



FIAP

Shift



Aula 01

HTML & CSS

Parte 1



O QUE O MATERIAL COBRE

1

INTRODUÇÃO AO HTML + EXTENSÕES PARA VSCODE

2

ESTRUTURA BÁSICA DO HTML

3

ESTRUTURA DE PASTAS E ARQUIVOS

4

TAGS HTML: TEXTOS, LINKS, IMAGENS E LISTAS

O QUE O MATERIAL COBRE

5

INTRODUÇÃO AO CSS

6

TIPOS DE SELETORES CSS

7

PROPRIEDADES CSS: FONTES E TEXTOS

8

SUGESTÕES DE EXERCÍCIOS

1

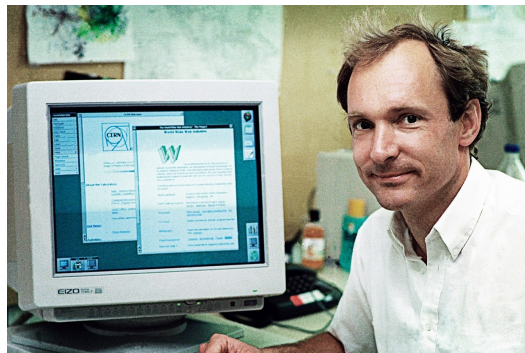
Introdução ao **HTML + Extensões**

O que é HTML e para que ele é usado?

- **HTML:** Hyper Text Markup Language;
- É a linguagem universal para criação de páginas para internet;
- Não é uma linguagem de programação, mas de marcação.

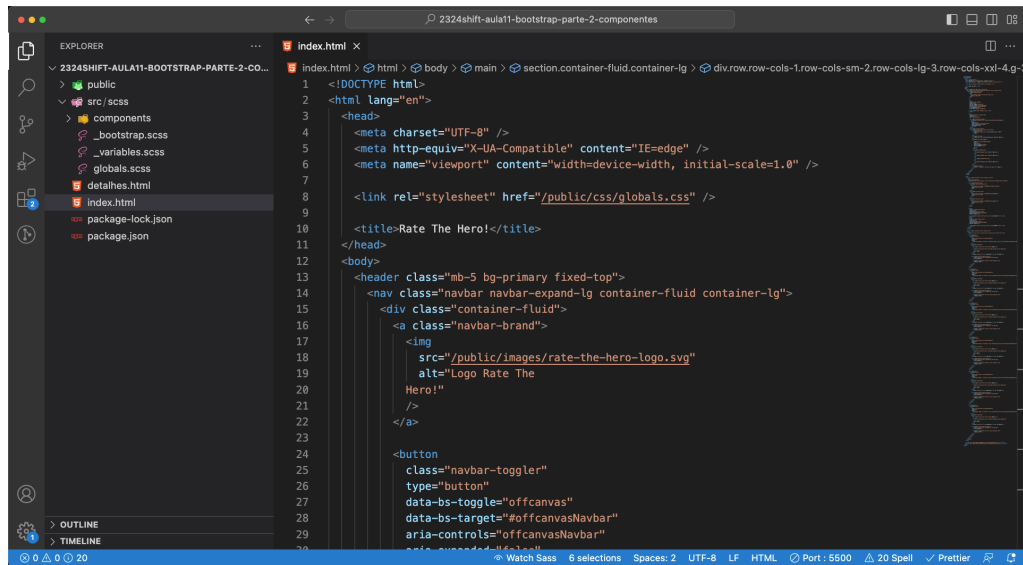
A primeira versão do HTML foi escrita por **Tim Berners-Lee** (foto) em 1993.

Desde então diversas versões foram construídas até chegarmos a que temos hoje, que é conhecido como **HTML5**.



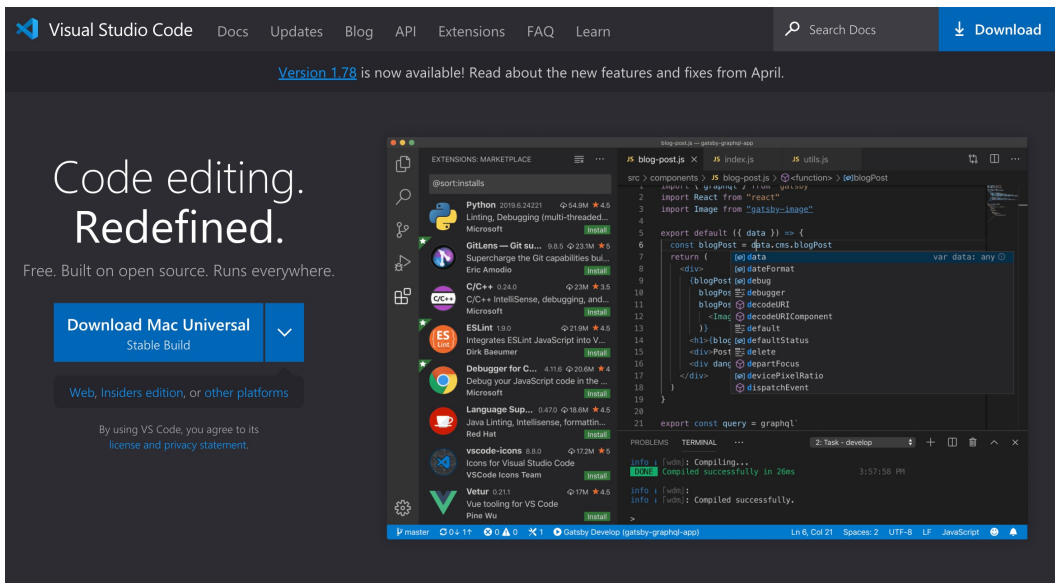
Como programar em HTML?

- É necessário um bloco de notas e arquivos com extensão **“.html”** ou **“.htm”**;
- Vamos usar um software próprio para programação web (ou seja, IDE): o **VSCode**;
- Esse software é **“Open Source”** ou seja: é gratuito e possui seu código fonte aberto, atualmente no GitHub.



Instalando o VSCode

- O VSCode é um software bem leve, simples de instalar e que roda nos principais sistema operacionais;
- Acesse: <https://code.visualstudio.com/>
- Automaticamente ele vai reconhecer seu sistema operacional;
- Clique em download e siga as instruções de instalação.



Opções principais do VSCode

Ao abrir o VSCode você vai se deparar com alguns botões no menu ao lado esquerdo, são eles:



1. Árvore de arquivos do projeto



2. Busca geral no projeto



3. Integração com o GIT



4. Debug e execução de comandos



5. Extensões para deixar seu VSCode melhor ainda

Vamos usar mais os itens 1 e 2 e pontualmente o item 5.

Extensões do VSCode

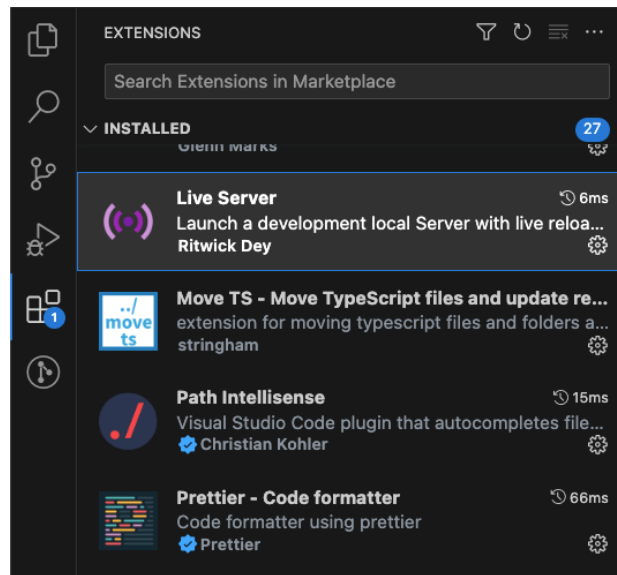
Para uma melhor performance e “dev experience” o VSCode conta com uma vasta biblioteca de extensões feitos por eles e pela comunidade

Você pode instalar extensões de acordo com o projeto que está trabalhando. Nós vamos instalar as seguintes:

- **Live Server:** para subir seu site em um servidor local
- **VSCode Icons:** para estilizar os icons do projeto de acordo com a extensão do arquivo ou nome da pasta
- **Prettier:** para formatar o código automaticamente

Você pode instalar quantas extensões quiser!

Digite o nome da extensão no campo de texto



Alguns atalhos do VSCode (Windows)

Atalho	Descrição
Ctrl + +	Aumentar zoom
Ctrl + -	Diminuir zoom
Ctrl + , (virgula)	Abre o painel de configurações (settings)
Ctrl + Alt + Setas para cima/baixo	Insere múltiplos cursores
Alt + Click botão esquerdo do mouse	Múltiplos cursores em partes separadas do código
Alt + Shift + F	Identação automática do código
Alt + Seta para cima/baixo	Mova a linha ou bloco de código onde o cursor está posicionado
Ctrl + K + C ou Ctrl + ; (Ponto e virgula)	Comenta uma ou mais linhas
Alt + Shift + Setas	Duplicar linha
Ctrl + Barra de Espaço	Abre o Intellisense
Ctrl +] (Fecha colchetes)	Divide a tela

Os atalhos podem mudar de acordo com o seu sistema operacional

Formatar ao Salvar

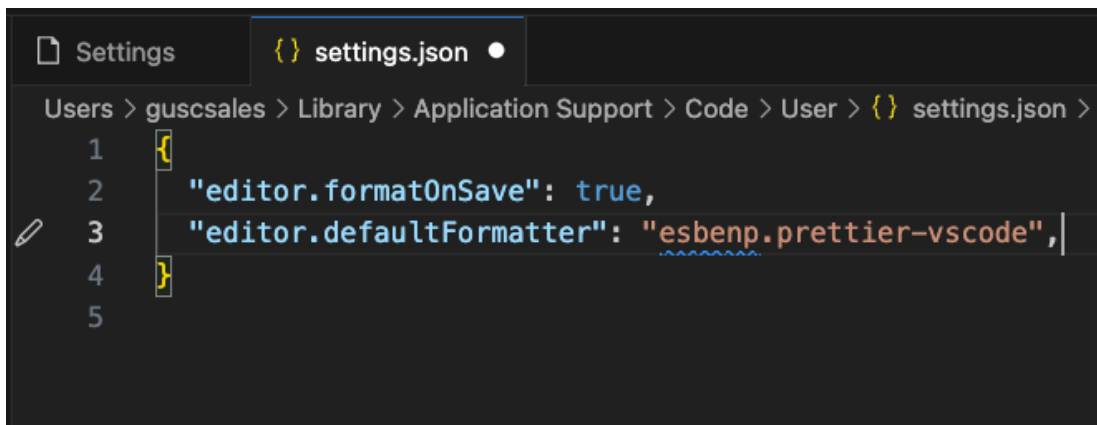
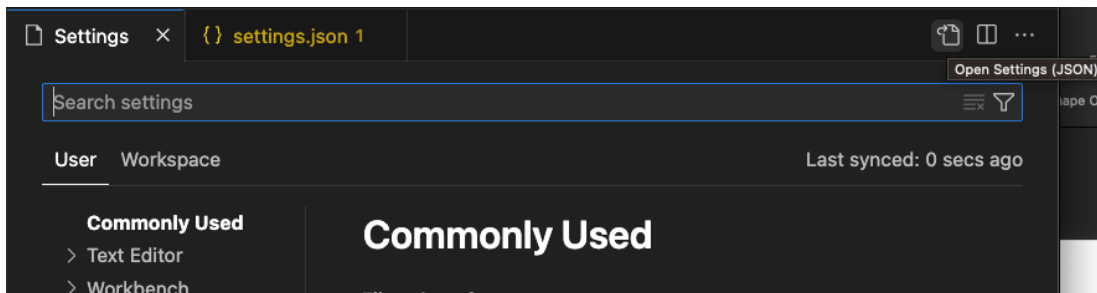
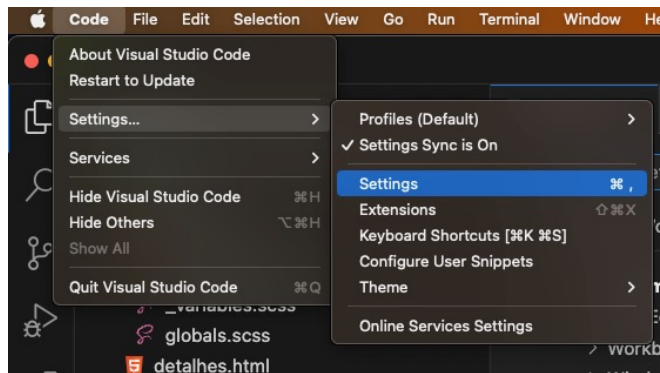
A indentação do código é parte fundamental do desenvolvimento, porque gera bastante organização no seu projeto.

Felizmente o VSCode juntamente com o prettier pode fazer a formatação automática ao salvar o arquivo.

Nas configurações do VSCode podemos colocar as seguintes propriedades:

```
{  
  "editor.formatOnSave": true,  
  "editor.defaultFormatter": "esbenp.prettier-vscode",  
}
```

Formatar ao Salvar – Passo a Passo



Se já existirem outras configurações, pode manter, apenas adicione as novas

2

Estrutura básica do **HTML**

HTML: Estrutura Básica

Todo documento HTML possui “tags” para definir sua estrutura e isso é uma regra da linguagem.

As tags em sua maioria são abertas e uma linha e ao final fechadas em outra linha, porém algumas são fechadas na mesma linha.

Exemplo de tag aberta/fechada em linhas separadas

```
<html>
...
</html>
```

Exemplo de tag aberta/fechada na mesma linha

```
<img ... />
```

HTML: Estrutura Básica

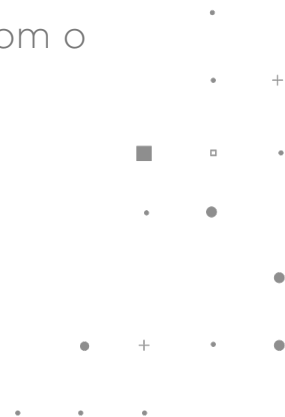
Ao iniciar uma página para o seu site, você deve seguir a estrutura básica do HTML, que contam com as tags essenciais para um site funcionar

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```


HTML: Estrutura Básica

<!DOCTYPE html>

É a primeira linha do arquivo HTML, e ela indica qual versão da linguagem será utilizada. Nesse caso o navegador vai utilizar a última versão que é o HTML 5. Em versões passadas essa linha era bem maior, com o tempo ela foi sendo refatorada.



HTML: Estrutura Básica

A tag principal que envolve toda a página HTML. Toda e qualquer tag do seu site deverá estar dentro da tag “<html>”.

<html lang="pt-br">

É interessante informar ao navegador qual o idioma da página através da propriedade “**lang**”, que nesse caso está com o idioma português do Brasil.

Seu fechamento é o “</html>” no final da página.



HTML: Estrutura Básica

<head>

Essa tag é algo que as pessoas em si não veem no site, ela é voltada mais para a máquina, porém é fundamental para um bom funcionamento do seu site.

Dentro dela é possível definir o título da página, os estilos, os metadados em geral e outros recursos. Você pode fechar essa tag com “</head>”.



HTML: Estrutura Básica

Essa tag precisa ficar dentro do <head> para o seu bom funcionamento, de preferência a primeira tag desse grupo. Ela indica qual cadeia de caracteres o site irá utilizar.

<meta charset="UTF-8" />

A **UTF-8** é a cadeia que funciona bem para o ocidente, exibindo corretamente as palavras com acentuação e caracteres especiais.

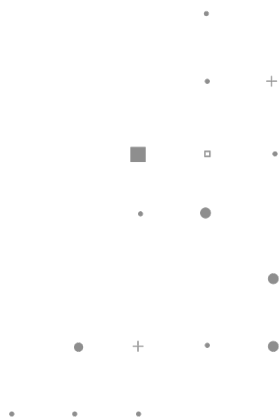
As meta tags são exemplos de tags que fecham na mesma linha.



HTML: Estrutura Básica

```
<meta name="viewport"  
content="width=device-  
width, initial-scale=1.0" />
```

Mais um exemplo de meta tag. Agora com a propriedade “name” e “content”. Essa tag é muito importante, porque define que seu site poderá ser visualizado de maneira responsiva em dispositivos móveis, como por exemplo iPhones e outros smartphones.



HTML: Estrutura Básica

<title>

Essa tag define o título do seu site, aquele título que irá aparecer na aba do navegador.

No caso dessa tag ficar vazia, a maioria dos browsers colocam a URL do seu site como título e consequentemente robôs como o Google irão ignorar o seu site e não vão ranqueá-lo, então a recomendação é preencher sempre essa tag.

Seu fechamento é o “**</title>**”.



HTML: Estrutura Básica

<body>

Saindo do âmbito de tags interpretadas por máquina, agora vamos para a parte onde as pessoas podem ver. Praticamente tudo o que for colocado dentro da tag body será visível para a pessoa que estiver acessando seu site.

Nessa tag você pode colocar texto, imagens, vídeos e muitos outros elementos.

Seu fechamento é o “**</body>**”.



PARA NÃO ESQUECER!

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```

Com exceção do **“DOCTYPE”**,
toda e qualquer tag a mais deve
obrigatoriamente estar dentro da
tag **“<html>”**, ou seja, a última
linha do seu site deve ser
“</html>”



3

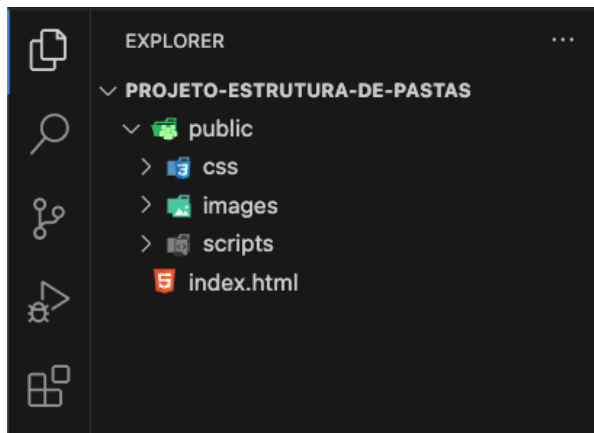
Estrutura de **Pastas e Arquivos**

Estrutura de Pastas e Arquivos

Ao iniciar um projeto é sempre importante procurar manter ele organizado. Mesmo que comece pequeno ele pode ganhar escala em algum momento, por isso é importante seguir uma estrutura de pastas para que você ou qualquer pessoa que trabalhe nesse projeto entenda e não se perca.



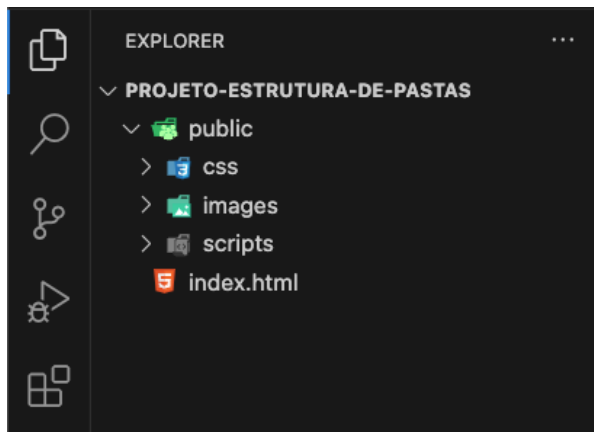
Estrutura de Pastas e Arquivos



A estrutura que vamos seguir possui a pasta do projeto, nesse caso “projeto-estrutura-de-pastas” (mas pode ser o nome do seu projeto) seguida de uma pasta public com outras pastas dentro.

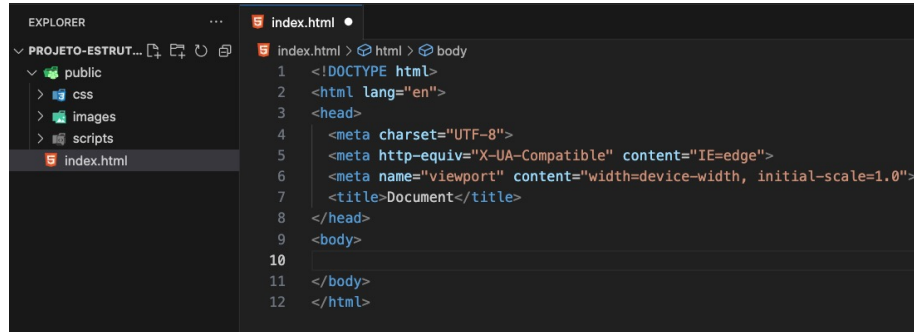
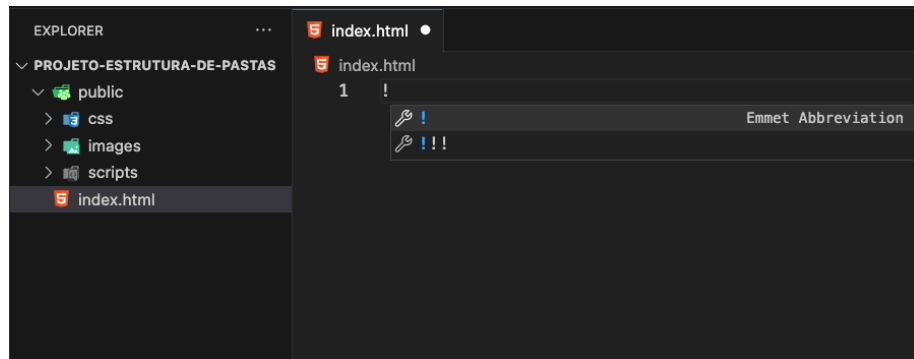


Estrutura de Pastas e Arquivos



- **public** – a pasta que engloba as subpastas que vão receber todos os arquivos externos que serão acessados por suas páginas HTML, exemplo: imagens, CSS, scripts, PDFs e qualquer tipo similar;
 - **css** – a pasta que irá armazenar os arquivos CSS;
 - **images** – a pasta que irá armazenar as imagens;
 - **scripts** – a pasta que irá armazenar arquivos Javascript.
- **index.html** – o arquivo inicial que o navegador irá acessar.

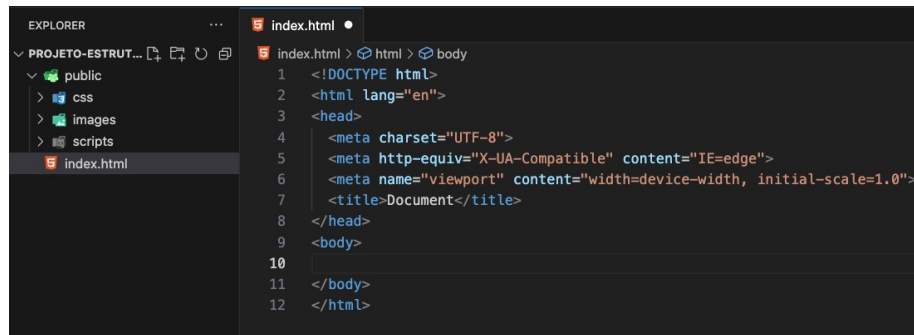
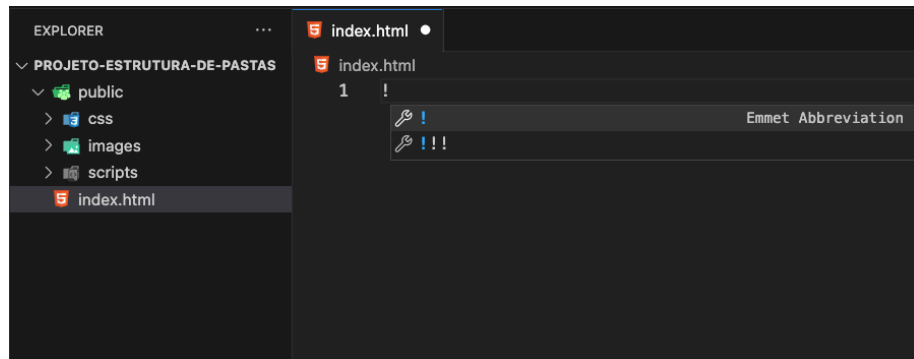
Estrutura de Pastas e Arquivos



Vamos criar a primeira página HTML agora. Para isso você irá abrir o arquivo **index.html** e irá digitar a estrutura inicial que vimos anteriormente.

Para facilitar o processo você pode digitar apenas uma **exclamação (!)** e apertar **Enter**.

Estrutura de Pastas e Arquivos

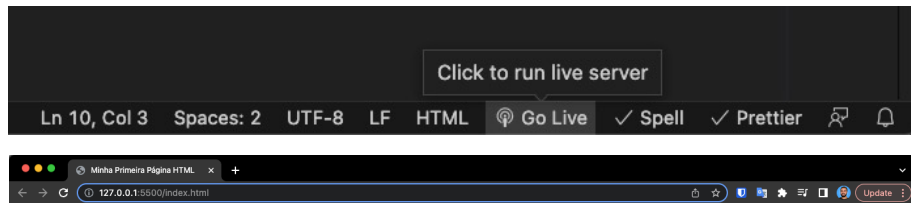


Vamos criar a primeira página HTML agora. Para isso você irá abrir o arquivo **index.html** e irá digitar a estrutura inicial que vimos anteriormente.

Para facilitar o processo você pode digitar apenas uma **exclamação (!)** e apertar **Enter**.

Dica importante: não esqueça de salvar 😊

Estrutura de Pastas e Arquivos



Inicie o servidor através do botão **“Go Live”** localizado na parte inferior do VSCode (você precisa da extensão live server).

Veja o resultado na página!

Altere a tag title e vá descobrindo aos poucos como o HTML funciona.

4

Tags HTML

Textos, Links, Imagens e Listas

Tags HTML: Headings (Cabeçalhos)

```
<h1>Heading Nível 1</h1>
```

```
<h2>Heading Nível 2</h2>
```

```
<h3>Heading Nível 3</h3>
```

```
<h4>Heading Nível 4</h4>
```

```
<h5>Heading Nível 5</h5>
```

```
<h6>Heading Nível 6</h6>
```

A tags “h”, que são definidas como **h1, h2, h3, h4, h5 e h6** (termina no h6), são referentes aos títulos que podem ser utilizados para os conteúdos.

Sendo a tag h1 a mais forte, que seria o título principal, passando por h2 que seria um sub-título e assim por diante até a h6.



Tags HTML: Parágrafos

```
<p>  
Lorem ipsum, dolor sit amet  
consectetur adipisicing elit.  
Soluta, inventore! Aspernatur  
quidem iure totam quod  
illum sint dicta.  
</p>
```

Ao utilizar a tag **<p>** você consegue criar parágrafos para o seu site, essa é uma das tags mais comuns, mas procure utiliza-la apenas para textos que fazem sentido serem parágrafos.

Dica: no VSCode digite “lorem” e clique enter para gerar Lorem Ipsum

Tags HTML: Span

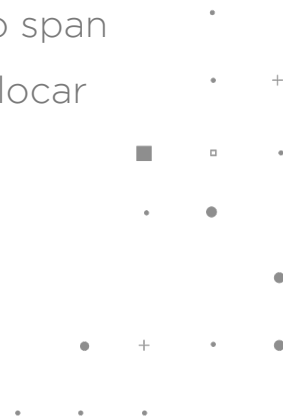
```
<span>Qualquer texto aqui</span>

<!-- Um caso de uso -->
<h1>
  Meu título
  <span>Meu subtítulo</span>
</h1>
```

Com a tag **** você pode criar qualquer tipo de texto separado, caso necessite.

Quando você se perguntar: “esse pequeno texto deve ser um parágrafo?”, provavelmente é um span que você irá utilizar.

Mais para frente vamos ver CSS e é aí que o span faz total sentido, uma vez que podemos colocar um estilo diferente para ele.



Tags HTML: Negrito, Itálico e Sublinhado

```
<strong>bold</strong>
```

```
<em>italic</em>
```

```
<u>underline</u>
```

Com as seguintes tags você pode formatar seu texto:

- strong – para deixar o texto em negrito
- em – para deixar o texto em itálico
- u – para deixar o texto em sublinhado

Mais para frente vamos ver como aplicar esses mesmos estilos, porém via CSS.



Tags HTML: Links

```
<!-- Link comum -->
<a href="sobre.html">
  Sobre
</a>

<!-- Link em nova aba -->
<a
  href="https://fiap.com.br"
  target="_blank"
>
  Ir para o site da FIAP
</a>
```

Você pode interligar páginas do seu site através de links (ou âncoras) utilizando a tag **<a>** com os seguintes atributos:

- **href** – com o caminho ou URL do destino
- **target (opcional)**
 - `_blank`: para abrir o link em uma nova aba



Tags HTML: Imagem

```

```

Para adicionar imagens ao seu site você pode utilizar a tag **** com as seguintes propriedades:

- **src** – o caminho da imagem, seja interno ou uma URL completa
- **alt** – um texto para servir de acessibilidade e para ser mostrado quando a imagem não carregar

A imagem é uma tag que fecha na mesma linha.

Tags HTML: Lista ordenada

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Para adicionar listas numéricas e ordenadas podemos utilizar a tag **** seguido da tag ****.

A tag é uma abreviação de “ordered list” e a representa um item dessa lista.

O resultado será:

1. Item 1
2. Item 2
3. Item 3



Tags HTML: Lista não ordenada

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Para adicionar listas não ordenadas podemos utilizar a tag **** seguido da tag ****.

A tag é uma abreviação de “unordered list” e a representa um item dessa lista.

O resultado será:

- **Item 1**
- **Item 2**
- **Item 3**



5

Introdução ao **CSS**

Nem só de HTML viverão os sites

- **CSS:** Cascading Style Sheet;
- O HTML cria a estrutura, mas com o CSS podemos dar estilo aos nossos sites;

Após o lançamento do HTML o programadores **Håkon Wium Lie** e **Bert Bos** perceberam que algo estava faltando.

E após trabalharem juntos, em 1996 eles lançam a primeira versão do CSS que hoje se encontra na versão 3.



Quer saber mais sobre a história do CSS? <https://www.w3.org/Style/CSS20/>



Uma das grandes vantagens do CSS é que podemos reutilizar os códigos de estilos para diversos elementos HTML.

Vamos supor que você tem vários elementos com a cor azul no seu site e quer mudar para vermelho, apenas uma alteração no CSS e tudo será atualizado. Isso é possível por conta dos tipos de seletores e das propriedades do CSS.

Mas antes, vamos falar um pouco sobre como declarar o CSS no seu site.

Declaração CSS: Inline

```
<h1 style="font-size: 16px; color: red">  
  Meu elemento HTML  
</h1>
```

A primeira coisa que temos que saber sobre o CSS são as maneiras que em que ele pode ser escrito.

Essa forma ao lado é a declaração **“inline”**, não deve ser a sua primeira escolha e ela deve utilizada para situações mais específicas, uma vez que você estiliza diretamente a tag.



Declaração CSS: Interno

```
<head>
  <style>
    h1 {
      font-size: 16px;
      color: red;
    }
  </style>
</head>
```

Nessa forma ao lado você pode escrever CSS diretamente na sua página HTML criando uma tag **<style>** dentro da tag <head> e então colocando as regras CSS dentro da tag style.

Essa também não deve ser sua primeira escolha para sites, eu recomendo para criação de template de e-mails customizados.

Declaração CSS: Externo

```
h1 {  
  font-size: 16px;  
  color: red;  
}
```

```
<head>  
  <link  
    rel="stylesheet"  
    href="/public/css/styles.css"  
  />  
</head>
```

Nessa forma ao lado você pode escrever CSS em um arquivo específico com a extensão **“.css”** que de acordo com nossa estrutura de pastas vai ficar dentro da **“public/css”**.

Para declarar o CSS no HTML é necessário usar a tag **<link>** com a propriedade **rel=“stylesheet”** e **href=“caminho_do_css”** dentro da tag **<head>**.

Essa deve ser sua primeira escolha ao criar CSS, porque criando arquivos você garante muito mais organização no seu projeto.

Declaração CSS: O efeito cascata

```
<head>
  <link
    rel="stylesheet"
    href="/public/css/globals.css"
  />
  <link
    rel="stylesheet"
    href="/public/css/home.css"
  />

  <style>
    h1 { color: red; }
  </style>
</head>
<body>
  <h1 style="color: blue">
    Meu título
  </h1>
</body>
```

O CSS é lido da esquerda para direita e de cima para baixo – guarde bem isso.

O navegador utiliza do efeito cascata para aplicar as regras no HTML. O padrão é:

1. **CSS padrão do navegador**
2. **CSS externo**
3. **CSS interno**
4. **CSS inline**

Se a mesma regra CSS for encontrada em múltiplos arquivos externos, o mais forte é o que tem a importação por último.

6

Tipos de seletores

CSS

Como criar uma regra CSS

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

O CSS é formado de **seletores**, **propriedades** e **valores** para cada propriedade.

Para fazer a declaração da regra você tem que primeiro dizer qual é o seletor, ou seja, o elemento HTML a ser estilizado.

Então abrir e fechar chaves (**{}**) e dentro das chaves entrar com a propriedade e o valor da mesma, finalizando com um **ponto e vírgula (;)**.



Dica Crucial!

**Não esqueça de colocar
ponto e vírgula (;)
no fim de cada linha do CSS**

Isso pode te evitar muita dor de cabeça... 😊

Tipos de Seletores: Elemento

```
<h1>  
  Meu título  
</h1>
```

```
h1 {  
  font-size: 22px;  
}
```

O seletor do tipo elemento é o que você pode declarar utilizando a tag HTML diretamente.

Vamos supor que você tem um `<h1>` no seu HTML, você pode declarar uma regra para o `h1` no css.

E isso serve para qualquer tag HTML!

Tipos de Seletores: ID

```
<h1 id="meu-id">
  Meu título
</h1>
```

```
#meu-id {
  font-size: 22px;
}
```

Para criar estilos mais específicos é possível utilizar o seletor do tipo ID. Ele deve ser declarado como uma propriedade **“id”** na tag HTML.

Já no CSS basta estilizar com o **cerquilha (#)** seguido do nome atribuído na propriedade HTML. Apenas no CSS como que você escreve o # como prefixo.

Qualquer tag HTML pode ter a propriedade id.

O ID obrigatoriamente deve ser um valor único, não pode ser repetido em múltiplas tags.

Tipos de Seletores: Classe

```
<h1 class="minha-classe">
  Meu título
</h1>
```

```
.minha-classe {
  font-size: 22px;
}
```

Outra forma de criar estilos mais específicos é através do seletor tipo classe. Ele deve ser declarado como uma propriedade **“class”** na tag HTML.

Já no CSS basta estilizar com o **ponto (.)** seguido do nome atribuído na propriedade HTML. Apenas no CSS como que você escreve o . como prefixo.

Qualquer tag HTML pode ter a propriedade class.

A classe pode ser repetida múltiplas vezes em diferentes tags HTML.

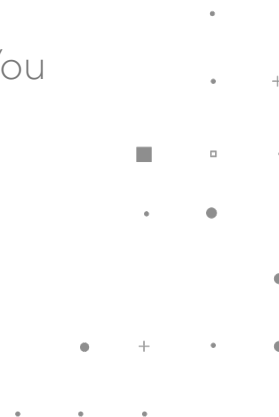
Agrupamento de Seletores

```
h1,  
h2,  
h3,  
#meu-id,  
.minha-classe {  
    font-size: 22px;  
}
```

É possível fazer agrupamento de seletores para que todos recebam a mesma regra CSS.

O agrupamento pode ser feito de maneira dinâmica, utilizando qualquer tipo de seletor de forma misturada.

Para agrupar basta separar as tags, ids e/ou classes por virgula e aplicar a regra.



7

Propriedades CSS

Fontes e Textos

Formatação de Texto: font-family

```
p {  
  font-family: Calibri, Arial;  
}
```

Define uma ou mais fontes a serem utilizadas na página.

É recomendado declarar mais de uma fonte nessa propriedade porque caso a primeira não for encontrada por algum motivo, ele vai para a segunda e assim por diante até a última.

No caso de nenhuma fonte declarada, o navegador utilizará a padrão dele.



Formatação de Texto: font-size

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

Define o tamanho da fonte para o elemento encaixado na regra.

Não esqueça de utilizar a unidade de medida, por exemplo pixels (px).



Formatação de Texto: font-weight

```
p {  
  font-weight: bold;  
}
```

Define a espessura da fonte. É bem parecido com a tag , mas aqui você pode definir vários tipos, de 100 até 900 com intervalos de 100 em 100.

Também é possível utilizar palavras-chave como normal, bold, bolder, lighter.

Atenção: nem todas as fontes possuem todas as variações de espessura.

Formatação de Texto: font-style

```
p {  
  font-style: italic;  
}
```

Define a direção da fonte. É bem parecido com a tag .

É possível utilizar palavras-chave como normal e italic.



Formatação de Texto: text-decoration

```
p {  
  text-decoration: underline;  
}
```

Define se a fonte é sublinhada. É bem parecido com a tag <u>.

É possível utilizar palavras-chave como none e underline.



Formatação de Texto: text-align

```
p {  
  text-align: center;  
}
```

Define o posicionamento do texto.

Pode ser utilizado com palavras-chave como:
right, left, center e justify.



Formatação de Texto: text-transform

```
p {  
  text-transform: uppercase;  
}
```

Define o padrão de caixa do texto. Como caixa baixa, caixa alta ou caixa alta apenas nas primeiras letras.

Pode ser utilizado com palavras-chave como: lowercase, uppercase e capitalize.



Formatação de Texto: line-height

```
p {  
  line-height: 26px;  
}
```

Define a altura da linha e pode ser utilizado para dar espaçamento entre linhas no elemento HTML.

Deve ser definido um valor numérico.



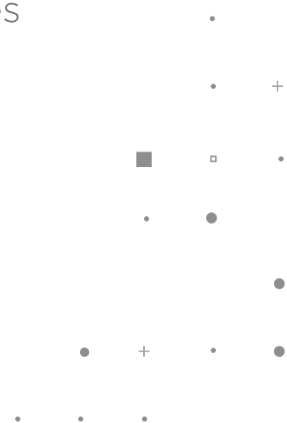
Formatação de Texto: color

```
p {  
  color: #000000;  
}
```

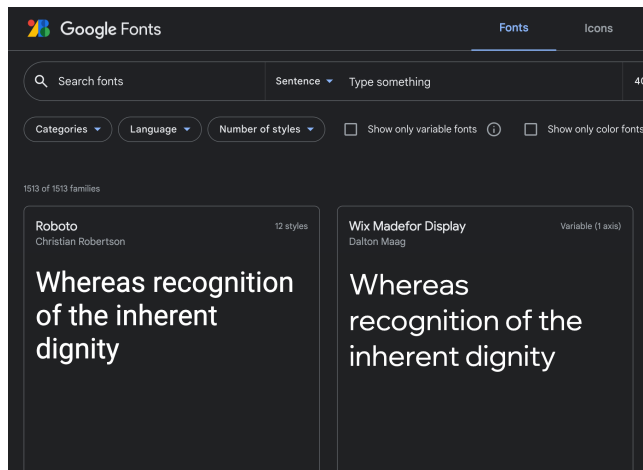
Define a cor do texto.

Pode ser utilizado com o nome da cor em inglês, por exemplo: red, green, blue.

O mais comum é ser utilizado com valores hexadecimais, por exemplo: #000000.



Fontes Personalizadas: Google Fonts

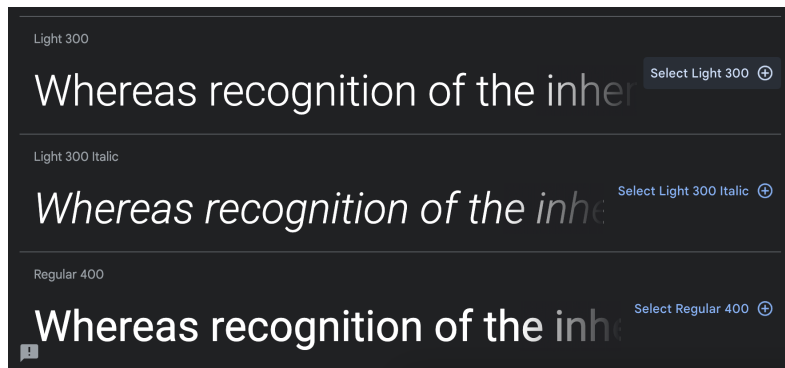


Uma ótima maneira de inserir fontes em seus projetos e sites é utilizando o Google Fonts. Ele possui muitas fontes gratuitas e é muito simples de implementar.

Acesse: <https://fonts.google.com>



Fontes Personalizadas: Google Fonts

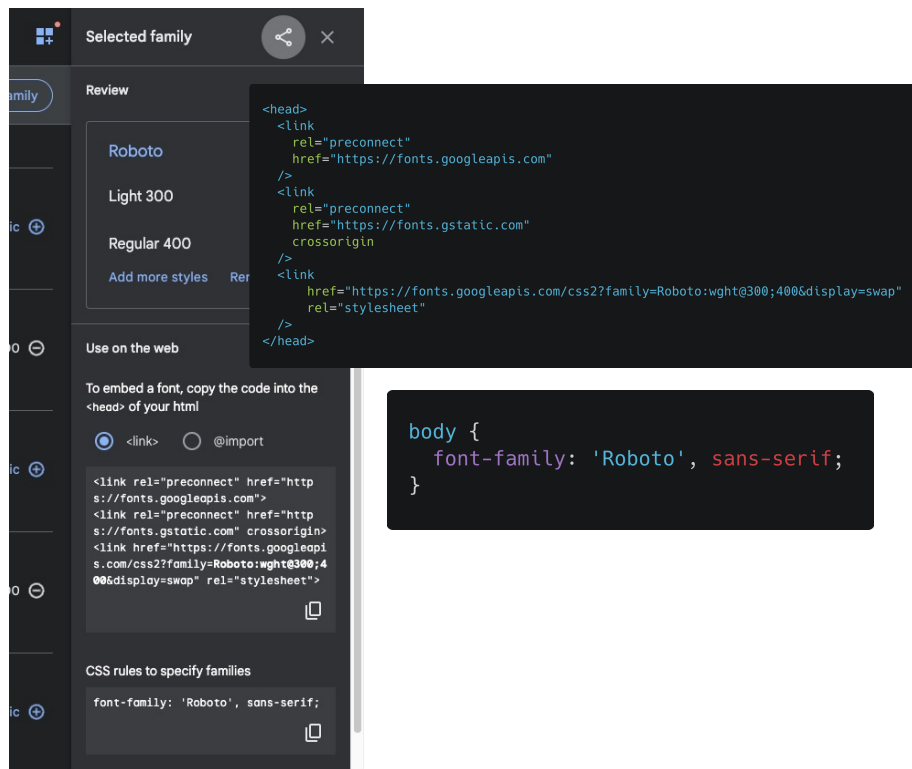


Selecione a fonte que você quer e logo após as variações de font-weight clicando em “Select tipo de fonte +”.

Dica: por questões de performance, procure não selecionar mais do que duas fontes para o seu projeto



Fontes Personalizadas: Google Fonts



Após escolher suas fontes e variações de espessura, clique no botão “Ver fontes selecionadas” no lado superior direito.

Você verá três tags **<link>** do HTML referente as fontes personalizadas. Pode copiar e colar em seu arquivo HTML.

Logo em seguida coloque a **font-family** abaixo na regra CSS que desejar.

Normalmente definimos a fonte de maneira geral no body.

8

Sugestões de **Exercícios**

Exercícios para treinar HTML e CSS

Crie um site que fale sobre você e sua carreira

- Crie páginas como “sobre mim”, “experiência”, “escolaridade” etc;
- Faça link entre essas páginas;
- Utilize as tags headings para separar as sessões da sua página;
- Utilize as tags HTML vistas hoje como parágrafos, listas e imagens;
- Coloque uma fonte personalizada e defina regras de CSS para seus elementos
 - Lembre de usar id e classe de acordo com o que for necessário.

Bons estudos! 😊

Bibliografia básica

W3SCHOOLS - HTML Introduction - 2023

University of Washington - A Brief History of HTML - 2023

W3 - A brief history of CSS until 2016 - 2016





OBRIGADO

@guscsales - gsales.io

FIAP

Copyright © 2023 | Professor Gustavo Campos Sales

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

