

# **Introdução ao desenvolvimento WEB**

# O que é HTML?

**HTML**



# HyperText Markup Language

- **Linguagem de estruturação e apresentação.**
- **Versão Atual: HTML 5**
- **Facilmente legível por humanos e compreendida por navegadores.**
- **Base para criar páginas da internet, aplicações híbridas, entre outros.**

# HyperText Markup Language

- O HTML é composto por Elementos, também conhecidos como TAGs.
- Dentro destas TAGs temos atributos e valores.
- Cada tipo de TAG tem seu atributo específico, e cada atributo tem seu valor específico.
- Por exemplo na TAG `<img>` utilizamos `href`, para indicar onde está a imagem, e na tag `<script>` utilizamos `src`, para indicar onde está o script.

# HyperText Markup Language

- **Sempre que abrir uma TAG, você deve fecha-la.**
- **E entre a TAG de abertura e a de fechamento você põe o conteúdo, que vai ser impresso na tela.**

# HyperText Markup Language

- **Exemplos de tags:**

- a – para links.
- img – para imagens.
- button – para botões.
- link – para importar outros documentos.
- p – para parágrafos.
- h\* – para cabeçalhos.
- br – para quebra de linhas.

# HyperText Markup Language

**<p>** Isto é um parágrafo **</p>**  
1 2 3

**1 - Abertura da tag.**

**2 - Texto que será impresso na tela.**

**3 - Fechamento da tag.**

# HyperText Markup Language

**<a href="https://uenp.edu.br">UENP</a>**  
1      2                                  3                                  4                                  5

**1 - Abertura da tag.**

**2 - Atributo href, que aponta para um endereço.**

**3 - Valor do atributo href.**

**4 - O que será impresso na tela.**

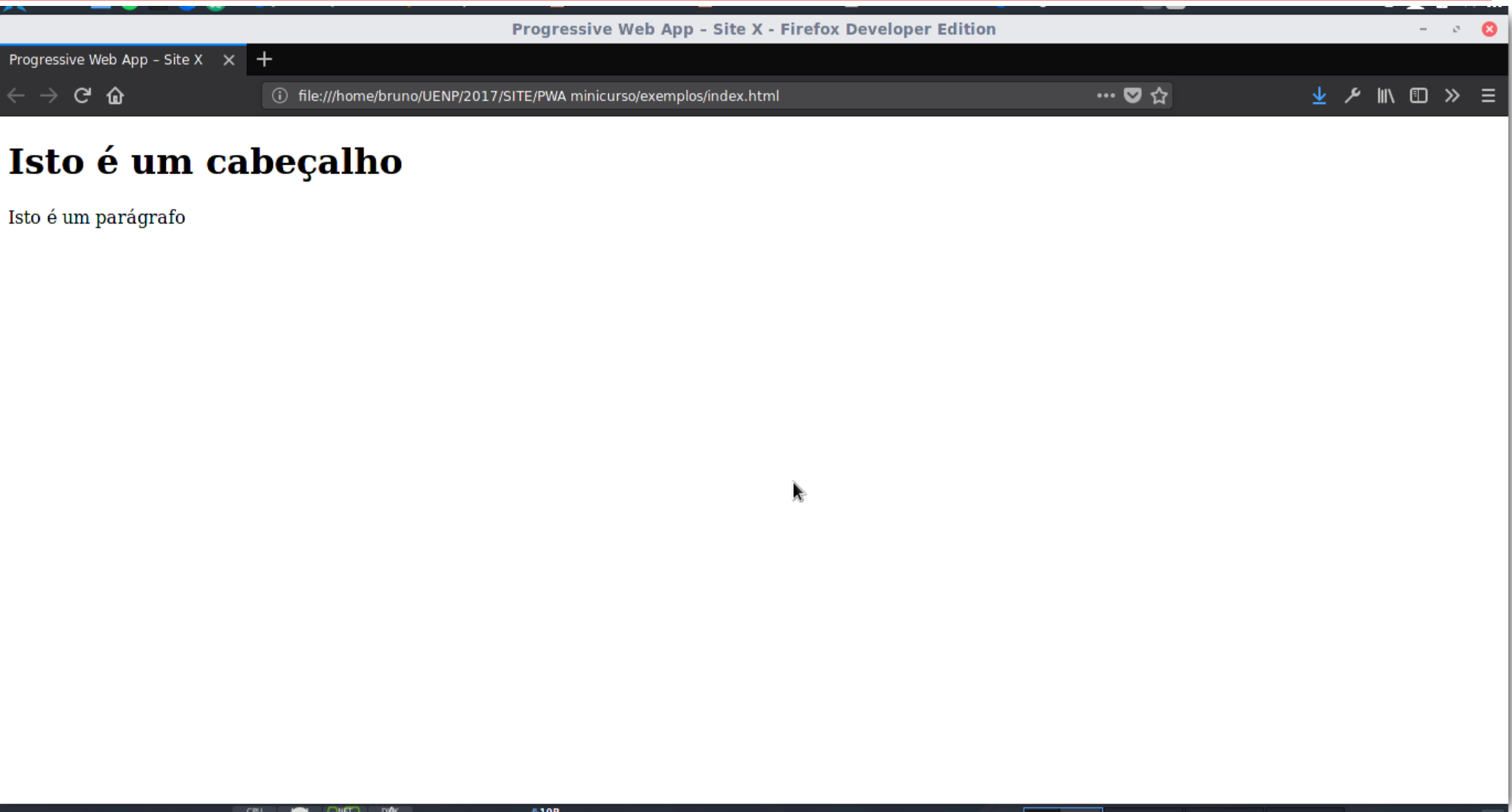
**5 - Fechamento da tag.**



# Exemplo de documento HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Progressive Web App - Site X</title>
</head>
<body>
  <h1>Isto é um cabeçalho</h1>
  <p>Isto é um parágrafo</p>
</body>
</html>
```

# Exemplo de documento HTML



# O que é CSS?



# Cascading Style Sheets

- **Define estilos (formatação) das páginas em HTML.**
- **Versão Atual: CSS 3**
- **Formatação de Fontes, Imagens, Bordas, Cores, etc.**
- **Pode ser utilizado para criar animações.**
- **Permite a formatação da estrutura criada pelo HTML.**

# Três formas de se por CSS no HTML.

- A primeira, e menos recomendada é por direto na tag, por exemplo:  
`<p style="color: blue; text-align: center"> Parágrafo </p>`
- A segunda, e não tão recomendada é por a tag `<style>`, por exemplo:  
`<style>  
p {  
 color: blue;  
 text-align: center;  
}  
</style>  
<p> Parágrafo</p>`
- A terceira e mais recomendada, é criar um documento css, e importá-lo no HTML, com:  
`<link rel="stylesheet" type="text/css" href="./css/style.css">`

# Exemplo de um arquivo CSS

```
body {  
    background-color: cornflowerblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
    text-shadow: 4px 4px 2px black;  
}  
  
p {  
    color: gold;  
    text-align: center;  
}
```

# Exemplo de um arquivo CSS



# O que é JavaScript?



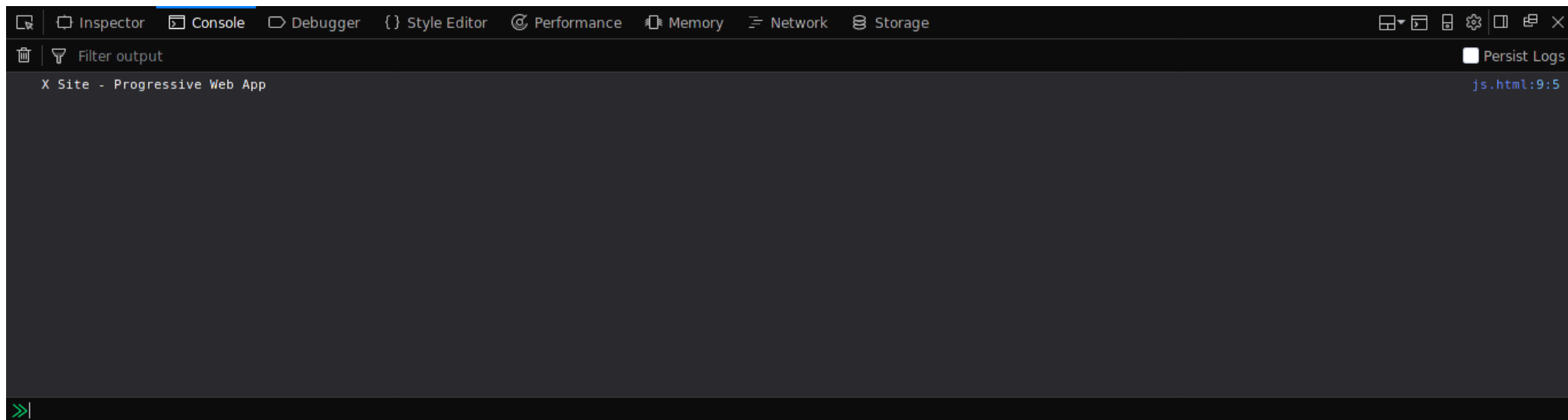


# JavaScript

- **Linguagem de programação da WEB e do HTML.**
- **Versão Atual: JavaScript 5**
- **Language interpretada multi-paradigma.**
- **Permite a criação de scripts nas páginas HTML tornando seu conteúdo dinâmico.**
- **Pode ser utilizado nos navegadores ou diretamente no sistema operacional(backend) como o NodeJS.**

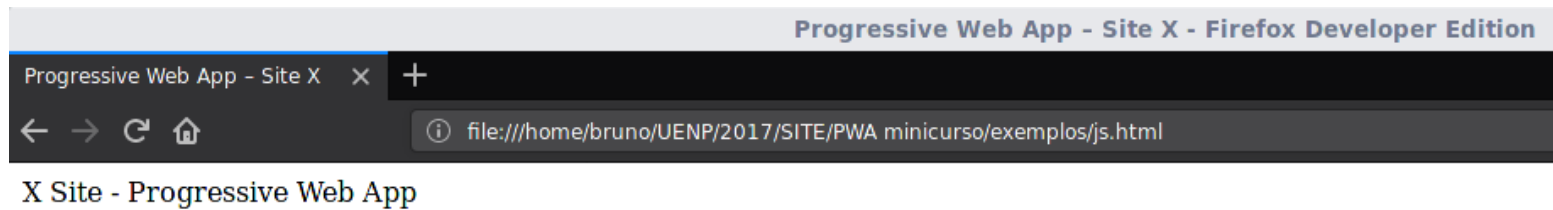
# Exemplo de código JavaScript

```
Console.log("X Site - Progressive Web App");
```

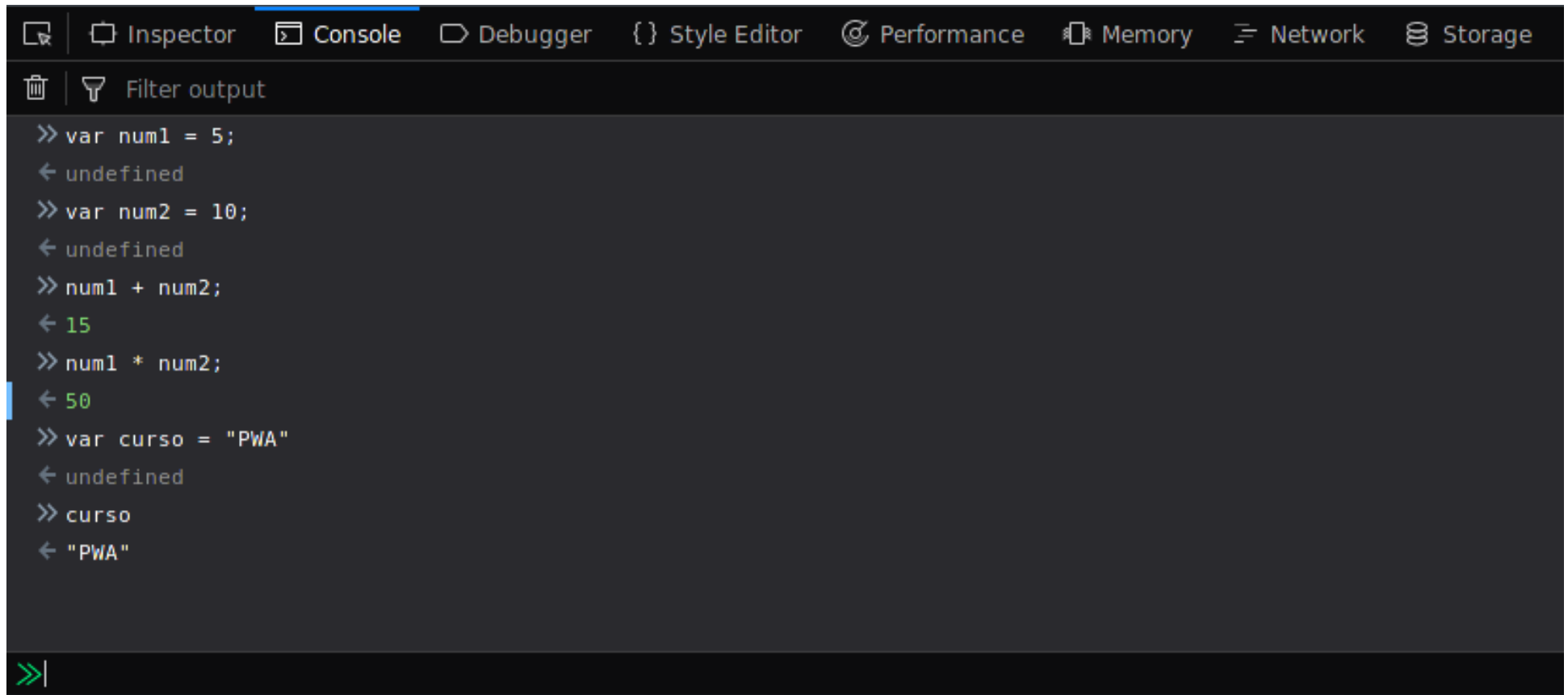


# Exemplo de código JavaScript

**Document.write**("X Site - Progressive Web App");



# Exemplo de código JavaScript



The image shows a screenshot of a web browser's developer console. The console is open, and the 'Console' tab is selected. The toolbar at the top includes icons for Inspector, Console, Debugger, Style Editor, Performance, Memory, Network, and Storage. Below the toolbar, there is a 'Filter output' section. The console displays the following JavaScript code and its output:

```
>> var num1 = 5;  
← undefined  
>> var num2 = 10;  
← undefined  
>> num1 + num2;  
← 15  
>> num1 * num2;  
← 50  
>> var curso = "PWA"  
← undefined  
>> curso  
← "PWA"
```

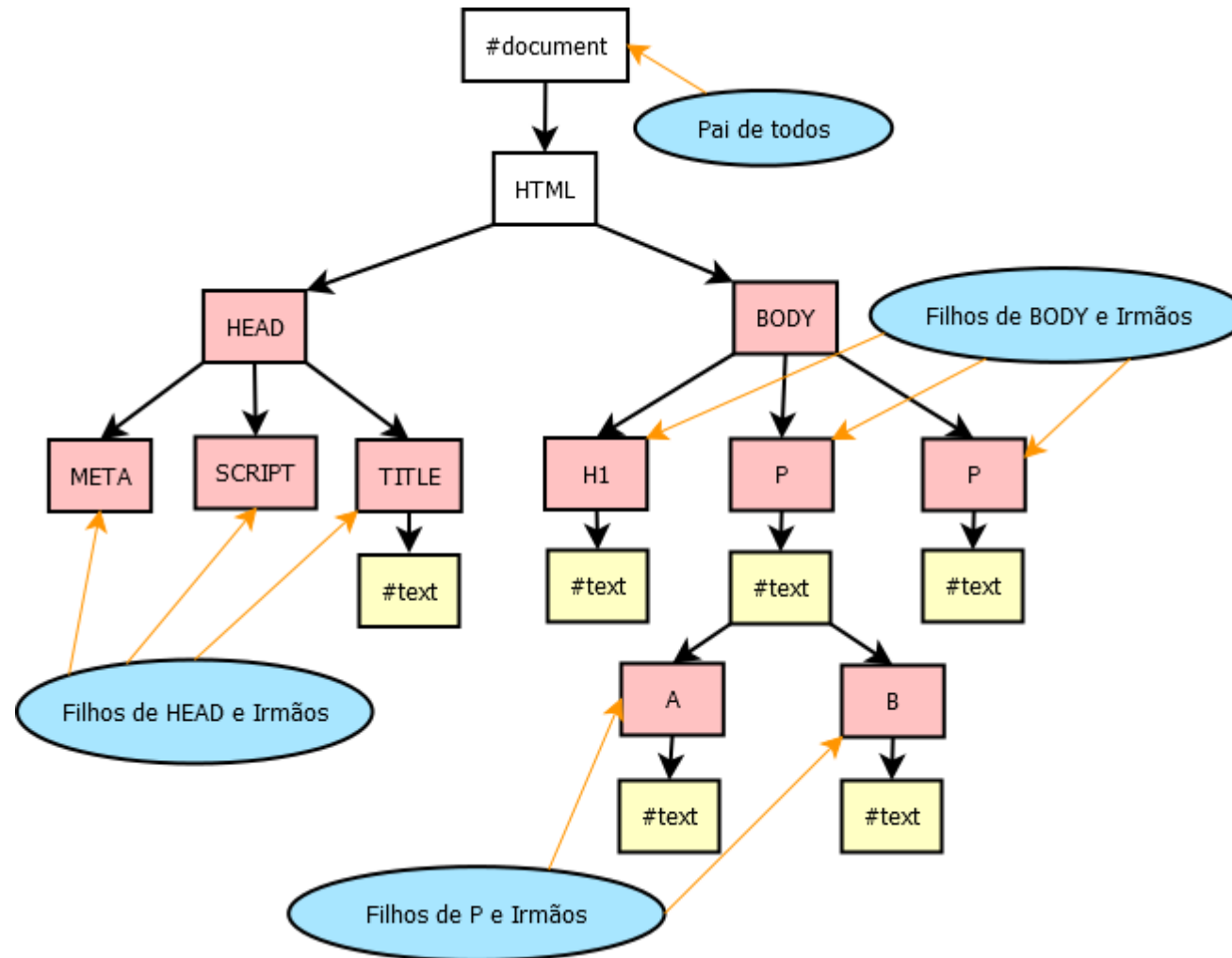
The console shows the execution of several JavaScript statements. The first two statements are variable declarations: `var num1 = 5;` and `var num2 = 10;`. The output for these is `undefined`. The third statement is an addition: `num1 + num2;`, with the output `15`. The fourth statement is a multiplication: `num1 * num2;`, with the output `50`. The fifth statement is another variable declaration: `var curso = "PWA"`, with the output `undefined`. The sixth statement is a reference to the variable: `curso`, with the output `"PWA"`. The console ends with a prompt `>>|`.

# Manipulando DOM

# Document Object Model

- O DOM é uma plataforma e interface neutra da linguagem que permite que programas e scripts acessem e atualizem dinamicamente o conteúdo, estrutura e estilo de um documento.
- Quando uma página é carregada, o navegador cria o **D**ocument **O**bject **M**odel da página.
- Com o Object Model o javascript pode manipular todos os elementos, atributos e valores do HTML, pode também mudar estilos CSS da página, pode criar novas TAGS HTML, e criar novos eventos HTML na página.

# Document Object Model



# Exemplo de DOM

```
function mostraData(){  
    document.getElementById("data").innerHTML = Date();  
}
```

- E no HTML:

```
<button type="button" onclick="mostraData()">Clique aqui</button>
```

```
<p id="data"></p>
```



# Manipulando BOM

# Browser Object Model

- Desde que os navegadores tem implementado(quase) os mesmos métodos e propriedades para interatividade com JavaScript, muitas vezes é referido como métodos e propriedades do BOM.
- Com o BOM e JavaScript, é possível manipular dados do navegador, como, tamanho, localização, cookies, saber quando a página carregou, e muito mais.

**O que é a programação para WEB?**

# Programação para WEB

- **Programação voltada para arquitetura cliente-servidor**
- **O servidor WEB é um software responsável por aceitar requisições HTTP.**
- **O cliente realiza as requisições através de um navegador.**
- **Infinidade de tecnologias disponíveis.**
- **Em geral: HTML, CSS e JavaScript.**

# Tecnologias Envolvidas na WEB(frontend)

