **no-repeat**.



CSS Float

Float é uma propriedade CSS que usamos para posicionar algo na tela, seja um elemento em bloco ou em linha.

Com a propriedade float definida, um elemento pode ser empurrado para esquerda ou para a direita, permitindo que os outros elementos possam organizar em torno dele.

HTML5
CSS3
JAVASCRIPT



um elemento com um posicionamento absolute ignora totalmente a propriedade float.



CSS

Float - Exemplo

```
.divA{  
  width: 300px;  
  float: left;  
  background: red;  
}
```

```
.divB{  
  width: 300px;  
  float: right;  
  background: blue;  
}
```

```
<div class="divA">  
  conteúdo da divA  
</div>
```

```
<div class="divB">  
  conteúdo da divB  
</div>
```

HTML5

CSS3

JAVASCRIPT



CSS Position

Funcionalidade utilizada para posicionar qualquer elemento em qualquer parte da página, existe três tipos:

Position Fixed

O `position: fixed;` irá fixar a posição do elemento na coordenada que você definir. À medida que a página é rolada, o elemento continua fixo na posição que você definiu e o conteúdo da página rola normalmente.

Geralmente é usado para fixar elementos como cabeçalhos ou sidebars.



CSS Position

Position Relative

Todos os positions precisam de um ponto para iniciar o cálculo da coordenada para assim posicionar o elemento na tela. Ao contrário do que muitos acham, esse ponto não é o ponto central do elemento, o ponto base é o canto superior esquerdo do elemento. A partir deste canto, o browser irá calcular a coordenada que você definiu e irá posicionar o elemento no viewport.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere eros vel risus varius et euismod dui tincidunt. Nulla et ipsum sem, eget pellentesque dolor. Suspendisse vel tellus nisi. Sed tincidunt adipiscing venenatis. Integer vestibulum nibh sed quam aliquet tincidunt. Mauris vel congue neque. Sed leo



O elemento com position relative se referencia pelo ponto superior esquerdo dele mesmo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere eros vel risus varius et euismod dui tincidunt. Nulla et ipsum sem, eget pellentesque dolor. Suspendisse vel tellus nisi. Sed tincidunt adipiscing venenatis. Integer vestibulum nibh sed quam aliquet tincidunt. Mauris vel congue neque. Sed leo

HTML5
CSS3
JAVASCRIPT

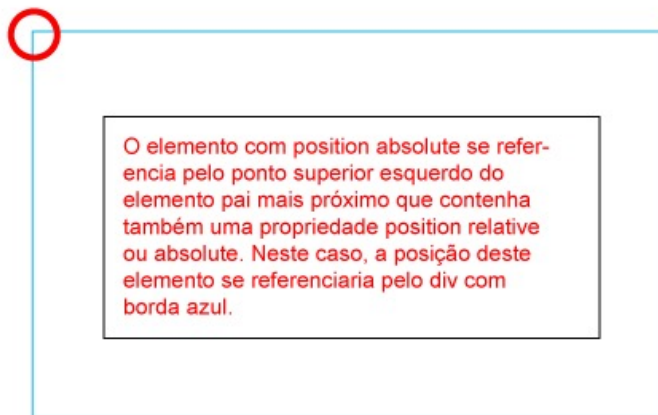


CSS Position

Position Absolute

O Position Absolute é um tanto diferente do Relative. Enquanto o elemento com Position Relative utiliza seu próprio canto para referenciar sua posição, o elemento com Position Absolute se utiliza do ponto superior esquerdo de outros elementos. Estes elementos são os parentes dele do elemento com position absolute. Mais especificamente o pai.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere eros vel risus varius et euismod dui tincidunt. Nulla et ipsum sem, eget pellentesque dolor. Suspendisse vel tellus nisi. Sed tincidunt adipiscing venenatis. Integer vestibulum nibh sed quam aliquet tincidunt. Mauris vel congue neque. Sed leo lectus, vulputate sed varius id, pellentesque at quam. Pellentesque tortor nisl, vestibulum ut rutrum



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere eros vel risus varius et euismod dui tincidunt. Nulla et ipsum sem, eget pellentesque dolor. Suspendisse vel tellus nisi. Sed tincidunt adipiscing venenatis. Integer vestibulum nibh sed quam aliquet tincidunt. Mauris vel congue neque. Sed leo lectus, vulputate sed varius id, pellentesque at quam. Pellentesque tortor nisl, vestibulum ut rutrum

HTML5
CSS3
JAVASCRIPT



CSS

Position - Exemplo Fixed

```
.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  background-color: white;  
}
```

```
<div class="fixed">  
  conteúdo da divA  
</div>
```

HTML5
CSS3
JAVASCRIPT



CSS

Position - Exemplo Relative

```
.relative1 {  
  position: relative;  
}  
  
.relative2 {  
  position: relative;  
  top: -20px;  
  left: 20px;  
  background-color: white;  
  width: 500px;  
}
```

```
<div class="relative1">  
  conteúdo da divA  
</div>
```

```
<div class="relative2">  
  conteúdo da divB  
</div>
```

HTML5
CSS3
JAVASCRIPT



CSS

Position - Exemplo Absolute

```
.relative {  
  position: relative;  
  width: 600px;  
  height: 400px;  
}
```

```
.absolute {  
  position: absolute;  
  top: 120px;  
  right: 0;  
  width: 300px;  
  height: 200px;  
}
```

```
<div class="relative">  
  relative  
</div>  
<div class="absolute">  
  absolute  
</div>
```

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **font-family:** Define a família da fonte utilizada.
Exemplos: font-family: Arial, Verdana, Geneva, sans-serif;
- **font-style:** Define a propriedades de estilos que podem ser: normal, italic ou oblique.
Exemplos:
font-style: italic;
font-style: normal;

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **font-weight** : Define a propriedade de intensidade de uma fonte na sequência de valores de '100' a '900', cada número indica uma fonte mais intensa (escura) que o valor anterior.
Exemplos:
font-weight: normal; (ou valores de 100 a 500)
font-weight: bold; (ou valores de 600 a 900)
- **font-size**: Define o tamanho da fonte.
Exemplo: font-size: 12pt;
- **letter-spacing**: Controla o espaçamento entre as letras de um texto. Exemplo: letter-spacing: 5px;

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **word-spacing:** Controla o espaçamento entre as palavras de um texto.
Exemplo: word-spacing: 10px;
- **line-height:** Controla a altura entre as linhas do texto de um paragrafo.
Exemplo: line-height: 25px;
- **text-align:** Controla o posicionamento horizontal do conteúdo de um elemento. Os valores possíveis são: left, right, center e justify.
Exemplo: text-align: center;

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **text-decoration:** Define um efeito decorativo no texto. Podendo entre eles ser: none (sem decoração); underline (sublinhado); line-through (linha cortando o texto); e blink (efeito piscante).
Exemplo: text-decoration: underline;
- **text-indent:** Define o tamanho da endentação para a primeira linha do texto contida em um bloco, ou seja, o deslocamento para a direita de um parágrafo.
Exemplo: text-indent: 20px;

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **text-transform:** Controla os efeitos de capitalização do texto. Com ela, podemos definir a caixa das letras (minúsculas e maiúsculas). Os valores possíveis são:
 - capitalize – transforma o primeiro caracter de cada palavra em maiúscula
 - uppercase – transforma todas as letras de todas as palavras em maiúsculas
 - lowercase – transforma todas as letras de todas as palavras em minúsculas
 - none – cancela algum valor que tenha sido herdado

Exemplo: text-transform: uppercase;

HTML5
CSS3
JAVASCRIPT



CSS

Principais Propriedades

- **color:** Define a cor do texto de um texto.
Exemplos:
 - `color: red;`
 - `color: rgb(255,0,0);`
 - `color: #FF0000;`
- **width:** Define o comprimento (largura) de um elemento.
Exemplo: `width: 100px;`
- **height:** Define a altura de um elemento.
Exemplo: `height: 50px;`

HTML5
CSS3
JAVASCRIPT



OBRIGAD@



by NUAGE
IT's everywhere.