

# FIAP

Abstract flowing lines in shades of pink and blue, resembling smoke or liquid, set against a dark background.

## Responsive Web Design

1<sup>o</sup>  
semestre  
2018

[www.fiap.com.br](http://www.fiap.com.br)

RENATO BORTOLIN  
[renatoboschini@gmail.com](mailto:renatoboschini@gmail.com)

# Agenda

1. História
2. Principais Características
3. Por que aprender JavaScript
4. Primeiros passos
5. Utilizando Comentários
6. Variáveis
7. Objeto Console
8. Tipo de Dados
9. Operadores
10. Concatenação, Alertas, .....

# História

No início dos anos 90, a empresa Netscape, criadora do navegador mais popular, criou o **LiveScript**, uma linguagem simples que permitia a execução de scripts contidos na página dentro do navegador. Aproveitando o sucesso do Java, que estava conquistando os espaço do mercado de desenvolvimento, a Netscape, logo rebatizou para **JavaScript**, num acordo com a Sun, para alavancar o uso das duas linguagens.

Sua concorrente a Microsoft, criou o **VBScript** no seu navegador Internet Explorer, e criou sua própria versão de script.



# Principais Características

## É uma linguagem interpretada

Isso significa que os comandos JavaScript são executados um a um diretamente por um interpretador, sem que haja necessidade de compilação como nas linguagens de programação convencionais.

No caso do JavaScript, quem faz o papel do interpretador é o navegador do usuário que, ao receber um arquivo html, executa o script na medida em que encontra os comandos JavaScript dentro do código HTML.

# Principais Características

## Javascript é uma linguagem baseada em objetos.

O projeto de programas orientados a objetos tenta tratar um programa como um conjunto de partes individuais (objetos) que executam tarefas distintas, e não como uma seqüência de comandos que executam uma tarefa específica. Objetos similares são agrupados em classes. Numa linguagem orientada a objetos você precisa construir (instanciar) todos os objetos que você for utilizar no seu programa. A linguagem JavaScript é dita “baseada em objetos” porque ela já possui um conjunto de objetos embutidos. Por exemplo, não é necessário construir um objeto para manusear datas e horas pois em JavaScript já existe um pronto para ser usado.



# Principais Características

## Sintaxe parecida com C, C++ e Java

Em termos de sintaxe, a linguagem javascript herdou suas principais construções e comandos da linguagem Java que por sua vez os havia herdado das linguagens C e C++. Devido ao escopo de aplicação limitado de javascript vários recursos existentes nestas outras linguagens foram deliberadamente omitidos.



# Principais Características

## Programação dirigida por eventos

Esta é uma característica muito importante de javascript. Sempre que alguma coisa que tenha um significado importante acontece ocorre um evento.

O clique num botão, o arrastar do mouse, o preenchimento de um campo de um formulário ou o envio do mesmo, são exemplos de eventos provocados pela interação do usuário com a página.

# Principais Características

## Case Sensitive

Case sensitive é um termo da língua inglesa que significa sensível ao tamanho, ou sensível a maiúsculas e minúsculas, ele é usado para indicar que há **diferenças** entre letras com caixa alta e com caixa normal.



# Por que aprender JavaScript?

JavaScript é uma linguagem importante, justamente por rodar no navegador (cliente), o que foi fundamental para a popularidade desta linguagem.

Ela ajuda a melhorar a experiência do usuário, além de resolver questões de performance, evitando que requisições sejam enviadas para o servidor.



# Primeiros Passos – Uso Interno

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4     <meta charset="UTF-8">
5     <title>JavaScript</title>
6
7     <script type="text/JavaScript">
8         //Código JavaScript
9     </script>
10
11 </head>
12 <body>
13
14 </body>
15 </html>
```

Um exemplo de uso de tag `<script>` dentro de um documento HTML4

# Primeiros Passos – Uso Interno

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>JavaScript</title>
6
7     <script>
8         // Código JavaScript
9     </script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

Um exemplo de uso de tag `<script>` dentro de um documento HTML5

# Primeiros Passos – Uso Interno

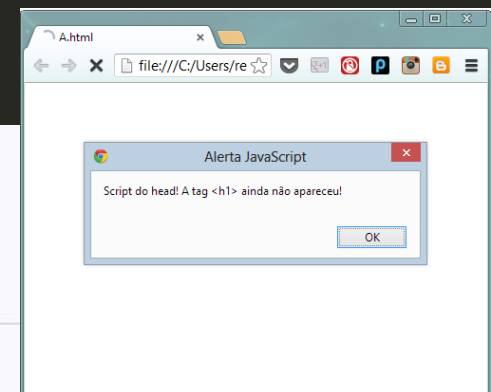
```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>JavaScript</title>
6
7     <script src="js/meucodigo.js"></script>
8
9 </head>
10 <body>
11
12 </body>
13 </html>
```

Um exemplo de uso de tag <script> fora de um documento HTML5

# Primeiros Passos – Uso Interno

A tag <script> pode ser declarada **dentro** da tag <head>, o código é lido imediatamente dentro do navegador.

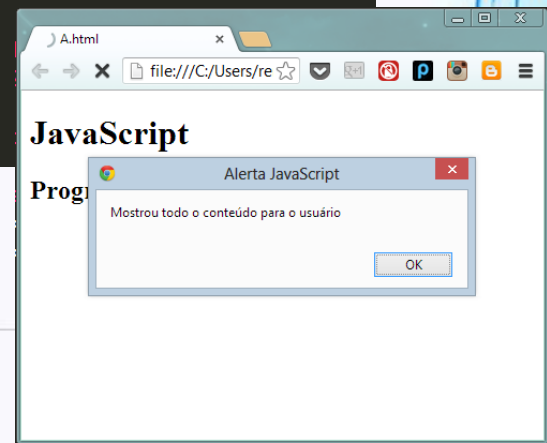
```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <script>
6         alert('Script do head! A tag <h1> ainda não apareceu!');
7     </script>
8 </head>
9 <body>
10     <h1>JavaScript</h1>
11     <h2>Programação front-end em JavaScript</h2>
12 </body>
13 </html>
```



# Primeiros Passos – Uso Interno

A tag <script> pode ser declarada **fora** da tag <head>, antes do fechamento da tag </body>.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5 </head>
6 <body>
7     <h1>JavaScript</h1>
8     <h2>Programação front-end em JavaScript</h2>
9
10    <script>
11        alert('Mostrou todo o conteúdo para o usuário');
12    </script>
13 </body>
14 </html>
```





# Utilizando Comentários

## Utilizando Comentário

Para escrever um comentários dentro do script basta utilizar:

`/* */` = comentário de bloco

`//` = comentário por linha

```
10      <script>
11          //Este é um comentário por linha
12          /*
13              Autor: Renato Bortolin
14              Data: 12/07/2013
15              Descrição: Estudo javascript
16          */
17          alert('Olá alunos');
18      </script>
```

# Variáveis

A abordagem do JavaScript quanto á criação de variáveis é bem direta.

```
10      <script>  
11          var mensagem = "Boa noite alunos";  
12          alert(mensagem);  
13      </script>  
14
```

É uma boa prática utilizar ponto e vírgula após as declarações.

# Variáveis

## Fracamente Tipada

Em linguagens estaticamente tipadas como JAVA, quando uma variável do tipo String é criada, ela morre como String. Em javascript, nada impede que uma variável receba um tipo diferente de seu tipo inicial.

```
10▼    <script>
11        var mensagem = "Boa noite alunos";
12        mensagem = 10;
13        alert(mensagem);
14    </script>
```

# Objeto Console

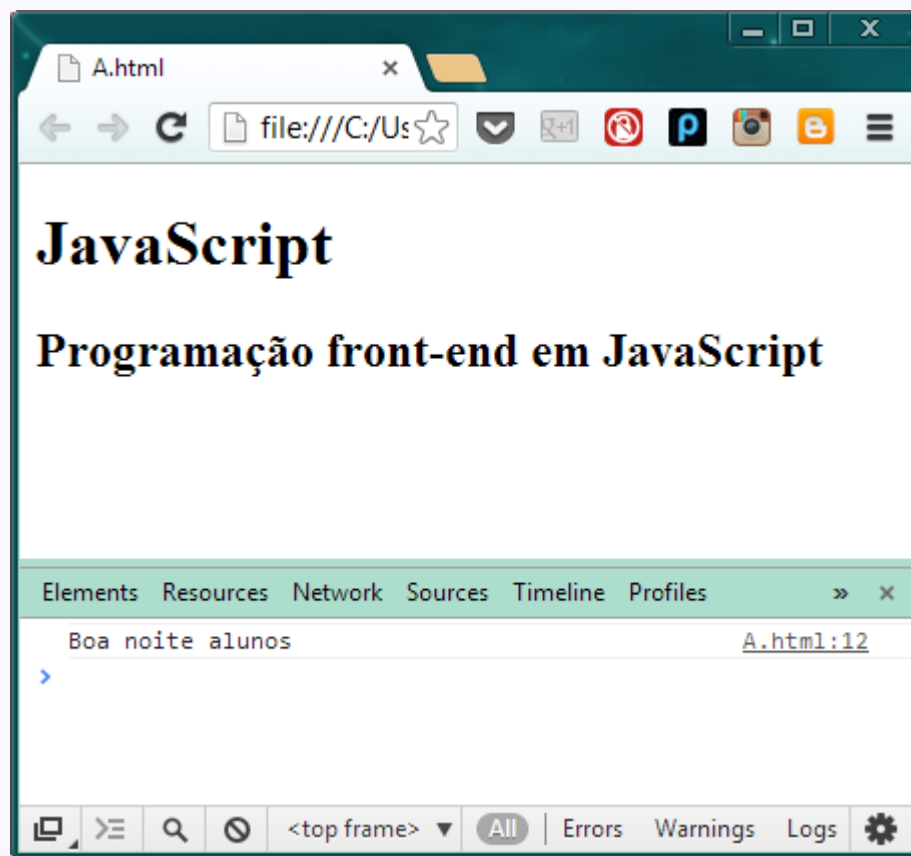
No lugar de mostrar um alerta com o valor digitado, o objeto console do navegador pode ser utilizado para imprimir em seu log.

Para ativar a saída, é necessário ativar o (debugger) do JavaScript do seu navegador.



# Objeto Console

Para ativar a saída, é  
Necessário ativar o  
(debugger) do JavaScript do  
seu navegador.



F12

FIAP

# Tipo de Dados

Os principais tipo de dados do JavaScript são:

- String
- Number
- Boolean

Há também um tipo complexo chamado de **Object**.



# Tipo de Dados

Uma string em JavaScript é utilizada para armazenar trechos de texto. Uma string pode ser criada de duas formas: Como literal ou como objeto String.

Aspas duplas ou simples podem ser utilizadas:

```
10  <script>
11      var empresa = "FIAP"; //literal primitiva
12      console.log("Minha empresa => " + empresa);
13      console.log("Do Tipo => " + typeof empresa) ;
14      console.log("Construtor => " + empresa.constructor)
15      //-----
16      var curso = new String("JavaScript"); //objeto String
17      console.log("Meu curso => " + curso)
18      console.log("Do Tipo => " + typeof curso)
19      console.log("Construtor => " + curso.constructor)
20  </script>
```

# Tipo de Dados

## String

Uma string em JavaScript é utilizada para armazenar trechos de texto. Uma string pode ser criada de duas formas: Como literal ou como objeto String.

Aspas duplas ou simples podem ser utilizadas:

```
10      <script>
11          var empresa = "FIAP"; //literal primitiva
12          console.log("Minha empresa => " + empresa);
13          console.log("Do Tipo => " + typeof empresa) ;
14          console.log("Construtor => " + empresa.constructor)
15          //-----
16          var curso = new String("JavaScript"); //objeto String
17          console.log("Meu curso => " + curso)
18          console.log("Do Tipo => " + typeof curso)
19          console.log("Construtor => " + curso.constructor)
20      </script>
```

# Tipo de Dados

## Atributos e Métodos do tipo String

```
10  <script>
11      var empresa = "FIAP Tecnologia"; //literal primitiva
12      console.log(empresa); // Nome da variável
13      console.log(empresa.length); // Tamanho da string
14
15      //Començando do 0, a posição onde começa a parte procurada
16      console.log(empresa.indexOf("Tecno")); //retorna 5
17
18      //Començando do 0, retorna o caracter da posição 5
19      console.log(empresa.charAt(5)); //retorna T
20
21      /*Extrai uma quantidade de caracteres a partir de
22      posição inicial */
23      console.log(empresa.substr(5, 6)); //retorna Tecnol
24
25      //Substitui a parte "Tecnologia" por "Digital"
26      console.log(empresa.replace("Tecnologia", "Digital"));
27
28  </script>
```

# Tipo de Dados

## Atributos e Métodos do tipo String

String é um objeto imutável

```
10     <script>
11         var empresa = "FIAP Tecnologia"; //literal primitiva
12         console.log(empresa);
13
14         empresa.replace("Tecnologia","Digital");
15         console.log(empresa);
16
17         entidade = empresa.replace("Tecnologia","Digital");
18         console.log(entidade);
19     </script>
```

# Tipo de Dados

## Number

Tem um único objeto para tratar de todos os números. Ele permite a realização de operações aritméticas com maior facilidade, pois todas as conversões necessárias são realizadas automaticamente.

```
10  <script>
11      var vinte = 20; //literal primitiva
12      var objVinte = new Number(20); //Objeto Number
13  </script>
```

# Tipo de Dados

## Number - Conversões

O JavaScript possui funções de conversões de string para number:

```
10▼  <script>
11      var textoInteiro = "20"; //Tipo String
12      console.log(typeof textoInteiro);
13
14      //Conversão
15      var inteiro = parseInt(textoInteiro, 10);
16      console.log(typeof inteiro);
17
18      var textoFloat = "10.22";
19      console.log(typeof textoInteiro);
20
21      //Conversão
22      var floats = parseFloat(textoFloat);
23      console.log(typeof floats);
24  </script>
```



# Tipo de Dados

## Number - Conversões

Number, assim como String, também são imutáveis. No exemplo abaixo, vamos alterar o número das casas decimais com a função toFixed. Esta função retorna uma string, mas para ela funcionar corretamente, seu retorno precisa ser capturado.

```
10      <script>
11          var milNumber = 1000;
12          var milString = milNumber.toFixed(2);
13          console.log(milString);
14      </script>
```

# Tipo de Dados

## Boolean

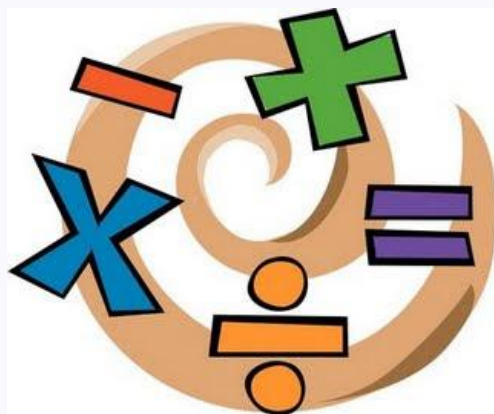
O tipo boolean em JavaScript, também tem seu tipo literal e seu tipo objeto.

O tipo boolean não trás métodos especiais, mas pode receber em seu construtor valores como (true ou false).

```
10      <script>
11          var tipo = true; //Tipo Literal
12          console.log(typeof tipo);
13
14          var tipo2 = new Boolean(false);
15          console.log(tipo2.valueOf());
16          console.log(tipo2.toString());
17
18      </script>
```

# Operadores

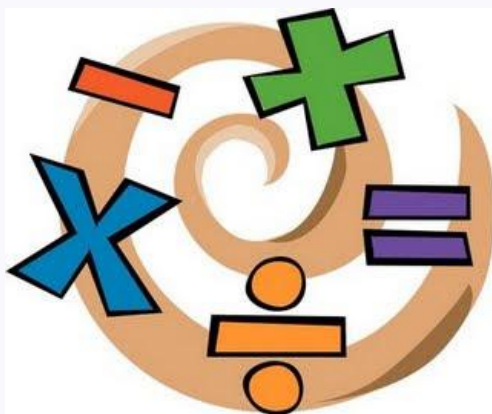
No JavaScript, podemos somar, subtrair, multiplicar, dividir como qualquer linguagem:



```
10▼    <script>
11        var idade = 10;
12        var idade2 = 30;
13        console.log(idade + idade2);
14
15        console.log(idade2 - idade);
16
17        console.log(idade * 2);
18
19        console.log(idade / 2);
20
21        console.log(idade % 2); //Módulo
22    </script>
```

# Operadores

Usando operadores de atribuição permite escrever menos como mostra no exemplo abaixo:



```
10▼ <script>
11     var idade = 10;
12     var idade2 = 30;
13
14     console.log(idade+=10);
15
16     console.log(idade-=5);
17
18     console.log(idade*=10);
19
20     console.log(idade/=10);|
21
22 </script>
```

# Operadores - Concatenação

É possível concatenar tipos diferentes.

## String com String

```
<script>
  var A = Renato;
  var B = Bortolin;
  console.log(A + B);
</script>
```

## String com Outro tipo de dados

```
<script>
  var NumA = 2;
  var NumB = 3;
  var Nome = "Fiap"
  console.log(NumA + NumB + Nome);
</script>
```

# Vamos exercitar

Vamos implementar o código que irá gerar uma String nesse formato a partir de um número.

Primeiramente vamos criar uma variável do tipo Number que será formatada.

Nosso código deve resultar na string “**R\$ 9,90**”. Logo, será necessário converter para string com duas casas decimais, adicionar “R\$ ” ao início e substituir o ponto e virgula.



# Vamos exercitar

## Resultado esperado

```
10  <script>
11      var numero = 9.9;
12      var formatado = numero.toFixed(2)
13
14      formatado = "R$ " + formatado;
15      formatado = formatado.replace(".", ",");
16
17      console.log(formatado);
18  </script>
```

```
18  </script>
19      console.log(formatado);
```

# Vamos exercitar

Resultado esperado em uma única linha.

```
10▼  <script>
11      var numero = 9.9;
12      var formatado = "R$ " + numero.toFixed(2).replace(".", ",");
13
14      console.log(formatado);
15  </script>
```

```
12  </script>
13  console.log(formatado);
```

# Vamos exercitar

Assim como é necessário formatar para mostrar ao usuário, também é comum precisarmos fazer o processo inverso, ou seja a partir de uma string formatada teremos que extrair o número.

Portanto, vamos iniciar declarando uma variável do tipo String.

# Vamos exercitar

Resultado esperado em uma única linha.

```
10  <script>
11      var valor = "R$ 155,89";
12      var valorExtraido = valor.replace("R$", "");
13      valorExtraido = valorExtraido.replace(",", ".");
14      var valor = parseFloat(valorExtraido);
15
16      console.log(valor);
17  </script>
```

```
J1  <\script>
J2      console.log(valor);
```

# Vamos exercitar

Resultado esperado em uma única linha.

```
10 <script>
11     var valor = "R$ 155,89";
12     var valor = parseFloat(valor.replace("R$", "").replace(",", "."));
13
14     console.log(valor);
15 </script>
```

```
12 </script>
13 console.log(valor);
```

# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```

## Igualdade

==

Verdadeiro se os operandos são iguais.

A==3 retorna Verdadeiro  
A==B retorna Falso

# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```

Diferente

!=

Verdadeiro se os  
operandos não são  
iguais.

A!=3 retorna Falso

A!=B retorna Verdadeiro



# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```



Verdadeiro se os operandos são iguais e do mesmo tipo.

A===3 retorna Verdadeiro  
A==="3" retorna Falso

# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```



Verdadeiro se os operandos são iguais e do mesmo tipo.

A!=3 retorna Verdadeiro  
A!="3" retorna Verdadeiro

# Operadores de Comparações

```
10      <script>
11          var A = 3
12          var B = 5
13      </script>
```

Maior quê



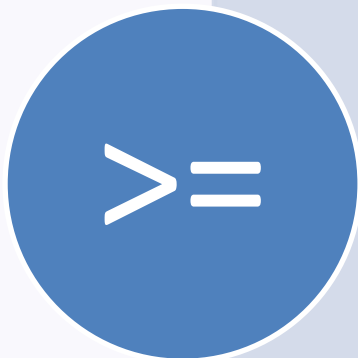
Verdadeiro se o  
operando esquerdo é  
maior que o direito.

A>B retorna falso  
B>A retorna verdadeiro

# Operadores de Comparações

```
10      <script>
11          var A = 3
12          var B = 5
13      </script>
```

Maior igual quê



Verdadeiro se o  
operando esquerdo é  
maior ou igual ao direito.

A >= 3; retorna Verdadeiro  
B >= 7 retorna Falso

# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```

Menor quê



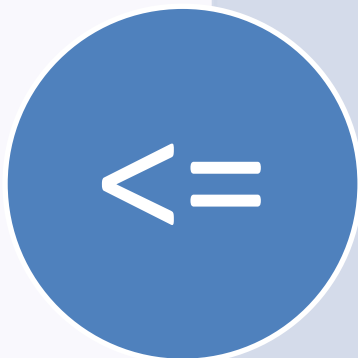
Verdadeiro se o  
operando esquerdo é  
maior ou igual ao direito.

A >= 3; retorna Verdadeiro  
B >= 7 retorna Falso

# Operadores de Comparações

```
10      <script>  
11          var A = 3  
12          var B = 5  
13      </script>
```

## Menor igual quê



Verdadeiro se o operando esquerdo é menor ou igual ao direito.

A <= 3; retorna Verdadeiro  
B <= 0 retorna Falso

# Alertas

Popup boxes no JavaScript nos permitem alertar o usuário sobre algum acontecimento, pedir-lhe uma confirmação para que uma determinada ação possa ser executada e pedir-lhe que nos informe algum dado.

Assim, podemos criar três tipos de popup boxes. São eles:

- **Alert box**
- **Confirm box**
- **Prompt box**



# Alertas

## alert

Esse tipo de popup box tem o objetivo de alertar o usuário sobre algo que ocorreu durante alguma ação por ele executada.

Um botão de OK é exibido junto a ela.

Para prosseguir com a navegação é necessário que o usuário clique nesse botão.

```
10      <script>
11          alert("Vai Corinthians!!!!")
12      </script>
```

# Alertas

## confirm

Com o Confirm, podemos ter a certeza se o usuário deseja ou não executar determinada ação, uma vez que esse box exibe dois botões: OK e Cancel.

Ao clicar em OK o box retorna true e ao clicar em Cancel o box retorna false.

```
10  <script>  
11      confirm("Você acha legal aprender JavaScript");  
12  </script>
```

# Alertas

## prompt

Existem 2 formas

Sem sugestão de resposta:

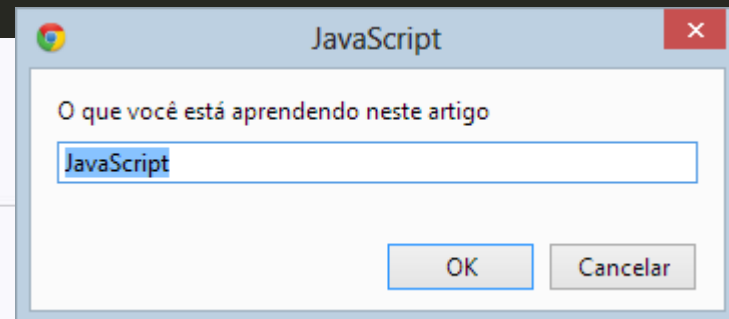
```
prompt("O que você está aprendendo neste artigo", "")
```

Com sugestão de resposta:

```
prompt("O que você está aprendendo neste artigo", "JavaScript")
```

Ambos possuem a mesma característica e função:  
solicitar uma informação do usuário.

```
10 <script>
11     prompt("O que você está aprendendo neste artigo", "JavaScript")
12 </script>
```



# Alertas

## Prompt

### Exemplo de validação

```
10      <script>
11          var idade = prompt("Qual é a sua idade?","");
12          if(idade == ""){
13              alert("Favor preencher sua idade!");
14          }else if (idade >= 18){
15              alert("Acesso aprovado");
16          }else{
17              alert("Acesso reprovado");
18          }
19      </script>
```

# Bloco Condicionais

## if

O **if** é a mais básica das expressões condicionais no JavaScript. Com ele, você pode decidir se quer executar uma ação ou não.

```
10▼    <script>
11        var num = 10;
12▼    if (num > 5)
13        {
14            alert("Maior que cinco!");
15        };
16    </script>
```

## If..else

Utilizamos essa declaração para que somente uma parte do código seja executada quando determinada condição for verdadeira e, outra parte, quando a condição for falsa.

```
10      <script>
11          var num = 10;
12          if (num >= 5)
13              {
14                  alert("Maior que cinco!");
15              }else{
16                  alert("Menor que cinco!");
17              };
18      </script>
```

# If...else If...else

Utilizamos essa declaração quando desejamos selecionar um bloco de código entre vários outros.

```
10▼ <script>
11     var estado = "SP";
12     if(estado == "SP"){
13         alert("Terra da Garoa");
14     }else if(estado == "MG"){
15         alert("Terra do Queijo");
16     }else{
17         alert("Informe o Estado SP ou MG");
18     }
19 </script>
```



# Bloco Condicionais

## If

Ainda é possível utilizar os operadores **&&** (e) e o **||** (ou):

```
10     <script>
11         var num = 10;
12         if (num == 5 || num > 7)
13             {
14                 alert("Atende as condições");
15             };
16
17         if (num >= 5 && num == 7)
18             {
19                 alert("Atende as condições");
20             };
21
22     </script>
```

# if ternário

```
10      <script>
11          var num = 25;
12          var retorno = num > 2 ? "sim" : "não";
13          alert(retorno);
14      </script>
```

# Bloco Condicionais

## switch

Utilizamos também essa declaração quando desejamos selecionar um bloco de código entre vários outros. É uma forma mais prática para alguns casos.

```
10      <script>
11          var estado = "SP";
12          switch(estado){
13              case "SP":
14                  alert("Terra da Garoa");
15                  break
16              case "MG":
17                  alert("Terra do Queijo");
18                  break
19              default:
20                  alert("Informe o Estado SP ou MG");
21          }
22      </script>
```

# Bloco Condicionais

## array

O array é útil quando precisamos trabalhar com diversos valores armazenados. Podemos trabalhar com array de duas formas.

```
10  <script>
11      var amigos = new Array();
12      amigos.push("João");
13      amigos.push("Maria");
14
15      //console.log(amigos);
16      console.log(amigos[0]);
17
18  </script>
```

```
10  <script>
11      var amigos = ["João", "Maria"];
12      //console.log(amigos);
13      console.log(amigos[0]);
14  </script>
```

# Bloco Condicionais

## array

### Outro exemplo

```
10 <script>
11     //Padrão Literal
12     var frutas_1 = ["uva", "melancia", "manga", "caju"];
13
14     //Construtor
15     var frutas_2 = new Array("kiwi", "laranja", "maça", "abacate");
16
17     console.dir(frutas_1);
18     console.dir(frutas_2);
19
20 </script>
```

```
21     console.log(frutas_1[2]);
22     console.log(frutas_2[3]);
```

# Bloco Condicionais

## array

Outros aspecto interessante é o tamanho do array, podemos adicionar quantos elementos quisermos que seu tamanho aumentará quando necessário. Vamos exercitar no console do browser.

```
var carros = []; //Criando um array
carros;          //Objeto criado em memória
carros.constructor;
carros.length;
carros.push("GOL");
carros.sort();   //ordem
carros.reverse(); //ordem reversa
carros.pop();    //remove o último elemento do array
carros.shift();  //remove o primeiro elemento do array
```

# Bloco Condicionais

## for

Muitas vezes precisamos executar um trecho de código repetidamente até que uma condição seja contemplada ou enquanto uma condição for verdadeira. Para isso o JavaScript oferece uma série de blocos de repetição. O mais comum é o FOR.

```
10     <script>
11         var numero = 1;
12         for(numero = 1; numero <= 10; numero++){
13             console.log("O número atual "+numero);
14         }
15     </script>
```



# Bloco Condicionais

## while

O bloco **while** executa determinado código repetitivamente enquanto uma condição for verdadeira. Diferente do bloco **for**.

```
10▼    <script>
11        var contador = 1;
12
13▼    while(contador <=10){
14        console.log(contador + " Fiap.....");
15        contador++;
16    }
17
18        console.log("Valor do contador: " + contador);
19    </script>
```

**</script>**

**Exercícios**

# Exercícios - 1

Criar uma variável que receba um número digitado e apresente os reajustes e descontos a serem obtidos através do número informando.

*Número com reajuste = **0,25***

*Número com desconto = **0,45***

## Exercícios - 2

Pedir um número e mostrar o seu sucessor e o seu antecessor!

## Exercícios - 3

Pedir 4 notas e calcular a média do aluno!

## Exercícios - 4

Comparação de valores.

Receba duas variáveis, e utilize os operadores de comparações, sem a utilização de IF.



## Exercícios - 5

Pedir um número ao usuário e verificar se é maior ou menor que 100, informar na tela a resposta.



## Exercícios - 6

Pedir ao usuário a digitação do seu sexo...  
Exibir se ele é menininho ou menininha.



## Exercícios - 7

Exibir uma mensagem de boas vindas,

(*Bom NOITE GaLeRaaa, hoje é Sexta-feira*)

Converter em letras maiúsculas - ***toUpperCase***

Converter em letras minúsculas – ***toLowerCase***

E apresentar na tela, e perguntar qual é o seu sexo (dê novo).

## Exercícios - 8

Pedir um número e informar se ele é par ou ímpar

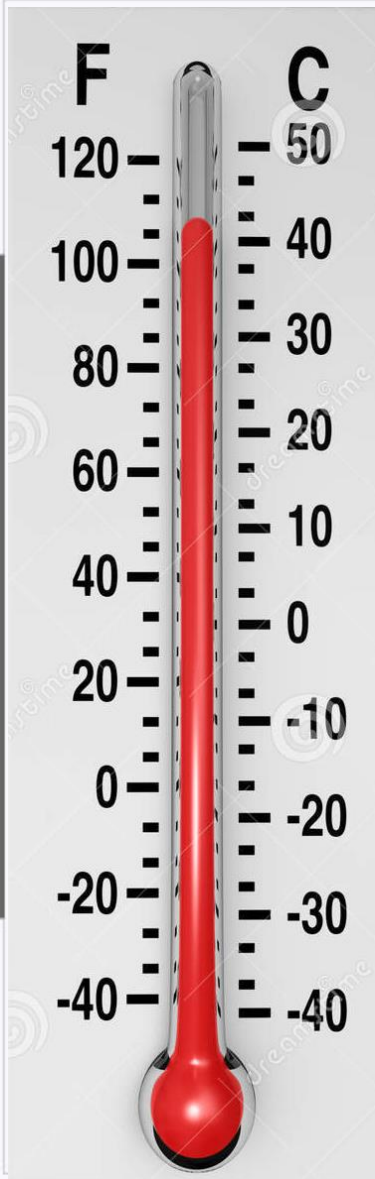


## Exercícios - 9

Pedir ao usuário digitar um número e informar:

- O número digitado foi: ?
- O sucessor do número é: ?
- O antecessor do número é: ?
- O dobro do número é: ?
- A metade do número é: ?

## Exercícios - 10



Informar uma temperatura em **Celsius** e apresentar o resultado em **Fahrenheit**

$$F = \frac{9}{5} C + 32$$



## Exercícios - 11

Crie um script que apresente o *nome do vendedor*, a *quantidade de calças vendidas*, o total de sua *comissão* e o seu *salário* no final do mês.



Comissão = **R\$ 25,00** (por peça vendida)  
Salário = **R\$ 1000,00** (por mês)

## Exercícios - 12

Crie um script que solicite os dados do funcionário como:

- Nome
- Cargo
- Idade
- Salário Bruto

E apresente:

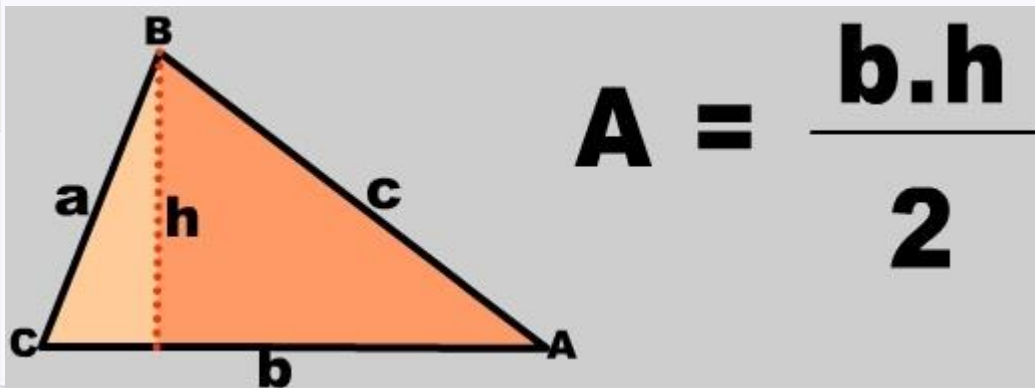
- Reajuste de **38%**
- Bônus de **20%**
- Desconto de **15%**
- Salário Final (Líquido)





## Exercícios - 13

Crie um script que apresente a área do triângulo, solicitando a base e a altura



## Exercícios - 14

Uma empresa tem para um determinado funcionário uma ficha contendo o **nome**, **número de horas trabalhadas** e o **número de dependentes de um funcionário**.

Considerando que a empresa paga **R\$ 98,00** por hora e **R\$ 80,00** por dependentes, apresentar:

- Nome do funcionário
- Salário bruto (sem descontos)
- Desconto de IR (0.05)
- Desconto de INSS (0.085)
- Salário Final

## Exercícios - 15

Pedir ao usuário um número, verificar se o mesmo é múltiplo de 8.

Caso verdade elevar esse número à potência 4, caso falso elevar à potência 5 e informar se é ou não múltiplo de 8.

## Exercícios - 16

Solicitar ao usuário a entrada de 03 números quaisquer.

Efetuar a exibição dos mesmos, bem como a indicação do maior número digitado e do menor número digitado

## Exercícios - 17

Escreva um script para entrar com 10 números e exibir a quantidade de números maiores que 10

## Exercícios - 18

Elaborar um script que peça a entrada de um valor que esteja entre a faixa de 1 a 9. Após a leitura do valor fornecido pelo usuário, o programa deverá indicar uma de duas mensagens: “O valor está na faixa permitida”, caso o usuário forneça o valor nesta faixa, ou a mensagem “O valor está fora da faixa permitida”, caso o usuário forneça valores menores que 1 ou maiores que 9.

## Exercícios - 19

Faça um script que receba um número e diga se este número está no intervalo entre 100 e 200.



## Exercícios - 20

Escreva um script que receba os valores de A, B e C.

Se a soma de  $A+B$  for maior que 20 então escreva o total de  $A+B$ .

Se a soma for menor que 40, leia o valor de C e em seguida escreva a soma de  $A + B + C$ .

# Exercícios - 21

O índice de massa corpórea (IMC) é calculado a partir da seguinte fórmula:

$$\text{IMC} = \text{Peso (kg)} / \text{Altura} * \text{Altura}$$

Monte um script para o usuário entrar com seu respectivo peso e altura. Após isso o script deverá calcular o IMC da pessoa e exibir uma das seguintes mensagens:

| IMC      | CLASSIFICAÇÃO  |
|----------|----------------|
| ATÉ 18,5 | Abaixo do peso |
| ATÉ 25   | Peso Normal    |
| ATÉ 30   | Sobrepeso      |
| ATÉ 35   | Obeso Leve     |
| ATÉ 40   | Obeso Moderado |
| ACIMA 40 | Obeso Mórbido  |



## Exercícios - 22

Faça um scrit que identifique qual é o gênero do animal, como por exemplo:

Mamífero = homem, baleia, golfinho, cachorro

Ovíparo = cobra, tartaruga, jacaré, avestruz

Usando switch

## Exercícios - 23

Exibir uma mensagem

*“EU ADORO O MEU PROFESSOR, ELE É 10”*

**10 vezes**



## Exercícios - 23

Fazer uma contagem de 1 até 10



## Exercícios - 24

Fazer uma contagem de 20 até 0



## Exercícios - 25

Fazer uma contagem de 0 até 1000 pulando de 5 em 5



## Exercícios - 26

Sortear e exibir 10 números entre 0 e 100

## Exercícios - 27

Sortear 10 números e exibir apenas os números pares

## Exercícios - 28

Pedindo 4 notas ao usuário e calcular sua média

## Exercícios - 29

Simular uma pesquisa para 100 pessoas que indica se ela é do sexo **masculino** ou **feminino**

exibir o resultado final

## Exercícios - 30

Simular uma pesquisa para 100 pessoas que indica se ela é do sexo **masculino** ou **feminino**

exibir o resultado final

## Exercícios - 31

Repita uma ação, até o usuário digitar o valor 7  
Usando o *while*

## Exercícios - 32

Usando o **while**

Faça um script perguntando para o usuário qual é o seu sexo (**M** ou **F**), caso digite algo errado, informe a mensagem “**Erro...sexo invalido**” e pergunte novamente.



## Exercícios - 33

Usando o *for* e *while*

Sortear e exibir 10 números pares

## Exercícios - 34

Usando o *array*

Faça um script onde o usuário informa o nome de **5** pessoas e suas respectivas idades.

- *Apresente todos os nomes e suas idades*
- *Apresente o nome e a idade da primeira pessoa que foi informada*

# FIAP



Copyright © **2018** Prof. Renato Bortolin Boschini  
Prof. Israel

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).