Javascript

Variável, tipo de dado e conversão

```
// Variável
var tw = "TreinaWeb";

// Redeclarando a variável
var tw = 20;
```

Javascript

Aritmética

```
var t = 10;
var w = 6;
var tw = 0;

// Soma
tw = t + w;

// Subtração
tw = t - w;

// Multiplicação
tw = t * w;
// Divisão
tw = t / w;

// Módulo
tw = t % w;
```

Javascript

Aritmética - Problemas com arredondamento

```
// Definição das variáveis
var t = 0.7 - 0.6; // Retorna: 0.09999999999999998
var w = 0.2 - 0.1; // Retorna: 0.1

// Comparações de igualdade:
console.log("Primeiro teste: "+(t == w)); // False
console.log("Segundo teste: "+(t == 0.1)); // False
console.log("Terceiro teste: "+(w == 0.1)); // True
console.log("Quarto teste: "+(t == 0.09999999999999998)); // True
```

Javascript

Objetos

```
// Criando um objeto
var objeto = {
    tw : 'TreinaWeb Cursos'
};
// Exibindo o valor da propriedade 'tw' do objeto 'objeto'
// utilizando ponto (.)
document.write("O retorno de 'objeto.tw' é: "+objeto.tw);
document.write("<br />");

// Exibindo o valor da propriedade 'tw' do objeto 'objeto'
// utilizando colchetes ([])
document.write("O retorno de 'objeto[\"tw\"]' é: "+objeto["tw"]);
```

Javascript

Criando objeto (exemplo 2)

```
// Criando um objeto
var objeto = {
    tw : "TreinaWeb"
};

document.write("O valor de 'tw' é: "+objeto.tw+"<br />"); // O valor da propriedade 'tw'

objeto.tw = "TreinaWeb Cursos"; // Configurando a propriedade 'tw'

document.write("O valor de 'tw' agora é: "+objeto.tw+"<br />"); // O valor de 'tw' após a configuração

// Primeiro utilizando um identificador. Para isso utilize o ponto (.) // Criando duas novas propriedades
objeto.x = 6;

// Agora utilizando uma string literal. Para isso utilize os colchetes ([])
objeto["y"] = 4;
// Exibindo o valor das novas propriedades
document.write("O valor da nova propriedade 'x' é: "+objeto.x+"<br />");
document.write("O valor da nova propriedade 'y' é: "+objeto.y);
```

Javascript

O operador DELETE remove uma propriedade do objeto:

```
// Criando um objeto
var obj = {
    prop_1 : 1,
    prop_2 : 2,
    prop_3 : 3
};
// Exibindo os valores do objeto 'obj'
document.write("O valor de 'prop_1' do objeto 'obj' é: "+obj.prop_1+"<br />");
document.write("O valor de 'prop_2' do objeto 'obj' é: "+obj.prop_2+"<br />");
document.write("O valor de 'prop_3' do objeto 'obj' é: "+obj.prop_3+"<br />");
document.write("<br />");

// Agora será excluída a propriedade 'prop_2' do objeto 'obj'
delete obj.prop_2;
// Os valores após deletar 'porp_2' são:
document.write("O valor de 'prop_1' do objeto 'obj' agora é: "+obj.prop_1+"<br />");
document.write("O valor de 'prop_2' do objeto 'obj' agora é: "+obj.prop_2+"<br />");
document.write("O valor de 'prop_3' do objeto 'obj' agora é: "+obj.prop_3+"<br />");
```

Javascript

O operador DELETE remove uma propriedade do objeto:

```
// Criando um objeto
var obj = {
    prop_1 : 1,
    prop_2 : 2,
    prop_3 : 3
};
// Exibindo os valores do objeto 'obj'
document.write("O valor de 'prop_1' do objeto 'obj' é: "+obj.prop_1+"<br />");
document.write("O valor de 'prop_2' do objeto 'obj' é: "+obj.prop_2+"<br />");
document.write("O valor de 'prop_3' do objeto 'obj' é: "+obj.prop_3+"<br />");
document.write("<br />");

// Agora será excluída a propriedade 'prop_2' do objeto 'obj'
delete obj.prop_2;
// Os valores após deletar 'porp_2' são:
document.write("O valor de 'prop_1' do objeto 'obj' agora é: "+obj.prop_1+"<br />");
document.write("O valor de 'prop_2' do objeto 'obj' agora é: "+obj.prop_2+"<br />");
document.write("O valor de 'prop_3' do objeto 'obj' agora é: "+obj.prop_3+"<br />");
```

Javascript

DOM -

Document Object Model – DOM, é uma API utilizada para representar e manipular o conteúdo dos documentos.

No DOM, os elementos aninhados de um documento HTML são representados como uma árvore de objetos.

A representação em árvore de um documento HTML contém nós representando marcações, ou elementos HTML, como <body> e <p>, e nós representando string de texto.

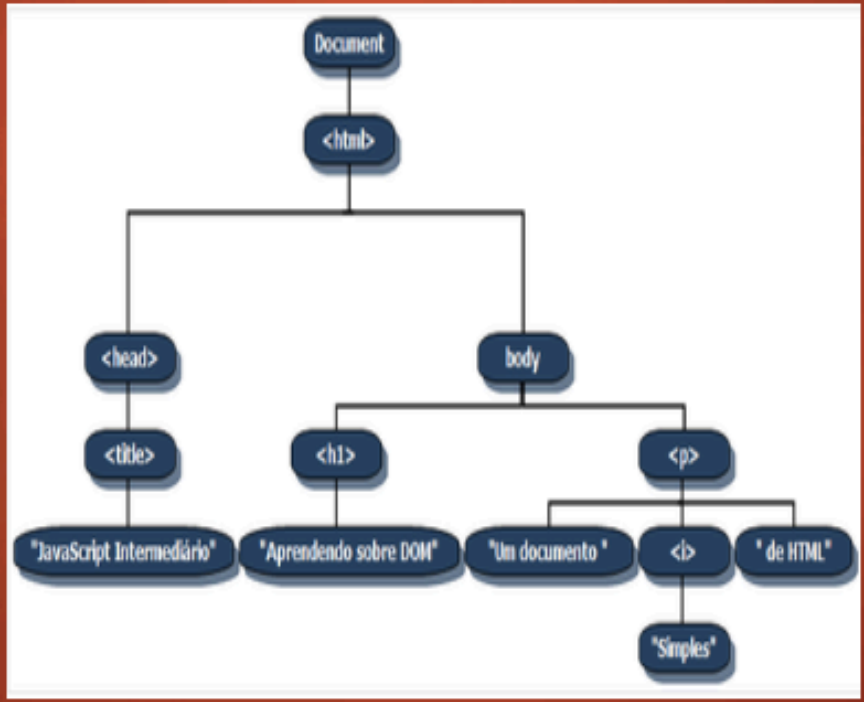
Javascript

DOM -

```
<html>
   <head>
      <title>JavaScript Intermediário</title>
   </head>
   <body>
      <h1>Aprendendo sobre DOM</h1>
      <p>Um documento <i>simples<i> de HTML</p>
   </body>
</html>
```

Javascript

A representação DOM do documento anterior é a árvore a seguir:



Javascript

Expressões Regulares

Uma expressão regular é um objeto que descreve um padrão de caracteres. A classe RegExp de JavaScript representa as expressões regulares, e tanto String quanto RegExp definem métodos que utilizam expressões regulares para executar funções poderosas de comparação de padrões e de localização e substituição de texto.

Javascript

Os objetos RegExp podem ser criados com a construtora RegExp().

```
var expRegular = new RegExp("s$");
```

Mas são criados mais frequentemente com o uso de uma sintaxe literal:

```
var expRegular = "/s$/";
```

Javascript

Método exec() de RegExp, que executa uma expressão regular numa string especificada.

```
// Criando uma variável com um texto para análise
var texto = 'TreinaWeb Cursos - JavaScript Avançado!'

// Criando uma expressão regular que retornará um array
var expRegularTrue = new RegExp("TreinaWeb");

// Criando uma expressão regular que retornará null
var expRegularFalse = new RegExp("treinaweb");

// Executando test()
// Retorno é um array
console.log(expRegularTrue.exec(texto));

// Retorno igual a null
console.log(expRegularFalse.exec(texto));
```