


JS

BEM-VINDOS AO MÓDULO



O que vimos até hoje nessas
maravilhosas **15 semanas** de
sprints, reuniões, diálogo,
vscode e pastel às sextas?



HTML BOOTSTRAP GIT

CSS RESPONSIVIDADE SCRUM

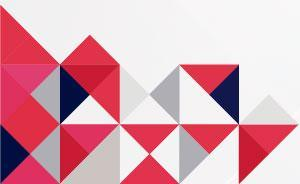
PHP MySQL PDO POO CRUD

ARQUITETURA LARAVEL
MVC



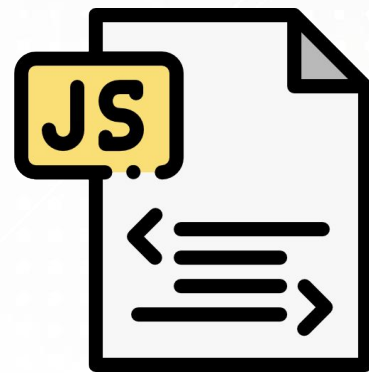


E **por que** cargas d'águas
ele faz parte do plano de
ensino da DH e **precisamos**
aprender JavaScript?





“Pelo sétimo ano consecutivo, o JavaScript é a linguagem de programação mais usada.”



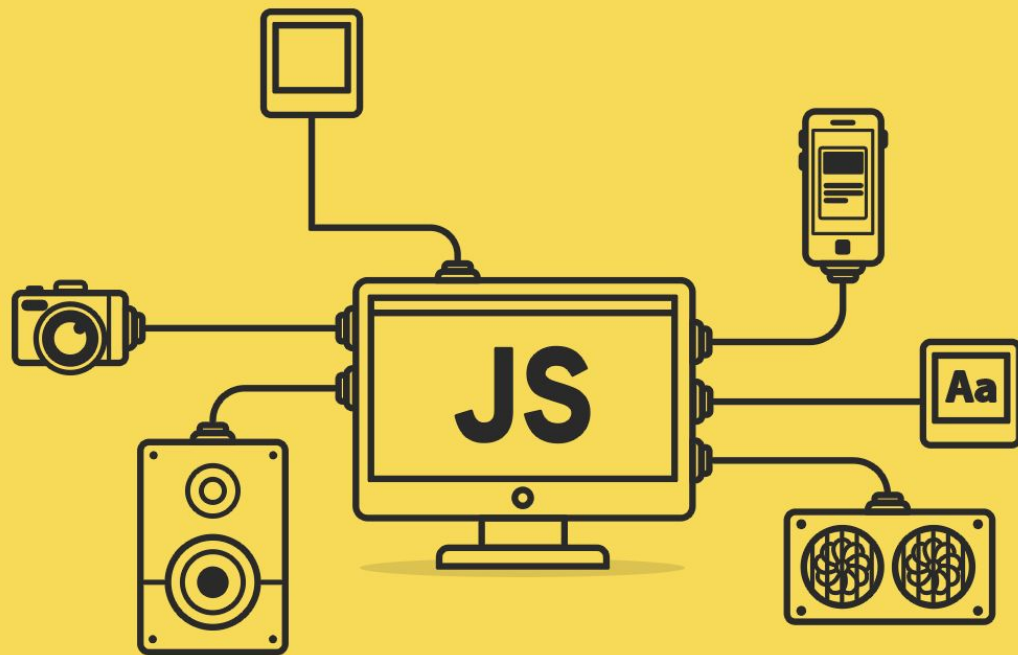
Fonte: Stackoverflow 2019 - Developer Survey Results (most popular technologies)



O que é JavaScript?

- Linguagem de programação
- Orientada a objetos / procedural
- Fracamente Tipada
- ECMAScript
- Imperativa

Para que serve o JS?



Para que serve o JS?



- Como é uma linguagem de programação, o JavaScript pode servir para gerenciar a lógica do back-end.
- Os navegadores trazem um mecanismo que permite executar JS no cliente (**client side**), tornando as páginas interativas.



JS pode servir para desenvolver...



Jogos

[CookieClicker](#)



Galerias

[Photoswipe](#)



Mapas

[Google Maps](#)



Buscadores

[Google](#)



Vídeos

[VideoJS](#)



Redes sociais

[Facebook](#)

E muito mais

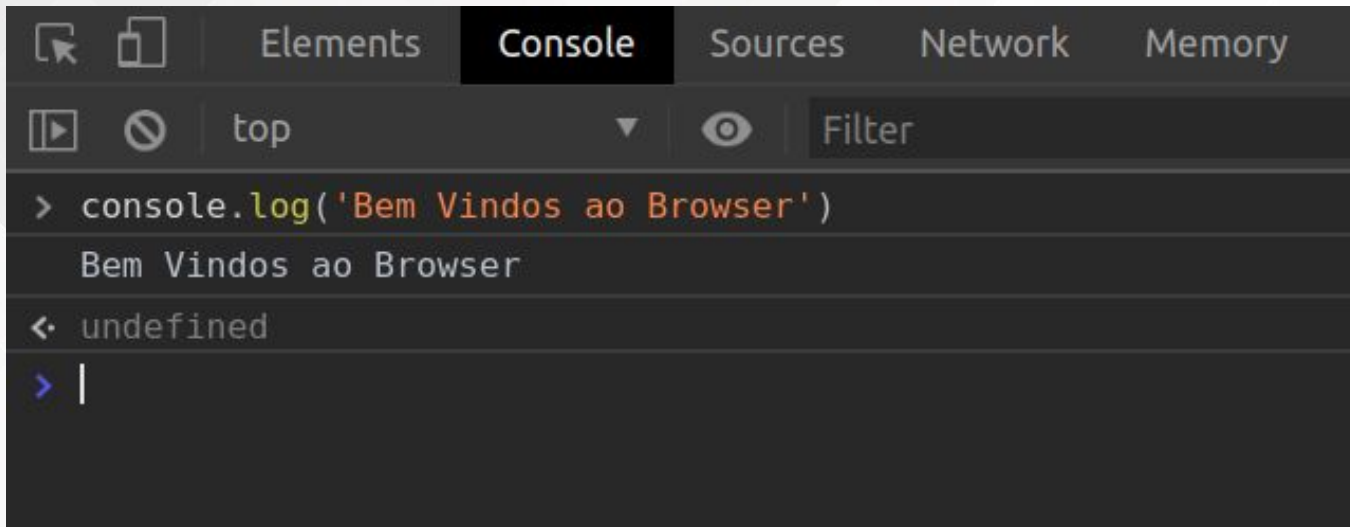
E-mails, bate-papos, gráficos de dados, relógios, calculadoras, formulários, etc...



A linha de comando

Os navegadores trazem uma linha de comando integrada para interpretar JavaScript.

Normalmente usamos F12 para abri-la e começar a escrever o código.



The image shows a screenshot of a web browser's developer console. At the top, there are tabs for 'Elements', 'Console', 'Sources', 'Network', and 'Memory'. The 'Console' tab is selected. Below the tabs, there are icons for a play button, a stop button, and a dropdown menu showing 'top'. To the right of these is a search icon and a 'Filter' input field. The console area contains the following text: a prompt character '>' followed by the command `console.log('Bem Vindos ao Browser')`, the output `Bem Vindos ao Browser`, a prompt character '<' followed by the output `undefined`, and a new prompt character '>' followed by a vertical bar cursor.

```
> console.log('Bem Vindos ao Browser')
Bem Vindos ao Browser
< undefined
> |
```

Variáveis

var nomeDaVariavel = valor

let nomeDaVariavel = valor

nomeDaVariavel = valor

Tipos básicos

"pepe"	string
10.7	number
true	boolean
{ }	object
[]	array
function() { }	function
null	null
undefined	undefined

Operadores aritméticos

<code>var num1 = 5</code>	
<code>var num2 = 3</code>	
<code>num1 + num2</code>	adição, subtração
<code>num1 * num2</code>	multiplicação, divisão
<code>num1 % 5</code>	módulo
<code>num1++</code>	incremento
<code>num1--</code>	decremento

Operadores de atribuição

<code>var numero = 10</code>	
<code>numero += 2</code>	<code>numero = numero + 2</code>
<code>numero -= 2</code>	<code>numero = numero - 2</code>
<code>numero *= 2</code>	<code>numero = numero * 2</code>
<code>numero /= 2</code>	<code>numero = numero / 2</code>

Operadores de comparação simples

==	igualdade de valor
!=	valor diferente
<=	menor e igual
>=	maior ou igual
==	igualdade de valor

Operadores de comparação estrita

===	igualdade de valor e tipo
!==	valor e tipo diferente

Operadores Lógicos

&&	Operador and (e)
	Operador or (ou)
!	Operador de negação

IF - Sintaxe

```
if (condição) {  
    // fazer isto caso a condição seja  
    verdadeira  
} else {  
    // fazer isto caso a condição seja  
    falsa  
}
```

IF - Exemplo

```
var diaDaSemana = 'Domingo'

if (diaDaSemana == 'Domingo') {
  console.log('Hoje é dia de churrasco!')
} else {
  console.log('Hoje não é dia de churrasco :(')
}
```

Resultado: Hoje é dia de churrasco!

IF ternário - Sintaxe

`condição ? expressão1 : expressão2`

`condição`

Qualquer expressão **booleana**

`expressão1`

Expressão retornada se `condição` for **true**

`expressão2`

Expressão retornada se `condição` for **false**

IF ternário - Exemplo

```
var diaDaSemana = 'Segunda-feira'
```

```
diaDaSemana == 'Domingo'
```

```
? console.log('hoje é dia de churrasco!')
```

```
: console.log('hoje não é dia de churrasco :(')
```

Resultado: Hoje não é dia de churrasco :(

Switch

```
var fruta = "mamão";

switch (fruta) {
  case "abacaxi":
    console.log("Me ajuda como descascar  
esse abacaxi?");
    break;
  case "mamão":
    console.log("É mamão com açúcar!");
    break;
  default:
    console.log("É outra fruta");
}
```

For - Sintaxe

```
for (inicio; condicao; incremento) {  
    // fazer isso enquanto a condição for  
    verdadeira  
}
```

For - Exemplo

```
for (let i = 0; i < 3; i++) {  
  console.log(`Olá ' + i)  
}
```

Resultado: Olá 0 Olá 1 Olá 2

For - Break

```
for (let i = 0; i < 4; i++) {  
  console.log('Olá ' + i);  
  if(i === 1){  
    break; // interrompe o loop FOR  
  }  
}
```

Resultado: Olá 0 Olá 1

While - Sintaxe

```
while (condição) {  
    // executar enquanto a condição  
    for verdadeira  
}
```

While - Exemplo

```
let numero = 0
while (numero < 3) {
  console.log('Olá ' + numero)
  numero++
}
```

Resultado: Olá 0 Olá 1 Olá 2

Do While - Sintaxe

```
do {  
    // código a ser executado  
} while (condição);
```

O do while irá executar o código que está entre o **do {}** antes de verificar a condição.

Próxima aula - JS + HTML

