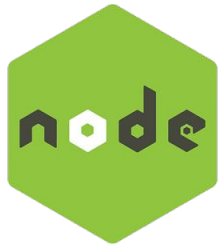
NodeJS



Prof. Celso Henrique Masotti

Necessidades Prévias

- Ter o node.JS instalado;
- Ter o gerenciador de pacotes (NPM) instalado;
- Ter um editor de códigos IDE(*), dê preferência ao “Visual Studio Code”
- Conhecer JavaScript básico.

(*) IDE, do inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo, como um editor de texto para a digitação de códigos.

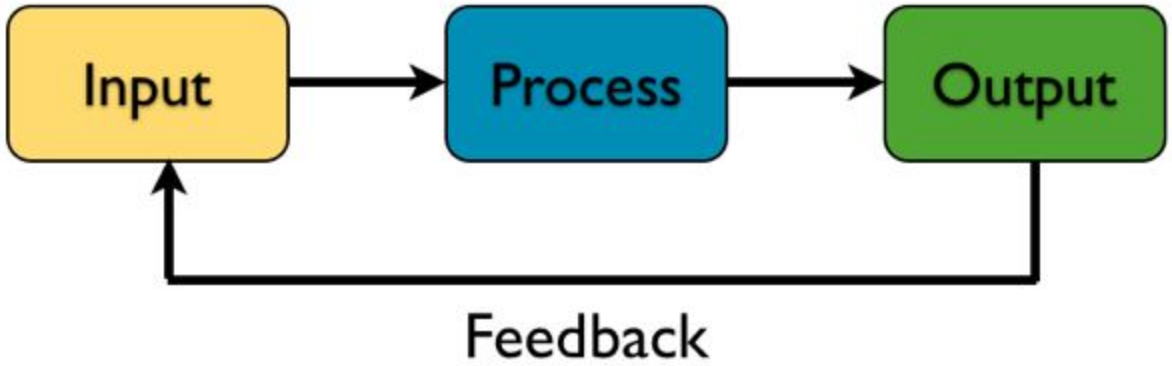
Você verá nesta aula:

Objetivo: conhecer os recursos output do javascript e recursos ampliados do objeto “console” e introdução ao DOM.

Output
Tipos de Saídas
innerHTML
document.write(); [1]
document.write(); [2]
window.alert();
console.log();
Objeto “console”
DOM

Output

Todo programa computacional possui uma entrada (input) de dados, processamento e uma saída de dados (output), podendo ou não retroalimentar o processo.



Tipos de Saídas

No javascript podemos oferecer a saída de dados (exibição) de diferentes formas.

Para escrever em um elemento HTML usamos o “**innerHTML**”.

Para escrever no documento HTML usamos o “**document.write()**”.

Para um output em objetos, como na caixa de alert, usamos o “**window.alert()**”.

Para escrever em um console (do navegador ou não), usamos “**console.log()**”.

innerHTML

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

document.write();

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
     document.write(5 + 6);
</script>

</body>
</html>
```

document.write();

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>
</html>
```

window.alert();

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
     window.alert(5 + 6);
</script>

</body>
</html>
```

console.log();

```
<!DOCTYPE html>
<html>
<body>

<script>
     console.log(5 + 6);
</script>

</body>
</html>
```

Objeto “console”

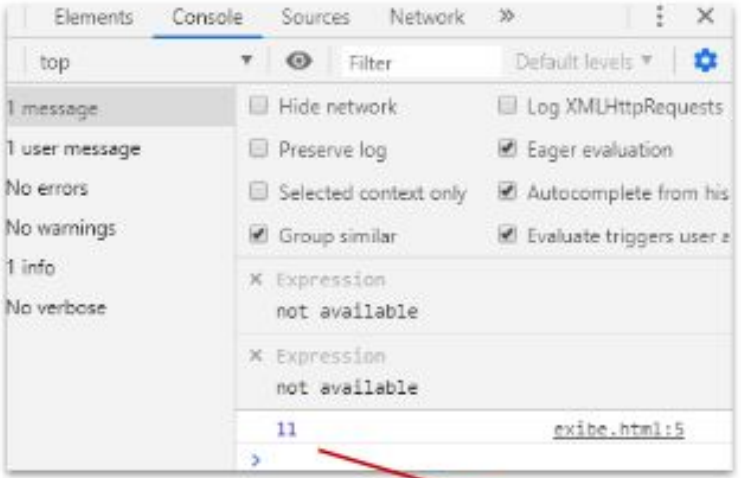
O *objeto* **console** fornece acesso ao console de debug do navegador ou do aplicativo à parte (como o prompt).

```
<!DOCTYPE html>
<html>
<body>

<script>
    console.log(5 + 6);
</script>

</body>
</html>
```

Resultado no console do Windows (prompt)



Resultado no console do navegador

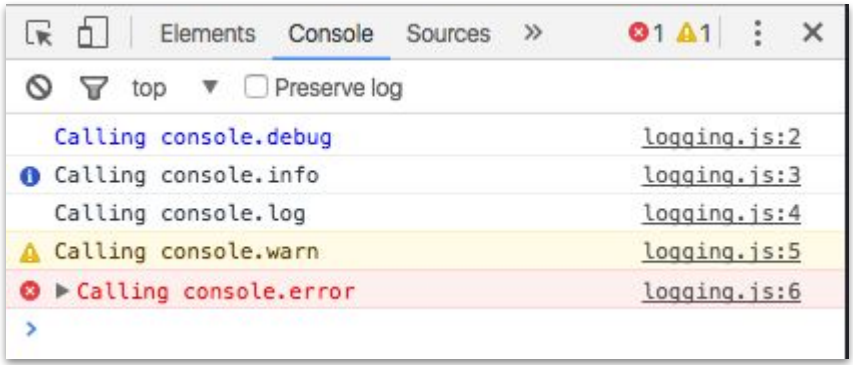
Objeto “console”

Este objeto documenta vários métodos disponíveis, como: console.*assert*(); console.*count*(), console.*debug*(), console.*dir*(), console.*error*(), console.*profileEnd*(), Console.*table*(), etc. (veja a lista completa)

Mas, no que tange aos outputs mais utilizados, podemos indicar quatro métodos:

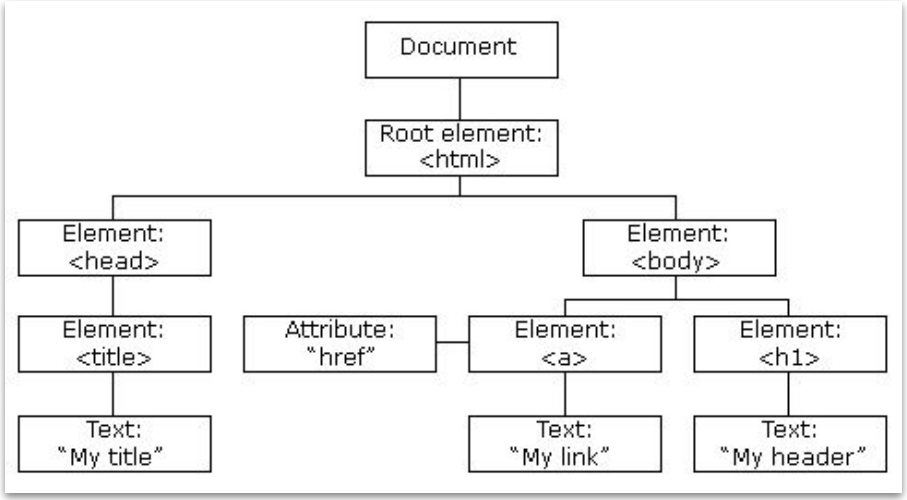
console.**log**() - imprime o texto ou o conteúdo de variáveis;
console.**error**() - emite uma mensagem de erro;
console.**warn**() - emite uma mensagem de alerta;
console.**info**() - imprime informações do registro;

No console do navegador é comum ícones a companhar a informação do output.

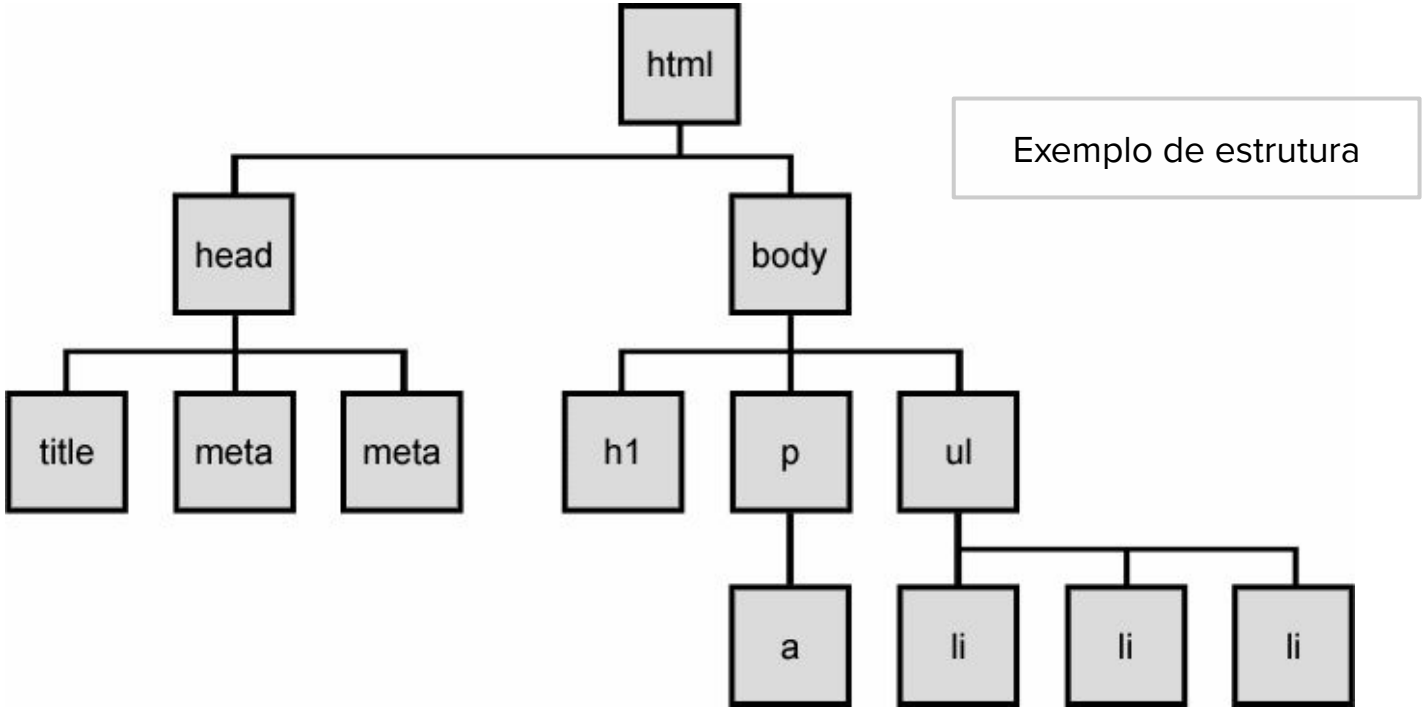


DOM

DOM vem do inglês “Document Object Model” e significa a árvore de objetos hierarquizados que compõe um arquivo HTML. Afinal, <body> é filha de <html>, já que obrigatoriamente <body> só pode existir no interior de <html>. Da mesma forma <p> só pode existir no interior de <body>, <title> no interior de <head> e todos eles no elemento “document”.
Estas palavras chaves denominadas “tags” e que formam um documento HTML podem ser acessadas e manipuladas pelo javascript associada ao DOM.



DOM



Exemplo de estrutura

DOM

Vimos um pouco atrás o comando “document.write()”. Se observar do ponto de vista do DOM perceberemos que estamos pedindo para que alguma coisa seja escrita no “document”.

Também podemos observar o DOM na linha de comando “alert(document.cookie);”, onde pedimos para que uma janela tipo “alert” exiba informações do “cookie” existente no “document”.

No exemplo “document.getElementById("demo").innerHTML = 5 + 6;” pedimos para que o resultado da soma (5 + 6) seja escrita em um objeto cuja identificação “id” tem o nome de “demo”.

Sabendo usar o DOM podemos dominar facilmente eventos de um documento HTML.

Obrigado

Profº Celso Henrique Masotti