



Entendimento do código-fonte

E

ntendimento do código-fonte

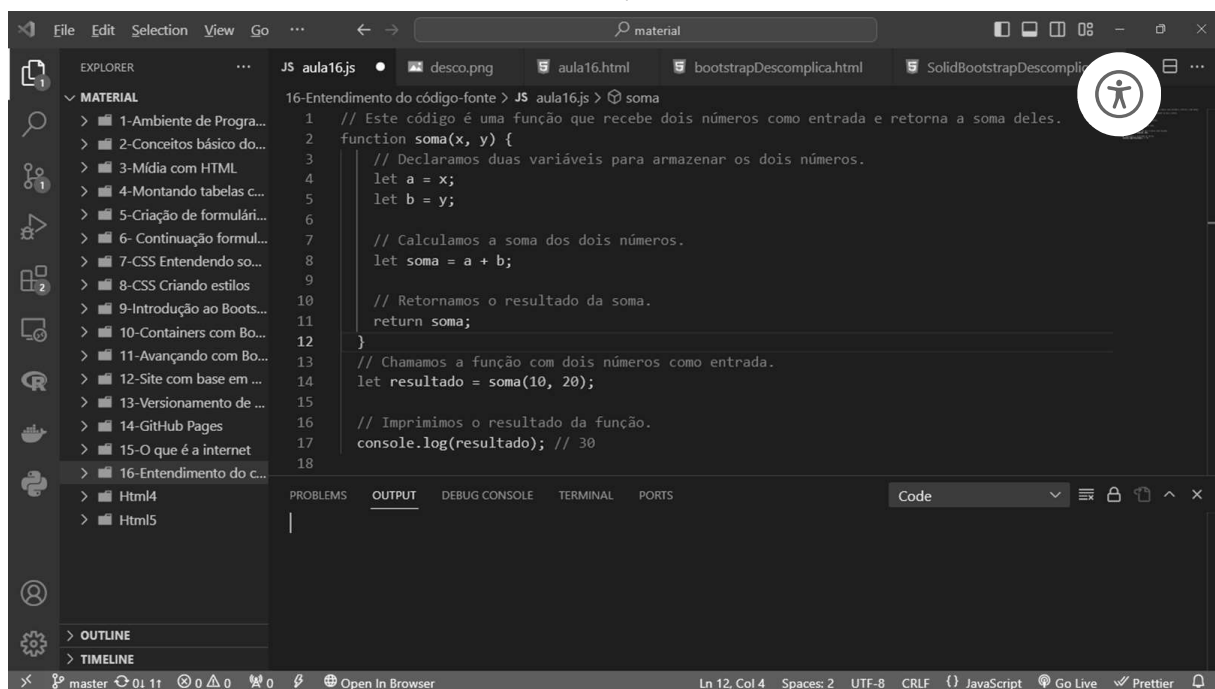
O código-fonte é o conjunto de instruções que um computador usa para executar um programa. Ele é escrito em uma linguagem de programação, que é um conjunto de regras que definem como as instruções devem ser escritas.

Para entender o código-fonte, é importante conhecer os conceitos básicos da linguagem de programação em que ele está escrito. Além disso, é importante praticar a leitura e a compreensão de código.

A leitura de código pode ser feita de forma ativa ou passiva, a ativa envolve a compreensão do significado do código, enquanto a passiva envolve apenas a compreensão da estrutura do código.

A compreensão do código-fonte pode ser aprimorada por meio de prática e exercícios. Existem muitos recursos disponíveis para auxiliar os desenvolvedores a aprender e praticar a leitura e a compreensão de código.





```
1 // Este código é uma função que recebe dois números como entrada e retorna a soma deles.
2 function soma(x, y) {
3   // Declaramos duas variáveis para armazenar os dois números.
4   let a = x;
5   let b = y;
6
7   // Calculamos a soma dos dois números.
8   let soma = a + b;
9
10  // Retornamos o resultado da soma.
11  return soma;
12 }
13 // Chamamos a função com dois números como entrada.
14 let resultado = soma(10, 20);
15
16 // Imprimimos o resultado da função.
17 console.log(resultado); // 30
18
```

Exemplo de um arquivo aula16.js na IDE VSCode.

1 - Por que é importante saber programar?

Saber programar é uma habilidade valiosa em um mundo cada vez mais digital. A programação é a base para o desenvolvimento de software, essencial para uma ampla gama de aplicações, incluindo sites, aplicativos, jogos e sistemas operacionais.

Aqui estão alguns dos benefícios de saber programar:

- Pode abrir novas oportunidades de carreira: a programação é uma habilidade altamente valorizada no mercado de trabalho. Os desenvolvedores de software são frequentemente bem remunerados e têm muitas oportunidades de emprego.
- Pode ajudar a resolver problemas: a programação pode ser usada para resolver problemas de forma criativa e eficiente. Os programadores são

capazes de automatizar tarefas, otimizar processos e criar novos produtos e serviços.



- Pode ser uma forma de expressão: a programação pode ser usada para criar arte, música, jogos e outras formas de entretenimento. Os programadores são capazes de dar vida às suas ideias e compartilhar suas criações com o mundo.


2 - Básico das linguagens de programação

Uma linguagem de programação é um conjunto de regras que definem como as instruções devem ser escritas para um computador poder entendê-las e executá-las. As linguagens de programação são usadas para criar software, que é o conjunto de instruções que um computador usa para realizar uma tarefa específica.

Os conceitos básicos de uma linguagem de programação incluem:

- **Variáveis:** são espaços de memória que armazenam dados.
- **Operadores:** são símbolos usados para realizar operações matemáticas, lógicas e de comparação.
- **Controle de fluxo:** é usado para controlar a ordem de execução das instruções.
- **Funções:** são blocos de código que podem ser reutilizados e chamados para executar uma tarefa específica.

3 - Tipos de linguagens de programação

Existem muitos tipos de linguagens de programação, cada uma com suas próprias características e aplicações. Os tipos mais comuns incluem: 

- **Linguagens de alto nível:** são mais fáceis de entender para humanos.
- **Linguagens de baixo nível:** são mais próximas da linguagem de máquina, que é o código que os computadores entendem diretamente.
- **Linguagens interpretadas:** são convertidas em código de máquina na hora da execução.
- **Linguagens compiladas:** são convertidas em código de máquina antes da execução.

4 - Aprender a programar

Aprender a programar pode ser uma tarefa desafiadora, mas também pode ser muito gratificante. Existem muitos recursos disponíveis para auxiliar as pessoas a aprender a programar, incluindo livros, cursos online e comunidades de programação.

Dicas para aprender a programar:

- **Comece com os fundamentos:** antes de começar a programar, é importante aprender os fundamentos da programação, como variáveis, operadores, controle de fluxo e funções.
- **Pratique regularmente:** a melhor maneira de aprender a programar é praticar regularmente. Reserve um tempo todos os dias para aprender e praticar novas habilidades.
- **Junte-se a uma comunidade:** existem muitas comunidades online e offline para programadores. Essas comunidades podem ser um ótimo lugar

para aprender, compartilhar ideias e obter ajuda.



5 - Investigar o código-fonte


Investigar o código-fonte é um processo de análise do código-fonte de um programa para entender seu funcionamento. Essa atividade pode ser realizada por desenvolvedores de software, pesquisadores de segurança, ou qualquer pessoa interessada em entender como um programa funciona.

Os objetivos da investigação do código-fonte podem variar conforme o contexto. Em geral, incluem:

- **Compreender o funcionamento do programa:** pode ser feito para fins de manutenção, depuração, ou simplesmente para aprender mais sobre um programa.
- **Identificar vulnerabilidades de segurança:** as vulnerabilidades de segurança podem ser usadas por atacantes para comprometer um sistema. A investigação do código-fonte pode ajudar a identificar essas vulnerabilidades antes que sejam exploradas.
- **Reutilizar código:** o código-fonte pode ser reutilizado em outros projetos. A investigação do código-fonte pode ajudar a entender como o código funciona e como ele pode ser adaptado para outros propósitos.

6 - Métodos para investigar o código-fonte


Existem vários métodos que podem ser usados para investigar o código-fonte. Alguns dos mais comuns incluem:

- **Leitura do código:** é o método mais básico de investigação. Isso envolve a leitura do código-fonte linha por linha para entender  seu significado.
- **Análise estática:** é um processo de análise do código-fonte sem executá-lo. Isso pode ser feito usando ferramentas automatizadas ou manualmente.
- **Análise dinâmica:** é um processo de análise do código-fonte executando-o. Isso pode ser feito usando um depurador ou um simulador.
- **Os testes de unidade e de integração:** são métodos usados para verificar se o código-fonte está funcionando conforme o esperado.

7 - Dicas para Investigação

A investigação do código-fonte pode ser uma tarefa desafiadora, mas também pode ser muito gratificante. Aqui estão algumas dicas para tornar a investigação mais eficaz:

- **Comece com o básico:** antes de começar a investigar o código-fonte de um programa complexo, é importante aprender os fundamentos da linguagem de programação em que ele está escrito.
- **Use ferramentas de apoio:** existem muitas ferramentas disponíveis para auxiliar na investigação do código-fonte. Essas ferramentas podem automatizar tarefas e tornar o processo mais eficiente.
- **Divida o código em partes:** em vez de tentar entender o código na totalidade, divida-o em partes menores e mais fáceis de entender.

- **Não tenha medo de perguntar:** se você não entender algo, não tenha medo de perguntar a um desenvolvedor experiente c  uma comunidade online.

Conteúdo Bônus

Ferramenta de inspecionar código do Google Chrome

A ferramenta de inspecionar código do Google Chrome é um recurso poderoso que permite aos desenvolvedores e usuários avançados visualizar e editar o código-fonte de uma página da web. Ela pode ser usada para solucionar problemas de desenvolvimento, personalizar páginas da web e aprender mais sobre como as páginas da web funcionam.

Como abrir a ferramenta de inspecionar código

Para abrir a ferramenta de inspecionar código, siga estas etapas:

1. Abra o navegador Google Chrome.
2. Navegue até uma página da web.
3. Clique com o botão direito do mouse em qualquer lugar da página e selecione "Inspecionar".

Abas da ferramenta de inspecionar código

A ferramenta de inspecionar código tem quatro abas principais:

- **Elements:** exibe o código HTML da página da web.

- **Styles:** exibe os estilos CSS da página da web.
- **Network:** exibe o tráfego de rede da página da web.
- **Application:** exibe informações sobre os recursos da página da web, como scripts e imagens.



Recursos da ferramenta de inspecionar código

A ferramenta de inspecionar código também inclui uma variedade de recursos que podem ser usados para visualizar e editar o código-fonte.

- **Seletor:** permite selecionar elementos da página da web.
- **Console:** permite executar comandos JavaScript.
- **Breakpoints:** permite interromper a execução de uma página da web em um ponto específico.
- **Sources:** permite visualizar e editar o código-fonte de arquivos externos.

Usos da ferramenta de inspecionar código

A ferramenta de inspecionar código pode ser usada para uma variedade de propósitos, incluindo:

- **Solucionar problemas de desenvolvimento:** pode ser usada para identificar e corrigir erros de código.
- **Personalizar páginas da web:** pode ser usada para adicionar ou remover elementos de uma página da web.

- **Aprender sobre desenvolvimento web:** pode ser usada para aprender sobre como as páginas da web funcionam.



Exemplos de uso da ferramenta de inspecionar código


Aqui estão alguns exemplos de como a ferramenta de inspecionar código pode ser usada:

- **Para solucionar um erro de HTML:** se você estiver vendo um erro de HTML em uma página da web, você pode usar a ferramenta de inspecionar código para identificar o código que está causando o erro.
- **Para adicionar um novo estilo CSS:** se você quiser adicionar um novo estilo CSS a uma página da web, você pode usar a ferramenta de inspecionar código para adicionar o estilo ao código CSS da página.
- **Para rastrear o tráfego de rede:** se você quiser rastrear o tráfego de rede de uma página da web, você pode usar a ferramenta de inspecionar código para visualizar o tráfego na guia “Network”.

Para aprofundar seu conhecimento sobre o assunto, procure o livro: **“Web Development with Google Chrome DevTools”**, de Emily Bache.

Referência Bibliográfica

BECK, Kent. **Padrões de implementação: um catálogo de padrões indispensável para o dia a dia do programador**. Porto Alegre: Bookman, 2013.

SANTOS, F.; SANTANA, J. R.; SILVA, P. R. S. de. Entendimento de código-fonte: desafios e abordagens. **Revista Brasileira de Computação**  **ada**, Passo Fundo, v. 11, n. 2, p. 1-15, 2019.

SILVA, L. O. da; SOUZA, M. A. de; NUNES, L. de S. Análise estática de código-fonte: uma revisão sistemática. **Revista Brasileira de Computação Aplicada**, Passo Fundo, v. 13, n. 2, p. 1-20, 2022.

Atividade Prática 16 — Entendimento do código-fonte

Título: Desvendando o Código-Fonte

Objetivo: Familiarizar os alunos com a leitura e compreensão de código-fonte, permitindo que identifiquem estruturas, funções e lógicas de programação.

Materiais e Ferramentas:

- IDE (Ambiente de Desenvolvimento Integrado) de sua preferência (sugestão: VSCode).
- Código-fonte exemplo (fornecido nesta atividade).
- Acesso à internet para consultas.

Instruções:

1. Abaixo, você encontrará um trecho de código-fonte em JavaScript. Seu objetivo é analisá-lo diretamente aqui.
2. Leia atentamente o código-fonte, linha por linha.
3. Identifique e anote:
 - Todas as variáveis e seus tipos.

- As funções presentes e suas respectivas funcionalidades.
- Os operadores utilizados.
- Estruturas de controle de fluxo (como loops e condicionais).



4. Em um documento à parte (pode ser um arquivo de texto ou um documento online), comente cada linha ou bloco de código, explicando em suas próprias palavras o que cada parte faz.

5. Identifique qualquer parte do código que você não entenda e destaque-a.

Código-fonte para análise:

javascript

```
function calcularAreaTriangulo(base, altura) {
```

```
    let area = (base * altura) / 2;
```

```
    return area;
```

```
}
```

```
let baseTriangulo = 10;
```

```
let alturaTriangulo = 5;
```

```
let area = calcularAreaTriangulo(baseTriangulo, alturaTriangulo);
```

```
console.log("A área do triângulo é: " + area + " unidades quadradas.");
```



Resolução Comentada:

- `function calcularAreaTriangulo(base, altura) {` - Esta linha define uma função chamada `calcularAreaTriangulo` que recebe dois parâmetros: `base` e `altura`.
- `let area = (base * altura) / 2;` - Aqui, calculamos a área de um triângulo usando a fórmula $(base * altura) / 2$ e armazenamos o resultado na variável `area`.
- `return area;` - Esta linha retorna o valor da variável `area`.
- `let baseTriangulo = 10;` - Declaramos uma variável chamada `baseTriangulo` e atribuímos o valor 10 a ela.
- `let alturaTriangulo = 5;` - Declaramos uma variável chamada `alturaTriangulo` e atribuímos o valor 5 a ela.
- `let area = calcularAreaTriangulo(baseTriangulo, alturaTriangulo);`
- Aqui, chamamos a função `calcularAreaTriangulo` com os valores de `baseTriangulo` e `alturaTriangulo` e armazenamos o resultado retornado na variável `area`.
- `console.log("A área do triângulo é: " + area + " unidades quadradas.");` - Esta linha exibe a área calculada no console.

Ir para exercício