Imersão Desenvolvimento Web

Modulo I - Dia 2

Taylane Brandão Renan Verissimo Laion Luiz

Indice:

- Estilização
- CSS
- GitHub

Estilização

Parte I

Tags de Estilo

```
<p>O pai tá <b>bonito</b>!</p>
```

- São **tags** com objetivo de **alterar** o visual dos textos de um documento.
- Podem alterar a exibição do texto e indicar sua **importância**.

Strong

```
<p>Um texto em <strong>negrito.</strong></p>
```

- A tag **** deixa o texto dentro dela em negrito.
- Dá ênfase ao que está dentro da tag, mostrando que é uma informação **importante** naquele contexto.

Bold

```
<p>O texto está em <b>negrito.</b></p>
```

- A tag **<b>** deixa o texto dentro dela em negrito.
- Diferente do ****, o texto em específico não possui muita importância, mas existe a necessidade de **separá-lo** visualmente.

Italic

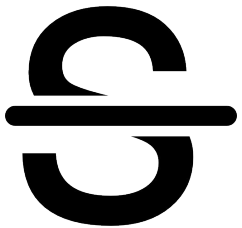
```
<p>O texto está em <i>itálico.</i></p>
```

- A tag **<i>** deixa o texto dentro dela em *itálico*.
- Normalmente usado para termos técnicos, expressões ou citações.

Strike

```
<p>O texto está <s>riscado.</s></p>
```

- A tag **<s>** deixa o texto dentro dela riscado.
- Normalmente representa itens que não estão mais **disponíveis** ou **relevantes**.

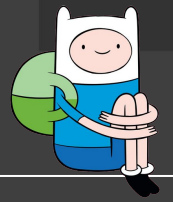


Small

```
<p>O texto está <small>pequeno.</small></p>
```

- A tag **<small>** deixa o texto dentro dela pequeno.
- Representa notas simples como dicas ou textos legais.

É hora de codar!



- Implemente cada uma das **5 tags** apresentadas e veja o resultado.

O atributo Estilo

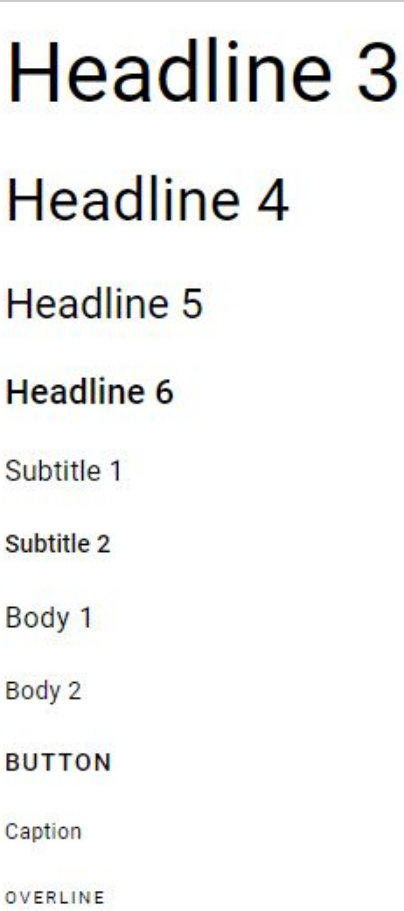
```
<p style="font-size: 1em">O pai tá bonito!</p>
```

- As **tags textuais** possuem o atributo **estilo**.
- Podem alterar a **fonte**, **tamanho** e **peso** de um **texto**.
- Ao estilizar dessa **forma**, é comumente chamado de estilo ***inline***.

Font-size

- O tamanho da fonte pode ser definido usando o `font-size`:

```
<p style="font-size: 1.2em;"></p>
```



Font-size

- Existem diversas unidades de **medidas** utilizadas na Web.
- A em ganhou força por ser **relativo** ao elemento **pai**.
- Caso o elemento pai tenha 14px, **1.2em** = 1.2 x 14 = 16.8px;

```
font-size: 1.2em;
```

```
font-size: x-small;
```

```
font-size: smaller;
```

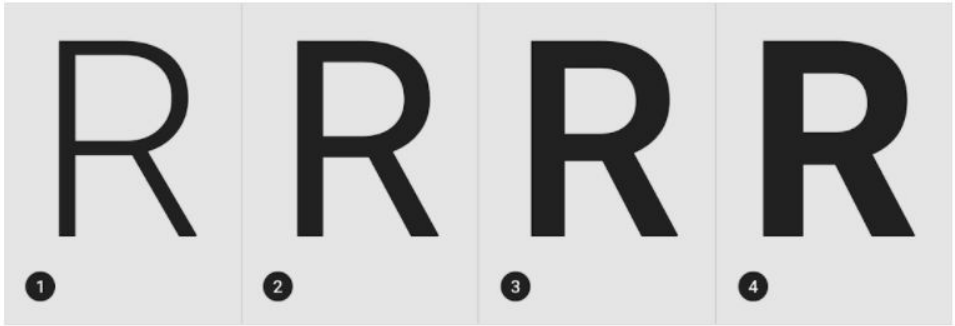
```
font-size: 12px;
```

```
font-size: 80%;
```

Font-weight

- O peso da fonte pode ser definido usando o `font-weight`:

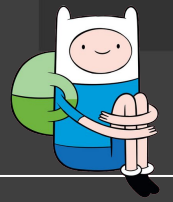
```
<p style="font-weight: normal;"></p>
```



Font-weight

- As fontes são divididas entre normal e bold.
- Algumas **fontes** possuem mais opções de peso.
- Então foi criado um **padrão** usando dos números 100 a 900 para representá-las.

É hora de codar!

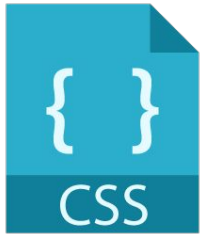


- Teste diferentes combinações de **tamanho de fonte** e **peso**;

CSS

Parte II

O que é CSS afinal ?



- Cascading Style Sheet;
- **Seleciona** e **estiliza** elementos dentro do HTML;
- Arquivo com a extensão .CSS;
- Bora testar?

Aplicando CSS ao HTML

- Crie um arquivo chamado estilo.css e adicione o **código** abaixo:

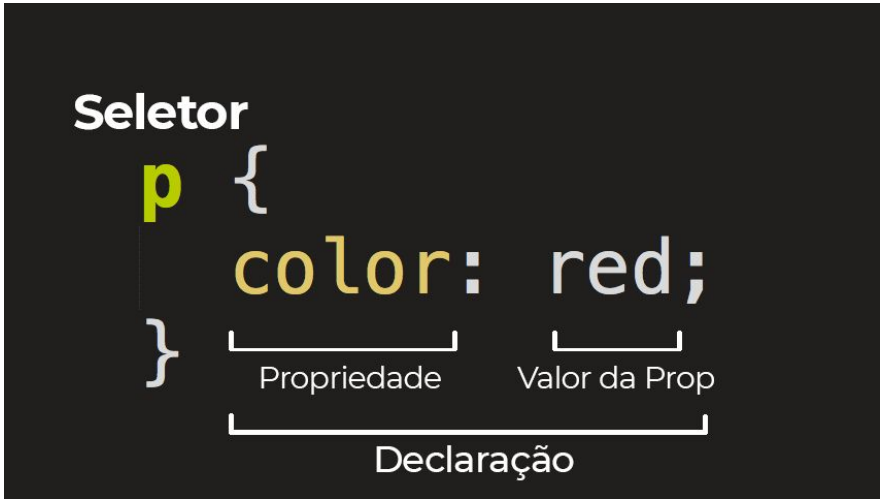
```css
p {
  color: red;
}
```

- Para ligar o HTML com CSS, usamos a tag `<link>` dentro da **<head>**:

```html
<head>
    <link href="estilo.css" rel="stylesheet">
</head>
```

O que aconteceu ?

- Neste caso, selecionamos todos os **parágrafos <p>** e aplicamos a cor **vermelha** ao texto.



Estrutura do CSS

- Seletor: Decide **quais elementos** no documento serão **estilizados**.
- Declaração: As **regras** de estilo que serão aplicadas. São criadas a partir de uma **propriedade** e seu **valor**.
- Propriedade: **Atributo** que será **estilizado**.
- Valor: **Um** dos **diversos valores** que tal prop pode assumir.

Tipos de Seletor

- Elemento: Todos os elementos com a **tag** escolhida.

```
p {color: red;}
```

- ID: Todos os elementos com o **id** definido.

```
#id {color: red;}
```

- Atributo: Todos os elementos com o **atributo** declarado.

```
img[src] {color: red;}
```

Seletor de Classe

- Representa um **conjunto de elementos** que compartilham das mesmas **características**.
- Referenciado usando um ponto: `.classe {color: red;}`
- Para **adicionar** uma classe a um elemento, use o atributo `class`:

```
<p class="classe"></p>
```

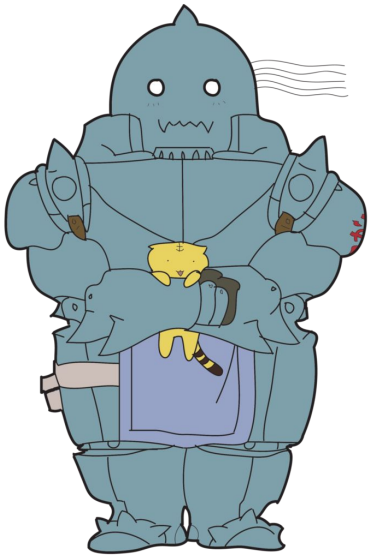
Seletor de Pseudo-Classe

```
.p:hover {color: red;}
```

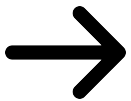
- Representa o **estado** de um elemento HTML.
- Os estados mais **comuns** são:
  - **:Hover** - Mouse **em cima** do elemento.
  - **:Active** - Quando o link está **ativo**.
  - **:Focus** - Um elemento recebe **foco**.

Equivalência

- Os estilos **inline** podem ser aplicadas também no CSS, mudando apenas a **sintaxe**:



```
<p style="color:red">
```



```
p {
  color: red;
}
```

Display

- Imagine que todo **elemento** funciona como uma caixa.
- O display indica o comportamento dessa caixa em relação aos **outros** elementos.

```
.ex1 {display: inline;}
.ex2 {display: block;}
.ex3 {display: inline-block;}
.ex4 {display: none;}
```

Inline

- Usado para deixar os elementos lado a lado.
- O tamanho é definido pelo conteúdo dentro dele.
- Não aceita valores de **largura** e **altura** (*width* e *height*).

Block

- O elemento irá ocupar **toda a linha**.
- Mais de um elemento ***block*** fará que o próximo **pule de linha**.
- **Aceita** valores de **largura** e **altura** (*width* e *height*).

Inline-block

- **Meio block, meio inline.**
- Usado para deixar os elementos lado a lado.
- Aceita valores de **largura** e **altura** (*width* e *height*).

display: none

```
.blue-book {
    display: none;
}
```

The blue book is removed completely from the stack

visibility: hidden

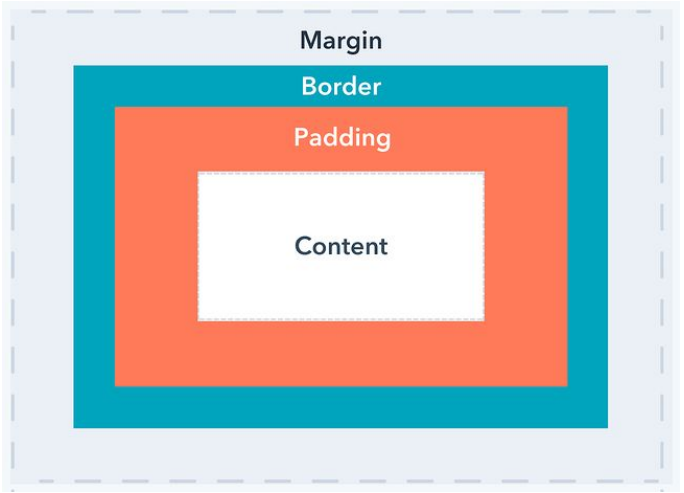
```
.blue-book {
    visibility: hidden;
}
```

The blue book is hidden visually, and its place is still reserved

Espaçamento

- Border são as **extremidades** do elemento.
- Margin cria um espaço entre a **borda** e os **elementos vizinhos**.
- Padding cria um espaço entre a **borda** e o **conteúdo**.

Margin == Margem



Padding == Preenchimento

Espaçamento

- Padding e Margin podem ser declarados usando um, dois, três ou quatro valores.
- 1 valor aplica todos os **4 lados**.
- 2 valores aplicam em **cima/baixo** e **esquerda/direita**.
- 3 valores aplicam **cima**, **esquerda/direita** e **baixo**.
- 4 valores aplicam **cima**, **direita**, **baixo** e **esquerda**. (Horário)

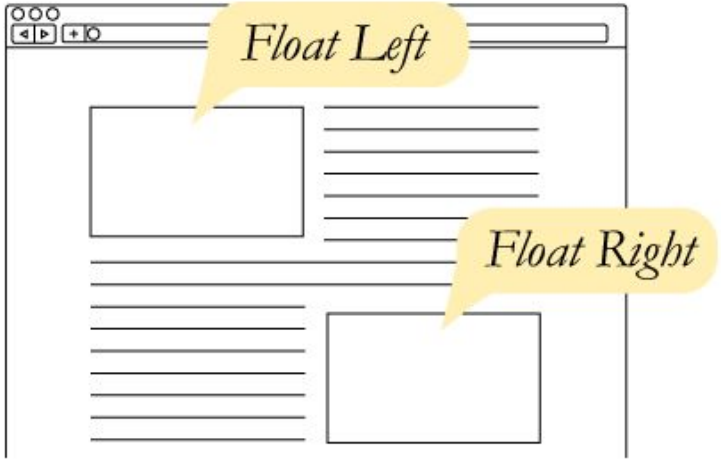
Espaçamento

- Os valores podem ser tanto **fixos** (px, em) quanto em **porcentagem**.
- A porcentagem é sempre referente a **largura** (width) do elemento.

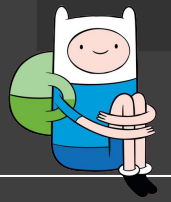
```
.ex1 {margin: 10px;}
.ex2 {margin: 1em 10px;}
.ex3 {margin: 1em 10px 10%;}
.ex4 {margin: 10px 5px 1em 10%;}
```

Float

- **Determina** como um elemento deveria **Flutuar**.
- As posições possíveis são:
  - `left == Esquerda;`
  - `Right == Direita;`
  - `none == Nenhuma;`
  - `inherit == Herdar;`



É hora de codar!



- Vamos organizar os **elementos** da página ?

GitHub

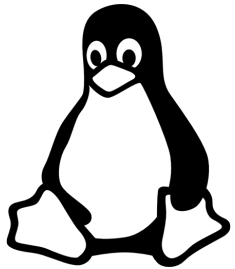
Parte III

O que é GitHub?

- Uma das maiores **plataformas** para salvar código de seus projetos.
- Utiliza tecnologia Git para diversas funcionalidades como o **controle de versão**.
- Facilita trabalhar com outros **devs** em um mesmo projeto.
- Diversos projetos Open Source!

Open Source

- Projetos onde o código é **aberto** para **todos**.
- É possível **contribuir**, **estudar**, **modificar** e até **distribuir**.
- Ótimo para **praticar** tanto **Git** quanto **desenvolvimento**.
- “Quem faria uma **doideira** dessa?”



Criando uma conta/repositório

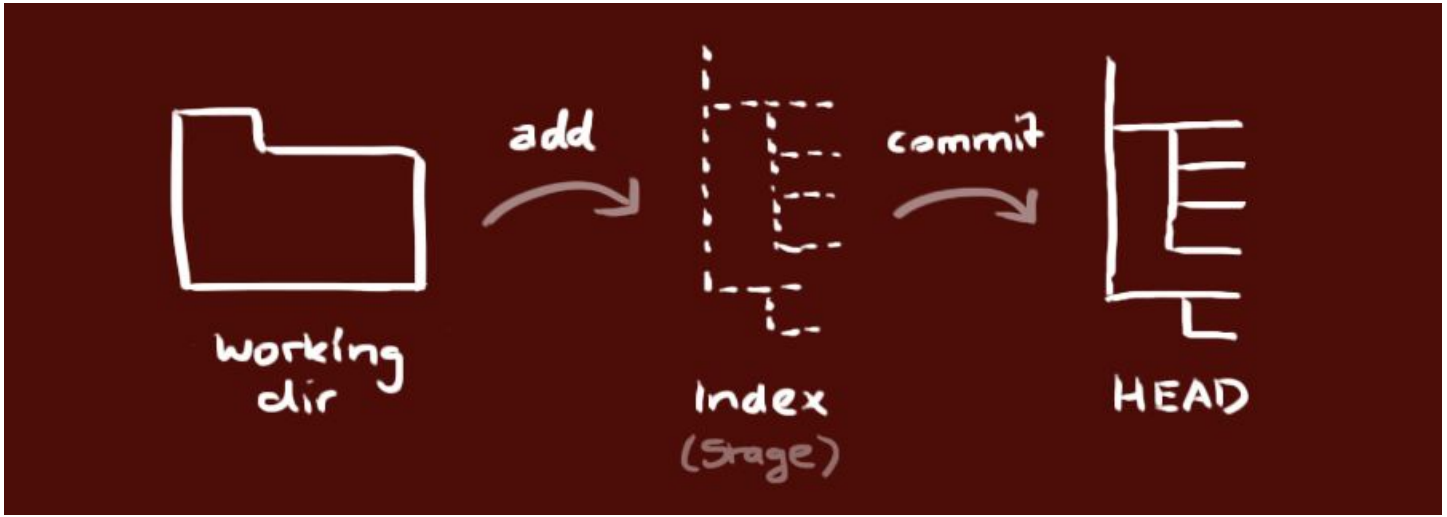
- Acesse **github.com**.
- Clique em **sign up** e preencha com seus dados.
- Clique em ***New***, escolha um **nome** e clique em ***New Repository***.
- Deixe essa página **aberta** pois vamos precisar dela.

Clonando!

- Para começar a usar um **repo**, é preciso fazer uma **cópia dele** na máquina. Para isso usamos o comando ***git clone***.
- Abra o **terminal** referente a pasta onde seu projeto **vai ficar**.
- **Copie o link** do seu repo na página que deixou **aberta**.
- Digite ***git clone <link do seu repositório>*** e dê **enter**.

Como funciona?

Agora sua **pasta** possui a seguinte **estrutura**:



Como funciona?

- O **primeiro** é a pasta onde **estão os arquivos** do seu projeto.
- O **segundo** é o ***stage***, onde ficam as alterações que você **pretende** enviar para o repo.
- O **terceiro** é o ***head***, onde ficam as alterações que você **confirmou** que vão para o repo.

Esses aqui devem subir

- Para enviar os arquivos para o ***stage***, usamos o comando ***git add***.
- Para adicionar um arquivo em **específico**: ***git add <nome do arquivo>***
- Para adicionar **todos** os arquivos alterados: ***git add** *.

Comita ai!

- Nesse momento, todos os arquivos estão no *stage*. Hora de transformá-los em um **commit**: ***git commit***.

- Para usá-lo, é necessário adicionar um **comentário** relacionado: ***git commit -m “Umas gambiarras muito doidas aqui.”***

- O **comentário** costuma ser um **resumo** referente ao que foi feito no *commit* em questão.

Tô subindo.

- Depois de ***comitar*** algo, é importante lembrar que eles ainda **estão na máquina**.

- Para enviá-las para seu repo no **GitHub**, use o comando ***git push***.

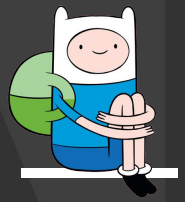
Caso seja a **primeira vez** subindo fazendo um *push*, será preciso adicionar suas **credenciais do GitHub** para vincular sua conta aos *commits* enviados. (*Uma janelinha vai surgir, não se preocupe.*)

Puxando!

- Caso esteja trabalhando num repo em time, outra pessoa pode ter **subido alterações** que **não** estão na sua máquina. O que fazer?

- Para **puxar** os últimos commits do seu repo para a **máquina**, use o comando ***git pull***.

É recomendado usar o *git pull* **antes** de fazer qualquer alteração no repo. **Evita dor de cabeça!**



É hora de ~~codar~~ subir!

- O objetivo é **subir os arquivos** criados nas últimas aulas para o **repo** que criamos anteriormente no GitHub.


VICTORY