

Programação Web 1

Cascading Style Sheets

Objetivos de aprendizagem

Conhecer os princípios básicos do CSS3

Agenda

- Por que CSS?
- Como CSS funciona?
- *Frameworks CSS*

Por que CSS?

Cascading Style Sheets

- *Layouts* de sites precisos
- Especificação centralizada
- Atualização rápida
- Acessibilidade

Possibilidades do CSS

CSS Zen Garden

Como CSS Funciona?

Aplicação CSS

- Elementos individuais (*inline*)

```
1 <h1 style="color: red; margin-top: 2em">Introdução</h1>
```

- Embutido no HTML (*embedded*)

```
1 <head>
2   <title>Document Title</title>
3   <style>
4     /* rules */
5   </style>
```


Aplicação CSS

- Arquivo externo (`.css`)

```
1 <head>
2   <title>Jen's Kitchen</title>
3   <link rel="stylesheet" href="kitchen.css" type="text/css">
4 </head>
```

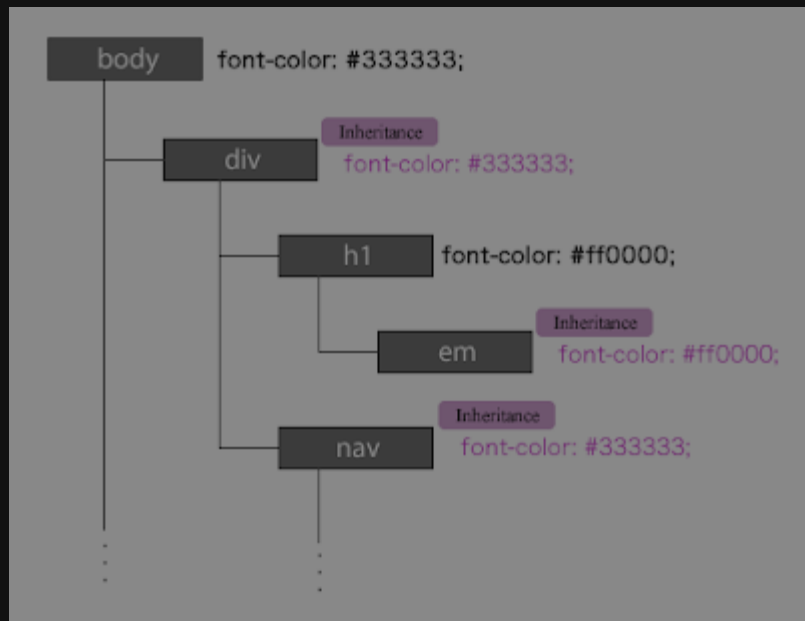
Estrutura CSS

- Folha de estilo (*sheets*)
 - Conjunto de regras (*rules*)
 - Seletores
 - Declaração (propriedade + valor)

Conceitos

- Herança
- Conflitos
- *Box Model*

Herança



Conflitos (Precedência)

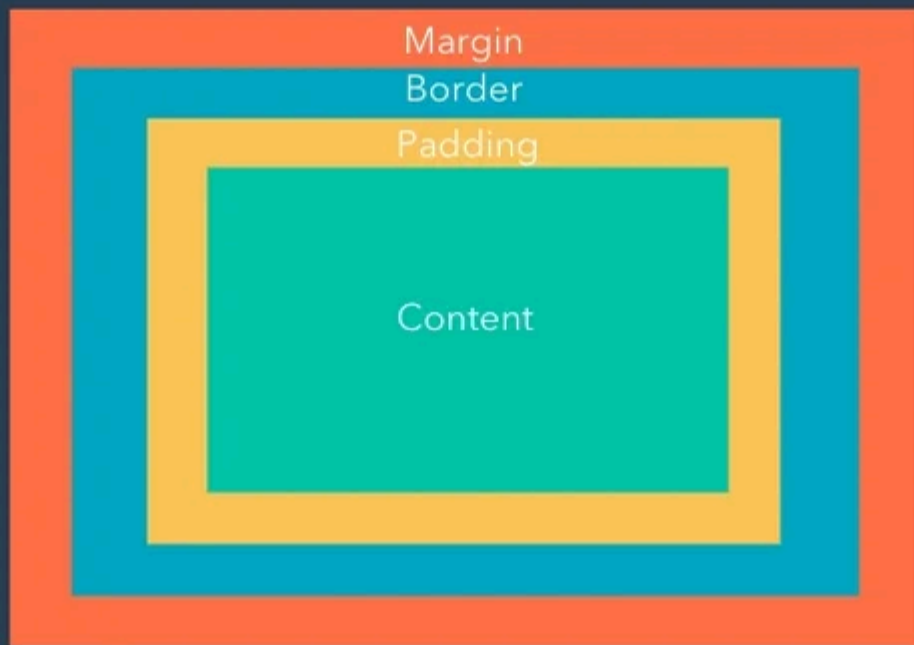
- Podem existir conflitos em definições de estilo
- As folhas de estilo possuem uma hierarquia para resolver tais conflitos:
 1. Definições do navegador
 2. Definições de estilo do usuário (definido no navegador)
 3. Arquivo externo (link)
 4. Estilos embutidos (`<style>`)
 5. Estilos *inline*
 6. Regras marcadas com `!important`

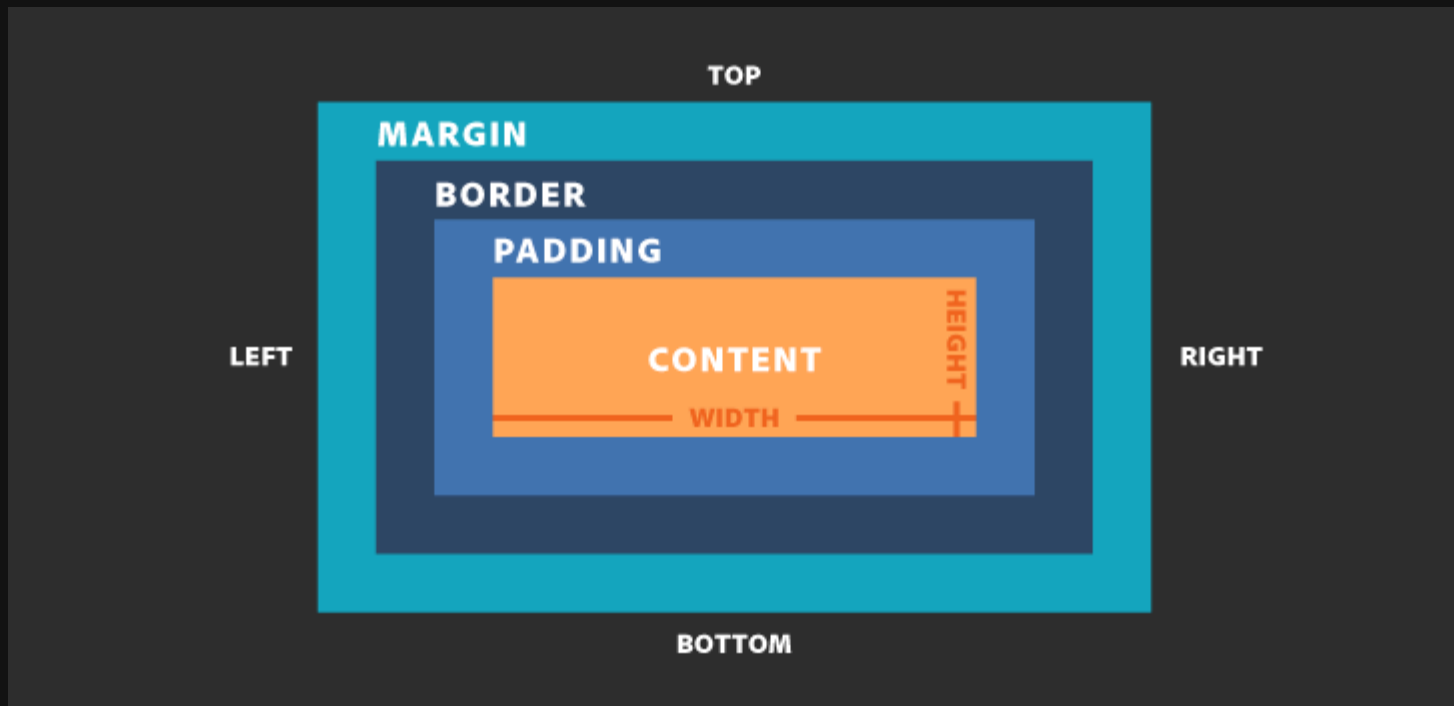
Exemplos

```
1  p {color: blue !important}
```

```
1  <style>  
2      p {color: red;}  
3      p {color: blue;}  
4      p {color: green;}  
5  </style>
```

CSS Box Model





Propriedade *Padding*

Exemplo

```
1  p {  
2      padding-top: 5px;  
3      padding-right: 10px;  
4      padding-bottom: 15px;  
5      padding-left: 20px;  
6  }
```

padding *Shorthands* (4 valores)

```
1 padding: top right bottom left
```

```
1 blockquote {  
2     padding: 1px 3px 1px 3px;  
3     background-color: #D098D4  
4 }
```

padding *Shorthands* (3 valores)

```
1 padding: top right+left bottom
```

```
1 blockquote {  
2     padding: 10px 30px 10px;  
3 }
```

padding *Shorthands* (2 valores)

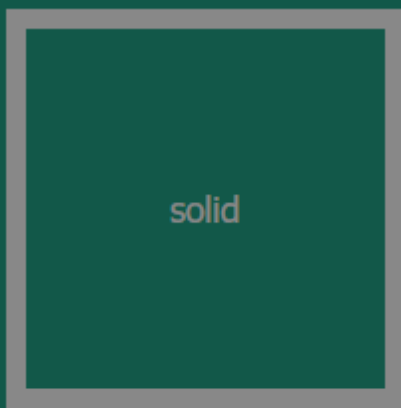
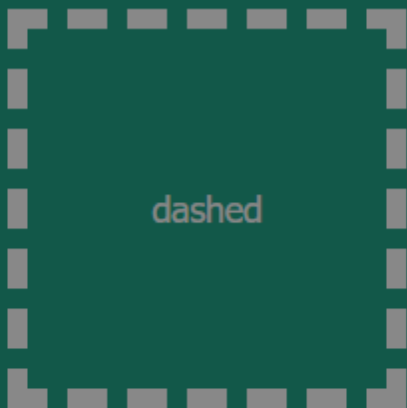
```
1 padding: top+bottom right+left
```

```
1 blockquote {  
2     padding: 10px 30px;  
3 }
```

padding *Shorthands* (1 valor)

```
1 padding: top+bottom+right+left
```

```
1 blockquote {  
2     padding: 15px;  
3 }
```



Propriedades *Border*

- `border-style`
- `border-width`
- `border-color`

```
1  p.one {  
2      border-style: solid;  
3      border-width: 5px;  
4      border-color: red;  
5  }
```

Shorthand border

```
1 p {  
2     border: 5px solid red;  
3 }
```

Especifica width, style e color

Propriedade *Margins*

Exemplo

```
1  p {  
2      margin-top: 5px;  
3      margin-right: 10px;  
4      margin-bottom: 15px;  
5      margin-left: 20px;  
6  }
```

Propriedade *Margins*

Exemplo

```
1  p {  
2      margin-top: 5px;  
3      margin-right: 10px;  
4      margin-bottom: 15px;  
5      margin-left: 20px;  
6  }
```

Disponíveis os *shorthands* de 4 valores até 1 valor

Propriedades para Fontes

Exemplos

```
1  /* set the font size to 12px and line height to 14px and
2  p {
3      font: 12px / 14px sans-serif;
4  }
5  /* set the font size to 80% of the parent element or default
6  Set the font family to sans-serif */
7  p {
8      font: 80% sans-serif;
9  }
10 /* set the font weight to bold,
11 the font style to italic,
```

Propriedades para Fontes

Live Sample (Shorthand fonts)

Melhores *Web Safe Fonts*

Fontes que são instaladas em todos os navegadores de todos os dispositivos

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

Como usar *Google Fonts*?

Alternativa às WSF

HTML

```
1  <head>
2    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
3    <style>
4      body {
5        font-family: "Sofia", sans-serif;
6      }
7    </style>
8  </head>
```

CSS

```
1  @import url('https://fonts.googleapis.com/css?family=Muli');
```

Como usar *Google Fonts*?

- Documentação *Google Fonts*
- Fontes disponíveis

Frameworks CSS

Frameworks CSS

Exemplos

1. **Tailwind CSS**

- Estilo: *Utility-first*
- Destaques: Total controle sobre estilos, performance com JIT, muito usado em projetos modernos e customizados

2. **Bootstrap 5**

- Estilo: Component-based
- Destaques: Fácil de usar, ideal para protótipos e painéis administrativos, grande comunidade

Frameworks CSS

Exemplos

3. **Foundation**

- Estilo: Component-based + responsivo
- Destaques: Alta acessibilidade, poderoso para projetos grandes, flexível

4. **Bulma**

- Estilo: CSS puro com classes utilitárias e componentes
- Destaques: Sintaxe simples, leve, sem dependência de JavaScript

Frameworks CSS

Exemplos

5. **Materialize**

- Estilo: *Material Design*
- Destaques: Visual moderno baseado no design do Google, fácil integração

6. **MUI (Material UI)**

- Estilo: *Component-based* para React (Material Design)
- Destaques: Altamente usado com React, componentes prontos e acessíveis, grande flexibilidade com tema

Tailwind CSS

- **Estilo:** *Utility-first* (baseado em classes utilitárias)
- **Popularidade:** Muito alto
- **Por que usar:** Permite personalizar rapidamente o *design* sem sair do HTML.
Ótimo para projetos modernos e altamente customizados.
- **Ideal para:** SPAs (*Single Page Applications*), sites modernos com design único.
- **Pontos fortes:**
 - Total controle sobre o estilo
 - Extensa documentação

Bootstrap

- **Estilo:** *Component-based* (classes prontas para UI)
- **Popularidade:** Ainda é um dos mais usados
- **Por que usar:** Rápido de implementar, especialmente para *backends* e protótipos.
- **Ideal para:** *Dashboards*, sites institucionais.
- **Pontos fortes:**
 - Sistema de grid poderoso
 - Componentes prontos como `navbar` , `cards` , etc

Perguntas

Exercício

Ver Quiz CSS na Aula 3.

Referências

- CSS Tutorial W3 Schools
- CSS Zen Garden