CSS – Cascading style sheets – Folhas de estilo em cascata

**Estilo CSS**

Para editar um estilo H1 dentro de um HTML, utilizamos um seletor, então vamos utilizar um seletor. Que deve iniciar com o que você quer editar, nesse caso o **H1** e abrir e fechar colchete

**H1{**

**}**

ps: cada linha é chamada de declaração. “font-family: Arial;” é uma declaração, “font-size:20pt;” é outra declaração

toda declaração tem “;” no final que é para finalizar a declaração e poder colocar outra.

Color: propriedade

Blue: valor

**Esse é o resultado final**

H1 {

font-family: Arial;

font-size: 20pt;

color: blue;

}

**Como criar degradê no CSS?**

Utiliza a propriedade chamada

Background-image:

Junto a função interna

Linear-gradient

**ficando:**

background-image: linear-gradient()

**Dentro dos parâmetros você deve colocar:**

DIREÇÃO, COR, COR

**Ficando:**

background-image: linear-gradient(to right, #FFFFFF, #000000)

• Direção: pode ser escrita em graus como “90deg”, “45deg” ou “to right” “to bottom” “to top” “to left”

**Aí possa ser que não fique bom, pois estará pequeno, então para ajeitar isso é bom fazer uma configuração global chamada:**

\*{

Height 100%;

}

\* - seletor global

Height = altura e o 100% indicará que utilizará sempre 100% da altura

**PS: PODE COLOCAR QUANTAS CORES VOCÊ QUISER NO DEGRADÊ, MAS O RECOMENDADO É DE 3 A 4 CORES QUE SEJAM SEMELHANTES, PARA NÃO FICAR ALGO ABSURDAMENTE GRITANTE HORRIVEL**

**Para que o degradê não fique rolando, utilize o comando:**

Background-attachment: fixed;

**Dentro do seletor onde foi colocado o degradê.**

Existe também, além do linear-gradient, o radial-gradient que é um gradiente meio circular

Body{

Background-image: radial-gradient(circle, #FFFFFF, #000000)

}

**Dá para definir o tanto que cada cor do ingrediente vai ocupar de espaço na tela colocando uma porcentagem logo após a cor**

background-image: linear-gradient(to right, #FFFFFF **5%**, #000000)

**Sempre que for utilizar o CSS, lembre de colocar a regra:**

@charset “UTF-8” – ela serve para que você não tenha problemas com caracteres especiais como acentuação e etc.

**Para arredondar um canto de caixa utilize o:**

Border-radius: 10px;

**Para colocar sombra numa caixa, utilize o:**

Box-shadow: 3px 7px 5px 10px black;

Parametrização:

1, 3 – quanto a sombra vai andar pro lado

2, 7px – quanto a sombra vai andar pra baixo

3, 5px – quanto a sombra vai espalhar

4, 10px; – um negócio la q esqueci

Black – cor da sombra

**Fontes**

Font-family: Arial, helvética, sans-serif;

São 3 fontes diferentes porque é uma **safe combination** caso o dispositivo que vá abrir não tenha Arial ele pula pra próxima. Caso queira descobrir as safe combinations vá até o W3C SCHOOL em: <http://www.w3bai.com/en-US/cssref/css_websafe_fonts.html>

Existem parâmetros genéricos que você pode colocar e todo dispositivo irá ler

Font-family: sans-serif; - todo dispositivo tem fontes sem serifa, ai o dispositivo vai escolher uma fonte sans-serif ao padrão dele

Font-family: serif; - todo dispositivo tem fontes serifadas, ai o dispositivo vai escolher uma fonte serif ao padrão dele

Font-family: monospace; - todo dispositivo tem fontes monoespaçadas, ai o dispositivo vai escolher uma fonte monoespaçada ao padrão dele

Usando o **Tamanho de Fontes**

Existem medidas absolutas e relativas

**Absolutas:** cm, mm, in, px, pt, pc

**Relativas:** em, ex, rem, vw, vh, %

**PS: basicamente só devemos utilizar: px e em. px é pixels e em é tamanho relativo ao tamanho da fonte definido.**

Exemplo:

Font-size: 10px;

**Os navegadores tem como padrão 16px de font-size.**

**Agora vamos falar sobre o peso, estilo e text-decoration**

**Peso da fonte:**

Fonte-weight: parâmetro;

Parâmetros: bold, normal, bolder, lighter ou de forma numérica de 100 a 900. Porém algumas fontes não tem essas variações escritas e muito menos as numéricas que seriam: 100, 200, 300... 900

**Estilo da fonte:**

Fonte-style: parâmetro;

Parâmetros: normal, itálico e etc...

**Decoração da fonte**

Text-decoration: parâmetro.

Parâmetros; underline (sublinhado), none e etc...

**Shorthands**

São atalhos para definir várias configurações em uma linha. Elas têm uma ordem correta a ser seguida.

**A shorthand para essas configurações de fontes que foram mostradas são:**

Font: style weight size Family

Exemplo:

Font: italic bold 1em Arial, sans-serif;

Se você não quiser por o itálico por exemplo, ou melhor, não mexer no estilo, pode fazer;

Font: bold 1em Arial, sans-serif; -> irá funcionar tranquilamente.

**Importando Fontes**

Existem alguns sites para pegar boas fontes, como:

<https://www.dafont.com/pt/>

<https://fonts.google.com/>

**Para colocar ela no seu site, existem por duas formas, externas e internas.**

**Externa:**

Basta pegar o código @import que o google.fonts oferece lá e copiar. A tag ficaria assim:

@import url(‘ ’);

**Interna:**

Após baixar a fonte se atente ao formato, existem dois tipos o otf e o ttf, para utilizar, utilize a tag:

@font-face{

Font-family: ‘deumnome’;

src: url(ocaminhodafontebaixa);

}

**Caso queira especificar o formato faça dessa forma:**

@font-face{

Font-family: ‘deumnome’;

src: url(ocaminhodafontebaixa) format();

}

Em **Format** coloque um dos seguintes parâmetros:

- opentype (otf)

- truetype (ttf)

- embedded-opentype

- truetype-aat (para Apple Advanced Typography)

**Como:**

@font-face{

Font-family: ‘fonte-teste’;

src: url(fontes/teste.otf) format(opentype);

}

**Na importação da fonte você pode especificar como quer ela já de cara, exemplo:**

@font-face{

Font-family: ‘fonte-teste’;

src: url(fontes/teste.otf) format(opentype);

font-weight: normal;

font-style: normal

}

**Para usar ela seria apenas colocar:**

h1{

font-family: ‘fonte-teste’, sans-serif;

}

**Para capturar fontes de um site que você esteja visitando temos:**

A extensão: **FONTS NINJA**

**Mas e se quiser capturar de uma imagem? Ai temos**

<https://www.fontsquirrel.com/>

<https://www.whatfontis.com/> - o melhorzinho!

<https://www.myfonts.com/>

**Alinhamentos de texto**

**Para alinhar o texto:**

Text-align: parâmetro;

Parâmetro: justify, center, right, left...

**Para dar indentação ao texto:**

Text-indent: 20px;

Você vai definir quantos pixels quer que o primeiro trecho do parágrafo tenha de espaçamento da margem.

**SELETORES!!!!!!!!!!!!!!!!!!!!!!**

**Identificando um ITEM único**

ID=”identificação”

**Como usar?** **Exemplo:**

<h1 id=”titulo1”>

</h1>

<h1 id=”titulo2”>

</h1>

<h1>

</h1>

Foi dado ao primeiro h1 um id chamado titulo1 e ao segundo um id chamado titulo2.

**NO CSS** basta especificar a formatação de cada um através de **#ID, exemplo:**

#titulo1{

}

**Se** fosse colocado dessa forma:

H1{

Background-color: green;

}

Todos os três H1 automaticamente teriam cor de fundo verde. Porém se eu fizer:

H1{

Background-color: green;

}

#titulo1{

Background-color:blue;

}

#titulo2{

Background-color:White;

}

O h1 **ID**entificado como titulo 1 ficaria com o fundo azul

O h1 **ID**entificado como titulo 2 ficaria com o fundo White

E o H1 sem **Id**entificação ficaria com a configuração geral feita para todos os H1 e teria fundo verde.

**Resumindo:**

Em HTML ID = em CSS #

Em HTML CLASS = em CSS .

**REGRA DO W3C, SIGA-A!**

Dentro de um mesmo documento, o ID ele só deve valer pra uma única modificação específica, não use o mesmo ID em outras coisas. Para isso use CLASS! Ela faz exatamente a configuração personalizada e serve exatamente para fazer a configuração em múltiplos ao mesmo tempo!

**Basicamente:**

ID – para um!!

Class – para vários!!

**Uma curiosidade:**

Se você tiver um H1 e um H2 com a CLASS igual:

<h1 class=”mudar”>

</h1>

<h2 class=”mudar”>

</h2>

Se você quiser mudar ambos de uma vez no CSS utilize:

.mudar{

}

Se quiser mudar só um utilize:

h1.mudar{

<!-- Aqui só ira mudar os H1 que estão com a classe mudar -->

}

**ATENÇÃO: É POSSÍVEL UTILIZAR DUAS CLASSES EM UMA MESMA TAG**

<h1 class=”destaque teste”>  
</h1>

Tanto destaque como teste são duas classes diferentes, não é necessário separar por vírgula. Ou seja, se aplicássemos o estilo:

.destaque{

Background-color:red;

}

.teste{

color:white;

}

O nosso h1 ficaria com o fundo vermelho e a cor da letra branca.

**ORDEM DE PRECEDÊNCIA**

De acordo com meus testes, a sobreposição funciona assim, da esquerda pra direita o mais forte sobrepõe:

ID > Class > seletor > configuração geral

Ou seja

**#teste > .teste > seletor > configuração geral ( \* )**

**PSEUDO CLASSES**

São tags que meio que configuram o estado de um determinado elemento e são determinadas por “:” como por exemplo:

Div:hover{

Background-color: red;

}

Quando eu passar o mouse em cima da div ela vai ficar com o background vermelho. Ou seja, hover significa que: quando o mouse tiver em cima vai acontecer algo.

**Para saber mais pseudo classes:**

<https://www.w3schools.com/cssref/css_selectors.asp>

**para acessar propriedades do CSS:**

<https://www.w3schools.com/cssref/>

**PSEUDO ELEMENTOS**

Tem também os pseudo elementos, que são parecido com pseudo classes e são representados por “::”. É fazer com que coisas antes ou depois apareçam. Exemplo:

Div::after {

Content: ‘zzz’;

}

Nesse exemplo, depois da minha div, vai aparecer **zzz**

**CHILDREN**

Representado por > faz com que você faça uma modificação em um filho de uma tag. Exemplo:

<div>

<p> a </p>

</div>

<p>b</p>

**No CSS:**

Div > p{

Background-color: blue;

}

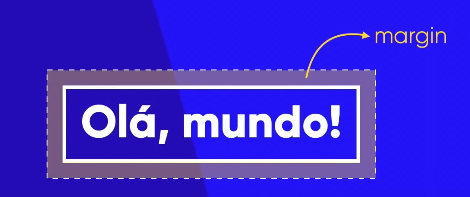
**Vai editar somente o <p> que tá dentro da div, que no caso é filho da div.** Ou seja: <p> a </p>

Vai ficar com o background-color: Blue;

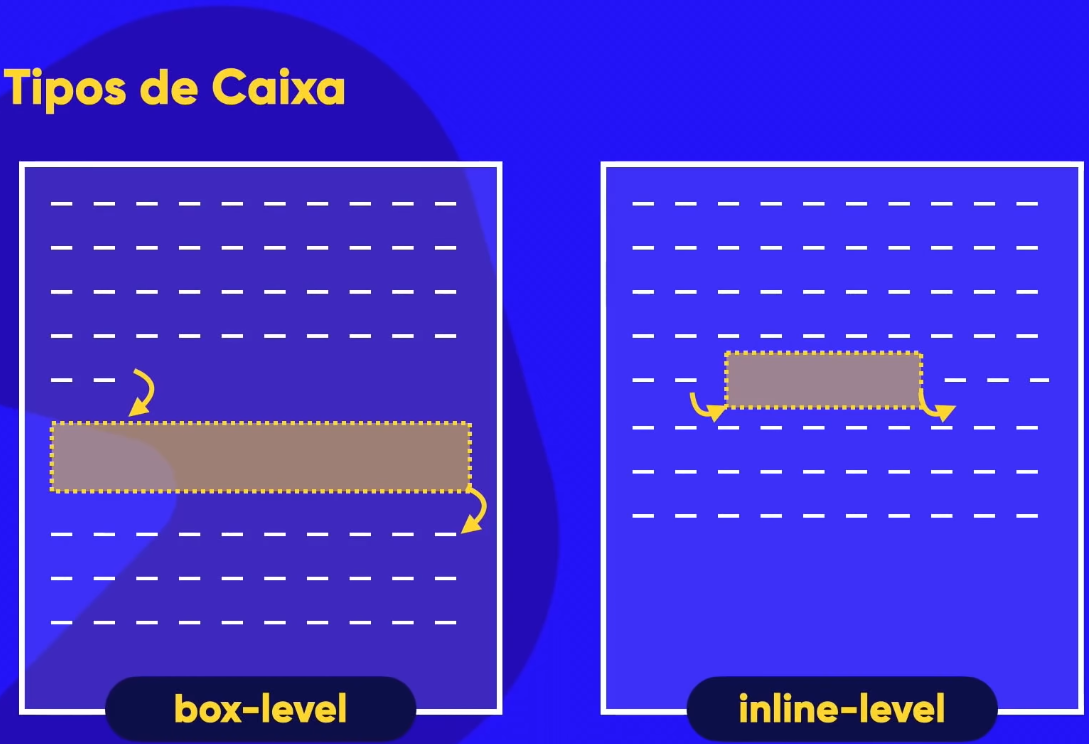
**MODELO DE CAIXAS**











**Algumas tags**

