



CSS I

CSS vem de **Cascading Style Sheets** (Folhas de Estilo em Cascata). É a ferramenta responsável pela **estilização da página**. Isto é, cabe ao CSS dizer ao navegador de internet como os elementos serão renderizados (exibidos) no site.

Como vimos anteriormente, para conectar nosso HTML ao nosso CSS precisamos utilizar uma tag chamada `link`. Esta tag diz para o navegador de internet que ele precisa carregar nosso arquivo CSS quando a página é carregada. Podemos adicionar quantos arquivos .css quisermos ao nosso projeto utilizando o `link`.

A tag `link` funciona quase como a tag `a`, mas só deve ser usada dentro da tag `head`. Ela tem três **atributos** principais:

`href`: assim como na tag `a`, o atributo href deve apontar para o caminho do nosso arquivo css, seja ele um arquivo local, ou vindo da internet;

`rel` é obrigatório e especifica a relação entre o documento atual (html) e o documento linkado (css). Normalmente este valor será `stylesheet` (folha de estilos);

`type` = tipo de mídia do documento. Se omitido, o navegador vai olhar para o valor do atributo `rel`. Se esse valor for `stylesheet`, o navegador irá assumir que o `type` é `text/css`. É esse valor que vamos usar aqui no curso.

Existem três formas de se utilizar o CSS em nossos projetos:

- Inline: descrevemos a propriedade diretamente na tag HTML.
- Externo ao elemento: toda a estilização é feita no mesmo arquivo da estrutura (HTML), mas em uma tag própria, a tag `style`.
- Externo ao arquivo: a estilização é feita em um arquivo externo, e conectada ao HTML.

Exemplos:

```
<!--estilização inline--> <p style="color:red;font-size:16px;">Olá</p>
```

A estilização inline é a de maior prioridade na hora de exibir nossa página. Ela sobreescreve as regras escritas das outras formas.

```
<!--estilização externa ao elemento--> <head> <style> p { color: red; font-size: 16px; } </style> </head> <body> <p>Olá</p> </body>
```

Aqui, colocamos a estilização dentro das tags style (que, por sua vez, ficam no head)

```
<!--estilização externa ao arquivo--> <head> <link rel="stylesheet" type="text/css" href="styles.css"> </head>
```

```
p{ color: red; font-size: 16px; }
```

Arquivo css chamado "styles.css"

Head do HTML conectando-se ao styles.css utilizando a tag `link`.

Recomendamos o uso da forma **externa ao arquivo**, pois é a mais organizada!

Seletores

Seletores são a base da sintaxe do CSS. Os seletores definem os elementos HTML em que o estilo é aplicado. Abaixo, a estrutura de como funciona um seletor:



Os seletores primários são aqueles que permitem identificar **um item diretamente** ou **todos os itens daquele tipo**.

Por enquanto, falaremos de três tipos de seletor:

- tags
- class
- id

tags

Podemos colocar diretamente uma tag HTML e **todos os elementos iniciados por ela** seguirão esta regra.

```
div { background-color: red; border: 1px solid black; }
```

class

Ao invés de colocarmos diretamente a tag que queremos estilizar, podemos atribuir uma **classe** aos elementos. Assim, **todos os elementos que tiverem esta classe seguirão as regras atribuídas a ela**.

- Indicamos uma classe no CSS começando por `.`

```
<p class="texto-primario">Texto Bonito</p>
```

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

id

Às vezes queremos colocar uma estilização específica em **um único elemento**. Para isto, atribuímos ao elemento um id que deve ser **ÚNICO** e que possua **significado claro**.

- Indicamos um id no CSS começando por `#`

```
<p id="texto-unico">Texto Unico</p>
```

```
#texto-unico { color: blue; border: 3px double orange; }
```

Na hierarquia dos seletores CSS, temos a seguinte ordem:

tags ⇒ classes ⇒ ids

Onde tags são a de menor força, e ids são os mais fortes.

Podemos adicionar quantas tags, classes e ids quisermos em nossos arquivos `.css`, e estes podem possuir quantas propriedades quisermos.

Extra: seletor universal *

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

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



Apesar de todos os elementos receberem a regra do `*`, caso essa mesma regra seja escrita em um elemento, a regra do elemento sobrescreverá a regra do seletor `*`.

Ou seja, a regra específica de cada elemento é MAIS FORTE que a regra do seletor universal `*`.

O box model

No CSS existem uma **infinitude** de propriedades. Nesta parte da aula, veremos 4, que formam o que chamamos de **box-model**:

- display;
- height e width;
- margin;
- padding;

- border;

Uma coisa muito importante sobre o CSS é que **nenhuma pessoa dev** sabe todas as propriedades CSS de cor. A gente sempre pode (e deve!) **consultar** algum material externo para lembrar da sintaxe exata. Aqui no curso vamos mostrar as mais comuns!

display

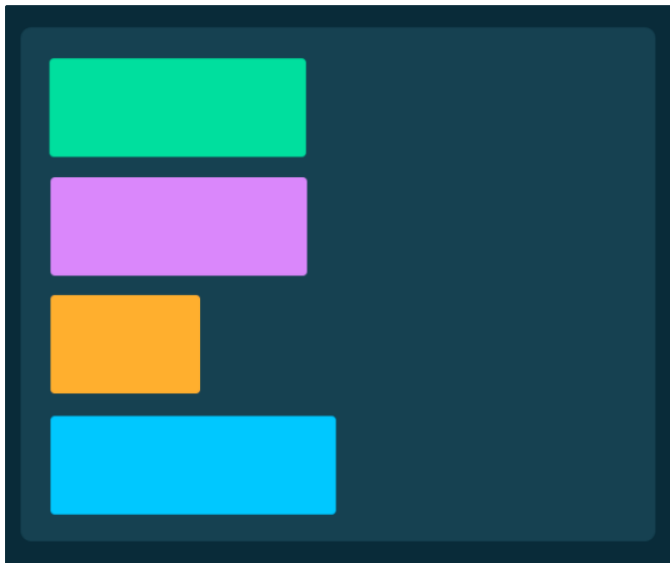
A propriedade `display` permite especificar o **comportamento** de **renderização** de um elemento. Existem algumas possibilidades. Por enquanto, vamos falar de:

- `block`
- `inline`
- `none`

`block` e `inline` são termos que já vimos, lá nas aulas de HTML. Nas aulas, vimos que as tags são normalmente block ou inline. Utilizando o CSS, podemos alterar o comportamento padrão das tags HTML.

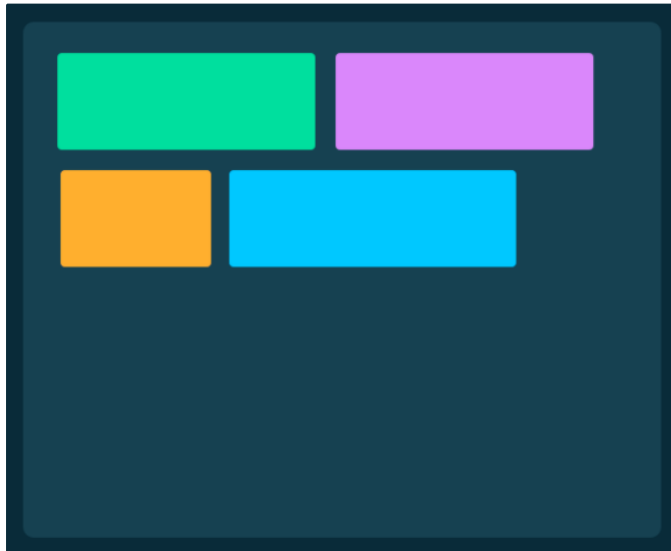
`display: block`

Faz com que o elemento ocupe toda a largura **disponível**. Um elemento block sempre começa numa **nova linha**.



`display: inline`

Faz com que o elemento ocupe toda a largura necessária. Um elemento inline não começa numa nova linha.



`display: none`

Faz com que o elemento não apareça na tela.



height e width

Height e width são, respectivamente, a **altura** e a **largura** dos nossos elementos. Essas propriedades têm valor padrão baseado no tipo de display que possuem, onde elementos `block` têm largura igual ao total de largura restante na tela, e altura igual aos limites do elemento que criamos. Elementos inline têm altura e largura iguais ao espaço que ocupam. As unidades de medida para altura e largura variam, mas o tipo mais comum é o **pixel** `px`.

margin

A propriedade **margin** representa o espaço que o elemento terá entre ele próprio e outros elementos.

```
div { margin-top: 5px; margin-bottom: 6px; margin-left: 10px; margin-right: 8px; }
```

padding

A propriedade **padding** faz com que o conteúdo dentro do elemento se **distancie da borda**.

```
div { padding-top: 5px; padding-bottom: 6px; padding-left: 10px; padding-right: 8px; }
```

border

Adiciona bordas ao elemento, permitindo especificar estilo, largura e cor da borda.

```
div { border-top: 1px solid black; border-bottom: 6px; border-left: 10px; border-right: 8px; }
```

Extra: abreviações

No CSS, temos sintaxes que permitem colocar os mesmos valores para **top-bottom** (topo-base) e para **right-left** (direita-esquerda). O primeiro argumento é para a **vertical** e o segundo para **horizontal**.

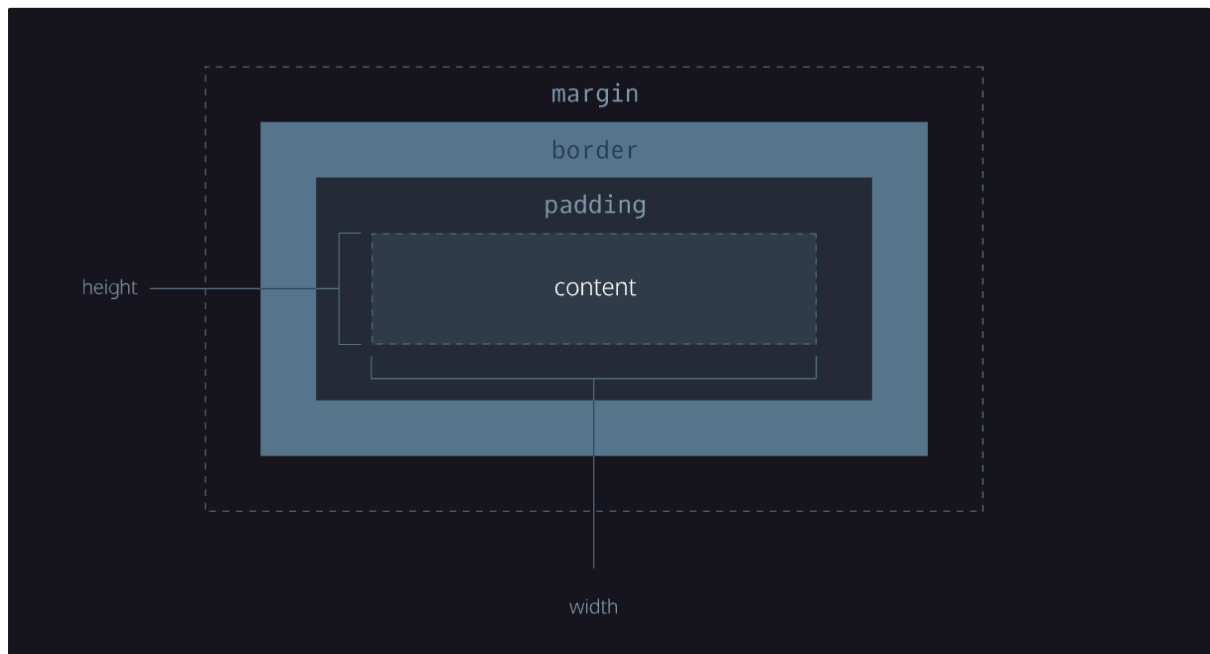
```
div { margin: 10px 5px; } /*neste exemplo, div tem 10px de margem para cima e para baixo, e 5px de margem dos lados*/
```

Há também uma sintaxe que permite colocar valores diferentes **em todas as direções**:

- A ordem das margens é: **top, right, bottom, left** (sentido horário: topo, direita, base, esquerda);

```
div { margin: 10px 5px 8px 20px; } /*neste exemplo, div tem 10px de  
margem para cima, 5 à direita, 8 para baixo e 20 à esquerda*/
```

Chamamos este conjunto de propriedades de **modelo de caixa**, e ele tem esta estrutura:



fonte: codecademy

Unidades de medida e fontes

Vamos ver algumas unidades que o CSS permite colocar em posições e tamanhos em geral. No geral, dividimos as unidades de medida entre unidades **absolutas** e unidades **relativas**. Aqui, vamos mostrar as mais comuns em cada tipo!

Unidades Absolutas

px

Indica a quantidade de pixels que deve ocupar. Pixels são o menor ponto de display de uma tela.

Unidades Relativas

%

Indica uma percentagem do tamanho relativo ao tamanho do elemento pai (por exemplo, a altura da div-filha é 50% do valor da altura da div-mãe).

vw

Cada 1vw equivale a 1% do comprimento da tela.

vh

Cada 1vh equivale a 1% da altura da tela.

Existem algumas outras unidades de medida, que são mais incomuns. Em caso de necessidade específica, mostraremos aqui no curso!

Fontes

Os arquivos CSS podem ser utilizados para alterar diversas propriedades dos elementos. Vamos dar uma olhada nas propriedades mais comuns:

font-family

Esta propriedade define qual é a fonte do nosso elemento.

font-size

Esta propriedade define o tamanho da fonte do nosso elemento. O tamanho padrão dos elementos de texto é 16px.

color

Esta propriedade determina a cor do texto do nosso elemento.

background-color

Esta propriedade é muito parecida com a **color**, mas define a cor do fundo do elemento que foi selecionado.

