

INTRODUÇÃO AO

JavaScript Descomplicado

L. GALLERANI



Copyright © 2020 by L. Gallerani

Diagramação

L. Gallerani

Capa

L. Gallerani

Revisão

L. Gallerani

G166j Gallerani L.

JavaScript descomplicado / Gallerani L. 2. ed - São Paulo, SP

Esta obra é uma produção independente.

Copyright[2020] by Gallerani L.

Todos os direitos desta edição reservados ao autor da obra.

1. Ciência da Computação
 2. Tecnologia
-

Índice para catálogo sistemático:

1. Ciência da Computação CDD 004
2. Tecnologia (ciências aplicadas) CDD 600

Alerta	9
Entrada e Saída de dados.....	13
Calculadora simples	17
Comparadores Lógicos	19
Comandos condicionais - If e Else	24
Comandos condicionais – CASE.....	27
Estruturas de Repetição - For	29
Estruturas de Repetição – While e Do.....	32
Utilizando funções.....	34

Introdução

JS ou JavaScript é uma linguagem de programação de script com tipagem dinâmica fraca, ou seja, não tem tipos de variáveis fixos, como o JAVA ou Delphi.

Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da Internet.

Onde traz dinamismo a páginas de internet.

Ambiente de trabalho

Para desenvolver páginas em HTML não é necessária nenhuma ferramenta específica, apenas o bloco de notas, mas é aconselhável utilizar editores de texto voltado para a programação, para facilitar o entendimento e releitura do código.

Pode ser utilizado o como já mencionado o bloco de notas, sublime text, visual studio code, php storm, notepad++, entre outros.

Nesta obra foi utilizado o sublime text para a edição e criação do HTML, caso deseje utilizar o mesmo pode encontrar para download em:

<https://www.sublimetext.com>

Alerta

Veremos um comando simples de alerta, abra o seu bloco de notas ou sublime e digite o comando abaixo:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title></title>
5  </head>
6  <body>
7  |
8  |   <script>
9  |       alert('Hello World from JavaScript');
10 |   </script>
11 </body>
12 </html>
```

Fonte: O autor

Depois de salvar sua página como aula1.html, abra no seu navegador dando um clique duplo e verá a mensagem criada.

Essa página diz

Hello World from JavaScript

OK

Fonte: O autor

Utilizaremos agora variáveis, estas guardaram algumas informações, como nomes, idade, altura, estado civil, ao final exibiremos os dados e o tipo de cada variável.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          var nome = 'Nome do Aluno';
10         var sobrenome='Sobrenome do Aluno';
11         var idade =21;
12         var altura = 1.78;
13         var casado = false;
14
15         alert(nome);
16         alert(sobrenome);
17         alert(idade);
18         alert(altura);
19         alert(casado);
20
21         alert('Tipo da variavel nome: '+typeof(sobrenome));
22         alert('Tipo da variavel sobrenome: '+typeof(sobrenome));
23         alert('Tipo da variavel idade: '+typeof(idade));
24         alert('Tipo da variavel altura: '+typeof(altura));
25         alert('Tipo da variavel casado: '+typeof(casado));
26
27     </script>
28 </body>
29 </html>
30
```

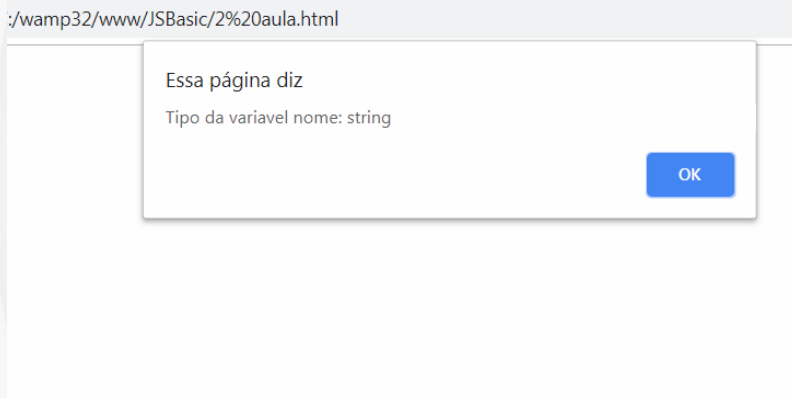
Fonte: O autor

Nas linhas 9 até 13 serão criadas estas variáveis e já informado os dados que serão utilizados.

Nas linhas 15 até 19 será usado o `alert` para exibir estes dados salvos.

E nas linhas 21 até 25 o `alert` é implementado com um texto, explicando do que se trata cada informação, deve-se usar aspas para identificar que se trata de um texto e não uma variável dentro do `alert`.

O resultado:



Fonte: O autor

Utilizaremos a função `toLowerCase()` e `toUpperCase`, para deixar textos em minúsculo e maiúsculo, a função `replace` para substituir letras ou números e a função `toString` para transformar variáveis do tipo número para texto.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          var nome = 'Nome do Aluno';
10         var sobrenome = 'Sobrenome do Aluno';
11
12         nome = nome.toLowerCase();
13         sobrenome=sobrenome.toUpperCase();
14
15         alert(nome);
16         alert(sobrenome);
17
18         var animal = 'Galo';
19         animal=animal.replace('l', 't');
20         alert(animal);
21
22         var numero = 10;
23         alert(numero.toString());
24     </script>
25 </body>
26 </html>
27
```

Fonte: O autor

Nas linhas 9 e 10 são criadas duas variáveis e atribuído seus valores.

Na linha 12 a variável nome recebe o nome com a função LowerCase para deixar o nome em minúsculo.

Na linha 13 a variável sobrenome recebe o sobrenome com a função UpperCase para deixar o sobrenome em maiúsculo.

Estas funções são muito uteis para pesquisas e comparações, para evitar erros ao comparar uma palavra com letras maiúscula de uma com letras maiúsculas.

Nas linhas 15 e 16 são exibidas estas variáveis com o alert.

Na linha 18 é criada uma variável chamada animal que recebe a palavra 'Galo'.

Na linha 19 é alterado a letra 'l' por 't' com a função replace, desta forma se pode ver que é possível alterar letras ou caracteres direto na programação, muito utilizado para geração de boletos bancários ou remover caracteres indesejados.

O resultado:



Fonte: O autor

Entrada e Saída de dados

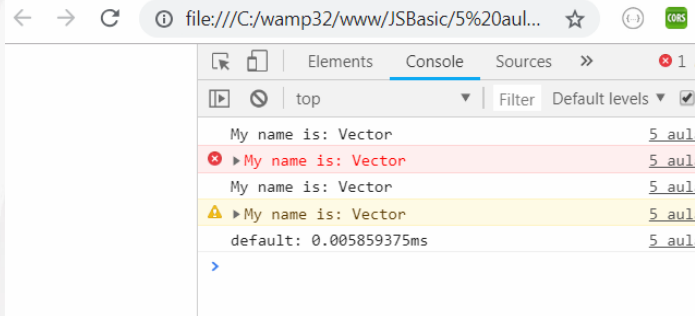
Aqui podemos ver uma entrada de dados usando o prompt e saídas de dados usando o confirm.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7      <script>
8          var name = prompt('Whats is your name?');
9          //alert('My name is: '+name); // ok
10         confirm('My name is: '+name); //ok e cancel
11         console.log('My name is: '+name); //F12 ou inspect
12         console.error('My name is: '+name);
13         console.info('My name is: '+name);
14         console.warn('My name is: '+name);
15         console.time();
16         console.timeEnd();
17     </script>
18 </body>
19 </html>
```

Fonte: O autor

Os comandos de console só poderão ser vistos ao teclar f12 e clicar no console no Google Chrome.

O resultado:



Fonte: O autor

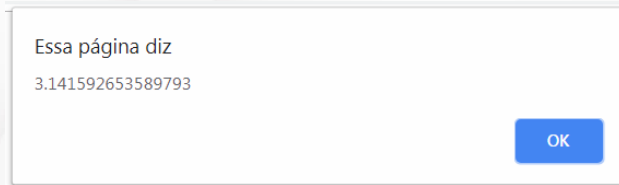
Utilizamos as funções:

- toString para converter número em texto.
- parseInt para converter um número real para inteiro.
- parseFloat para transformar um número inteiro para real.
- Math.random() para criar um número aleatório.
- Math.round para arredondar até 6 casas decimais.
- Math.trunc para exibir as 6 primeiras casas decimais, sem arredondamento.
- Math.PI para ver o valor do PI.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          var numero = 10.20;
10         var numero2 = 5;
11         numero.toString();
12         numero = parseInt(numero);
13         numero2 = parseFloat(numero2).toFixed(2);
14         alert(numero);
15         alert(numero2);
16
17         var aleatorio= Math.random();
18         alert(aleatorio);
19
20         /* Arredondado que não passa de 6*/
21         var aleatorioArredondado= Math.round(Math.random()*6);
22         alert(aleatorioArredondado);
23
24         /* truncar até 6*/
25         var aleatorioTruncado= Math.trunc(Math.random()*6);
26         alert(aleatorioTruncado);
27
28         var ValorDePi = Math.PI;
29         alert(ValorDePi);
30
31     </script>
32 </body>
33 </html>
```

Fonte: O autor

O resultado:



Fonte: O autor

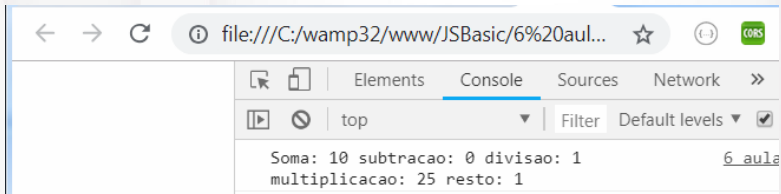
Calculadora simples

Aqui podemos ver uma calculadora simples.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7   <script>
8     /* Arithmetics */
9     var soma = 5 + 5;
10    var subtracao = 5 - 5;
11    var divisao = 5 / 5;
12    var multiplicacao = 5 * 5;
13    var resto = 3 % 2;
14
15    console.log("Soma: "+soma, "subtracao: "+subtracao, "divisao: "+divisao, "multiplicacao: "+
16      multiplicacao,"resto: "+resto);
17  </script>
18 </body>
19 </html>
20
```

Fonte: O autor

O resultado:



Fonte: O autor

Comparadores Lógicos

Comparadores lógicos são utilizados em todas as linguagens de programação e no JS, não é diferente.

Para comparar se um valor é igual ao outro é utilizado o `'=='`. Ao testar a variável A com o valor de 50 e B com o texto 50, o resultado será `'true'`(verdadeiro), pois corresponde ao mesmo valor.

Para comparar se um valor é igual ao outro e o tipo de variável é o mesmo se utiliza `'==='`. Ao testar a variável A com o valor de 50 e B com o texto 50, o resultado será `'false'`(falso), pois corresponde ao mesmo valor, mas a variável A recebeu um número, enquanto a variável B recebeu um texto.

O contrário é possível, para ver se um valor não é igual ao outro, se utiliza `'!=='`.

O contrário é possível, para ver se um valor e o tipo não é igual ao outro, se utiliza `'!=='`.

Caso queira ver se a variável A é menor que B, utilize `'<'`. No caso de suas variáveis com o valor 50, o resultado será `'false'`.

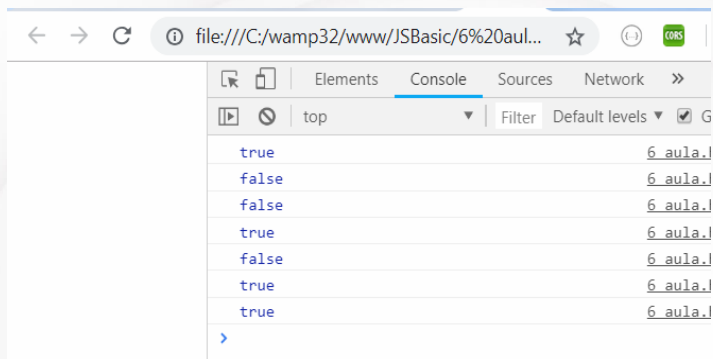
Menor ou igual, `'<='`. No caso de duas variáveis com o valor 50 será `'true'`, não são valores menores, mas são iguais.

Maior utilize `'>'` ou maior ou igual utilize `'>='`. No caso de duas variáveis com o valor 50 será `'true'`, não são valores maiores, mas são iguais.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <script>
    /* comparação */
    var a = 50;
    var b = '50';
    var c = a == b; //verifica o valor
    console.log(c);
    var d = a === b; //verifica o valor e o tipo
    console.log(d);
    var e = a != b; //verifica se o valor não é igual
    console.log(e);
    var f = a !== b; //verifica se o valor e o tipo não é igual
    console.log(f);
    var g = a < b;
    console.log(g);
    var h = a <= b;
    console.log(h);
    var i = a >= b;
    console.log(i);
  </script>
</body>
</html>
```

Fonte: O autor

O resultado ao teclar f12 e ir em Console do google chrome:



Fonte: O autor

O primeiro resultado é true, porque a variável A e B possuem o valor 50.

O segundo resultado é false, porque a variável A assume o valor inteiro 50, já a variável B recebe '50', entre aspas, assumindo o valor de string, ou seja, texto.

O terceiro resultado é false, porque a variável A e B possuem o valor 50.

O quarto resultado é true, porque a variável A assume o valor inteiro 50, já a variável B recebe '50', entre aspas, assumindo o valor de string, ou seja, um número inteiro é diferente de um texto.

O quinto resultado é false, porque a variável A não possui um valor menor que B.

O sexto resultado é true, porque a variável A possui um valor menor ou igual a B.

O sétimo resultado é true, porque a variável A possui um valor maior ou igual a B.

Aqui podemos ver outros comparadores lógicos.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          /* o peradores logicos
10             && and
11             || or
12             ! not
13         */
14         var valor1 = 50;
15         var valor2 = 10;
16         var valor3 = 600;
17
18         var l = (valor1>valor2)&&(valor3<valor2);
19         console.log(l);
20
21         var l2 = (valor1>valor2)|| (valor3<valor2);
22         console.log(l2);
23
24         var l3 = !(valor1>valor2)|| (valor3<valor2);
25         console.log(l3);
26     </script>
27 </body>
28 </html>

```

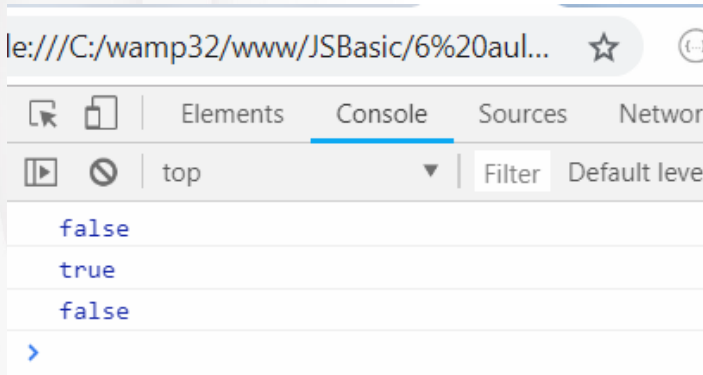
Fonte: O autor

Na linha 18 é possível ver o operando && que é entendível como ‘and’ ou em português ‘e’, desta forma só dará o resultado como ‘true’ caso a condição valor1 > valor2 for verdadeiro e valor3 < valor2 também for verdadeiro.

Já na linha 21 é possível ver o comando `||` que é entendível como 'or', em português 'ou' desta forma dará o resultado como 'true' caso a condição `valor1 > valor2` for verdadeiro ou `valor3 < valor2` for verdadeiro.

Já na linha 24 é possível ver o comando `!` entendível como 'not', em português 'não', desta forma inverte o resultado, no caso de verdadeiro pra falso ou vice versa.

O resultado:



Fonte: O autor

O primeiro resultado é false, porque 50 é maior que 10 E (&&) 600 não é menor que 10. (50>10) E (600<10)

O primeiro resultado é true, porque 50 é maior que 10 OU (||) 600 não é menor que 10. (50>10) OU (600<10)

O primeiro resultado é false, porque 50 é **não(!)** maior que 10 OU 600 não é menor que 10. ! (50>10) OU (600<10)

Comandos condicionais - If e Else

Criaremos uma calculadora usando comandos condicionais IF, Else IF e Else. Estrutura usadas para duas ou três resultados possíveis, mas pode ser usada para inúmeros resultados possíveis.


```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7      <script>
8          var num1 = prompt('Number 1: ');
9          var num2 = prompt('Number 2: ');
10         var op = prompt('Operation: ');
11
12         num1 = parseFloat(num1);
13         num2 = parseFloat(num2);
14
15         if(op == "+"){
16             console.log(num1 + num2);
17         }else if(op == "-"){
18             console.log(num1 - num2);
19         }else {
20             console.error('Operation not found!');
21         }
22
23     </script>
24 </body>
25 </html>
```

Fonte: O autor

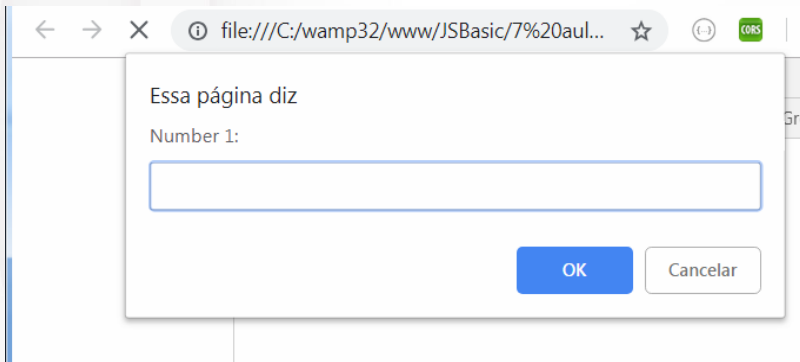
O comando 'If' significa 'Se', o 'Else IF' significa 'Senão Se' e o 'Else' significa 'Senão'.

Desta forma na linha 15 é feita uma comparação, se a variável op for igual ao sinal de soma (+), na linha 16 é realizada a soma.

Mas caso não seja o sinal de soma, na linha 17, é feita outra comparação, se a variável `op` for igual ao sinal de subtração (-), na linha 18 realiza a subtração.

Mas caso nenhum desses testes seja verdadeiro, irá para a linha 19, onde será a clausura `senão`, onde exibirá um texto de operação não encontrada.

O resultado:



Fonte: O autor

Comandos condicionais – CASE

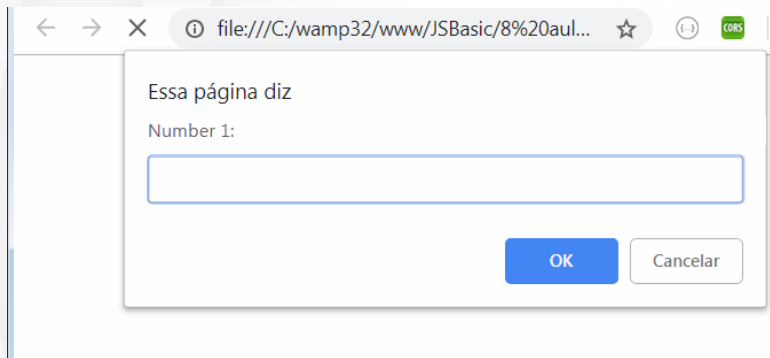
Criaremos agora usando a estrutura case, esta estrutura é usada quando existem várias opções possíveis de resultado.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          var num1 = prompt('Number 1: ');
10         var num2 = prompt('Number 2: ');
11         var op = prompt('Operation: ');
12
13         num1 = parseFloat(num1);
14         num2 = parseFloat(num2);
15
16         /* switch(expression){
17             case x:
18                 default:
19         }
20         */
21         switch(op){
22             case '+':
23                 console.log(num1+num2);
24                 break;
25             case '-':
26                 console.log(num1-num2);
27                 break;
28             case '/':
29                 console.log(num1/num2);
30                 break;
31             case '*':
32                 console.log(num1*num2);
33                 break;
34             default:
35                 console.error('Error');
36         }
37
38     </script>
39 </body>
40 </html>
```

Fonte: O autor

Apesar da estrutura IF, ser muito utilizada, uma outra semelhante é o case, utilizada para quando existem muitas opções, 'Case' significa 'Caso' em português.

O resultado:



Fonte: O autor

Estruturas de Repetição - For

Estruturas de Repetição, são utilizadas quando queremos repetir comandos, sem a necessidade de repetir estes diversas vezes.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          for(i=0; i <= 5; i++){
10             console.log(i);
11         }
12     </script>
13 </body>
14 </html>
```

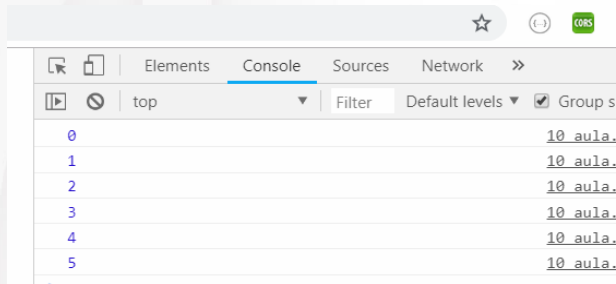
Fonte: O autor

Na linha 9 é possível ver que existe um contador (i), ele contará a quantidade de repetições, este no exemplo se inicia em 0.

Irá repetir 5 vezes, isso é, enquanto i for menor ou igual a 5. Na verdade 6 vezes, pois computadores iniciam com 0 e não com 1.

E `i++` é o sinal que a cada repetição o contador recebe 1. Desta forma ele inicia em 0 e vai atribuindo 1 cada vez que o código entre chaves se repetir.

O resultado:



Fonte: O autor

Aqui podemos ver um programa onde recebemos o valor de duas notas e calculamos a média para 2 alunos.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5      <meta charset="utf-8">
6  </head>
7  <body>
8      <script>
9          for(i=1; i <= 2; i++){
10              var nota1=prompt('Primeira Nota');
11              var nota2=prompt('Segunda Nota');
12              var media=(parseFloat(nota1)+parseFloat(nota2))/2;
13              alert("média é de: "+media);
14          }
15      </script>
16  </body>
17  </html>
```

Fonte: O autor

O resultado:

Essa página diz
média é de: 7.5

OK

Fonte: O autor

Estruturas de Repetição – While e Do

Estruturas repetição While e Do. Principal diferença do Do, é que sempre será executado uma vez, enquanto o for e while só executa se a condição for verdadeira.

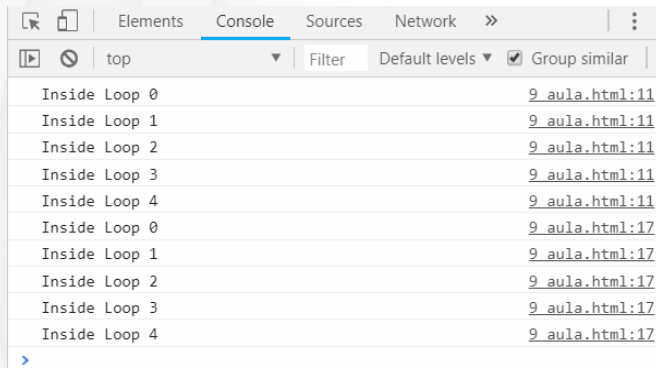
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          var i = 0;
10         while (i<5){
11             console.log('Inside Loop '+i);
12             i++;
13         }
14
15         var j = 0;
16         do{
17             console.log('Inside Loop '+j);
18             j++;
19         }while(j<5);
20     </script>
21 </body>
22 </html>
```

Fonte: O autor

Note que é necessário decalcar um contador, como na linha 9 e 15, e incrementar este, como na linha 12 e 18.

Caso não coloque um contador no while ou do, ele criará um laço infinito e este só irá para quando fechar o seu programa.

O resultado:



Inside Loop 0	9 aula.html:11
Inside Loop 1	9 aula.html:11
Inside Loop 2	9 aula.html:11
Inside Loop 3	9 aula.html:11
Inside Loop 4	9 aula.html:11
Inside Loop 0	9 aula.html:17
Inside Loop 1	9 aula.html:17
Inside Loop 2	9 aula.html:17
Inside Loop 3	9 aula.html:17
Inside Loop 4	9 aula.html:17

Fonte: O autor

Desafio faça um programa que receba a nota de 2 alunos e calcule a média usando o for.

Utilizando funções

Funções são muito utilizadas para facilitar a programação separando em blocos os códigos.

Evitam ter que digitar a mesma função em partes diferentes do seu código, podendo ser chamada diversas vezes, mesmo sendo criada uma única vez.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8      <script>
9          function sayHello(){
10              console.log('Hello from Function');
11          }
12          sayHello();
13
14          function square(){
15              return number * number;
16          }
17
18          //-----
19          var number=4;
20          var b = square();
21          console.log(b);
22
23
24      </script>
25  </body>
26  </html>
```

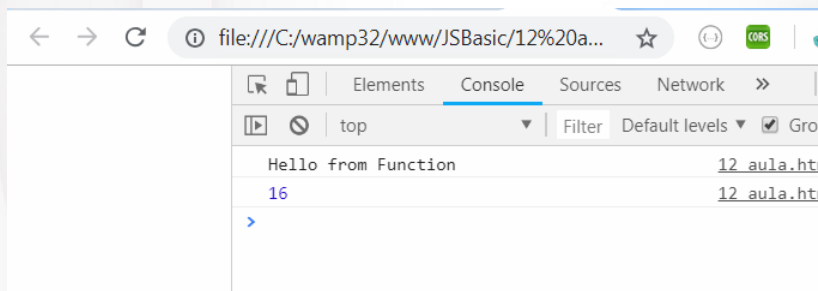
Fonte: O autor

Na linha 9 é criado uma função chamada 'sayHello' e abaixo é descrito o comando para exibir um texto ao usuário.

Ao digitar SayHello() na linha 12 ou em qualquer linha do seu programa exibirá essa função.

Na função square é utilizada para retornar um número multiplicada por ele mesmo.

O resultado:



Fonte: O autor