



**DIO
CEZAR
GO**

CSS3



AGENDA

- o que é?
- quais as novidades?
- uma rápida revisão;
- id e class;
- seletores complexos;
- pseudo-seletores;
- exemplos rápidos - gradiente;
- exemplos rápidos - transform;
- exemplos rápidos - transições;
- exemplos rápidos - animações;
- exemplos rápidos - font-face;
- mais no material complementar;
- pseudo-seletores;
- propriedade float;
- propriedade position;
- position: fixed;
- position: relative;
- position: absolute;
- z-index;
- exemplo ao vivo;
- materiais complementares;

O QUE É?

- o **CSS formata a informação** entregue pelo HTML;
 - essa informação pode ser qualquer coisa:
 - imagem, texto, vídeo, áudio...
- o **HTML5** trouxe poucas alterações;
 - novas tags;
 - alteração do significado de algumas tags;
 - outras foram descontinuadas;
- já o **CSS3** trouxe mudanças drásticas para a manipulação visual dos elementos do HTML;

O QUE É?

- com as versões anteriores nós podíamos formatar algumas características básicas:
 - cores;
 - backgrounds;
 - fonts;
 - margins;
 - paddings;
 - e uma maneira bem artesanal de controlar os posicionamentos;

O QUE É?

- a idéia do **CSS3**: trazer mais funcionalidades e menos dependências de tecnologias “auxiliares”:
 - *Photoshop* para criar detalhes de *layout*;
 - *JavaScript* para tratarmos comportamentos ou manipularmos elementos específicos no *HTML*;
 - acessibilidade e *SEO*;

QUAIS AS NOVIDADES?

- com as atualizações do **CSS3**, conseguimos:
 - selecionar primeiro e último elemento;
 - selecionar elementos pares ou ímpares;
 - selecionarmos elementos específicos de um determinado grupo de elementos;
 - gradiente em textos e elementos;
 - bordas arredondadas;
 - sombras em texto e elementos;
 - manipulação de opacidade;
 - controle de rotação;
 - controle de perspectiva;
 - animação;
 - estruturação independente da posição no código HTML;

UMA RÁPIDA REVISÃO

a sintaxe dos seletores é bem simples:

SINTAXE BÁSICA DE UM SELETOR EM CSS

```
seletor{  
    propriedade: valor;  
}
```

- *seletor* → é um forma de indicar qual elemento será modificado;
- *propriedade* → é a característica que deseja modificar no elemento;
- *valor* → é o valor referente a esta característica;

UMA RÁPIDA REVISÃO

- o que é um seletor?
 - um seletor representa uma estrutura;
 - essa estrutura é usada como uma condição para determinar quais elementos de um grupo de elementos serão formatados;
 - seletores agrupados e encadeados são a base do CSS;
 - e acredite... você só aprende por *osmose* durante o dia-a-dia de desenvolvimento;

DICA → sempre pesquise uma forma de selecionar elementos específicos para fixar cada uma de suas utilidades.

UMA RÁPIDA REVISÃO

um exemplo de seletor encadeado:

SELETOR ENCADEADO

```
div p strong a {  
    color: red;  
}
```

*este seletor formata o link (**a**), que está dentro de um **strong**, que está dentro de **P** e que por sua vez está dentro de um **DIV**.*

UMA RÁPIDA REVISÃO

um exemplo de seletor agrupado:

SELETOR AGRUPADO

```
strong, em, span {  
    color: red;  
}
```

*este seletor formata todos elementos **strong**, **em** ou **span***

DICA → agrupe elementos separados por vírgula para que herdem a mesma formatação.

ID E CLASS

- é a forma básica de se nomear os elementos em HTML5;
- um **ID** deve ser atribuído à um elemento único;
 - nunca deve ser repetido;
 - exemplo de utilização: `<div id="my-id"></div>`;
- uma **class** deve ser atribuída a elementos que têm propriedades em comum:
 - podem ser utilizadas por mais de um elemento;
 - pode-se ter mais de uma classe por elemento;
 - exemplo de utilização: `<div class="color-blue big-size"></div>`

SELETORES COMPLEXOS

- os seletores básicos são para cobrir suas necessidades básicas de formatação de elementos;
- mas o que fará você escrever menos código, e principalmente menos *JavaScript* são os seletores *complexos*;
 - eles selecionam elementos que talvez você precisaria fazer algum *script* em JavaScript para poder marcá-lo com uma CLASS ou um ID para então você formatá-lo.

SELETORES COMPLEXOS

Imagine que você tenha um título (**h1**) seguido de um parágrafo (**p**).
você precisa selecionar todos os parágrafos que vem depois de um título **h1**. Com os seletores complexos você fará assim:

SELETOR AGRUPADO

```
h1 + p {  
    color:red;  
}
```

tornam-se muito úteis ao trabalhar com animações e pseudo-seletores

*lista atualizada pelo **W3C** → <http://www.w3.org/TR/css3-selectors/#selectors>*

SELETORES COMPLEXOS

PADRÃO	SIGNIFICADO	CSS
elemento[atr]	Elemento com um atributo específico.	2
elemento[atr="x"]	Elemento que tenha um atributo com um valor específico igual a "x".	2
elemento[atr~="x"]	Elemento com um atributo cujo valor é uma lista separada por espaços, sendo que um dos valores é "x".	2
elemento[atr^="x"]	Elemento com um atributo cujo valor comece exatamente com string "x".	3
elemento[atr\$="x"]	Elemento com um atributo cujo valor termina exatamente com string "x".	3

SELETORES COMPLEXOS

PADRÃO	SIGNIFICADO	CSS
elemento[atr*="x"]	Elemento com um atributo cujo valor contenha a string "x".	3
elemento[atr ="en"]	Um elemento que tem o atributo específico com o valor que é separado por hífen começando com EN (da esquerda para direita).	2
elemento:root	Elemento root do documento. Normalmente o HTML.	3
elemento:nth-child(n)	Selecione um objeto N de um determinado elemento.	3
elemento:nth-last-child(n)	Seleciona um objeto N começando pelo último objeto do elemento.	3

SELETORES COMPLEXOS

PADRÃO	SIGNIFICADO	CSS
elemento:empty	Seleciona um elemento vazio, sem filhos. Incluindo elementos de texto.	3
elemento:enabled elemento:disabled	Seleciona um elemento de interface que esteja habilitado ou desabilidade, como selects, checkbox, radio button etc.	3
elemento:checked	Seleciona elementos que estão checados, como radio buttons e checkboxes.	3
E > F	Seleciona os elementos E que são filhos diretos de F.	2
E + F	Seleciona um elemento F que precede imediatamente o elemento E.	2

PSEUDO SELETORES

especificam um estado especial do elemento selecionado;

PSEUDO SELETORES

```
a:hover {  
    color:red;  
}
```

PRINCIPAIS PSEUDO SELETORES

PADRÃO	SIGNIFICADO
elemento:active	aplica as regras quando um elemento é ativado. Por exemplo, quando clicamos em um link e não soltamos o botão do mouse.
elemento:hover	quando passamos o mouse em cima do elemento.
elemento:visited	quando o link é visitado.
elemento:focus	quando um elemento recebe foco. Muito utilizado em campos de texto.

EXEMPLOS RÁPIDOS - GRADIENTE

o **que é?** → possibilita a criação de um background degradê;

COMO FAZER?

```
div {  
    background: red;  
    background: -webkit-linear-gradient(red, yellow);  
    background: -o-linear-gradient(red, yellow);  
    background: -moz-linear-gradient(red, yellow);  
    background: linear-gradient(red, yellow);  
}
```

EXEMPLOS RÁPIDOS - TRANSFORM

o que é? → manipula a forma com que o elemento aparecerá na tela;

scale → modifica as dimensões de um elemento;

skew → modifica os ângulos de um elemento;

translation → move o elemento no eixo X e Y;

rotate → rotaciona o elemento;

COMO FAZER?

```
div {  
  transform: scale(1.5) skew(30deg) rotate(20deg);  
  -webkit-transform: scale(1.5) skew(30deg) rotate(20deg);  
  -moz-transform: scale(1.5) skew(30deg) rotate(20deg);  
  -o-transform: scale(1.5) skew(30deg) rotate(20deg);  
}
```

EXEMPLOS RÁPIDOS - TRANSIÇÕES

o que é? → possibilita a criação de transições;

COMO FAZER?

```
div {  
    color: white;  
    background: gray;  
    transition: all 0.5s linear;  
    -webkit-transition: all 0.5s linear;  
}  
div:hover {  
    color: white;  
    background: gray;  
    transition: all 0.5s linear;  
    -webkit-transition: all 0.5s linear;  
}
```

EXEMPLOS RÁPIDOS - ANIMAÇÕES

o que é? → possibilita criar animações em css;

DEFININDO A ANIMAÇÃO

```
@-webkit-keyframes rotate{  
  from {  
    -webkit-transform:rotate(0deg);  
  }  
  to {  
    -webkit-transform:rotate(360deg);  
  }  
}
```

APLICANDO A ANIMAÇÃO

```
div:hover{  
  -webkit-animation-name: rotate;  
  -webkit-animation-duration: 0.5s;  
  -webkit-animation-timing-function: linear;  
  -webkit-animation-iteration-count: infinite;  
  -webkit-animation-direction: alternate;  
}
```

EXEMPLOS RÁPIDOS - FONT-FACE

o que é? → possibilita utilizar uma fonte específica em seu site;

COMO FAZER?

```
@font-face {  
    font-family: 'helveticoneue';  
    src: url(helveticaNeueLTStd-UltLt.otf);  
}  
  
div{  
    font-family: 'helveticoneue';  
}
```

VEJAM MAIS NO MATERIAL COMPLEMENTAR

- columns;
- bordas;
- múltiplos background;
- cores;

PROPRIEDADE FLOAT

- é a propriedade que define onde um elemento deverá “**flutuar**”
- é um tanto quanto imprevisível;
- por definição:
 - `float:left` → alinha o elemento a esquerda;
 - `float:right` → alinha o elemento a direita;
 - `float:none` → restaura o valor original;

PROPRIEDADE POSITION

- **por que ainda utilizar?**
 - retrocompatibilidade;
- **problemas conhecidos** → centralização de elementos;
- essencialmente **não** é só ela quem diagrama os *layouts*;
 - isso se faz em conjunto com a propriedade: *float*;

PROPRIEDADE POSITION

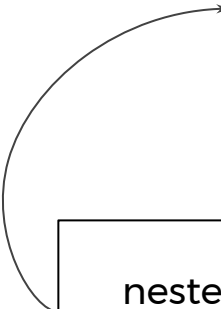
- **coordenadas!**

- são a base para o entendimento dos elementos em css através de: `relative`, `absolute` e `fixed`;
- utilizam as propriedades: `top`, `left`, `right` ou `bottom`;
- todos os valores de `position` trabalham com essas coordenadas;
- obviamente se você define `left`, não precisa definir `right`;
- e o mesmo serve para `top` e `bottom`;

PROPRIEDADE POSITION

UTILIZANDO AS COORDENADAS PARA POSICIONAR UMA DIV

```
div {  
  position: absolute;  
  top: 150px;  
  left: 150px;  
}
```



neste caso, a div está posicionada a *150px* do topo e *150px* da esquerda

agora podemos falar dos tipos de position: **relative**, **absolute** e **fixed**

PROPRIEDADE POSITION : FIXED

- esta propriedade irá **fixar** a posição do elemento na coordenada que você definir;
- a medida que se faz o *scroll* o elemento continuará fixo na posição que foi definida;
- o conteúdo continua rolando normalmente;
- geralmente é utilizado para fixar cabeçalhos ou sidebars;

PROPRIEDADE POSITION : FIXED

EXEMPLO DE UTILIZAÇÃO DO FIXED

```
div#sidebar {  
    width: 200px;  
    height: 300px;  
    background: #75AD45;  
    border: 2px solid black;  
    position: fixed;  
    top: 30px;  
    right: 30px;  
    color: white;  
    padding: 10px;  
}
```

PROPRIEDADE POSITION : RELATIVE

- todos os *positions* precisam de um **ponto** para iniciar o cálculo da coordenada;
- utilizam este ponto para posicionar o elemento na tela;
- este ponto é o canto **superior esquerdo** do elemento;
- a partir deste ponto, o navegador irá calcular a coordenada que você definiu e irá posicionar o elemento no *viewport* (elemento externo ao em questão).
- o **position relative** posiciona o elemento em relação a **si mesmo**;
- utilizado para posicionar elementos lado a lado, e deixá-los quebrar a linha automaticamente (quando não houver mais espaço);
- utilizado quando não se necessita de um posicionamento preciso, ele se orienta por seu próprio ponto de referência;

PROPRIEDADE POSITION : ABSOLUTE

- o **position absolute** posiciona o elemento em relação ao seu pai (que contenha um `position absolute` ou `relative`);
- se nenhum div pai for encontrado, utiliza como referência o body;
- posiciona exatamente o elemento com base nos atributos de posicionamento: `top`, `left`, `right` ou `bottom`;

PROPRIEDADE Z-INDEX

- ocasionalmente você precisará colocar um elemento acima do outro;
- funciona apenas para elementos com position: fixed ou absolute;
- isso se consegue com a propriedade z-index;
- quem tiver maior número fica acima;

EXEMPLO Z-INDEX

```
div {  
    position: absolute;  
    top: 150px;  
    left: 150px;  
    z-index: 150;  
}
```

EXEMPLO AO VIVO

- criar um exemplo de utilização do pseudo seletor **hover**;
- criar um exemplo de div com background **gradiente**;
- criar um exemplo com a utilização da propriedade **transform**;
- criar um exemplo com a utilização da propriedade **transition**;
- criar um exemplo com a utilização da propriedade **animation**;
- criar um exemplo com a utilização da propriedade **font-face**;
- criar um exemplo com a utilização de: **float**, **relative**, **absolute**, **z-index** e **fixed**;

MATERIAIS COMPLEMENTARES

- <https://goo.gl/6VzDj4>
- <https://goo.gl/bHhALv>
- <https://goo.gl/Y2KXxK>
- <https://goo.gl/zDxSoz>
- <https://goo.gl/CDfSMV>
- <https://goo.gl/NUApxh>
- <https://goo.gl/kssk6Z>