

# Hello World

**Módulo HTML5,CSS3 E JS**

Relembrando...

# CSS

1. Background
  - a. Background-color, background-image, background-repeat, background-position, background-attachment, background-shorthand
2. Espaçamento
  - a. Letter-spacing, margin, padding
3. Hover, Opacity
4. Border
  - a. Border-width, border-color, border-radius, border shorthand,
5. Box Shadow



Sobre a aula anterior...

# Dúvidas?



# CSS

Aula 06



# 1. Posicionamento

Iremos aprender a posicionar os elementos no CSS.



# Position

---

O CSS possui uma propriedade `position`, onde podem ser aplicados os valores abaixo:

- `Static`
- `Relative`
- `Fixed`
- `Absolute`

Após aplicado o valor, pode-se posicionar o elemento usando as propriedades `top`, `bottom`, `left` e `right`.

# Position: static

---

É o valor padrão, que faz com que os elementos apareçam na ordem em que estão descritos no documento HTML.

Por ser o valor padrão, não aceita as propriedades de posicionamento.



# Position: relative

---

Esse valor faz com que o cálculo para onde o elemento deve se posicionar seja relativo ao seu próprio canto superior esquerdo.

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```

# Position: fixed

---

O valor fixed faz com que elementos fiquem fixados na página, mesmo com a utilização do scroll. Pode ser utilizado para definir cabeçalhos de páginas.

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

# Position: absolute

---

Diferente do "relative", o absolute faz o posicionamento a partir do canto superior esquerdo do container ao qual ele está inserido.

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```



O elemento com position absolute se referencia pelo ponto superior esquerdo do elemento pai mais próximo que contenha também uma propriedade position relative ou absolute. Neste caso, a posição deste elemento se referenciaria pelo div com borda azul.

# Z-index

---

A propriedade `z-index` define a ordem que os elementos vão aparecer na página, permitindo sobreposição entre eles.

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

# Time to code!

— — —

Aula06/exercicio01.html

Aula06/estilo01.css

# Time to code

---

1. Faça uma página HTML que contenha um header fixo (pode ser um menu de navegação)
2. Adicione borda inferior ao menu
3. Ele deve ser fixo ao rolar página
4. Dentro da página, faça duas divs que conterão altura 1000 e uma cor de fundo diferente em casa
5. Use opacidade para as cores usadas
6. Posicione o menu de navegação da forma como o exemplo ilustrado no próximo slide

# Time to code - Resultado Esperado

[Home](#) | [Como funciona](#) | [Quem somos](#) | [Contato](#)

## 2. Visibilidade

Iremos aprender a remover ou esconder elementos com CSS.

---



# Display: none

---

Propriedade utilizada para remover elementos do HTML via CSS. Normalmente utilizada em Javascript para retirar ou recolocar elementos HTML sem precisar ficar recriando o mesmo.

# Visibility: hidden

---

Propriedade utilizada para esconder elementos HTML com uso de CSS. Ele ainda ocupará espaço no browser, mas não estará mais visível.

# Time to code!

---

Aula06/exercicio02.html

Aula06/estilo02.css

# Time to code

---

1. Usando o CodePen, coloque uma div e um título abaixo dela.
2. Adicione altura e cor de fundo para a div
3. Utilize ambas as propriedades de visibilidade, e verifique o comportamento.

# 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!

— — —

Aula06/exercicio03.html

Aula06/estilo03.css



# 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 desenvolvedor consiga ter controle sobre o tamanho de tela ao qual vai lidar, através do elemento "meta" no "head" da página.

Um site otimizado para mobile deve conter esta tag

```
<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..

# 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)

