

Programação de Aplicativos



JavaScript



JavaScript é uma linguagem de programação interpretada, de tipagem dinâmica e multiparadigma (suporta programação funcional, orientada a objetos e procedural).



JavaScript



Junto do HTML e o CSS, o JavaScript conclui os três pilares da WEB.

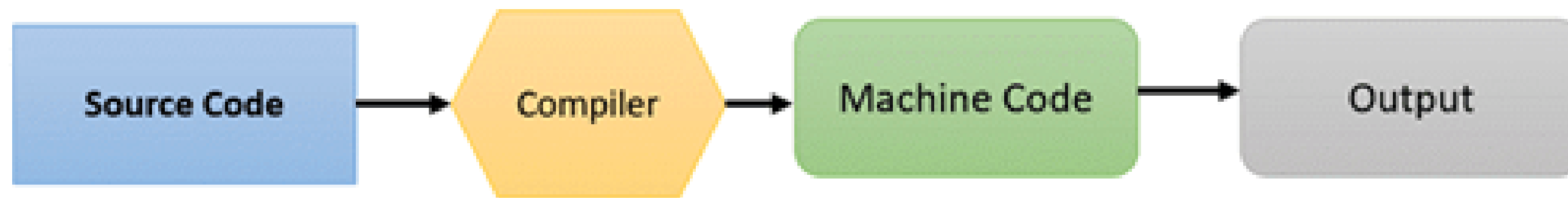


Interpretado x Compilado



Linguagens de programação possuem diferentes formas de serem processadas. As duas formas que temos, são a compilação e a interpretação.

How Compiler Works



© guru99.com

How Interpreter Works



JavaScript



- É executado, em sua maior parte, dentro do navegador (embora hoje também seja utilizada no servidor via Node.js).
- Tem como principal objetivo manipular o comportamento de páginas web, tornando-as dinâmicas e interativas.

Client Side



É tudo que acontece no navegador do usuário (Chrome, Firefox, Edge) e é possível ser visualizado.

The process of client side scripting



Sintaxe



Variáveis:

```
var cidade = "São Paulo"; // escopo de função ou global
let idade = 30;           // escopo de bloco
const PI = 3.14159;       // constante (não pode ser reatribuída)
```

Sintaxe



Tipos primitivos (Não declarados):

```
let nome = "Carla";           // string
let altura = 1.65;            // number
let ativo = true;             // boolean
let indefinido;               // undefined
let vazio = null;             // null
```


Sintaxe



Tipos primitivos (Não declarados):

```
// Array  
let frutas = ["maçã", "banana", "uva"];  
  
// Objeto  
let pessoa = {  
  nome: "João",  
  idade: 28,  
  profissao: "Engenheiro"  
};
```

Sintaxe



Tipos primitivos (Não declarados):

```
// Função tradicional  
function soma(a, b) {  
    return a + b;  
}  
  
// Função anônima atribuída  
const mult = function (a, b) {  
    return a * b;  
};  
  
// Arrow function  
const sub = (a, b) => a - b;
```

Sintaxe



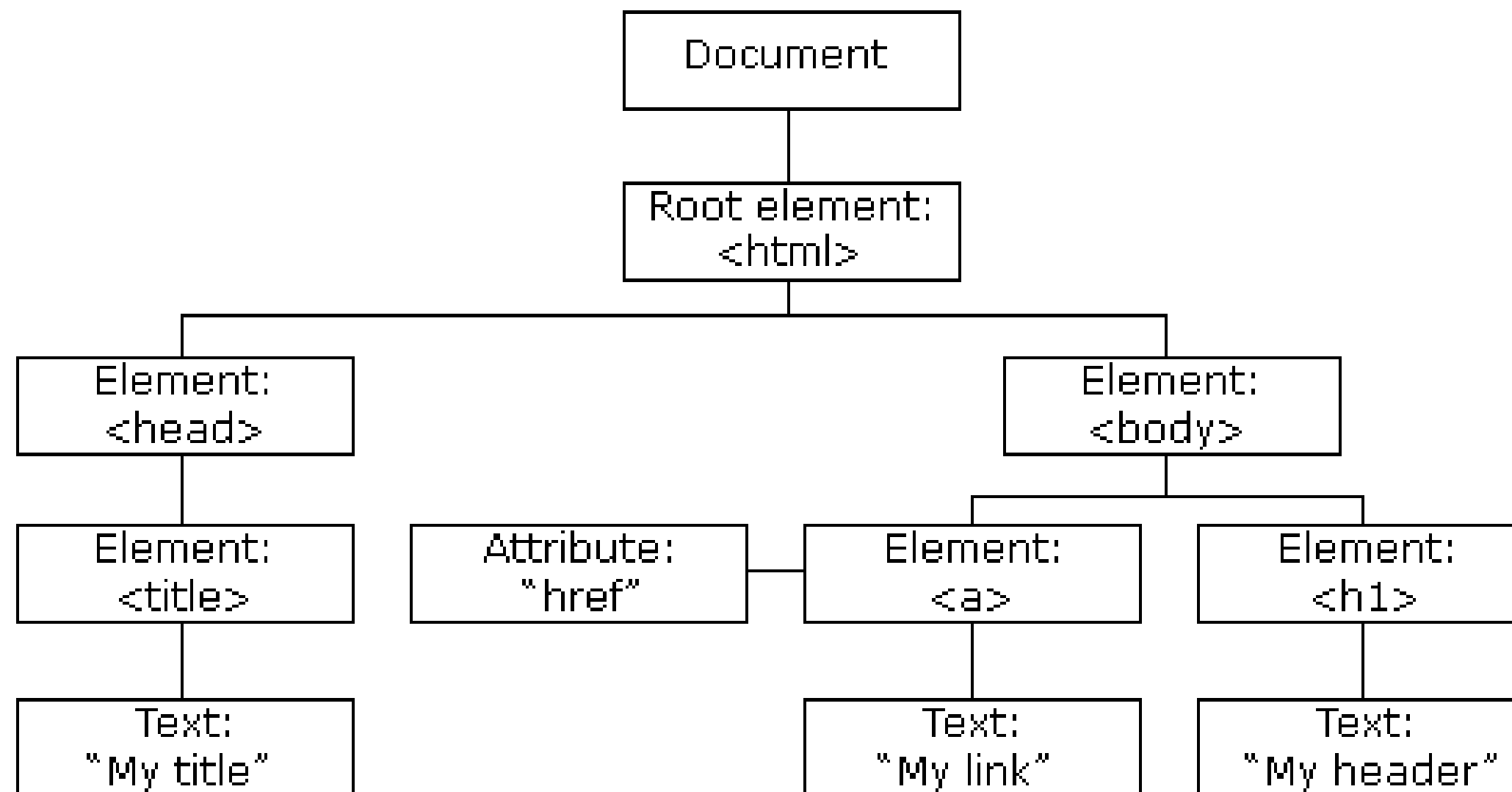
Principais métodos de array → dontpad.com/js/arrayexample



DOM



DOM (Document Object Model) é uma interface de programação que os navegadores usam para representar e interagir com páginas web.



DOM



- O DOM é a representação da página em forma de árvore de objetos.
- O JavaScript é a linguagem que manipula essa árvore.
- Tudo que altera a página em tempo real, responde a eventos ou cria elementos novos é JS interagindo com o DOM.

Incluindo JavaScript



```
<body>
  <script>
    |   console.log("Bom dia!!!")
  </script>
  <!-- Ou... -->
  |   <script src="scripts/bomdia.js"></script>
</body>
```

Selecionando elementos



- `getElementById("id")` → pega um elemento pelo seu id.
- `getElementsByClassName("classe")` → pega todos elementos com determinada classe.
- `querySelector("seletor")` → retorna o primeiro elemento que bate com o seletor CSS.
- `querySelectorAll("seletor")` → retorna todos elementos que batem com o seletor.

Alterando conteúdo e atributos



- `innerText` → muda ou lê o texto visível do elemento.
- `innerHTML` → altera ou lê o conteúdo HTML dentro do elemento.
- `setAttribute("atributo", "valor")` → muda atributos, tipo `src` ou `href`.
- `getAttribute("atributo")` → lê atributos existentes.

GeeksforGeeks

Add Element Type

Add Element Value

1. List Item 1
2. CHILD 2

Gerenciando estilos



- `element.style.propriedade = "valor"` → altera CSS diretamente.
- `classList.add("classe"), .remove("classe"), .toggle("classe")` → controla classes CSS dinamicamente.

GeeksforGeeks

Click on the button to change the style attribute

Click here

Style Attribute Changed

Adicionando e removendo elementos



- `document.createElement("tag")` → cria um novo elemento.
- `appendChild`, `prepend` → adicionam elementos na árvore DOM.
- `remove()` → remove um elemento da página.

GeeksforGeeks

DOM `appendChild()` Method

- Computer Network
- Data Structures using C

Submit



OBRIGADO!