

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
PARAÍBA

# Introdução à Linguagem Script para web

## Introdução à Linguagem de Script para web



Javascript

Javascript/ECMA+

/ECMA+

Prof. Mr. Fabio Abrantes Diniz  
fabio.abrantes.diniz@gmail.com

Introdução

# Introdução

## O que é Programação Web?

- Processo de criar **sites, sistemas e aplicativos** que rodam na internet.



# Introdução

## Introdução

- **Importância**



- **Transformação Digital**

- Empresas **dependem de sistemas web** para



## ■ **Acessibilidade Global**

- Qualquer pessoa com internet pode acessar aplicativos.



## ■ **Inovação e Novas Tecnologias**

- Aplicações evoluem com **IA, IoT, Blockchain e Web3.**



## ■ **Oportunidades Profissionais**

- Alta demanda por **desenvolvedores web full-stack, front-end e back-end.**



# Introdução

## **Mercado de programação**

- Vagas

- [Mais de 790 mil até 2025 - Brasscom](#)

- Global

- Trabalhar em qualquer lugar do mundo

- Tempo de Experiência

- Cerca de 2 anos



## Introdução

### Áreas de atuação e salários

- Podemos aprender programação em diversas

## áreas de atuação:

- WEB, Mobile, Dados, Inteligência artificial (A.I)
- Internet das coisas (IoT) e outras.
- O **salário** é de acordo com a vaga e a experiência da pessoa.
  - Podem iniciar entre 2 a 3 salários mínimos e chegar a mais de 10, 20



## Introdução

### Habilidades

- Hard skills

- Habilidades técnicas e conceitos da área
- Soft skills
  - Habilidades comportamentais:
    - comunicação, resolução de problemas, gerenciamento de tempo, trabalho em equipe ...



# Introdução

**O que é programar?**

- **Codar, Desenvolver, Programar** • Resolver problemas do mundo real por meio das linhas de código
- **Falar com o computador**
  - Aprender linguagens para dar ordens ao computador
- **Aprender Lógica de programação**
  - Algoritmos => programas



# Introdução

## Principais tipos de lógica de programação



# web

- **Programação imperativa:**

- Dar ordens - Seguir passos e sequências – ✓se preocupa **como** o programa deve executar tarefas. ▪ Ex: Javascript, typescript, java, C, C++, ... ●

## **Programação declarativa:**

- Conjunto de regras definidas.
  - ✓O foco está em **o que** precisa ser feito, e não em **como** fazer
- Ex: html, css, Javascript, typescript, reactjs, sql, ...



# Introdução

# Principais tipos de programação web

- **Programação imperativa:**

- Usa **estruturas de controle** como loops (**for**, **while**) e condicionais (**if**, **switch**)



- **Subtipos:**

- **Programação Estruturada:** Usa loops, condicionais e sub-rotinas.
- **Programação Orientada a Objetos (POO):** Baseada em classes, objetos, herança e encapsulamento.





# Introdução

## Principais tipos de programação

### web • **Programação declarativa:**

- O programador **define regras e condições**, sem especificar cada passo.



### ■ **Subtipos:**

- ✓ **Programação Funcional:** Baseada em funções puras, imutabilidade e composição.
  - Ex: reactjs





# Introdução

## Principais tipos de programação web • **Programação declarativa:**

- Html e css é declarativo,
  - ✓ Pois, descrevem a **estrutura** e **aparência** sem indicar como o navegador deve processá-los.





<https://terrific-swordtail-92f.notion.site/Principais-elementos-HTML1c7e9e7fd1d780e2b0aff18f232542c1?pvs=4>



# Introdução

## Diferença entre **linguagens de Programação** e **Linguagens de Script**

- Linguagens de Scripts

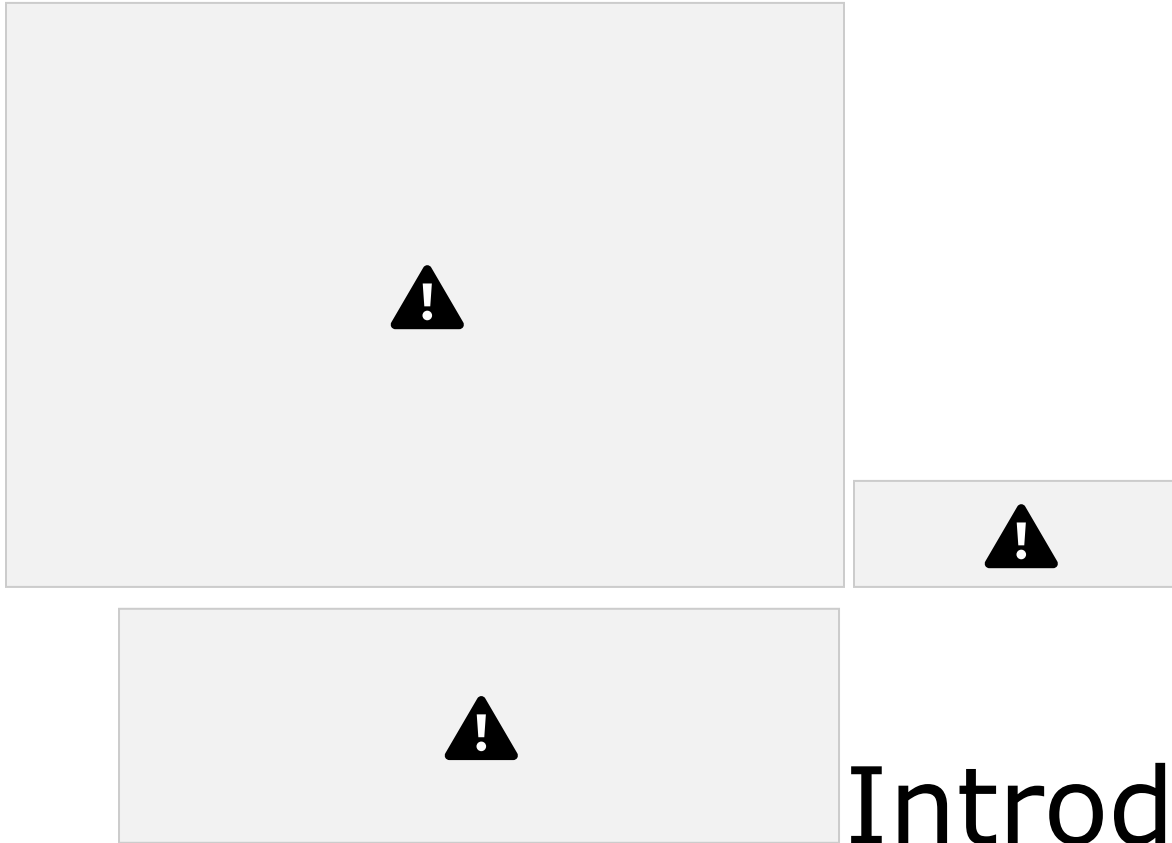
✓ São um **subconjunto das linguagens de programação**, mais voltadas para **automação, desenvolvimento web e produtividade**



# Introdução

## **Internet**

- Rede mundial de computadores
- Os computadores estão conectados entre si
- Transferência de dados



# Introdução

## Como funciona a internet (web)?

- **Internet**

- A comunicação na internet é possível graças **ao protocolo TCP/IP**,
  - ✓ Contém regras que permite os computadores

compartilharem dados entre si.



# Introdução

**Como funciona a internet (web)? •**

**TCP (*Transmission Control Protocol*)**

- “É responsável por garantir que os **dados sejam transmitidos de forma precisa e confiável entre os computadores**. Divide os dados em pacotes e os envia



através da internet, garantindo que todos os pacotes cheguem ao destino correto e na ordem correta.” **Andrew S. Tanenbaum**



## Introdução

**Como funciona a internet (web)?** •

### **IP (*Internet Protocol*)**

- “é responsável por garantir que os pacotes de dados cheguem ao destino correto. Atribui um endereço IP único

**a cada dispositivo conectado à internet** e usa esse endereço para encaminhar os pacotes de dados para o destino correto” **Andrew S. Tanenbaum**



## Introdução

**Como funciona a internet (web)?** •

**Outros protocolos que trabalha junto com o TCP/IP**

- **DNS** (*Domain Name System*)

✓ Responsável por traduzir os endereços de nomes de domínio, como [www.example.com](http://www.example.com), para endereços **IP**, – permitindo que os usuários acessem sites com nomes fáceis de lembrar.

- **HTTP** (*Hypertext Transfer Protocol*)

✓ é usado para transferir dados na *World Wide Web*.



# Introdução

## Como funciona a internet (web)?





# Introdução

**Como funciona a internet (web)? Parei  
aqui**

**Modelo cliente x servidor**





# Introdução

## Programação WEB

- **Front-end**

- A parte visual das coisas, a interação com o usuário

- **Back-end**

- A parte lógica das coisas, as regras de negócio

- **Fullstack**

- Front e Back juntos





# Introdução

**Como funciona a internet  
(web)? Modelo cliente x servidor**

**Front-end** **Back-end**





# Introdução

## Tecnologias do front-end



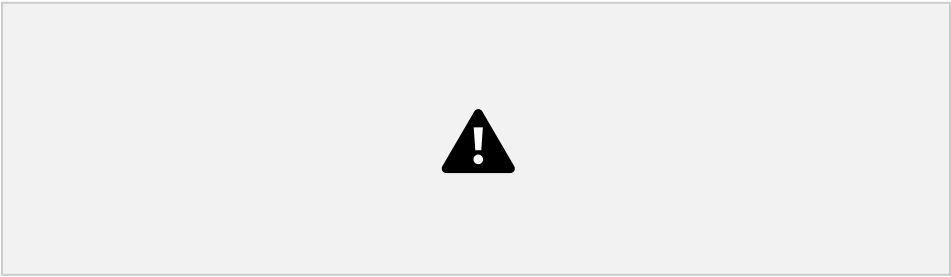
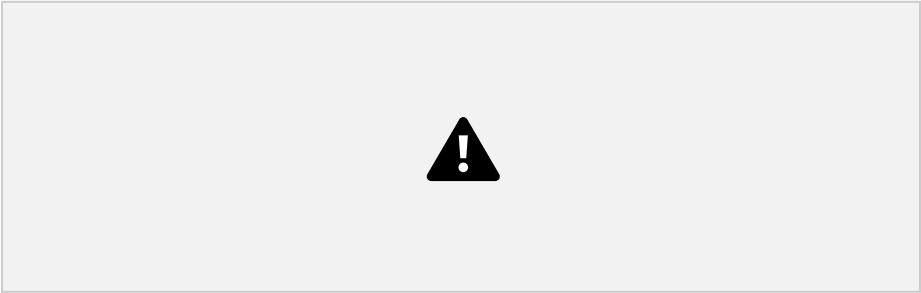
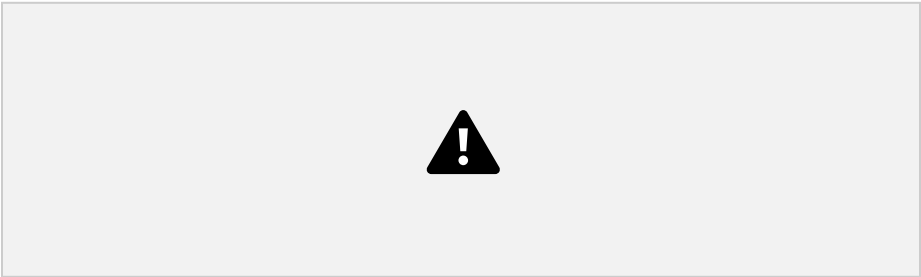
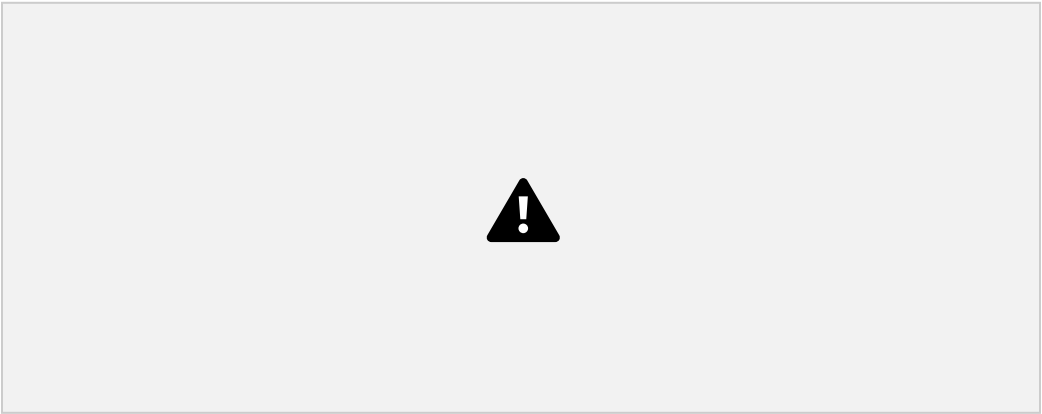




# Introdução

## **Tecnologias do back-end**

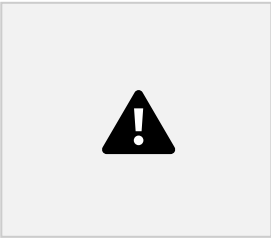


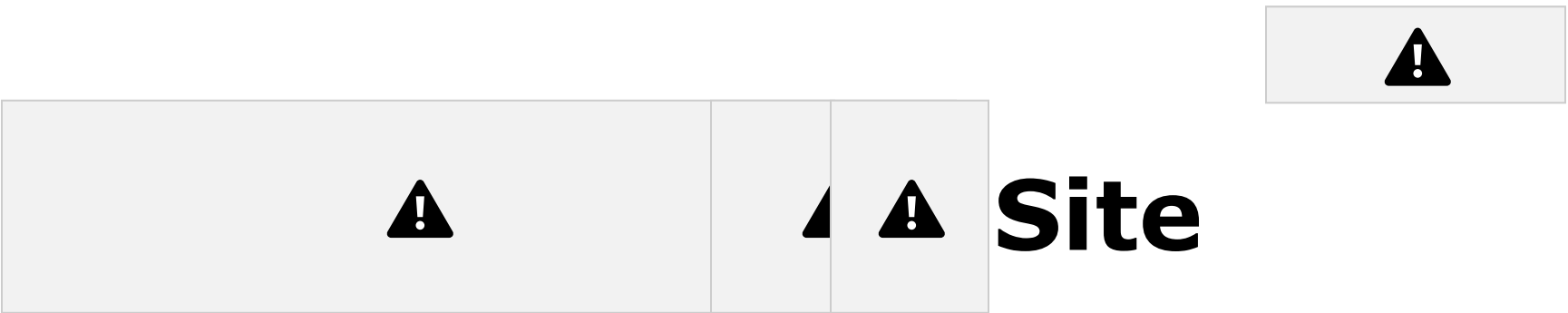




**Webpage vs**

**Website**





# Estático vs Site Dinâmico





# Aplicativos Web

- É um software que roda diretamente em um navegador
  - site bancário, um e-commerce ou um sistema de gestão



# **Aplicação Web Tradicional vs SPA (*Single Page Application*)**







The diagram at the top illustrates two types of web applications. On the left, a traditional multi-page application is shown with a vertical stack of four pages. The top page is highlighted with a black border and contains a warning icon. On the right, a single-page application (SPA) is shown with a single page highlighted by a black border, also containing a warning icon.

# **Aplicação Web Tradicional vs SPA (*Single Page Application*)**





# **Aplicação Web Tradicional vs SPA (*Single Page Application*)**



Aa





# Aplicação Web Tradicional vs SPA (*Single Page Application*)

- **Quando Escolher cada uma?**
  - **Aplicação Web Tradicional** → Se precisar de um site mais simples, com foco em carregamento rápido.
  - **SPA** → Se for um sistema interativo, como um dashboard, onde a experiência do usuário importa mais que o SEO.



# Introdução



# Introdução



Introdução







# Introdução





# Introdução

- **São a mesma coisa.**

- A gente conhece por javascript das antigas que patentearam.
  - ✓ Mas o nome oficial hoje é **ECMAScript**
- **ECMAScript (ES)** é a especificação que define a linguagem JavaScript
  - ✓ Foi criada pela **ECMA International**
    - Padronizar o JavaScript
    - Garantir compatibilidade entre navegadores.



# Evolução

do



## JavaScript (ECMAScript)







# Introdução

- De 2009 a 2015
  - Foram acrescentado muitas *features*.
- As novas *features* acrescentadas não invalidam as anteriores.
- Quando essas *features* são implementadas os browser ainda não dão suporte.

<https://compat-table.github.io/compat-table/es2016plus>  
/ <https://caniuse.com/>





# Typescript

- **Uma linguagem de programação**

- Fortemente **tipada** que se baseia em **JavaScript**
- Criado pela Microsoft
- Documentação

✓ <https://www.typescriptlang.org/pt/>

- Tutorial

✓ <https://terrific-swordtail-92f.notion.site/Typescript4f29dce1268c4daab4a82f7a5733a996?pvs=4>







# Vantagens Typescript



- **Vantagens**





# Transpiladores



## Typescript



- Converte código

**TypeScript** para **JavaScript puro**

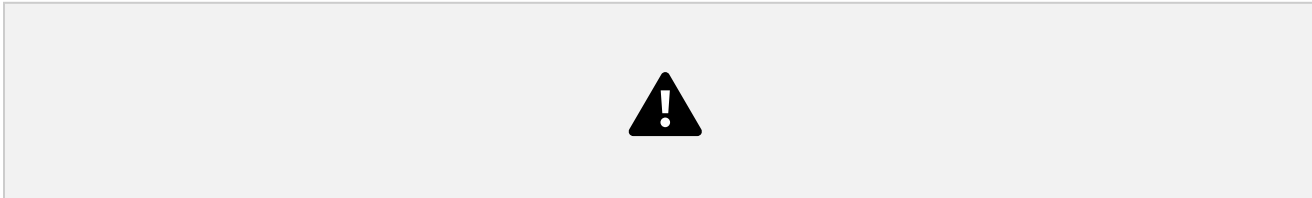
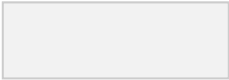
- Tornando-o compatível com **navegadores** e runtimes como **Node.js**



**Transpilando**

# Typescript usando TSC

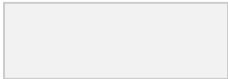
- **TypeScript** para **JavaScript puro** com erros de tipagem

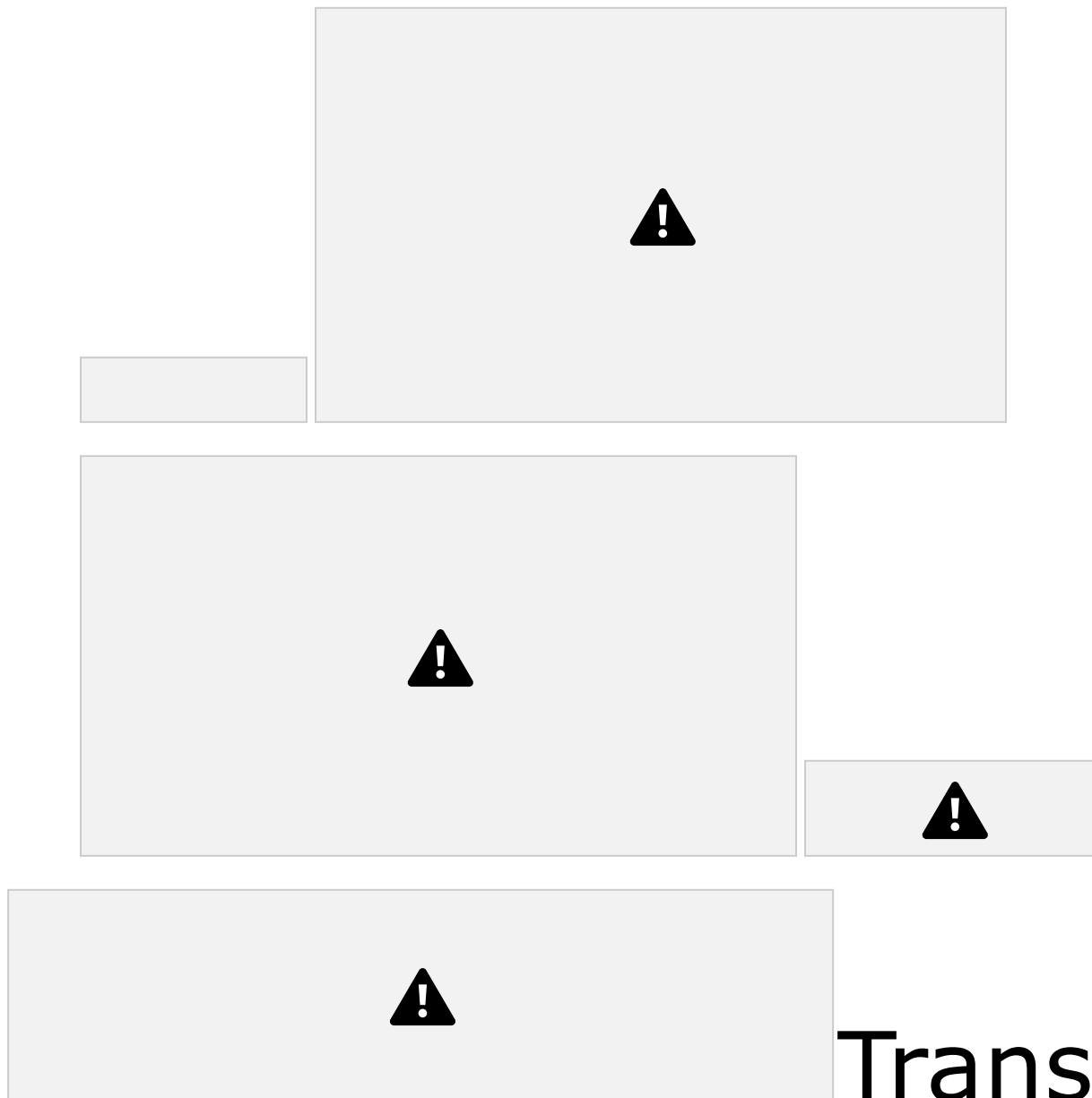




# Transpilando Typescript usando TSC

- **TypeScript** para **JavaScript puro** **sem** erros de tipagem





# Transpiladores

# Javascript



- O **Babel** é conhecido por transpilar **JavaScript moderno** para versões Javascript mais **antigas**
  - Garantindo compatibilidade com navegadores antigos



<https://babeljs.io/>



**Use do**



**Javascript/typescript no  
github**





[https://madnight.github.io/githut/#/pull\\_requests/2020/2](https://madnight.github.io/githut/#/pull_requests/2020/2)



Escrevendo

# código JavaScript



- Código "Espaguete"

```
<html
  xmlns="http://www.w3.org/1999/
x html">
<head>
  <title>Exemplo
JavaScript</title> <script
type="text/javascript">
  var nome = "Joao";
  var sobrenome = "Ricardo";
  nomeCompleto = nome + " " +
    sobrenome;

  document.write("Olá, " +
    nomeCompleto);
</script>
</head>
<body>
</body>
</html>
```

- Incluindo arquivos .js

```
<html
```

```
xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exemplo JavaScript</title>
  <script src="exemplo.js" type="text/javascript" />
</head>
<body>
</body>
</html>
```

No arquivo exemplo.js:

```
var nome = "Joao";
var sobrenome = "Ricardo";
nomeCompleto = nome + " " + sobrenome;

document.write("Olá, " + nomeCompleto);
```





# Escrevendo código JavaScript



■ **O resultado será o mesmo**

## ■ **Dica: Usar um arquivos.js**

- Melhor performance
- Melhor manutenção de código
- Uma das boas práticas para se ter um JavaScript não-obstrusivo



# Escrevendo código JavaScript



## • ONDE COLOCAR O CÓDIGO JAVASCRIPT NO HTML

- No **head** tem dois problemas:

- ✓ diminui a performance e não consegue manipular os elementos da DOM (tags).

- O ideal é colocar na última linha que finaliza **body**

