

Semana 1 - Aula 05

CSS - Parte II

Labenu_



O que vamos ver hoje?

- Combinadores
- Seletores Especiais
- Centralização Horizontal
- Propriedade position
 - static
 - relative
 - fixed
 - absolute
 - sticky

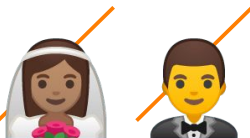


Combinadores

Labenu_



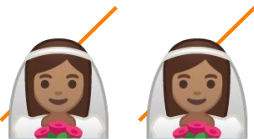
Combinadores



- Combinadores (**combinators**) são estruturas da sintaxe do CSS que permitem **combinar os seletores**. Hoje, veremos:
 - Múltiplas Condições
 - Seleção Múltipla
 - Descendentes
 - Filhos



Múltiplas Condições



- Podemos especificar ainda mais o nosso seletor
- Por exemplo: "quero que todos os **h1** com classe **texto** tenham tais características"

```
h1.texto {  
  color: green;  
  border: 1px dotted black;  
}
```

Vamos ver na prática!



Seleção Múltipla

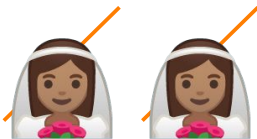
- Podemos aplicar a **mesma regra** para mais de um seletor, separando-os por **vírgula**
- Abaixo indica um código de CSS que será aplicado em todos os os elementos h1, h2 e h3

```
h1, h2, h3 {  
  color: green;  
  background-color: orange;  
}
```

Vamos ver na prática!



Filhos



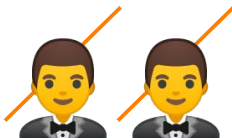
- Conseguimos especificar o seletor para que ele só atinja os **filhos diretos de um elemento específico**
- Usamos o operador **>**, desta forma:

```
div > p {  
    color: green;  
}
```

Vamos ver na prática!



Descendência



- Além de pedir só os filhos, podemos pedir para estilizar **todos os tipos** de um determinado elemento que estiverem **dentro** de outro
- Para isso, separamos os dois por um espaço

```
div p {  
    color: green;  
}
```

Vamos ver na prática!





Exercício 1

```
<h1>Sou um título GRANDÃO</h1>
<h2>Sou um título GRANDE</h2>
<h3>Sou um título MÉDIO</h3>
<h4>Sou um título PEQUENO</h4>
<h5>Sou um título PEQUENINO</h5>
<h6>Sou um título MINÚSCULO</h6>
<p> Sou um parágrafo </p>
<h2 class="texto-com-background">
| Sou um título GRANDE
</h2>
<h2
| class="texto-com-background"
| id="texto-com-outra-cor">
| Sou um título GRANDE
</h2>
```

Sou um título GRANDÃO

Sou um título GRANDE

Sou um título MÉDIO

Sou um título PEQUENO

Sou um título PEQUENINO

Sou um título MINÚSCULO

Sou um parágrafo

Sou um título GRANDE

Sou um título GRANDE



Seletores Especiais

Labenu_



Seletores Especiais

- Existem alguns seletores que tem funcionalidades e **responsabilidades especiais**
- Vamos falar de dois especificamente:
 - O seletor *
 - Pseudo-classes



Seletores *



- Permite que sejam atribuídos um conjunto de regras de estilização para **todos** os elementos

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Vamos ver na prática!



Seletores *



- Os navegadores tem alguns **estilos padrão** para os elementos HTML
- Utilizamos o seletor * para **remover esta estilização padrão**, permitindo que mexamos nos elementos crus



Seletores de Pseudo-Classes

- São seletores que permitem modificar a estilização do elemento dependendo de **estados especiais**
 - :active
 - :hover
 - :first-child e :last-child
 - :first-of-type e :last-of-type
 - :nth-of-type e :nth-child



Seletores de Pseudo-Classes

- **:active** é a pseudo-classe que representa quando o elemento é ativo pelo usuário (normalmente, pelo click)
- **:hover** é a pseudo-classe que representa quando o mouse está por cima do elemento em questão

Vamos ver na prática!



Seletores de Pseudo-Classes

- **:first-child** pega o elemento que for o primeiro filho de alguma tag
- **:last-child** pega o elemento que for o último filho de alguma tag

Vamos ver na prática!



Seletores de Pseudo-Classes

- **:first-of-type** pega o elemento que for o primeiro deste tipo dentro de uma tag
- **:last-of-type** pega o elemento que for o primeiro deste tipo dentro de uma tag

Vamos ver na prática!



Seletores de Pseudo-Classes

- **:nth-of-type/nth-child** pegar o n-ésimo elemento da pseudo-classe em questão
 - :nth-of-type(1) é igual a :first-of-type
 - :nth-child(1) é igual a :first-child

Vamos ver na prática!





Exercício 2

- Escreva o arquivo CSS necessário para montar a página ao lado
- Você já possui o body do HTML e não deve mudá-lo

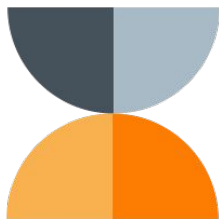
Abre google
Abre google

Olha só um paragrafo bonitinho

Mais um bonitinho

Esse já é feinho

```
<div>
  <a href="https://www.google.com.br">Abre google</a>
  <div>
    <span>
      <a href="https://www.google.com.br">Abre google</a>
    </span>
  </div>
  <p> Olha só um paragrafo bonitinho </p>
  <p> Mais um bonitinho</p>
  <span>
    <p>Esse já é feinho </p>
  </span>
</div>
```



Pausa para relaxar 🤔

10 min

- Combinadores
 - Múltiplas Condições
 - Seleção Múltipla
 - Descendência
 - Filho
- Pseudo-Classes
 - :hover e :active
 - :first-child e :last-child
 - :first-of-type e :last-of-type
 - :nth-of-type e :nth-child



Centralização Horizontal

Labenu_



Centralização

- **text-align**
 - Permite alinhar os descendentes à partir do elemento ascendente
 - **Funciona para:** texto, imagens e elementos **inline**
 - **Opções:**
 - center, right, left
 - start, end (sentidos da escrita)

Vamos ver na prática!



Centralização

- **margin**
 - Para elementos **block** com largura definida, podemos usar a propriedade **margin**
 - Dizemos para ele completar as margens da esquerda e da direita com o valor **auto**

Vamos ver na prática! 





Exercício 3

- Vamos retomar o código do remember e centralizar os botões e a tabela
- Além disso, vamos estilizar nossa tabela com mais cores

Nome	Preço
Jogos de PS4	R\$100,00
Jogos de Nintendo Switch	R\$150,00
Jogos de XBOX	R\$110,00
Jogos de PC	R\$70,00



Position

Labenu_



Position

- **position:**
 - Permite indicar o método de posicionamento do elemento
 - static
 - relative
 - fixed
 - absolute
 - sticky



Position

- **position: fixed**
 - Posiciona o elemento em relação à tela toda
 - Utilizamos as propriedades:
 - top
 - bottom
 - left
 - right

Vamos ver na prática!



Position

- **position: relative**

- Posiciona o elemento em relação à posição natural dele
- Utilizamos as propriedades:
 - top
 - bottom
 - left
 - right

Vamos ver na prática!



Position

- **position: absolute**
 - Posiciona o elemento em relação à outro elemento *relative* que o englobe
 - Utilizamos as propriedades:
 - top
 - bottom
 - left
 - right

Vamos ver na prática!



Position

- **position: sticky**
 - Posiciona o elemento de tal forma que ele fique scrollando até chegar em uma posição. Ao chegar nela, fica "grudado")
 - Para indicar onde ele gruda, usamos as seguintes propriedades:
 - top, bottom, left e right

Vamos ver na prática!



Position

- **position: static**
 - Posição padrão do CSS para todos os elementos
 - Não conseguimos usar as propriedades que vimos nos outros casos
 - ~~top~~, ~~bottom~~, ~~left~~ e ~~right~~

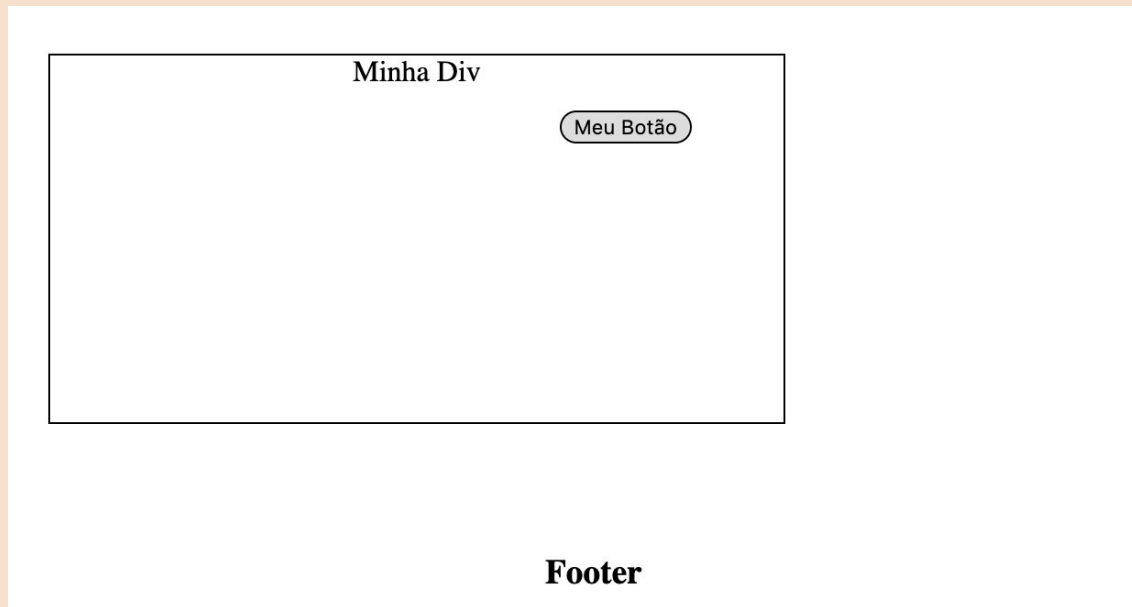
Vamos ver na prática!





Exercício 4

- Recriar utilizando HTML e CSS a tela mostrada abaixo



Resumo

Labenu_



Resumo

- **Combinadores**

- Múltiplas Condições
- Seleção Múltipla
- Descendência
- Filho



Resumo

- **Seletores especiais**

- *
- Pseudo-Classes
 - :active
 - :hover
 - :first-child e :last-child
 - :first-of-type e :last-of-type
 - :nth-of-type e :nth-child

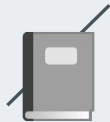


Resumo

- **text-align** permite alinhar textos, imagens e elementos inline **horizontalmente**
- Para centralizar elementos **block-level**, nós devemos colocar um :
 - **width** fixo
 - utilizar **margin-left: auto; margin-right: auto**



Resumo



- **position** permite trocar o método de posicionamento de um elemento:
 - **fixed**
 - **relative**
 - **absolute**
 - **sticky**
 - **static**
- Utilizamos as propriedades: **top**, **right**, **bottom**, **left** para realizar o posicionamento



Dúvidas? 

Labenu_





Obrigado(a)!