





# CSS e JavaScript

## Desenvolvimento Web I

### Objetivos:

-  Aprender fundamentos de CSS e JavaScript
-  Estilizar páginas web
-  Criar layouts modernos e responsivos
-  Aplicar boas práticas no desenvolvimento

# Atividade aula anterior

## ##Estilizando com CSS

Na proxima aula vamos aprender a estilizar as páginas com CSS.

Pesquisar sobre os principais frameworks de CSS:

- Bootstrap: <https://getbootstrap.com/>
- Materialize: <https://materializecss.com/>
- Bulma: <https://bulma.io/>
- Foundation: <https://get.foundation/>

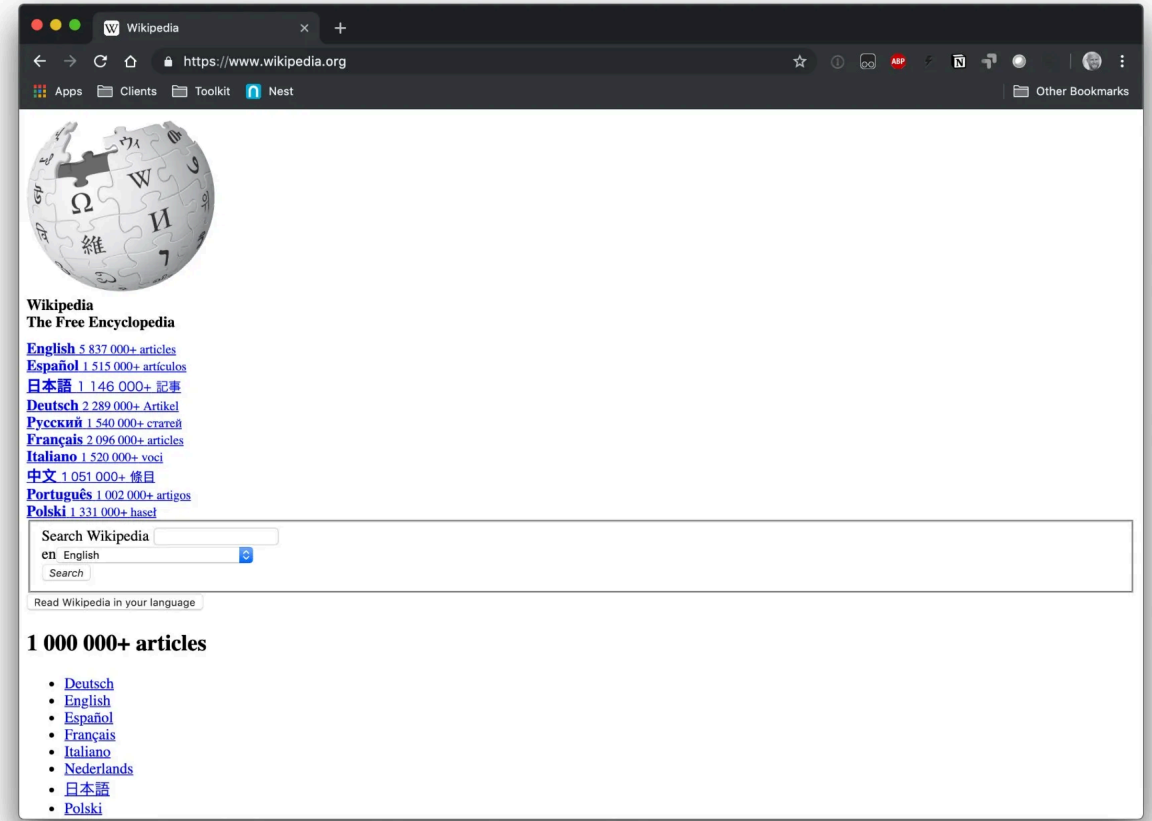
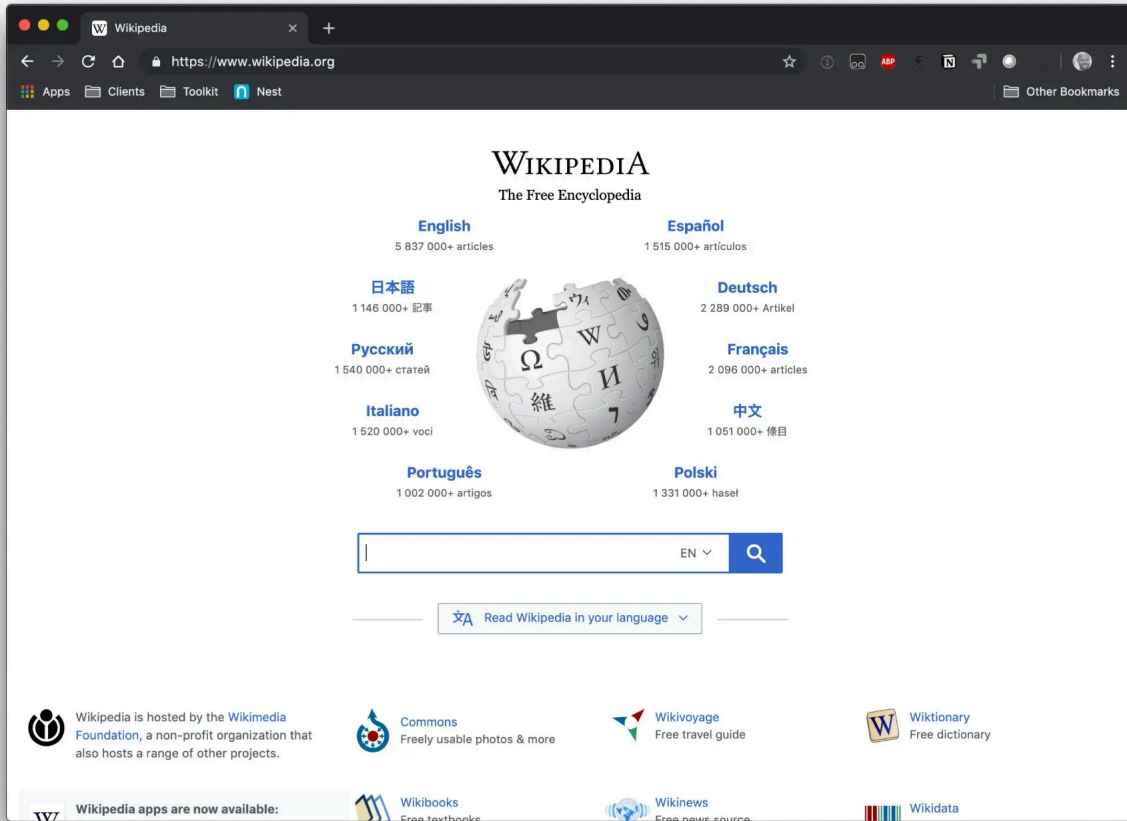
Escolher um para utilizar no projeto ao longo do curso, justificar a escolha.

## O que é CSS?

O CSS (Cascading Style Sheets) proposto em 1994.

O World Wide Web Consortium (W3C) adotou em 1996.

- CSS (Cascading Style Sheets) é usado para estilizar HTML
- O HTML não é suficiente para formatar toda a página, apenas os elementos
- O CSS permite estilizar a página inteira, com cores, fontes, tamanhos, etc.
- Animar elementos, criar layouts responsivos, etc.
- Separa o conteúdo da apresentação



# Sintaxe CSS

```
seletor {  
  propriedade: valor;  
}
```

Exemplo:

```
<style>  
  h1 {  
    color: red;  
  }  
</style>  
  
<h1>Olá, mundo!</h1>
```



# Seletores CSS

- **Elemento** ( `p` )
- **Classe** ( `.classe` )
- **ID** ( `#id` )

Os seletores podem ser combinados para estilizar elementos específicos.

# Seletores CSS

Exemplo de seletores:

## Básicos:





- `h1 { color: blue; }`
- `.classe { font-size: 18px; }`
- `#id { background-color: gray; }`

## Avançados:

- `div > p {}` → Filhos diretos
- `ul li:nth-child(odd) {}` → Ímpares
- `input[type="text"] {}` → Elemento específico

# Propriedades de Cores

## Cor do texto e fundo

-  Nome (red, blue, green)
-  Hexadecimal (#ff0000, #00ff00, #0000ff)
-  RGB (rgb(255, 0, 0))
-  Outros: RGBA e HSL

```
color: red; /* Cor do texto */  
background-color: #f4f4f4; /* Cor do fundo */  
opacity: 0.8; /* Transparência */
```



## Como aplicar CSS?

1. **Inline** ( `style="color: red;"` )
2. **Interno** ( `<style>` dentro do `<head>` )
3. **Externo** ( `<link rel="stylesheet" href="style.css">` ) -- **RECOMENDADO**



## Propriedades de Texto

```
font-family: Arial, sans-serif; /* Tipo de fonte */  
font-size: 16px;                /* Tamanho da fonte */  
font-weight: bold;              /* Peso da fonte */  
text-align: center;             /* Alinhamento */  
text-transform: uppercase;      /* Transformação do texto */  
line-height: 1.5;               /* Altura da linha */  
letter-spacing: 2px;            /* Espaçamento entre letras */
```



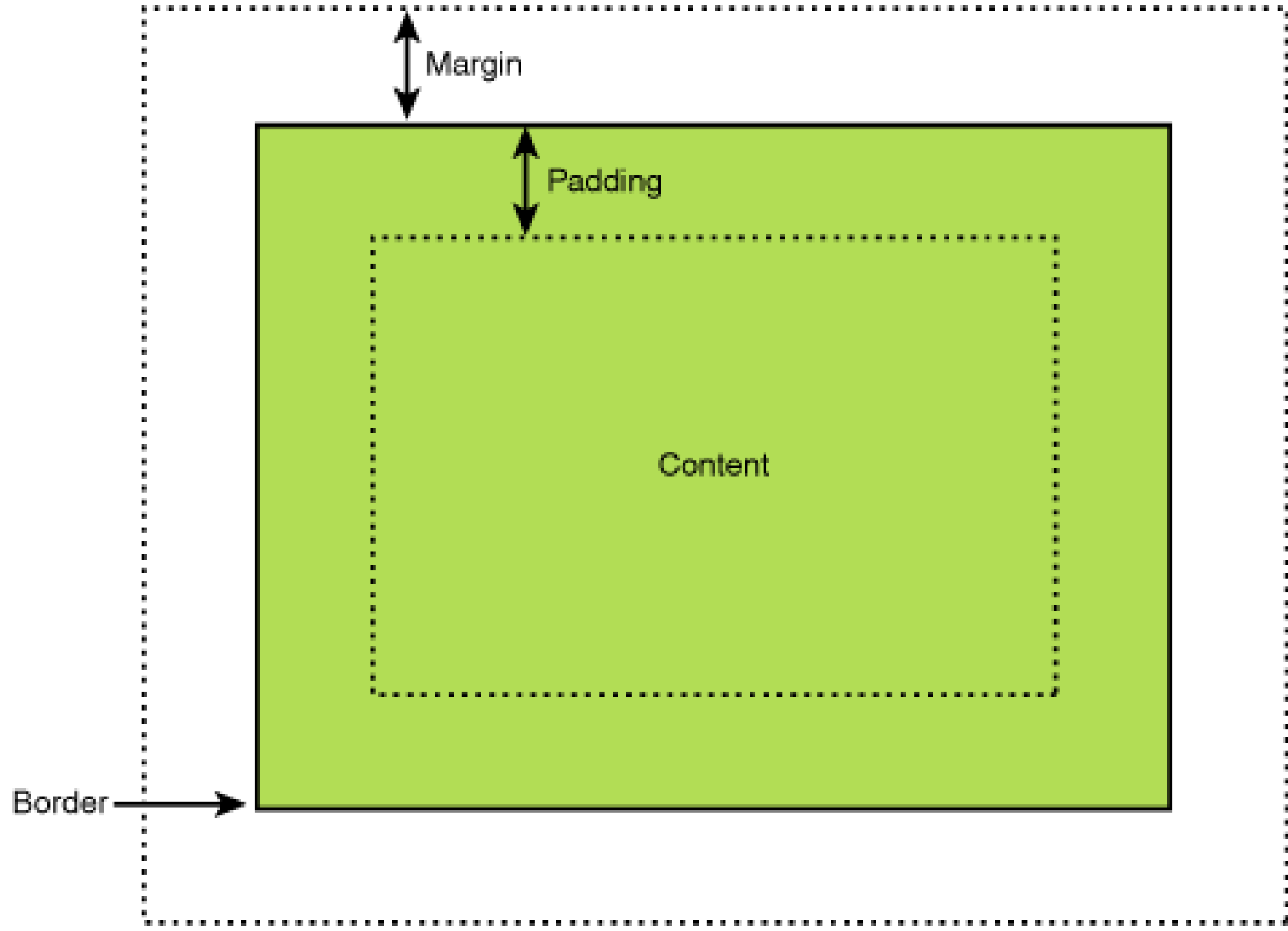
# Margens e Espaçamentos

```
margin: 10px; /* Espaço externo */  
padding: 15px; /* Espaço interno */  
border: 2px solid black; /* Borda */  
width: 300px; /* Largura do elemento */  
height: 200px; /* Altura do elemento */
```



Shorthand (atalho):

```
margin: 10px 20px 30px 40px; /* Top Right Bottom Left */  
padding: 5px 10px; /* Top/Bottom - Left/Right */
```

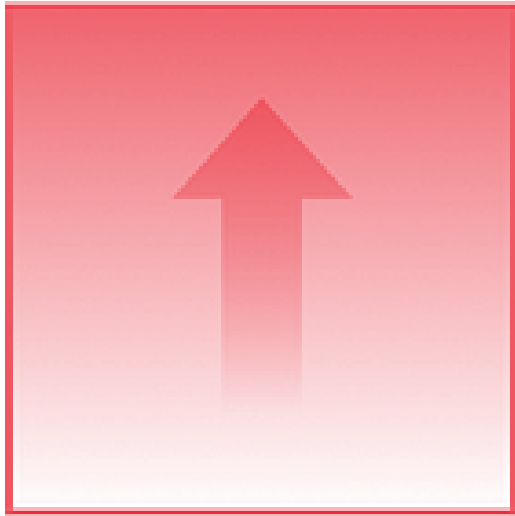




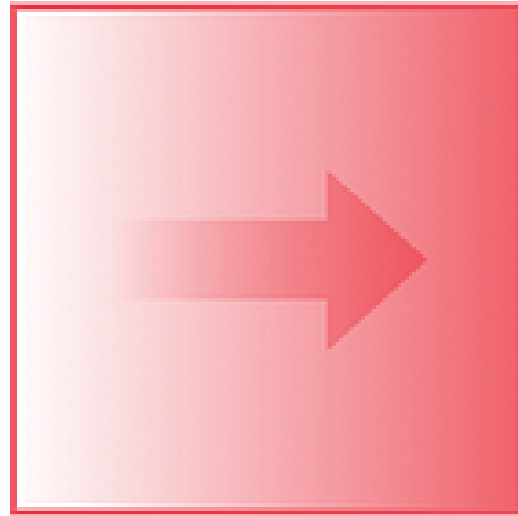
## Fundos e Gradientes

```
background-color: lightgray; /* Cor de fundo */  
background-image: url('bg.jpg'); /* Imagem de fundo */  
background-size: cover; /* Tamanho da imagem */  
background-position: center; /* Posição da imagem */  
background-repeat: no-repeat; /* Repetição da imagem */  
background: linear-gradient(to right, red, blue); /* Gradiente linear */
```

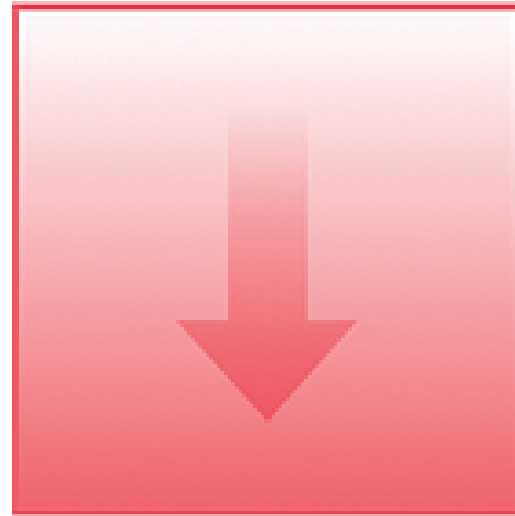
to top



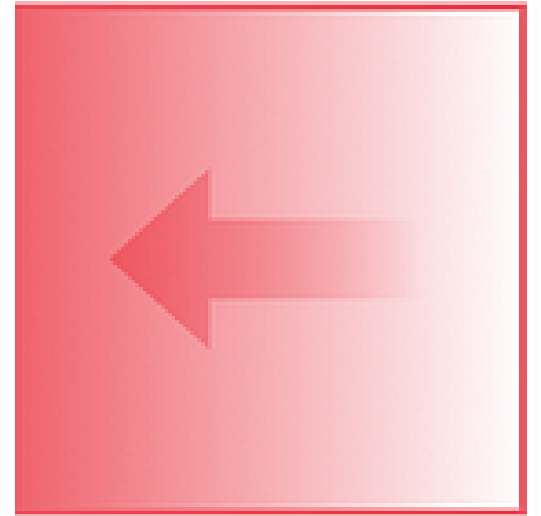
to right



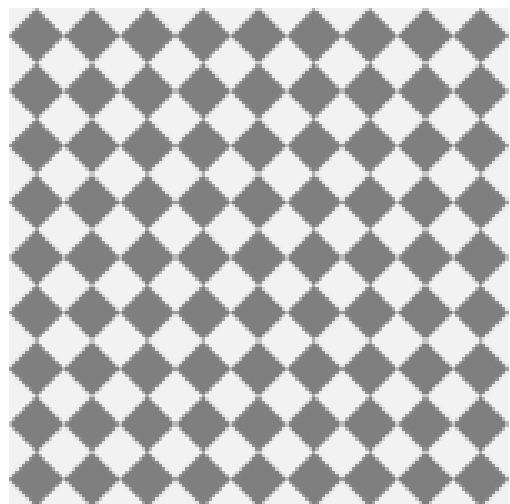
to bottom



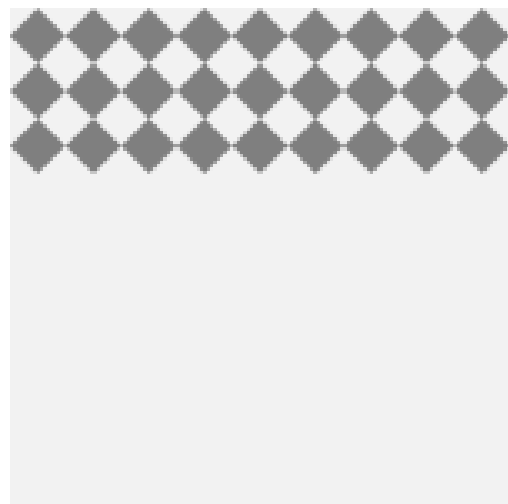
to left



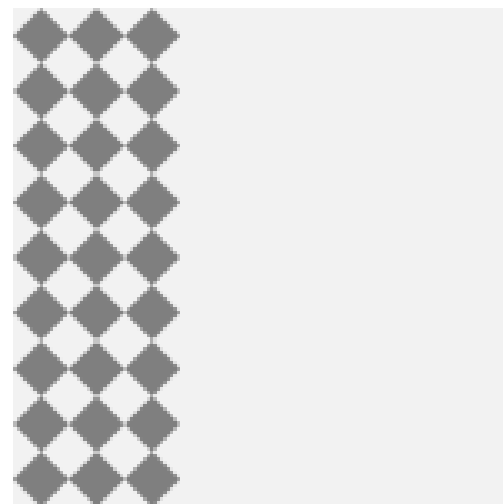




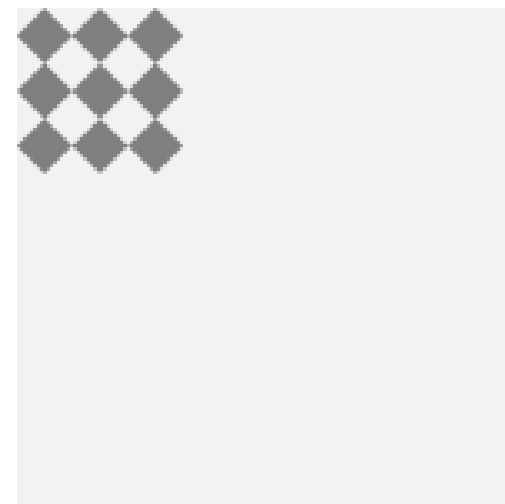
**repeat**



**repeat-x**



**repeat-y**




**no-repeat**

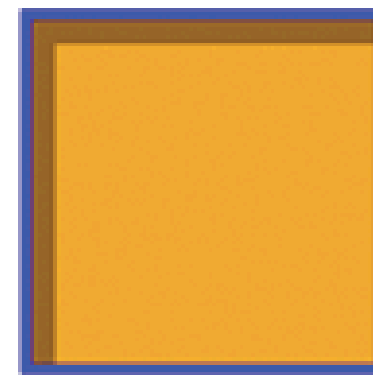
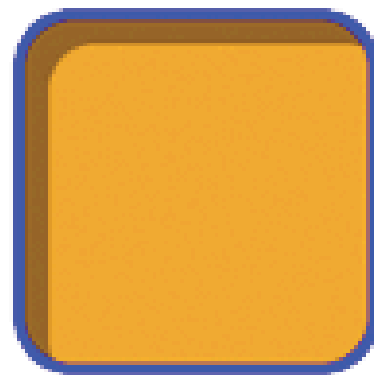
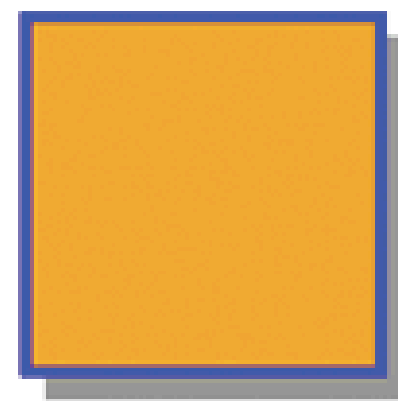
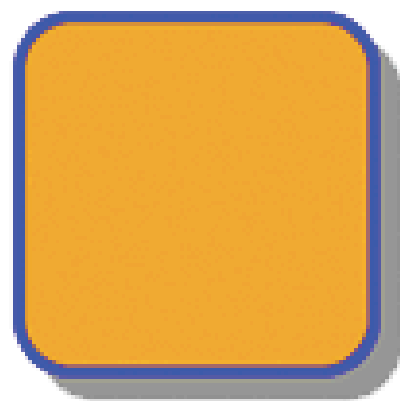
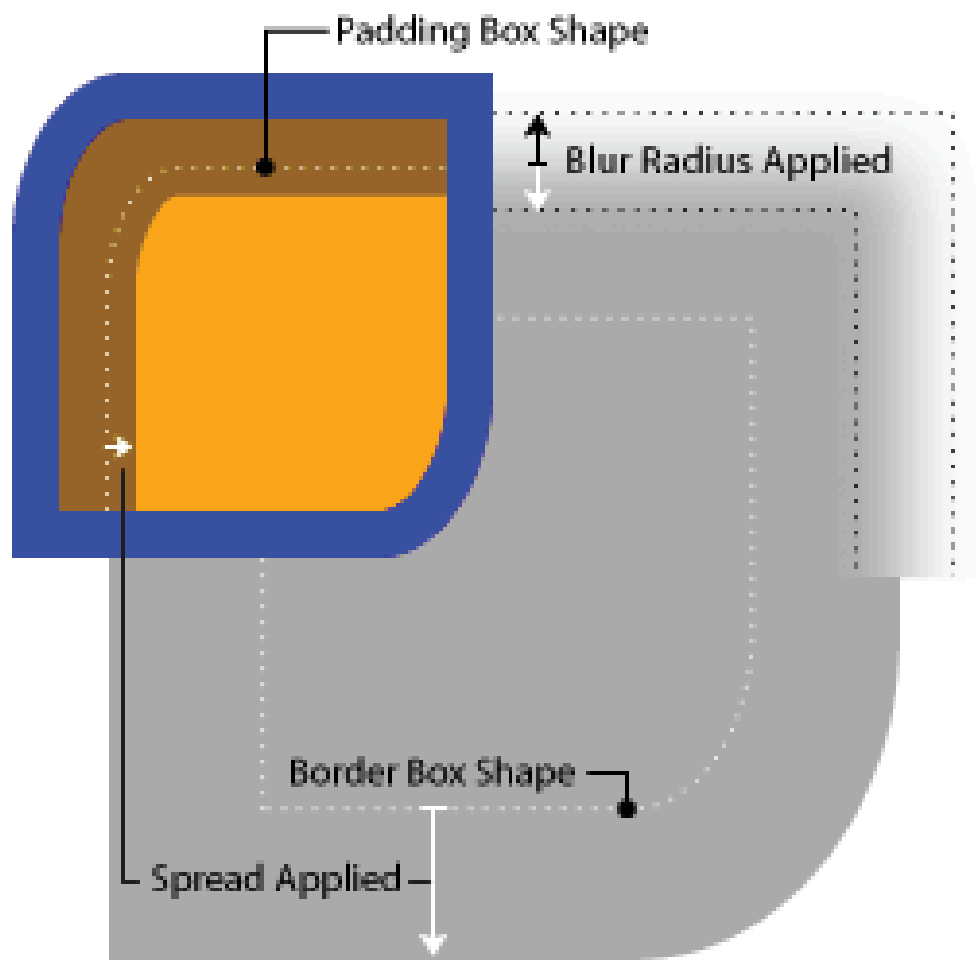


## Bordas e Sombras

```
border: 2px solid black; /* Tipo de borda */  
border-radius: 10px; /* Arredondamento */  
box-shadow: 5px 5px 10px gray; /* Sombra externa */  
text-shadow: 2px 2px 5px red; /* Sombra do texto */
```

 Tipos de bordas:

 solid | dashed | dotted | double





# Display e Posicionamento

## 1. Propriedade display

```
display: block;    /* Ocupa toda a largura */  
display: inline;   /* Ocupa apenas o necessário */  
display: flex;     /* Layout flexível */  
display: grid;     /* Layout em grade */  
display: none;     /* Esconde o elemento */
```



# CSS Display Example

Block



Inline



Inline-Block




None

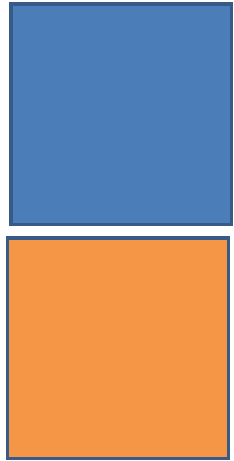


## 2. Posicionamento

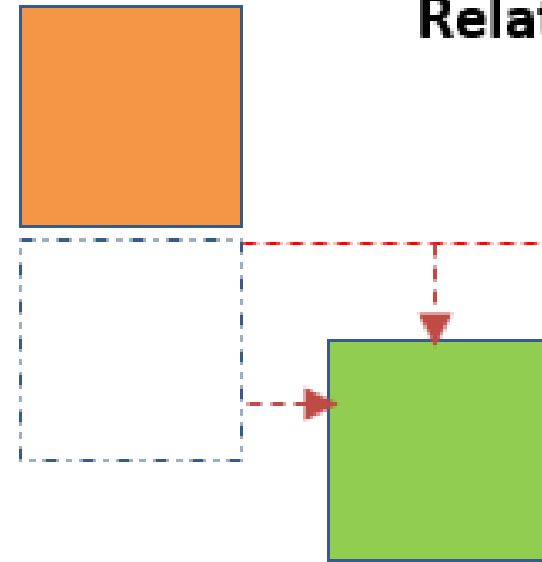
```
position: static; /* Padrão */  
position: relative; /* Relativo ao local original */  
position: absolute; /* Posição fixa na página */  
position: fixed; /* Fixo na tela */
```

 Exemplo de fixo: `position: fixed; bottom: 0;` (Rodapé fixo)

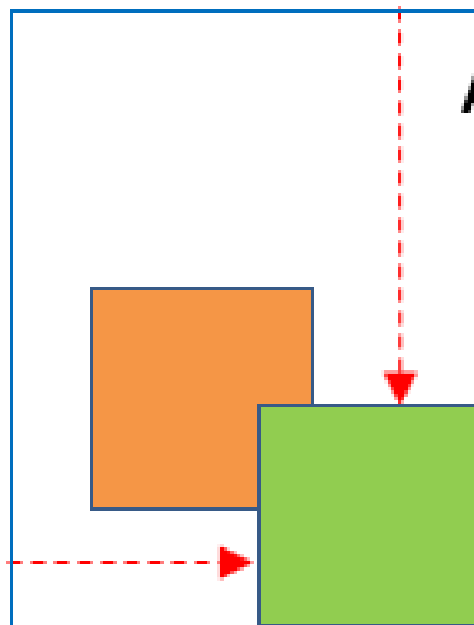
**Static Position**



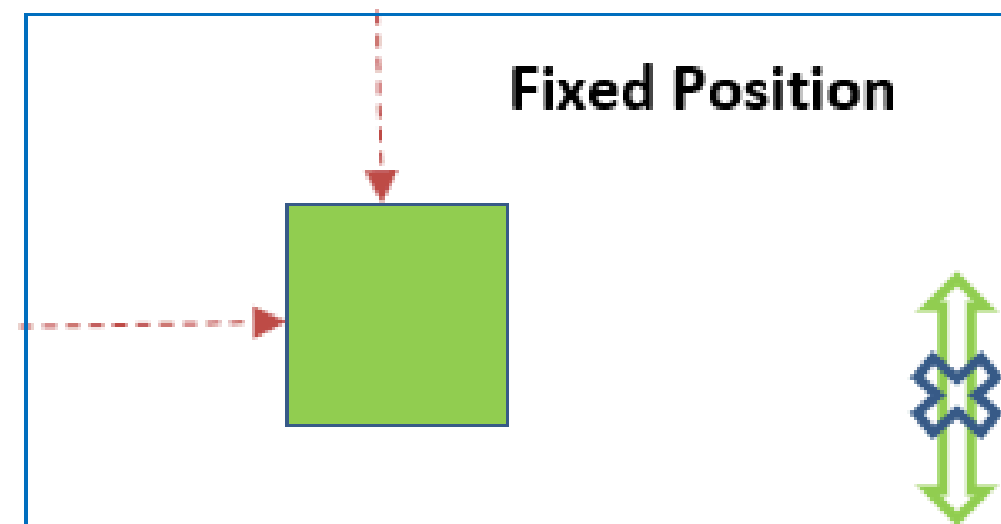
**Relative Position**



**Absolute Position**



**Fixed Position**



# Layouts com Flexbox

```
display: flex;  
justify-content: center; /* Alinhamento horizontal */  
align-items: center; /* Alinhamento vertical */  
flex-direction: row; /* Direção dos itens */  
gap: 10px; /* Espaço entre elementos */
```

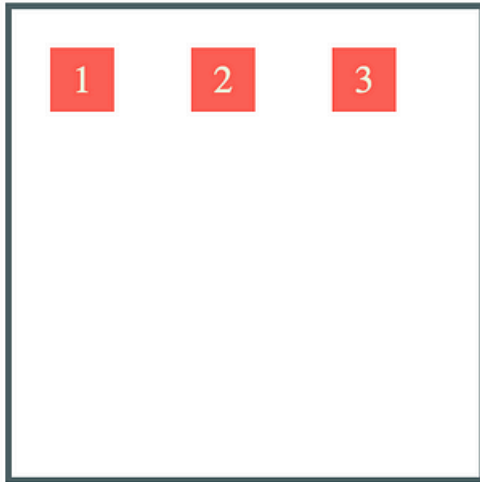
 Principais valores:

 ✓ row | row-reverse | column | column-reverse

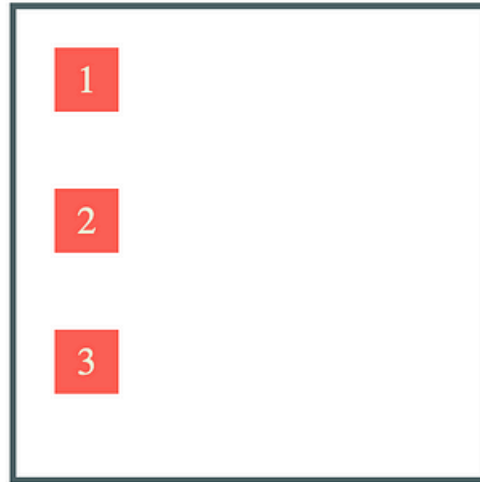
 ✓ justify-content: flex-start | center | flex-end | space-between

 ✓ align-items: flex-start | center | flex-end | stretch

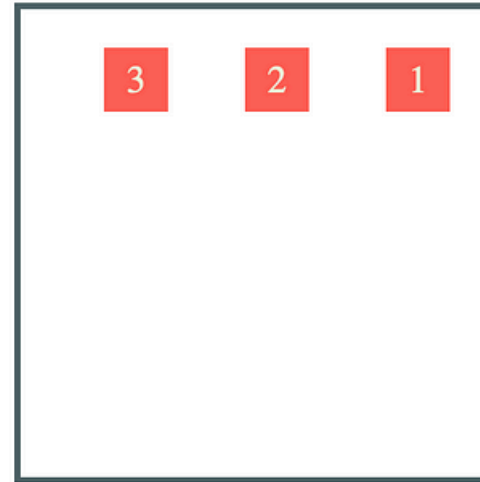
**display: flex;**  
**flex-direction: row;**



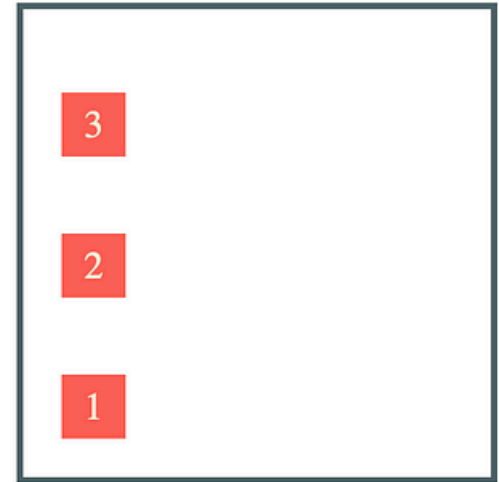
**display: flex;**  
**flex-direction: column;**



**display: flex;**  
**flex-direction: row-reverse;**



**display: flex;**  
**flex-direction: column-reverse;**





## Layouts com Grid

```
display: grid;  
grid-template-columns: repeat(3, 1fr); /* 3 colunas iguais */  
grid-template-rows: auto 200px;      /* Duas linhas */  
gap: 10px;                            /* Espaço entre os itens */
```

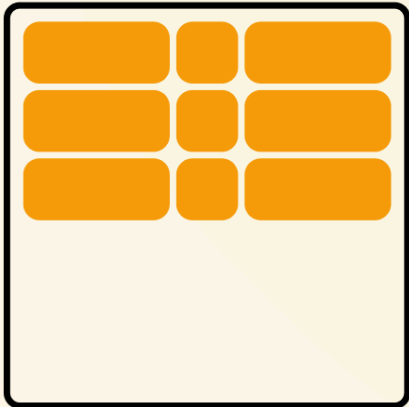
 Valores úteis:

 grid-template-columns: 1fr 2fr 1fr;

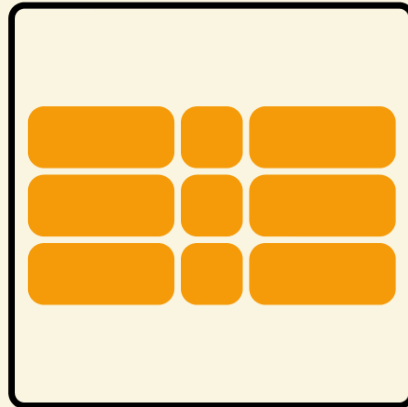
 grid-column: 1 / 3; (Elemento ocupa 2 colunas)

# CSS Grid Layout

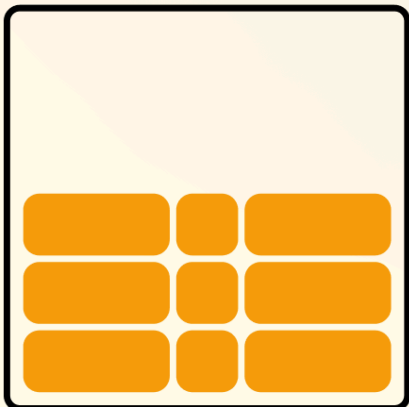
## align-content



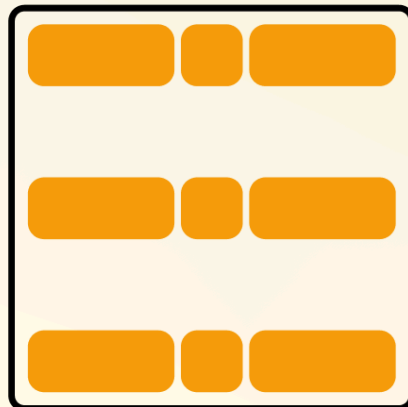
**start**



**center**

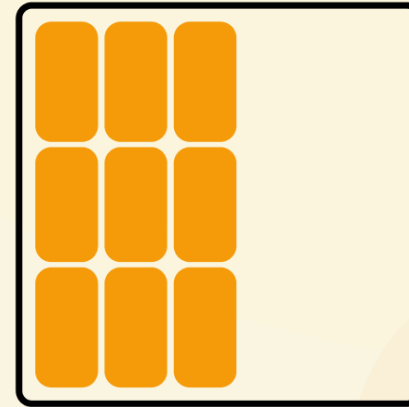


**end**

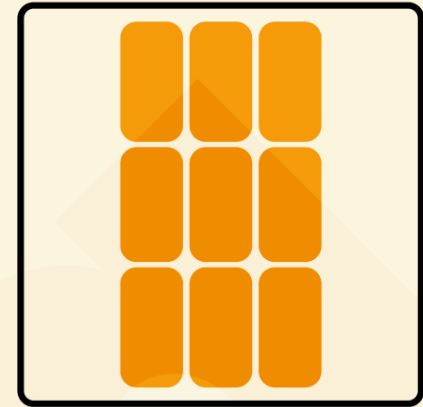


**space-between**

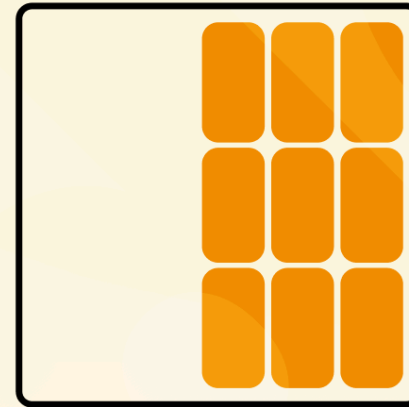
## justify-content



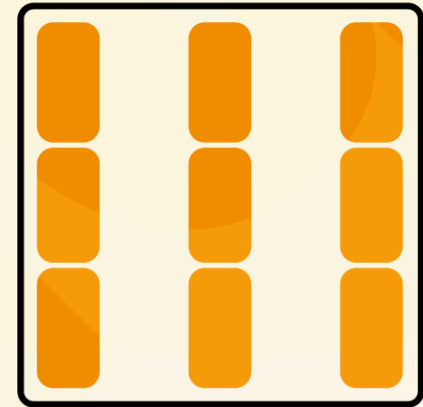
**start**



**center**



**end**



**space-between**



# Animações e Transições

```
button {  
  background-color: blue;  
  transition: background-color 0.5s ease-in-out;  
}  
button:hover {  
  background-color: red;  
}
```



# Responsividade

```
@media screen and (max-width: 768px) {  
  body {  
    background-color: lightgray;  
  }  
}
```

# Atividade Prática

Criar um arquivo HTML chamado `css.html` que deve conter pelo menos um dos seguintes elementos:

- Título
- Parágrafos
- Lista
- Imagem
- Links
- Botões

Customizar o arquivo com CSS, utilizando pelo menos 5 propriedades diferentes.

- Cor de texto
- Cor de fundo
- Tamanho da fonte
- Espaçamento
- Borda
- Alinhamento
- Display

