

# Anotações de CSS

## básico ao avançado

Autor: João Pedro Meneguesso

Obs: relógio → direções top, right, bottom, left

Obs 2: Tudo que está em vermelho é um valor da propriedade

# BACKGROUND:

background-image: url (link ou arquivo.jpg) ; --> "puxa a imagem"

background-repeat: no-repeat; --> repetição do background (no-repeat = sem repetir)

background-position: (valorEsquerda valorTopo) ou (center center);  
--> posição da imagem

background-size: cover; --> tamanho/preenchimento (cover = preencher)

background-attachment: fixed ; --> fixar o background (fixed = valor)

## BORDER:

border: 5px solid black; (espessura/ estilo da borda/ cor da borda)  
--> shorthand

border-radius: valor em px ou % ; --> cantos da borda ficam redondos (reloginho)

border-style: dotted solid; --> estilo da borda, shorthand (reloginho antes do solid)

# MARGIN E PADDING

margin: 10px 10px 10px 10px; --> espaçamento externo  
(topo, direita, em baixo, esquerda - relógio)

padding: 10px 10px 10px 10px; --> espaçamento interno (relógio)

# BOX SHADOW

box-shadow: 2px 2px 2px black; --> efeito de sombra

(Distancia da sombra na horizontal, vertical, desfoque, cor da sombra)

# SELETORES E COMBINADORES

"," --> aplicar uma regra para 1 ou mais seletores

">" --> aplicar uma regra para os filhos de uma tag

"+" --> aplicar uma regra no primeiro irmão da tag que eu selecionei

"~" --> aplicar uma regra em todos os irmãos

# TEXT STYLES

text-align: (**center, right, justify etc**); --> alinhamento de texto

font-weight: **100-900 (bold etc)** ; --> intensidade da fonte

font-size: **valor em px**; --> tamanho da letra

font-style:(**normal, italic etc**); --> estilo da letra

font-variant: (**small-caps etc**) ; --> variação da fonte {as vezes não tem}

text-transform:(**capitalize, lowercase etc**); -->deixar a letra com "Caixa" "ALTA" ou "baixa"

text-decoration: (**none, underline, overline**) --> decoração do texto

text-indent: **valor em px**; --> adiciona um espaço no começo do paragrafo

letter-spacing: **valor em px**; --> espaçamento entre as letras da palavra

word-spacing: **valor em px**; --> espaçamento entre as palavras

line-height: **valor em px**; --> tamanho de uma altura de uma linha para outra

text-shadow: **aceita valores do box-shadow**; --> sombra da palavra

## DISPLAY:

display: **block**; --> se comporta como uma "caixa", ocupa a tela toda, consigo setar uma largura e altura para ela.

display: **inline** --> não se comporta como uma caixa, o conteúdo que está dentro dele define seu tamanho, tudo que ficar como inline vai ficar lado a lado

display: **inline-block** --> ele vai ser inline mas uma caixa tbm, ele anula o espaço da caixa que ocupa a tela toda



# VISIBIILITY, OPACITY, DISPLAY NONE

display: **none** ; --> remove o elemento da tela

visibility: **hidden** ; --> esconde o elemento, mas não o espaço ocupado por ele

opacity: **0 até 1** ; --> opacidade

# OVERFLOW (vazar para fora)

overflow: **hidden**; --> esconde o conteúdo vazado

overflow: **scroll**; --> permite o uso do scroll no conteúdo vazado

overflow-x: **hidden**; --> esconder o eixo X do scroll

overflow: **auto**; --> identifica automaticamente

# PSEUDO CLASSES

**:hover** --> usuario irá passar o mouse por cima do elemento selecionado, depois ocorrerá a ação

pseudo classe só para a tag "<a>" {

**:visited** --> usuario visitou o link

**:active** --> quando o link for ativo

}

**:first-child** --> primeiro filho da tag

**:last-child** --> ultimo filho da tag

**:nth-child()** --> escolho um filho pelo numero

(ex: exemplo2 **p:nth-child(3)**)

# PSEUDO ELEMENTOS

**::first-line** --> primeira linha do paragrafo

**::first-letter** --> primeira letra

@codigo exemplo para melhor compreensão@

**::before**{ --> "antes"

content: 'teste';                   \*texto\*

background: url(link.com) ; \*imagem\*

background-size: cover;       \*imagem aparece no espaço definido\*

display: inline-block;       \*fica lado a lado\*

width: 10px;                   \*largura\*

height: 10px;                 \*altura\*

content: ""; -->               \*obrigatorio no before mesmo que não receba nada\*

}

**::after**{ --> "depois"

mesma logica do before

}

**::selection**{ --> muda a cor do texto quando selecionado

color: cor que eu quiser;

background: cor que eu quiser;

# POSITION FIXED, ABSOLUTE, RELATIVE

position: **fixed** ; --> o elemento fica fixo e acompanha o scroll da tela

top: **0**; --> distancia do topo(utiliza-se com o position){tem tbm  
bottom, left, right}

position: **absolute**; --> fica parado completamente no lugar

position: **relative**; --> todos os elementos filhos absolutos vão ficar  
relativos ao pai e não ao body

# VARIÁVEIS

Para declarar uma variavel primeiro usamos nas primeiras linhas do arquivo CSS o :root{}

```
:root{  
    --nome-da-variavel ou --nomeDaVariavel: seu valor pode ser cor,  
    tamanho etc;  
}
```

Exemplo de aplicação

```
:root{  
    --colorPrimary: red;  
}
```

```
p{  
    color: var(--colorPrimary); --> a cor da letra do parágrafo vai  
    receber vermelho  
}
```

# IMPORT

`@import "arquivo.css";` --> o import vai ser util quando eu tenho varios arquivos de CSS

# FILTER

Aplica filtros de estilo em uma determinada coisa olhar na documentação do w3schools para descobrir mais filtros

```
img{  
  filter: blur(valor em px); --> funciona em qualquer coisa  
}
```



# UNIDADES DE MEDIDA

olhar o w3schools

cm, mm, in, pc, etc --> impressão de documentos

pixel --> padrão, porem quando usar responsividade ele não é muito util

Unidades relativas --> são relativas, dependem de alguma coisa

% --> valor em porcentagem

em --> depende do fontsize definido no elemento

rem --> não depende do elemento pai e sim do :root

vw --> viewPort width --> area visivel do meu navegador (width)

vh --> viewPort height

# TRANSITION

transition-duration: **500ms**(boa pratica usar só o ms); --> duração da transição

transition-property: **background-color**; --> a propriedade que eu quero priorizar para a transição {valor padrão é all}

transition-timing-function: **ease** {olhar o w3schools para ver os outros valores};--> é como a velocidade da transição se comporta

transition-delay: **500ms**; --> tempo de atraso para começar a transição

transition: **property**, **duration**, **timing-function**, **delay** ; --> shorthand

# TRANSFORM

transform: **translate()** **rotate()** **scale()** ; --> transformar{olhar o w3schools} [todos os efeitos do transform tem que ficar só em um transform]

transform: **translate(50px, 100px)**; --> parecido com o margin, porem ele é fluido, não ocupa o espaço que nem no margin

transform: **rotate(20deg)**; --> girar o elemento

transform: **scale(2, 3)**; --> altera a escala, aumenta de acordo com o width e height[o numero escolhido é multiplicado por eles], scaleX - scaleY

transform: **skew(20deg, 50deg)**; --> torcer, pode ser skewX e skewY

transform: **matrix(1)**; --> combinar todas as transformações anteriores {ordem scaleX, skewY, skewX, scaleY, translateX, translateY}

transform-origin: **20% 40%**; --> alterar o eixo{ordem X, Y}

# ANIMATIONS

Para criar animação no css --> @keyframes nomeDaAnimação

Exemplo na prática :

```
.box{  
  width: 300px;  
  background-color: black;  
  
  animation-name: minhaAnimacao; --> vou colocar aonde eu  
quero colocar a animação  
  
  animation-duration: 5000ms; --> duração da animação  
(obrigatorio se eu quiser usar o animation)  
  
  animation-iteration-count: 2 ou infinite etc; --> quantidade de  
interações/animações  
  
  animation-delay: 5000ms; --> atraso para começar a animação  
  animation-timing-function: ease-in; --> mesma logica do transition  
  animation-direction: alternate; --> praticar para entender melhor  
como funciona.  
}
```

```
@keyframes minhaAnimacao {  
  from{  
    background-color: black;  
  }  
  
  to{  
    background-color: red;  
    transform: rotate(360deg);  
  }  
}
```

PODE USAR % EX:

```
0%{  
  Background-color: black  
}
```

Entre 0 e 100 posso colocar um valor e fazer algo acontecer nesse intervalo entre eles

```
100%{  
  Background-color: red;  
}  
}
```