



Front End Estático

---

Turma 1178 | Vem Ser Tech

# Desenvolvimento Web

## → Objetivo da Aula

- Conhecer Position CSS
- Compreender FlexBox.
- Compreender Grid layout
- Compreender Media Query
- Exercício em Aula

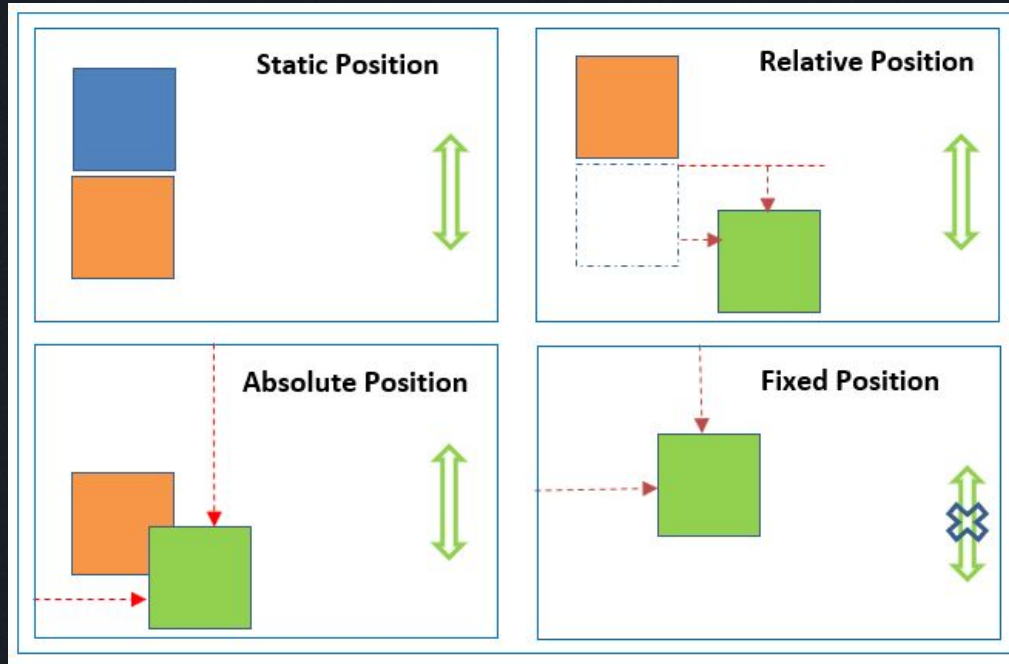
## Introdução Position CSS

# Propriedades Position

- **static:** Valor padrão, onde o elemento segue o fluxo natural da página.
- **relative:** Permite deslocar o elemento em relação à sua posição original no fluxo do documento, mantendo o espaço original.
- **absolute:** Remove o elemento do fluxo normal do documento, posicionando-o em relação ao ancestral mais próximo com position diferente de static.
- **fixed:** Similar ao absolute, mas o elemento é posicionado em relação à viewport e permanece fixo durante a rolagem.
- **sticky:** Combina características de relative e fixed, permitindo que o elemento se fixe em uma posição enquanto o usuário rola a página.



## Position CSS



## Introdução FlexBox

## O que é FlexBox?

- Um modelo de layout para melhorar a utilização do CSS3
- Ideal para criar **layouts de uma dimensão** (horizontal ou vertical), como menus, barras de navegação, ou galerias.
- **Layouts Dinâmicos**, diversidade de monitores e telas.
- Configurado por duas **Propriedades Flex Container, Flex Items**.

## Flex Container

- O elemento pai ao qual as regras de flexbox são aplicadas. Esse contêiner define um contexto de flexbox para seus filhos.



## Flex Container

Propriedades :

- `display: flex;` - Define o contêiner como um contêiner flexível.
- `flex-direction:` Define a direção dos itens no contêiner (`row`, `row-reverse`, `column`, `column-reverse`).
- `justify-content:` Alinha os itens ao longo do eixo principal (`flex-start`, `flex-end`, `center`, `space-between`, `space-around`).
- `align-items:` Alinha os itens ao longo do eixo transversal (`flex-start`, `flex-end`, `center`, `stretch`, `baseline`).
- `flex-wrap:` Controla se os itens devem quebrar para uma nova linha (`nowrap`, `wrap`, `wrap-reverse`).
- `gap:` Controla os espaços entre os itens( 10px, 20px)

## Flex Items

- Os **elementos filhos diretos** do Flex Container, que são dispostos de acordo com as regras definidas pelo Flexbox.

## Flex Items

Propriedades :

- **flex-grow:** Define quanto um item deve crescer em relação aos outros itens.
- **flex-shrink:** Define quanto um item deve encolher em relação aos outros itens.
- **flex-basis:** Define o tamanho inicial de um item antes da distribuição do espaço restante.
- **align-self:** Alinha individualmente um item ao longo do eixo transversal, sobrescrevendo align-items.



# Introdução linguagem HTML

Introdução HTML





## Exercício

- Escrever uma **página HTML**. A página deve conter:
  - Um Menu utilizando FlexBox
  - Uma Lista de imagens no mínimo 10 imagens
  - Deve se adaptar ao monitores de diferentes tamanhos
  - A primeira imagem deve ser maior que as outras, para mostrar que ela é a imagem principal.

## Introdução Grid Layout

## O que é Grid Layout?

- Um modelo de layout para melhorar a utilização do CSS3
- Ideal para criar `layouts de Linhas e Colunas`, site de notícias, portais de conteúdos
- `Layouts Dinâmicos`, diversidade de monitores e telas.
- Configurado por duas `Propriedades Grid Container, Grid Items`.

## Grid Container

- O **elemento pai** ao qual as regras de Grid são aplicadas. Esse contêiner define um contexto de Grid para seus filhos.



## grid Container

Propriedades :

- `display: grid;` - Define o contêiner como um contêiner grid.
- `grid-template-columns:` Define a quantidade de colunas do meu container
- `grid-column-gap:` Define o espaço entre as colunas.
- `grid-template-rows:` Define as propriedades de linhas explicitamente
- `grid-auto-rows:` Controla propriedades de linhas além do determinado
- `grid-gap:` Controla os espaços entre os itens( 10px, 20px)

## Grid items

- O **elemento filhos** ao qual as regras de Grid são aplicadas. Tem suas propriedades e reagem de acordo com o container.

## grid Items

Propriedades :

- `grid-column` - Define o início e término da coluna
- `grid-row`: Define o início e término da linha
- `justify-self`: Define o alinhamento horizontalmente
- `align-self`: Define o alinhamento verticalmente

## Introdução Media query



## Media Query

- **Media Queries** são uma parte fundamental do CSS que permite aplicar estilos específicos com base nas características do dispositivo, como largura da tela, resolução, orientação, entre outras.

## Media Query

```
@media (condition) {  
    /* Estilos aplicados quando a condição é verdadeira */  
}
```

# Introdução linguagem CSS

Definição

## Media Query

```
/* Estilos aplicados apenas quando a largura da tela for menor ou igual a 768px (tablets e smartphones) */
```

```
@media (max-width: 768px) {  
  body {  
    font-size: 14px;  
    background-color: lightgray;  
  }  
}
```

```
/* Estilos aplicados apenas quando a largura da tela for maior que 768px (desktops) */
```

```
@media (min-width: 769px) {  
  body {  
    font-size: 18px;  
    background-color: white;  
  }  
}
```

## Media Query

```
/* Estilos aplicados apenas em dispositivos com orientação de paisagem */
```

```
@media (orientation: landscape) {
```

```
  .container {
```

```
    width: 80%;
```

```
    margin: 0 auto;
```

```
  }
```

```
}
```

```
/* Estilos aplicados apenas em dispositivos com orientação de retrato */
```

```
@media (orientation: portrait) {
```

```
  .container {
```

```
    width: 100%;
```

```
    padding: 10px;
```

```
  }
```

```
}
```



# Introdução linguagem HTML

Introdução HTML



# Desenvolvimento Web

## → Validação Objetivo da Aula

- Conhecer Position CSS
- Compreender FlexBox.
- Compreender Grid layout
- Compreender Media Query
- Exercício em Aula