# Criar uma nova pasta

Você sabia que há diferentes maneiras de **criar e gerenciar projetos de site?**

Algumas variações ocorrem dependendo das ferramentas específicas que você tem e das preferências da sua organização.

**Ao criar um site**, é comum que a estrutura do projeto sofra alterações ao longo do tempo, conforme ele se torna mais complicado.

A chave é manter um semblante de organização e há estratégias para ajudar com isso!

Projetos grandes, geralmente, exigem um maior grau de cuidado e atenção para que as muitas pessoas envolvidas possam manter tudo organizado!

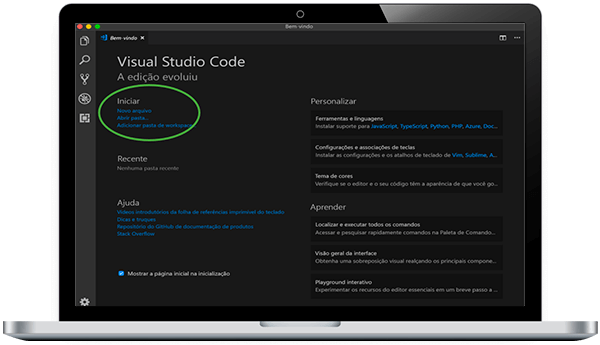
**Aqui você criará uma estrutura de projeto simples que inclui três arquivos: um HTML, um CSS e um JavaScript.**

Você também adicionará uma extensão do **Visual Studio Code** para simplificar a execução do site no navegador.

Vamos iniciar pela **criação de uma nova pasta**, seguindo os passos abaixo:

1. Abra o [**Visual Studio Code**](https://code.visualstudio.com/).

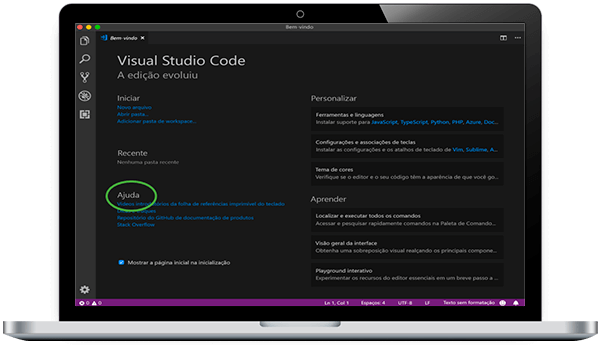
Quando você abre o Visual Studio Code, a página inicial é aberta.



Observe que você pode criar um arquivo ou abrir uma pasta.

Você também pode fazer isso acessando **Arquivo > Novo Arquivo**.

Se não estiver visível, você poderá exibir a janela de **Boas-vindas** navegando para **Ajuda > Boas-vindas**.

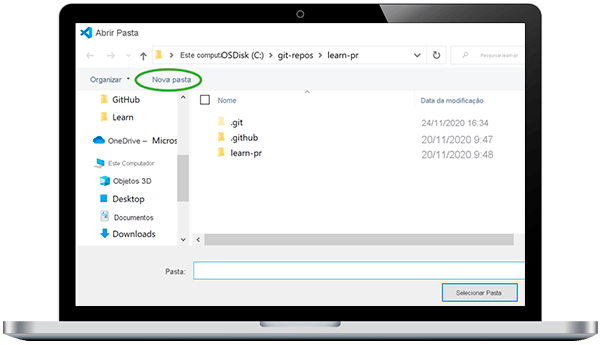


Como alternativa, selecione **Exibir > Paleta de Comandos** ou o atalho de teclado Shift + Command + P.

Em seguida, no campo de pesquisa, digite **> Ajuda: Boas-vindas**.

1. Selecione **Abrir pasta** na seção Iniciar da Página Inicial ou acesse **Pasta > Abrir**.

Ao abrir uma pasta, o sistema operacional fornece um botão para criar uma **Nova Pasta**.



1. Acesse o local onde você deseja criar a pasta e selecione **Nova Pasta**.

Nomeie a pasta como **site-simples**.

**Criar alguns arquivos**

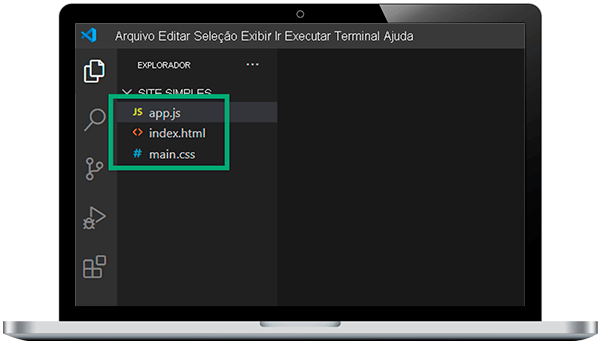
1. Selecione **Arquivo > Novo Arquivo**.
2. Salve o arquivo usando CTRL + S (Windows) ou Command + S (macOS).

Atribua um nome ao arquivo **index.html**.

1. Repita as etapas anteriores para criar mais dois arquivos, **main.css** e **app.js**.

Quando tiver terminado, a pasta do projeto conterá os seguintes arquivos, compondo seu site da Web:

* + index.html
  + main.css
  + app.js



É possível criar um site todo em um **arquivo de texto ou arquivo HTML**.

Aqui você está usando:

* Um arquivo **HTML** para a estrutura da página e para o conteúdo
* Um arquivo **CSS** para apresentação e aplicação de estilos
* Um arquivo **JavaScript** para comportamentos e interatividade

Configurar três arquivos ajuda a nos mantermos organizados!

Isso é um exemplo de melhoria progressiva.

**Se o JavaScript não estiver habilitado ou não for compatível, o CSS e o HTML ainda funcionarão.**

No entanto, se o CSS não funcionar, pelo menos o conteúdo HTML será exibido.

# nstalar extensões ou pacotes

Você pode estender a funcionalidade do **Visual Studio Code** usando o [**Marketplace de extensões**](https://marketplace.visualstudio.com/VSCode).

Tenha em mente que essas são **funcionalidades desenvolvidas pela comunidade**.

Você pode instalar extensões individualmente em seu editor ou instalar várias ao mesmo tempo, com a linha de comando.

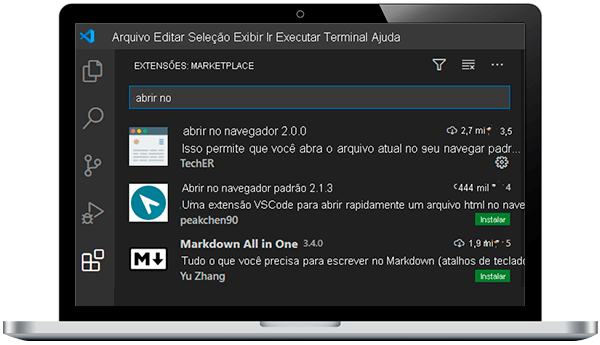
Para o desenvolvimento para Web, tudo que você precisa agora é da [**open-in-browser**](https://marketplace.visualstudio.com/items?itemName=techer.open-in-browser).

Essa extensão ajuda você a abrir rapidamente o site no navegador padrão, em vez de copiar e colar a URL do arquivo no navegador.

## Para instalar essa extensão

1. Selecione o ícone de Extensões na **Barra de Atividade (painel esquerdo)**.
2. Digite **“open in” (abrir no)** na barra de pesquisa.
3. Selecione [**open in browser**](https://marketplace.visualstudio.com/items?itemName=techer.open-in-browser) **(abrir no navegador 2.0.0)**, publicada pela TechER, e escolha **Instalar**.

A extensão é instalada!



A instalação e a configuração demoram um pouco, mas elas só precisam ser realizadas uma única vez.

**Agora estamos prontos para criar um site!**

**Adicionar um código HTML**

No momento, você tem **um arquivo HTML vazio**.

**Vamos adicionar um pouco de código a ele?**

A meta é usar HTML (linguagem de marcação de hipertexto) para descrever a página da Web que o navegador exibirá.

Não seria ótimo ter um modelo inicial?

Você aprenderá a **adicionar o conteúdo HTML básico**, **abrirá a página HTML no navegador** e terá seu primeiro contato com as **ferramentas de desenvolvedor**.

É isso o que veremos a seguir!

**Clique** no botão para acessar o conteúdo.

**Curiosidade**

1. No **Visual Studio Code**, abra **index.html**, selecionando o arquivo **index.html**.

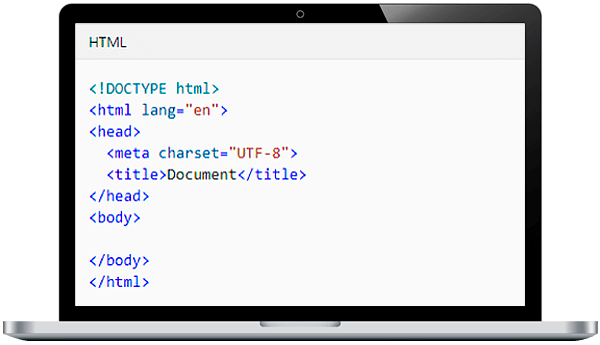
Selecione a página **index.html** e digite **html:5**.

Depois, clique em **Enter**.

O código do modelo HTML5 é adicionado ao arquivo.

1. Edite seu código para que ele fique parecido com o exibido a seguir.

Depois, salve o arquivo com **Control + S** (Windows) ou **Command + S** (macOS).



**Clique** no botão para acessar o conteúdo.

**Atenção**

A marca **meta** **indica informações de metadados** que normalmente não ficarão visíveis para o visualizador, a menos que ele use o próprio navegador para exibir o código-fonte.

**Metaelementos** ou **metamarcas** fornecem informações descritivas sobre a página da Web. Por exemplo, eles ajudam os mecanismos de pesquisa a exibir resultados para as pessoas.

O conjunto de caracteres **UTF-8** pode parecer insignificante, mas é crucial para estabelecer o modo como os computadores interpretam caracteres. Se estiver ausente, o conjunto de caracteres poderá levar a um comprometimento da segurança.

Há uma grande quantidade de informações de histórico e técnicas por trás do conjunto de caracteres.

O panorama é que esse clichê de texto fornece alguns padrões pertinentes.

**Editar o cabeçalho**

O título de uma página da Web é exibido na parte superior de uma janela do navegador e é significativo de diferentes maneiras. Por exemplo, o título é utilizado por mecanismos de pesquisa e exibido neles.

**Agora, vamos adicionar um título!**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Clique** no botão para acessar o conteúdo.

**Curiosidade**

Para estilizar os elementos HTML na página, você pode escrever o código **CSS diretamente no cabeçalho do site**, que é chamado de **CSS interno**.

No entanto, separar a estrutura HTML e o estilo CSS é uma prática recomendada.

Ter uma página de CSS separada é chamado de **CSS externo**.

**Clique** no botão para acessar o conteúdo.

**Dica**

1. No **Visual Studio Code**, adicione uma linha em branco após o elemento **<title>**.

Digite **link** e pressione a tecla **Enter (Return)**.

1. Atualize o **href** para **main.css** e salve o arquivo com **Control + S** (Windows) ou **Command + S** (macOS).

Texto, Quadro de comunicações

Descrição gerada automaticamente

**Editar o corpo**

Vamos começar preenchendo o elemento corpo.

1. Adicione um cabeçalho **<h1>**, parágrafo **<p>** e crie um item de lista **<li>**.
2. Edite ou copie e cole o seu código para que ele fique conforme demonstrado na imagem.

Texto

Descrição gerada automaticamente

**Curiosidade**

Um atributo de ID (usado no **<p>**) pode ser empregado para estilizar **um elemento**, enquanto o atributo de classe (usado no **<li>**) é usado para estilizar **todos os elementos** da mesma classe.

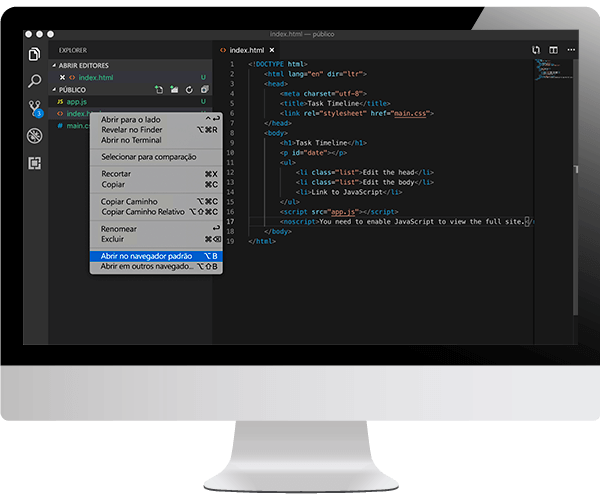
Cada elemento **<li>** de item de lista é agrupado em uma lista não ordenada **<ul>**.

# Abrir no navegador

Você pode visualizar sua página da Web localmente, abrindo o arquivo HTML em um navegador.

Em vez de um endereço de site que começa com **https://**, seu navegador aponta para o caminho de arquivo local.

Por exemplo, o caminho pode ser semelhante ao seguinte: ***file://Users/username/Desktop/public/index.html***.



Para visualizar usando o **Visual Studio Code**, selecione **index.html** e, em seguida, **Abrir no Navegador Padrão** ou use o atalho de teclado **Alt + B**.

A página da Web é aberta no navegador padrão.

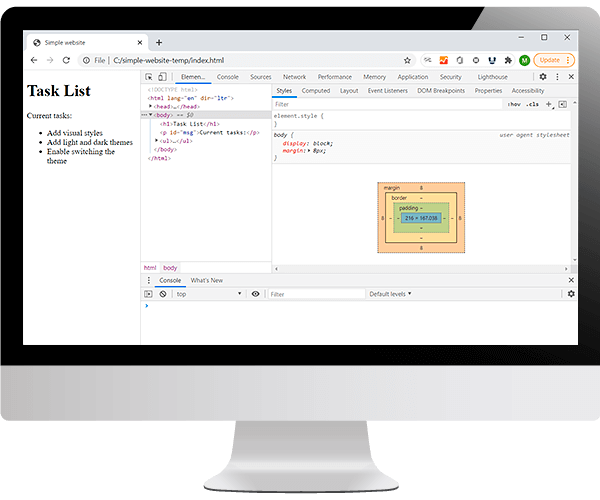
**Atenção**

Se estiver com problemas, verifique se você está  
selecionando diretamente no ícone ou texto do nome do arquivo.

**Exibir a página usando ferramentas de desenvolvedor**

Você pode inspecionar uma página da Web usando as ferramentas de desenvolvedor em seu navegador.

1. Abra as ferramentas para desenvolvedores.
   * Se o seu navegador é o **Microsoft Edge**, pressione o atalho de teclado de **Ferramentas para Desenvolvedores (FN + F12)**. Como alternativa, exiba **Configurações** e mais com **Alt + X** e selecione **Ferramentas de Desenvolvedor**.
   * Se o seu navegador é o **Google Chrome**, pressione o atalho de teclado das **Ferramentas para Desenvolvedores F12 ou (Option + Command + I)**.
2. Selecione a guia **Elementos**.
3. Role sobre os elementos HTML e selecione-os.
4. Abra os triângulos de divulgação.



A guia **Elementos** das ferramentas para desenvolvedores mostra o **DOM** (Modelo de Objeto do Documento) como **renderizado** no navegador.

Durante a depuração, geralmente, é importante ver como o navegador interpreta o código-fonte.

Inspecionar a página em um navegador fornece todos os tipos de informações úteis e pode ajudá-lo a solucionar problemas.

Você também pode exibir detalhes de CSS com o inspetor, mas isso você verá mais adiante neste curso.

# O que é CSS?

**CSS** é a sigla para *Cascading Style Sheets* (em português, Folhas de Estilo em Cascata), que permite que você especifique qual deve ser a **aparência de sua página**.

A ideia básica é direcionar ao código HTML, depois definir qual o estilo deve ser. Por exemplo, você pode marcar uma caixa e aplicar à ela cantos arredondados ou uma tela de fundo gradiente.

O CSS é responsável por determinar a **aparência dos hiperlinks** e como eles respondem ao **interagir** com eles – embora você defina o endereço de destino em HTML.

No **CSS3**, você também pode criar efeitos de animação sofisticados.

A seguir, você aprenderá a aplicar estilos de CSS a elementos de página, adicionará um código CSS para dar suporte a temas claros e escuros e verificará os resultados nas ferramentas para desenvolvedores.

**CSS externo**

No exercício anterior sobre HTML, você vinculou um arquivo CSS externo por meio de HTML.

Um benefício do **CSS externo** é que **várias páginas HTML podem ser vinculadas ao mesmo arquivo CSS**.

Se você fizer uma alteração ao CSS, seu estilo será atualizado para cada página.

Texto

Descrição gerada automaticamente

Designar um arquivo HTML para a estrutura da página e um arquivo CSS para a definição de estilos e arquivos JavaScript para interação ou eventos é chamado de **separação de interesses**.

Como mencionamos anteriormente, você também pode escrever CSS diretamente em HTML, o que é chamado de **CSS interno**.

Mesmo para um site básico, há tantas regras de CSS que a página HTML pode ficar desorganizada muito rapidamente. Com mais de uma página, o mesmo CSS seria repetido com frequência e se tornaria difícil de gerenciar.

# Regras de CSS

Para entender como funcionam as regras de CSS, imagine que você tenha uma chave antiga e uma série de portas em um corredor longo.

Primeiro, você seleciona uma porta. Então, você destranca a porta com a chave.

Depois de ter acesso a uma sala, você pode decorá-la como quiser. Você pode pintar as paredes de azul ou definir o piso como sendo de madeira.

Você seleciona uma sala e define as regras sobre como ela deve parecer.

**Você pode aplicar o mesmo estilo a várias salas**, que é exatamente como funciona o CSS!

1. No **Visual Studio Code**, abra o arquivo **main.css** e digite:

body {

font-family: monospace;

}

ul {

font-family: helvética;

}

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Você pode usar regras de CSS para aplicar estilos ao HTML.

O elemento de lista não ordenada **ul {}** é um seletor que seleciona o elemento HTML **<ul>** ao qual aplicará os estilos.

A declaração é **font-family: helvetica** e informa o que esse estilo deve ser.

O nome da propriedade é **font-family** e o valor é **helvetica**. A propriedade e o valor juntos formam um par chave-valor.

O que você selecionará é um elemento existente que definimos anteriormente em HTML **(<body> e <ul>)**.

**Seletores**

***ID*** e ***seletores* de classe** permitem aplicar estilos a nomes de atributos personalizados em seu HTML.

Uma ID é usada para definir o estilo de um elemento, enquanto classes podem definir o estilo de vários elementos.

1. Copie o código a seguir em seu arquivo CSS, após o seletor **<ul>** adicionado anteriormente.

li {

list-style: circle;

}

.list {

list-style: square;

}

#msg {

font-family: monospace;

}

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

1. Salve seu trabalho pressionando o atalho de teclado **Control + S** (Windows) ou **Command + S** (macOS).

**Atenção**

Os atributos personalizados são chamados de **.list** e **#msg**.

A lista do seletor de classe tem um prefixo de período,

enquanto o seletor de ID é msg e tem um prefixo de sinal de libra.

Os nomes são arbitrários, desde que eles correspondam ao que você definiu no HTML.

**Adicionar um tema claro**

Agora, vamos adicionar suporte para temas de cores em seu site!

Comece definindo um tema de cor clara. Para o tema claro, use códigos de cor hexadecimais para a cor da fonte (#000000, preto) e para a tela de fundo (#00FF00, um matiz verde).

1. **Adicione o código** a seguir ao final do arquivo CSS:

.light-theme {

color: #000000;

background: #00FF00;

}

Texto, Quadro de comunicações

Descrição gerada automaticamente

No arquivo HTML, atualize o elemento **<body>** com um nome de classe **light-theme** e o seletor de classe para tema claro aplicará os estilos corretamente.

1. **Adicione o código** a seguir ao arquivo:

<body class=“light-theme”>

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Exibir no navegador**

Para visualizar usando o **Visual Studio Code**, selecione **index.html** e, em seguida, **Abrir no Navegador Padrão**.

A página da Web é aberta no navegador padrão. Observe que o tema claro que usa uma tela de fundo verde é exibido.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

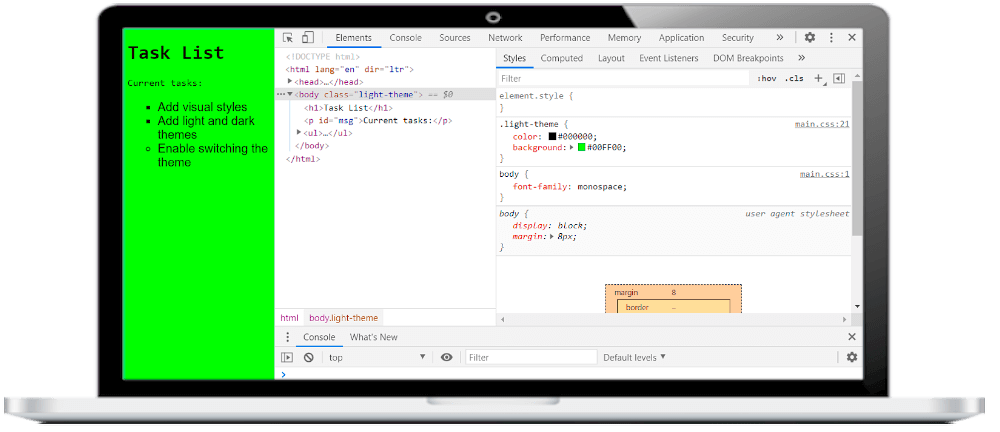
**Exibir CSS aplicado**

1. Abra as ferramentas para desenvolvedores.
   * No **Microsoft Edge**, pressione o atalho de teclado de **Ferramentas para Desenvolvedores** **(FN + F12)**.

Como alternativa, exiba **Configurações** e mais com **Alt + X** e selecione **Ferramentas de Desenvolvedor**.

* + No **Chrome**, pressione o atalho de teclado das **Ferramentas para Desenvolvedores (Option + Command + I) (F12 também funciona)**.

1. Selecione a guia **Estilos**.
2. Selecione a guia **Elementos**.
3. Role sobre os elementos HTML e selecione-os.
4. Selecione o elemento **<body>**. Observe o **light-theme** aplicado.
5. Abra os triângulos de divulgação.
6. Selecione um elemento **<li>**. Observe o **font-family: helvetica**; de estilo personalizado, que substitui o estilo do elemento **<body>**.



**Adicionar um tema escuro**

Para o tema escuro, defina as configurações como preparação para a próxima unidade, na qual você habilitará a alternância de temas na página da Web.

Para configurar o suporte a um tema escuro execute os seguintes passos:

1. Adicione algumas constantes à raiz da página no arquivo CSS.

Texto, Quadro de comunicações

Descrição gerada automaticamente

**Dica**

O seletor de **:root** representa o elemento **<html>** na página HTML.

Para esse tipo de tarefa, uma melhor prática é definir um conjunto de variáveis CSS globais no elemento **:root**.

Aqui, você define três variáveis de cor anexadas à raiz da página.

1. No final do arquivo CSS, adicione o seletor de **dark-theme** e atualize o seletor de **light-theme**.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

No código anterior, você deve definir algumas variáveis novas, que são **bg** e **fontColor**, para especificar a cor da tela de fundo e a cor da fonte.

Use a palavra-chave **var** para especificar as **variáveis** que serão usadas como valores de propriedade.

Você definiu os valores anteriormente no seletor de **:root**.

1. Em seguida, após o seletor de **:root** no arquivo CSS, adicione o seguinte código para substituir o seletor de **body** atual:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

O seletor **\*** é universal e se aplica a todos os elementos da página (exceto quando um seletor de elemento mais específico o substitui).

Aqui, você o usa com o objetivo de definir a propriedade de cor (**color**) padrão para todos os elementos de página.

Para as propriedades **color** e **background**, especifique as variáveis definidas nos seletores de tema claro e escuro.

1. Remova o seletor de **#msg** em seu CSS, para que a mesma fonte possa ser aplicada a todos os elementos.
2. Para exibir o tema escuro, edite manualmente o tema padrão no elemento **<body>** como tema escuro (**dark-theme**) e exiba a página no navegador.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

1. Edite o elemento **<body>** para alternar o padrão de volta para o tema claro.

No próximo módulo, você usará o JavaScript para fornecer interatividade e suporte à alternância de temas.

**O que é JavaScript?**

O **JavaScript** (ou *ECMAScript*) é uma linguagem de programação que ajuda você a adicionar interatividade à sua página da Web.

Quando você seleciona um botão, o JavaScript é o código que define o evento ou o comportamento que ocorrerá, assim como abrir uma janela pop-up.

Usando JavaScript, você pode adicionar ou remover conteúdo, como o texto da página da Web, sem recarregá-lo.

Como um desenvolvedor para a Web, você pode usar o navegador para testar e obter comentários sobre seus scripts.

Neste módulo, você configurará o arquivo JavaScript, criará um botão para alternar entre temas claros e escuros e conectará o botão ao código JavaScript que executa a alternância de tema real.

Quando isso for feito, você verificará o projeto concluído nas ferramentas para desenvolvedores.

**Vincular ao JavaScript**

Assim como o CSS, você pode adicionar JavaScript diretamente à página HTML, mas recomendamos a você não fazer isso.

Por exemplo, você pode criar um alerta de pop-up adicionando **<script>alert('Hello World')</script>** em qualquer lugar no corpo.

A marca de script **<script>** nos permitirá vincular a um arquivo JavaScript externo.

1. No **Visual Studio Code**, digite **script:src** e pressione a tecla **Enter (Return)**.
2. Ajuste o elemento de script para que tenha a aparência a seguir. Coloque-o logo após a lista.

Texto, Aplicativo, Quadro de comunicações

Descrição gerada automaticamente

**Clique** no botão para acessar o conteúdo.

**Curiosidade**

O elemento de script pode ser colocado no **<head>** ou em outro lugar no **<body>**.

No entanto, colocar **<script>** no final da seção **<body>** permite que todo o

conteúdo da página seja exibido na tela primeiro, carregando o script em seguida.

**Adicionar tolerância de falhas**

1. No arquivo HTML, adicione o elemento **<noscript>**, que pode ser usado para mostrar uma mensagem, caso o JavaScript seja desativado.

Texto

Descrição gerada automaticamente

**Dica**

Usar o elemento **<noscript>** é um exemplo de tolerância a falhas ou degradação gradual.

Podemos detectar quando um recurso não é compatível

ou não está disponível e nos planejar para essa eventualidade.

1. Salve seu trabalho pressionando o atalho de teclado **Control + S** (Windows) ou **Command + S** (macOS).

**Configurar o modo estrito**

Quando você começa a usar o **JavaScript**, o foco inicial, geralmente, é trabalhar com números, matemática, manipulação de texto, datas e armazenamento de informações.

Às vezes, o JavaScript faz suposições sobre o tipo de dados que você insere. Atribuição, matemática ou igualdade lógica podem gerar resultados inesperados.

O JavaScript tenta ser amigável e ele tenta fazer com que o código funcione e forneça uma solução, mesmo que o resultado venha a ser um erro.

Para combater essas deficiências, você pode ativar o **modo estrito**, que reduz erros silenciosos, melhora o desempenho, fornece mais avisos e menos recursos não seguros.

1. No **Visual Studio Code**, abra o arquivo app.js e digite: ‘use strict’

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Dica**

Geralmente, você pode omitir o ponto e vírgula em JavaScript, mas é uma boa prática

revisar quando ele é necessário, antes de tomar qualquer decisão sobre isso.

# Adicionar um botão

Você precisa elaborar alguma forma de permitir que o usuário alterne entre o tema claro e escuro em sua página da Web.

Nesse cenário, isso é feito com um elemento de botão, que é o que veremos a seguir!

1. Na página HTML, adicione um elemento **<button>**.

Coloque o botão no final da lista dentro de um elemento **<div>**.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

1. No arquivo CSS, adicione um seletor para o botão.

Para tornar as cores do botão diferentes das cores gerais dos temas claro e escuro, defina as propriedades **color** e **background-color** no seletor de botão.

Esse seletor, específico do botão, substitui o seletor universal (\*) usado para aplicar cores de fonte ao arquivo CSS.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Em seguida, adicione algumas regras para moldar o tamanho, a forma, a aparência e o posicionamento do botão.

O CSS a seguir cria um botão arredondado à direita do título da página.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Para melhorar a aparência do botão, adicione um seletor de pseudoclasse, **btn:focus**, após o seletor de botão.

Ao definir a regra de **outline-style** como **none**, você elimina um contorno retangular quando o botão é selecionado (recebe foco).

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

1. Em seguida, atualize o CSS para os temas claro e escuro.

Defina algumas novas variáveis, **btnBg** e **btnFontColor**, para especificar as cores da tela de fundo e da fonte do botão.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

**Adicionar um manipulador de eventos**

Para fazer com que o botão realize alguma tarefa ao ser pressionado, você precisará de um **manipulador de eventos** no arquivo JavaScript.

Para um botão, você precisa de um manipulador do evento **click**. A função do manipulador de eventos é executada quando o evento **click** ocorre.

Antes de adicionar o manipulador de eventos, você precisa de uma **referência para o botão**.

1. No arquivo JavaScript, use **document.querySelector** para obter a referência do botão.

Uma imagem contendo Gráfico de dispersão

Descrição gerada automaticamente

1. Em seguida, adicione o ouvinte e o manipulador do evento **click**.

No código a seguir, você adiciona um ouvinte para o evento **click**.

A função passada para o ouvinte de eventos é seu manipulador de eventos real.

Texto

Descrição gerada automaticamente

No código anterior, você usou o método **toggle** para alternar o elemento para a classe **dark-theme**.

Isso aplica, automaticamente, os estilos de tema escuro em vez do tema claro. No entanto, o rótulo do botão também precisa ser atualizado para mostrar o tema correto, portanto, você precisa adicionar uma instrução **if** para verificar o tema atual e atualizar o rótulo do botão.

Confira como deve ser o código JavaScript completo:

Texto

Descrição gerada automaticamente

**Dica**

É uma convenção JavaScript usar letras concatenadas para nomes de variáveis

com mais de uma palavra – por exemplo, a variável **className**.

# Mensagem de console

Você pode criar uma mensagem oculta que não aparecerá na sua página da Web. No entanto, o que você escreve no console aparecerá nas ferramentas de desenvolvedor do navegador.

Usar mensagens do console pode ser muito útil para ver o resultado do código.

1. Adicione uma chamada para **console.log** após a instrução **if**, mas dentro do ouvinte de eventos.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

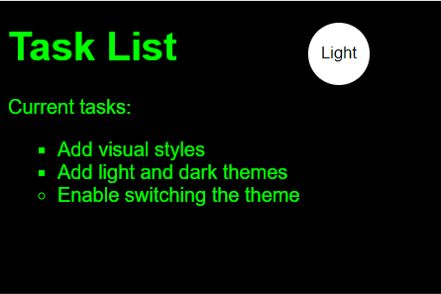
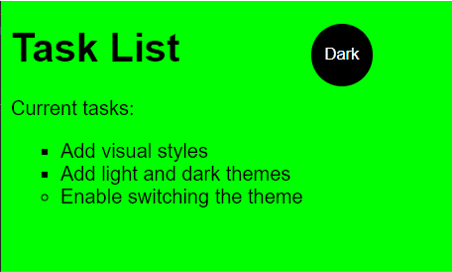
1. No **Visual Studio Code**, quando estiver no arquivo JavaScript, você pode usar o preenchimento automático do **log** digitando **console.log** e pressionando Enter (Return).

Você pode definir uma cadeia de caracteres de texto com aspas simples ou duplas ao redor do texto.

**Abrir no navegador**

Assim como antes, mesmo que você estivesse apenas editando o arquivo **app.js**, para visualizar as alterações, você deveria selecionar o arquivo **index.html**.

1. Para visualizar, selecione **index.html** e **Abrir no Navegador Padrão**. A página da Web é aberta no navegador padrão.
2. Clique no novo botão Escuro, para alternar para o tema escuro.
3. Verifique se tudo está correto.



**Verificar a página nas ferramentas para desenvolvedores**

1. Abra as ferramentas para desenvolvedores.
   * No **Microsoft Edge**, o atalho de teclado para **Ferramentas de Desenvolvedor** é F12 (FN + F12).

Como alternativa, você pode exibir **Configurações e mais** com Alt + X e selecionar **Ferramentas de Desenvolvedor**.

* + No **Chrome** do macOS, o atalho de teclado para o **Console** nas Ferramentas do desenvolvedor é Option + Command + J (F12 também funciona).

1. Selecione a guia **Estilos**.
2. Selecione a guia **Elementos**.
3. Selecione o elemento **<body>**.

Na guia **Estilos**, examine o tema aplicado.

Caso o tema atual seja escuro, os estilos de **dark-theme** estão aplicados.

Verifique se o tema escuro está selecionado.

1. Selecione a guia **Console** para ver a mensagem **console.log** "nome da classe atual: light-theme dark-theme".

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Curiosidade**

Usando o console, você tem uma visão interessante de como

a alternância de temas CSS é tratada.

Ambos os nomes de classe são aplicados ao elemento

quando você alterna para tema escuro.

No entanto, o último nome de classe aplicado, o tema escuro, tem precedência.

Na guia **Estilos**, você pode ver que as regras de tema escuro substituem

as regras de tema claro, que são mostradas usando o texto tachado.