

**Olá!**

# CSS

## Aula 04

---

1.

# Definições

Vamos  
introduzir a  
ideia do CSS e  
seus propósitos



```
.content {  
  background-color: red;  
  width: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}  
  
.content p {  
  font-size: 70px;  
  color: yellow;  
}
```



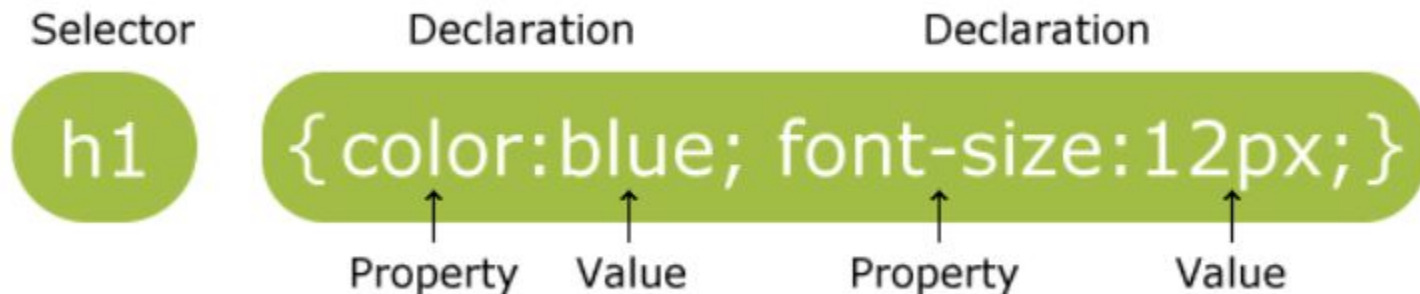
## O que é CSS?

O Cascading Style Sheet (CSS) são utilizados para definir estilos para as páginas web.

# Sintaxe

---

Um conjunto de regras em CSS consiste em um seletor e um bloco de declaração.



# Comentários



Comentários podem ser feitos em linhas ou em blocos usando `/* (...) */`.

```
/* comentário em linha */
```

```
/* comentário  
em  
bloco */
```

# Inserindo estilo na página web



Existem três formas de inserção de CSS em uma página HTML:

- Estilo em linha
- Folha de estilo interna
- Folha de estilo externa

# Estilo em linha



O estilo vem no atributo "style" dentro da tag de abertura de um elemento HTML.

```
<p style="color: gray;"> Teste de estilo </p>
```



# Folha de estilo interna



O estilo vem em uma tag `<style>` dentro do `<head>` do HTML, contendo todo o CSS utilizado.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: linen;
      }
      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1> Teste de Estilo </h1>
  </body>
</html>
```

**Teste de Estilo**

# Folha de estilo externa



O estilo vem em um arquivo separado, com extensão .css, e no HTML é referenciado por uma tag `<link>`, em seu head.

# Folha de estilo externa

```
<!DOCTYPE html>
<html>
<head>
  <title>Design Culture</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="content">
    <p> Teste de estilo </p>
  </div>
</body>
</html>
```

```
.content {
  background-color: red;
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
}
```

—————→ styles.css.

```
.content p {
  font-size: 70px;
  color: yellow;
}
```



# Exercício 1

1. Utilize um HTML de exercícios passados (podendo utilizar as resoluções que estão no github).
2. Crie uma folha de estilos externa e referencie nessa página HTML
3. Adicione um *background-color* para qualquer elemento presente em seu HTML.

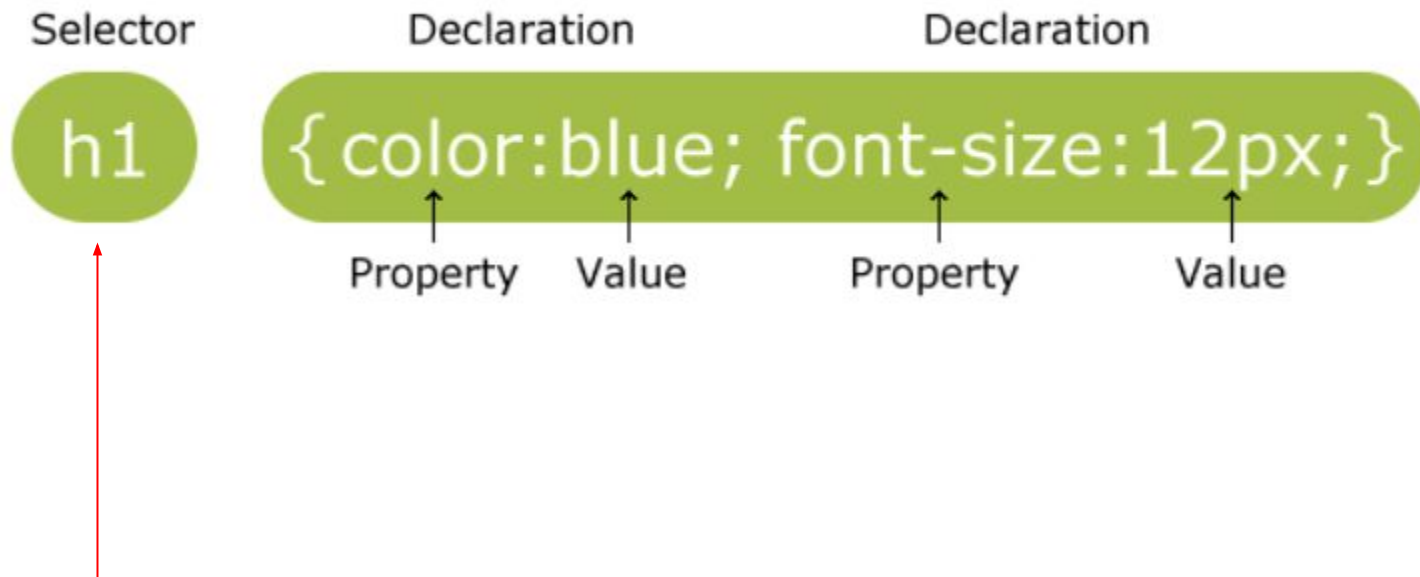
# 2.

## Seletores

Vamos  
desvendar  
como relacionar  
o CSS aos  
elementos  
HTML.

# Seletores

Seletores são os elementos identificadores do HTML ao qual se deseja atribuir estilo. Eles podem ser o próprio elemento HTML, uma classe, id dentre outros.



# Seletor por nome de elemento



Os nomes dos elementos (tags) podem ser usados para servir de ponte entre o CSS e o HTML. O exemplo abaixo retrata essa relação:

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```



# Seletor por Id



O seletor por id busca elementos HTML através do valor que foi atribuído a eles no atributo "id". No CSS, a forma de indicar que estamos estilizando a partir de um id é pelo "#" no início do seletor.

```
#title {  
    color: white;  
    text-align: center;  
}
```

# Seletor por classe



O seletor por classe busca elementos HTML através do valor que foi atribuído a eles no atributo "class". No CSS, a forma de indicar que estamos estilizando a partir de uma classe é pelo "." no início do seletor.

```
.title {  
    color: white;  
    text-align: center;  
}
```

# Agrupamento de seletores



Comumente, desejamos estilizar elementos muito específicos na árvore da página, ou colocar o mesmo estilo para diversos elementos. Para isso, os seletores podem ser agrupados, com a finalidade de reduzir a quantidade de código e deixar o mesmo mais legível.

# Agrupamento de seletores



Separando por vírgulas indica que se deseja pegar todos os elementos indicados e aplicar o mesmo estilo

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Separando por espaço indica que se deseja aplicar o estilo à todas as tags <p> que estão dentro de tags <h1>

```
h1 p {  
  text-align: center;  
  color: red;  
}
```

# Agrupamento de seletores



Separando por maior que indica que se deseja pegar todos os elementos `<p>` que tem como elemento pai o `<h1>`.

```
h1 > p {  
  text-align: center;  
  color: red;  
}
```

Separando por mais indica que se deseja aplicar o estilo à todas as tags `<p>` que vem imediatamente após as tags `<h1>`

```
h1 + p {  
  text-align: center;  
  color: red;  
}
```

# Seletores



A fim de facilitar a visualização do funcionamento de seletores, existe o [Selector Tester CSS](#) pode ser utilizado, pois dá uma referência gráfica ao que você está de fato selecionando.



Desafio

1. Utilizando o CodePen, crie uma div principal que terá uma cor de fundo e conterá em seu interior:
    - a. Outra div, ao qual deverá possuir uma distância (margem) em relação a sua div pai, uma cor de fundo diferente do seu pai e deverá conter 3 textos em seu interior:
      - i. Um deverá ficar alinhado à esquerda
      - ii. Outro deverá ficar no centro
      - iii. Outro deverá ficar alinhado à direita
      - iv. Todos deverão ter a cor branca
  2. As bordas das duas div citadas deverão ser exibidas, e com alguma coloração.
- Lembrando: o CodePen já considera a estrutura básica do HTML, logo, basta apenas elaborar o conteúdo do "body" em seu campo de HTML.



**Título Esquerda**

**Título Meio**

**Título Direita**

# 3.

## Fontes

Iremos aprender a estilizar os textos do nosso site.

# Fontes



O CSS possui várias propriedades para tratamento de fontes, dentre eles estão:

- Font-style
- Font-variant
- Font-weight
- Font-size
- Font-family

# Font-style



Especifica o estilo da fonte, aceitando três tipos: normal (padrão), itálico e oblíquo.

```
p.normal {  
    font-style: normal;  
}
```

```
p.italic {  
    font-style: italic;  
}
```

```
p.oblique {  
    font-style: oblique;  
}
```

# Font-variant



Utilizado para colocar o texto em "caps lock", mas em um tamanho menor, mantendo as propriedades das letras que já estavam em "caps lock" anteriormente.

```
p.normal {  
    font-variant: normal;  
}
```

```
p.small {  
    font-variant: small-caps;  
}
```

# Font-weight



Utilizado para especificar o volume da fonte.

```
p.normal {  
    font-weight: normal;  
}
```

```
p.thick {  
    font-weight: bold;  
}
```

# Font-size



Utilizado para especificar o tamanho da fonte.

```
h1 {  
    font-size: 40px;  
}
```

```
h2 {  
    font-size: 30px;  
}
```

```
p {  
    font-size: 14px;  
}
```

# Font-family



Utilizado para especificar a fonte que será utilizada. Pode receber mais de um nome em seu valor, por segurança, para caso o browser não encontre utilizando um dos nomes, pode tentar os demais. Nomes com mais de uma palavra devem vir entre aspas.

```
p {  
    font-family: "Times New Roman", Times, serif;  
}
```



# CSS Font-face



Utilizado para adicionar fontes de arquivos locais/remotos de forma segura, sem mais a preocupação de ela ser ou não uma fonte "web-safe".

```
@font-face {  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}  
  
div {  
    font-family: myFirstFont;  
}
```

4.

# Espaçamento

Iremos  
aprender a  
espaçar nossos  
elementos  
HTML.

# Espaçamento entre letras



O CSS possui uma propriedade chamada "letter-spacing", a qual pode ser atribuídos valores de espaçamento.

```
h1 {  
    letter-spacing: 2px;  
}  
  
h2 {  
    letter-spacing: -3px;  
}
```

# Margem



As margens de elementos HTML podem ser trabalhadas através da propriedade "margin" do CSS. Ela pode receber de um a quatro valores.

```
margin: 25px 50px 75px 100px;
```

```
margin: 25px 50px 75px;
```

```
margin: 25px 50px;
```

```
margin: 25px;
```

```
margin-left: 20px;
```

```
margin-bottom: 20px;
```

```
margin-top: 20px;
```

```
margin-right: 20px;
```

# Margem



- Quatro valores: superior, direita, inferior, esquerda
- Três valores: superior, direita e esquerda, inferior
- Dois valores: superior e inferior, direita e esquerda
- Um valor: todas as margens recebem mesmo valor

# Preenchimento

---

Os preenchimentos de elementos HTML podem ser trabalhadas através da propriedade "padding" do CSS. Ela pode receber de um a quatro valores.

```
padding: 25px 50px 75px 100px;
```

```
padding: 25px 50px 75px;
```

```
padding: 25px 50px;
```

```
padding: 25px;
```

```
padding-left: 20px;
```

```
padding-bottom: 20px;
```

```
padding-top: 20px;
```

```
padding-right: 20px;
```

# Preenchimento



- Quatro valores: superior, direita, inferior, esquerda
- Três valores: superior, direita e esquerda, inferior
- Dois valores: superior e inferior, direita e esquerda
- Um valor: todas as margens recebem mesmo valor

# 5.

## **Estilos condicionais e dimensões de tela**

Iremos  
aprender a  
adaptar nosso  
CSS para  
determinadas  
condições.



# Media Queries



São trechos do CSS que são apenas aplicados em caso de validade da condição inserida. É utilizado "@media" para indicar que o bloco CSS se trata de um condicional.

```
@media (max-width: 700px) {  
    div {  
        font-size: 100px;  
    }  
}
```

# Media Queries



Também é possível, por exemplo, definir estilos específicos para determinado tipo de orientação de tela (retrato / paisagem). O tipo "screen" indica que o estilo deve ser aplicado à telas coloridas e resoluções adequadas.

```
@media only screen and (orientation:
landscape) {
    body {
        background-color: lightblue;
    }
}
```



# Exercício 2

1. Utilize media queries para fazer um estilo para um HTML básico, onde com menos de 400px de largura a cor de fundo deve ser vermelha e com mais de 400px deverá ser roxa.

# Viewport



O viewport foi adicionado ao HTML5 para que o web developer consiga ter controle sobre o tamanho de tela ao qual vai lidar, através do elemento "meta" no "head" da página.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



Indica que a largura deverá ser a largura do dispositivo ao qual a página está aberta.



Define o zoom inicial da página ao ser aberto pelo navegador..

# 5.

## Unidades

Iremos identificar quando é melhor usar cada tipo de unidade.

# Unidades



O CSS possui diversas propriedades que definem tamanhos, e para elas existem unidades de medidas aceitas para seus valores.

Quando o valor para uma dessas propriedades for 0, a unidade poderá ser omitida.

Existem dois tipos de unidades de comprimento no CSS: unidades relativas e unidades absolutas.

# Comprimentos relativos



em	Em relação ao tamanho da fonte do elemento (2em significa 2 vezes o tamanho da fonte atual) <a href="#">Try it</a>
ex	Relativo ao x-height da fonte atual (raramente usado) <a href="#">Try it</a>
ch	Em relação à largura do "0" (zero)
rem	Em relação ao tamanho da fonte do elemento raiz
vw	Em relação a 1% da largura da viewport * <a href="#">Try it</a>
vh	Em relação a 1% da altura da viewport * <a href="#">Try it</a>
vmin	Em relação a 1% da dimensão menor da viewport * <a href="#">Try it</a>
vmax	Em relação a 1% da dimensão maior da viewport *



# Comprimientos absolutos



cm	centímetros <a href="#">Try it</a>
mm	milímetros <a href="#">Try it</a>
in	polegadas (1in = 96px = 2.54cm) <a href="#">Try it</a>
px *	pixels (1px = 1/96th of 1in) <a href="#">Try it</a>
pt	points (1pt = 1/72 of 1in) <a href="#">Try it</a>
pc	picas (1pc = 12 pt)



Dúvidas?



[github.com/drcabral](https://github.com/drcabral)



[twitter.com/DrCabrales](https://twitter.com/DrCabrales)



[diogo.cabral.dev@gmail.com](mailto:diogo.cabral.dev@gmail.com)