

# Hello World

**Módulo HTML5,CSS3 E JS**

Relembrando...

# CSS

## 1. Definições CSS

- a. Comentário, Seletor, Estilo Inline, Folha interna e externa

## 2. Seletores

- a. Espaço, vírgula, maior que e mais

## 3. Fontes

- a. Font-style, Font-variant, Font-weight, Font-size, Font-family



Sobre a aula anterior...

# Dúvidas?



# CSS

## Aula 05

Iremos aprender a  
trabalhar com o nosso  
background.



# Background

---

O CSS possui propriedades para trabalharmos com o conteúdo de fundo do nosso site. É possível manipular coisas como:

- Cor de fundo
- Imagem de fundo
- Fundo em repetição
- Posição

# Background-color

---

A propriedade background-color define a cor de fundo para o seu site, e é descrita da seguinte forma:

```
body {  
    background-color: lightblue;  
}
```

Pode-se definir a cor no CSS de três diferentes formas:

- Uma cor pré-definida (Exemplo: "red")
- Um valor HEX (Exemplo: "#b3b3ff")
- Um valor RGB (Exemplo: "rgb(255,0,0)")

# Background-image

---

A propriedade `background-image` atribui uma imagem como plano de fundo para o site. Por padrão, a imagem é repetida até que cubra todo o elemento ao qual o background está sendo alterado.

```
body {  
    background-image: url("bgdesert.jpg");  
}
```

- A repetição pode ser trabalhada através da propriedade `background-repeat`, que pode receber os valores `"repeat-y"`, `"repeat-x"` e `"no-repeat"`.



# Background-position

---

A propriedade `background-position` define a posição da imagem no background.

```
body {  
    background-position: right top;  
}
```

# Background-attachment

---

A propriedade background-attachment define que a posição da imagem no background será fixada (e não se movimentará com o scroll da página).

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

# Background Shorthand

---

Para diminuir a quantidade de código escrita para o background, é possível definir tudo de uma única vez, utilizando apenas "background" como propriedade.

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

# Time to code!

— — —

`exercicio01.html`

# Time to code

— — —

1. Usando o Codepen, crie uma página que conterá dois backgrounds:
  - a. O container externo será toda a página (mas definiremos uma altura de 1400px)
  - b. O Container interno ao primeiro terá o background com a altura de 700px;
  - c. No primeiro, crie um **background-image** com repetições horizontais e verticais e posição fixa no site. Utilize se possível uma imagem de textura.
  - d. No segundo, crie um **background-color** e **background-image** sem repetições e
2. Coloque um Título apenas para ilustrar do que se trata o site.
  - a. Adicione estilo ao título para ficar coerente com o padrão da sua página

# Time to code - exemplo

Testando Background



# Espaçamento

Iremos aprender a  
espaçar nossos  
elementos HTML.

---

# Espaçamento de 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

# Time to code!

— — —

`exercicio02.html`

# Time to code

---

1. Usando o CodePen, faça duas div que conterão:
  - a. Um background diferente cada (podendo ser imagem ou cor)
  - b. A primeira conterá uma altura de 500px e uma margem em todas as direções de 200px
  - c. A segunda conterá uma altura de 500px e um preenchimento de 200px em todas as direções.
  - d. Digite algum texto dentro dessas divs (não precisa estar dentro de alguma tag)
2. Observem o comportamento obtido

# 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;
    }
}
```

# Time to code!

---

`exercicio03.html`

# Time to code

---


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..

# 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 *

# Comprimentos 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?

---

# Contato

— — —



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



[litaa.olivera@gmail.com](mailto:litaa.olivera@gmail.com)



[linkedin.com/in/litaaoliveira](https://linkedin.com/in/litaaoliveira)



[@liitacherry](https://twitter.com/liitacherry)

