

# Curso de JavaScript jQuery

<https://goo.gl/32aGMK>

## Módulo 1 - O Básico e Necessário

### HTML

[O que é](#)

[Não são funções do HTML](#)

[São funções do HTML](#)

[Sintaxe e Nomeclatura](#)

[HTML 5](#)

[DOCTYPE](#)

[Exercício: Layout HTML](#)

[Monte um layout simples em HTML 5 com um cabeçalho, rodapé, menu de navegação e conteúdo.](#)

### CSS

[O que é](#)

[Não são funções do CSS](#)

[São funções do CSS](#)

[Aonde colocar CSS](#)

[Sintaxe e Nomeclatura](#)

[Seletores](#)

[CSS 3](#)

[Exercício: Layout com CSS](#)

[Aplique CSS no layout HTML 5 que você montou.](#)

### JavaScript Nativo

[O que JavaScript pode fazer](#)

[Onde posso colocar JavaScript](#)

[Dentro da Página](#)

[Em arquivos externos](#)

[Mostrando informações](#)

[Depurando código JavaScript](#)

[Funções](#)

[Palavras Reservadas](#)

[Variáveis](#)

[Operadores](#)

[Operadores Aritiméticos](#)

[Operadores de Atribuição](#)

[Operadores de Comparação](#)

[Operadores Lógicos](#)

[Operadores de Bit \(Bitwise\)](#)

[Precedência de Operadores](#)

[Tipos de Dados](#)

[Strings](#)

[Números](#)

[Datas](#)

[Objetos](#)

[Arrays](#)

[Objeto Math](#)[Eventos](#)[Exercício: Triângulo](#)[Monte uma página HTML com três inputs texto e um botão.](#)[Exercício: Fizz Buzz](#)[Faça uma programa que liste os número de 1 a 100](#)[Exercício: Báskara](#)[Exercício: Fibonacci](#)[Exercício: Fatorial](#)[Exercício: Ordenação](#)[Exercício: Idade](#)

## Módulo 1 - O Básico e Necessário

*“You must unlearn what you have learned.”*

Yoda - The Empire Striked Back

Como um dos principais objetivo do jQuery é manipular HTML e CSS, é importante relembrar alguns conceitos antes de iniciarmos.

### HTML

#### O que é

O HTML (HyperText Markup Language) é usado para definir o **conteúdo** de uma página, ou seja sua estrutura, tópicos, assuntos, textos e imagens. É através dele que expressamos as ideias e assuntos que queremos exibir em nosso site.

#### Não são funções do HTML

- Posicionar elementos na tela
- Formatar estilos de elementos
- Formatar o layout e aparência do site

#### São funções do HTML

- Compor o conteúdo da página.
- Através do uso de tags semânticas, permitir uma leitura mais inteligente do conteúdo.
- Estruturar o conteúdo da página através de seções.

#### Sintaxe e Nomeclatura

- Tag: é como chamamos um comandos HTML, como o “div” ou o “img”.
- Atributo: é como chamados uma informação complementar contida dentro de uma tag, como o “style” ou o “src”.



Estrutura básica de uma tag HTML

## HTML 5

Em 2014 foi lançado o HTML 5<sup>1</sup>, que proporcionou novas tags, que nos fornecem mais opções semânticas, permitindo a criação de páginas mais organizadas, legíveis e portanto, facilitando a indexação e localização de informações pelos mecanismos de busca como Google, Bing, etc.

### DOCTYPE

- A tag DOCTYPE não é uma tag HTML.
- O DOCTYPE irá instruir o navegador sobre qual versão do HTML a página foi escrita, assim o navegador saberá como ler e interpretar esse conteúdo<sup>2</sup>.
- O DOCTYPE deve ser a primeira linha dentro de um arquivo HTML.
- É altamente recomendado que você inclua a declaração de DOCTYPE em todas as suas páginas HTML.
- Você pode usar ferramentas de validação<sup>3</sup> para verificar se o HTML que você escreveu está de acordo com os padrões estabelecidos.
- Veja abaixo a declaração básica de uma página em HTML 5

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Título do Documento</title>
</head>

<body>
  Conteúdo do documento
</body>

</html>
```

Estrutura básica de um documento HTML 5.

### Debate: Web Semântica



- Você sabe como os buscadores como Google e Bing funcionam?
- Por que houve tanta preocupação em criar o HTML 5 de forma mais semântica?

### Exercício: Layout HTML



- Monte um layout simples em HTML 5 com um cabeçalho, rodapé, menu de navegação e conteúdo.

## CSS

### O que é

O CSS (Cascading Style Sheets) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.<sup>4</sup>

### Não são funções do CSS

- Definir o conteúdo de uma página.
- Definir o comportamento de uma página.

<sup>1</sup> [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

<sup>2</sup> [http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp)

<sup>3</sup> [http://www.w3schools.com/website/web\\_validate.asp](http://www.w3schools.com/website/web_validate.asp)

<sup>4</sup> [http://pt.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://pt.wikipedia.org/wiki/Cascading_Style_Sheets)

- Executar a lógica ou comandos/instruções de um programa.
- Executar efeitos de animação sobre elementos da página.

### São funções do CSS

- Formatar e estilizar o conteúdo que foi escrito em HTML ou XML.
- Posicionar elementos na tela e definir o layout de uma página.

### Aonde colocar CSS

- Dentro de elemento HTML, também chamado de “inline”:

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

- Dentro da tag “head” do seu documento HTML

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

- Em arquivos externos que serão vinculados a sua página através da tag “link” dentro do “head” do HTML.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

### Sintaxe e Nomeclatura

- Seletor: é como chamamos a regra que irá selecionar os elementos que terão seu estilo alterado.
- Declaração: é como chamamos uma regra CSS.
- Uma declaração é uma lista de pares “propriedade” e “valor”.

<u>seletor</u>	<u>declaração</u>	<u>declaração</u>
h1	{ color:red; font-size:14px;}	
	↑     ↑	↑     ↑
	propriedade valor	propriedade valor

Sintaxe básica de um comando em CSS

### Seletores

Existem diversas forma de selecionar os elementos para aplicar uma regra de CSS.

- Seleção pela tag

```
p {  
  text-align: center;  
  color: red;  
}
```

- 
- Seleciona todos elementos HTML da tag "p".

- Seleção pelo ID

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

- 
- Seleciona o elemento com o ID "para1"

- Seleção pela classe CSS

```
.center {  
  text-align: center;  
  color: red;  
}
```

- 
- Seleciona todos os elementos que tem o valor "center" no atributo "class".

```
p.center {  
  text-align: center;  
  color: red;  
}
```

- 
- Seleciona todos os elementos HTML "p" que possuem a classe CSS "center".

### CSS 3

- CSS 3 ainda está em desenvolvimento, mas navegadores modernos já oferecem um grande suporte a essa nova versão da especificação.
- Consulte esse [link](http://www.w3schools.com/cssref/css3_browsersupport.asp)<sup>5</sup> para verificar quais recursos do CSS 3 estão disponíveis em cada navegador.
- Consulte esse [link](http://css3test.com/)<sup>6</sup> para testar quais recursos do CSS 3 o seu navegador suporta.
- Consulte esse [link](http://caniuse.com/)<sup>7</sup> para verificar quais navegadores suportam algum determinado recurso do CSS 3, HTML 5 e JavaScript.

#### Um passo a frente: Frameworks CSS

- Atualmente é muito comum utilizar frameworks CSS como o [Bootstrap](http://getbootstrap.com/)<sup>8</sup> e o [Foundation](http://foundation.zurb.com/)<sup>9</sup> para facilitar o desenvolvimento Web.
- Veja uma comparação de alguns frameworks CSS através desse [link](http://usablica.github.io/front-end-frameworks/compare.html)<sup>10</sup>.
- Leia [esse artigo](http://www.infoq.com/br/news/2009/07/dry-css-less-yass)<sup>11</sup> para mais informações.

<sup>5</sup> [http://www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp)

<sup>6</sup> <http://css3test.com/>

<sup>7</sup> <http://caniuse.com/>

<sup>8</sup> <http://getbootstrap.com/>

<sup>9</sup> <http://foundation.zurb.com/>

<sup>10</sup> <http://usablica.github.io/front-end-frameworks/compare.html>

<sup>11</sup> <http://www.infoq.com/br/news/2009/07/dry-css-less-yass>



### Um passo a frente: Programando CSS com LESS ou SASS

- CSS as vezes pode ser muito repetitivo.
- Não haveriam maneiras de melhorar isso?
- Sim! Através de LESS<sup>12</sup> e SASS<sup>13</sup>.
- Leia [esse artigo](#)<sup>14</sup> para mais informações.



### Debate: HTML 5 e CSS 3 vs Flash

- Você conhece Flash?
- Como Flash e HTML 5 e CSS 3 se comparam?
- Qual será o futuro?



### Exercício: Layout com CSS

- Aplique CSS no layout HTML 5 que você montou.
- Posicione o menu a esquerda e o conteúdo no centro.

## JavaScript Nativo

Antes de iniciarmos com o estudo do framework jQuery, vamos revisar o JavaScript nativo e fixar alguns conceitos básicos.

O HTML nos dá o conteúdo, o CSS nos ajuda a formatar e estilizar esse conteúdo para apresentação, e o JavaScript irá nos auxiliar a definir o comportamento desse conteúdo quando o usuário interage com a página.

### O que JavaScript pode fazer

- Acessar elementos HTML da página e alterar o seu conteúdo:
  - [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_inner\\_html](http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_inner_html)
- Alterar atributos de elementos HTML:
  - [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_myfirst](http://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst)
- Alterar o estilos CSS de elementos HTML
  - [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_style](http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_style)
- Validar dados de formulários:
  - [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_validate](http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_validate)

### Onde posso colocar JavaScript

Você pode inserir código JavaScript em diversos locais de sua página, vejamos alguns exemplos.

#### Dentro da Página

Você pode adicionar código javascript nas áreas de “head” e “body” de sua página através da tag “script”.

```
<script>
document.getElementById("demo").innerHTML = "Meu primeiro JavaScript";
</script>
```

Exemplo de código javascript.

Note que não é mais necessário incluir a tag “<script type=“text/javascript”>”. JavaScript é a linguagem padrão de script para HTML.

<sup>12</sup> <http://lesscss.loopinfinite.com.br/>

<sup>13</sup> <http://sass-lang.com/>

<sup>14</sup> <http://www.infoq.com/br/news/2009/07/dry-css-less-yass>

### Em arquivos externos

- Scripts podem ser colocados em arquivos externos a sua página. Eles devem ser salvos com a extensão “.js”.
- Arquivos externos são úteis quando você usa o mesmo código em diversas páginas.
- Você pode vincular arquivos externos em ambos os elementos “head” e “body”.
- O navegador irá carregá-los exatamente no mesmo lugar em que a tag “script” foi colocada.
- Para definir um script externo a página use o atributo “src” da tag “script”.
- Vantagens de utilizar arquivos externos:
  - Separa o código HTML do JS.
  - Facilita a leitura e manutenção do código.
  - Arquivos JS podem ser salvos em cache pelo navegador e diminuem o tempo de carregamento da página.

```
<!DOCTYPE html>
<html>
<body>
<script src="meuScript.js"></script>
</body>
</html>
```

Exemplo de arquivo JS externo a página.

### Mostrando informações

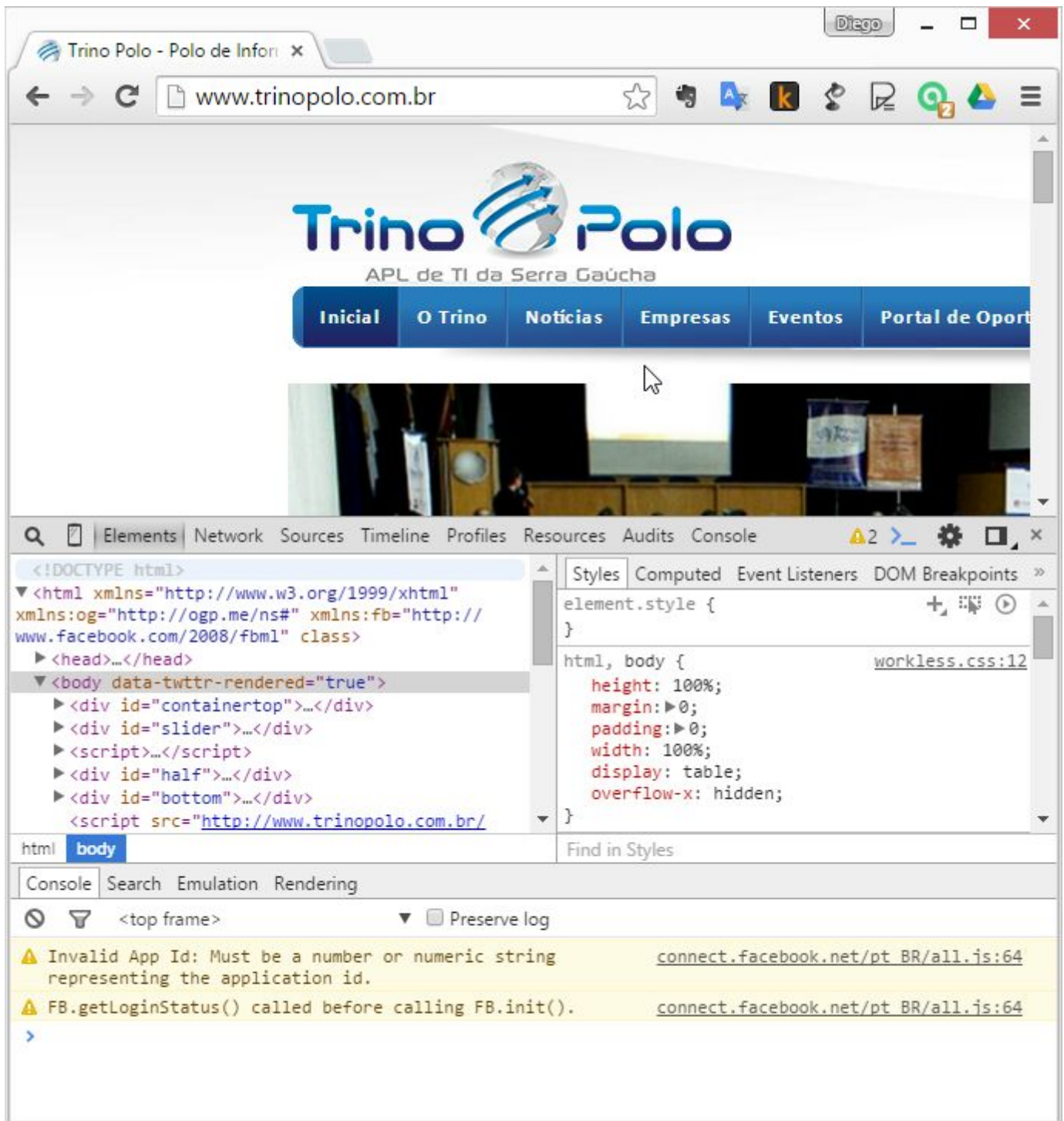
JavaScript pode mostrar informações de 4 maneiras diferentes:

Comando	Descrição
window.alert()	Mostra uma caixa de alerta no navegador. <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_alert">http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_alert</a>
document.write()	Escreve no documento (página) atual.  Se o comando estiver no meio do carregamento da página, o conteúdo será escrito no local em que o comando for colocado. <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write">http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write</a>  Se o comando for executado depois que a página terminar de carregar, todo o conteúdo da página será substituído pelo comando. <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write_over">http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write_over</a>
document.getElementById("demo").innerHTML	Substitui o conteúdo HTML de um elemento com o id “demo” <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_dom">http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_dom</a>
console.log()	Gera uma mensagem de log na console do seu navegador. Ative a console através da tecla de atalho “F12” para ver a mensagem. <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_console">http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_console</a>

### Depurando código JavaScript

A maioria dos navegadores atuais já inclui um depurador de código JavaScript embutido, você precisa apenas pressionar a tecla de atalho F12 para exibir o painel com as ferramentas de desenvolvimento. Use o Firefox ou o Chrome e pressione F12 para ver o painel de ferramenta. O painel de ferramentas permite que você defina “break points” no seu código para acompanhar a execução do código JavaScript.





Painel de ferramentas de desenvolvimento do Google Chrome.

## Funções

- Você pode agrupar comandos JS em blocos de código através de chaves “{” e “}”.
- Esses blocos de código podem ter um nome e parâmetros de entrada.
- Também podem retornar um valor ao final de sua execução.
- Esses blocos de código são chamados de “funções” e podem ser executados a qualquer momento depois de sua declaração através da “chamada” do seu nome.
- Veja um exemplo de função:
  - [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_function\\_return](http://www.w3schools.com/js/tryit.asp?filename=tryjs_function_return)
- Mais informações sobre funções:
  - [http://www.w3schools.com/js/js\\_functions.asp](http://www.w3schools.com/js/js_functions.asp)



```
var x = multiplicar(4, 3);

function multiplicar(a, b) {
    return a * b;
}
```

O resultado em "x" será:

12

Exemplo de função multiplicando dois números.

## Palavras Reservadas

Palavras reservadas são exclusivas de linguagem e não podem ser usadas como nomes de variáveis ou de funções. Clique sobre o nome da palavra reservada para ver mais informações. Para uma lista completa das palavras reservadas consulte [esse link](http://www.w3schools.com/js/js_reserved.asp)<sup>15</sup>.

Palavra	Descrição	Exemplo
<a href="#">break</a>	Termina a execução de um switch ou um loop.	<pre>for (i = 0; i &lt; 10; i++) {     if (i === 3) { break; }     text += "The number is " + i + "&lt;br&gt;"; }</pre>
<a href="#">continue</a>	Pula a interação atual de um loop e retornar para o início do loop.	<pre>for (i = 0; i &lt; 10; i++) {     if (i === 3) { continue; }     text += "The number is " + i + "&lt;br&gt;"; }</pre>
<a href="#">do ... while</a>	Executa um bloco de comandos até que a condição seja verdadeira.	<p>Exemplo 1 - While ... do</p> <pre>while (i &lt; 10) {     text += "The number is " + i;     i++; }</pre> <p>Exemplo 2 - Do ... while</p> <pre>do {     text += "The number is " + i;     i++; } while (i &lt; 10);</pre>
<a href="#">for</a>	Executa um bloco de comandos enquanto a expressão for verdadeira.	<p>Exemplo 1 - Array</p> <pre>for (i = 0; i &lt; cars.length; i++) {     text += cars[i] + "&lt;br&gt;"; }</pre>

<sup>15</sup> [http://www.w3schools.com/js/js\\_reserved.asp](http://www.w3schools.com/js/js_reserved.asp)

		<p>Exemplo 2 - Objeto</p> <pre>var person = {fname:"John", lname:"Doe", age:25};  var text = ""; var x; for (x in person) {     text += person[x]; }</pre>
<a href="#">function</a>	Declara uma função.	<pre>function toCelsius(fahrenheit) {     return (5/9) * (fahrenheit-32); } document.getElementById("demo").innerHTML = toCelsius(32);</pre>
<a href="#">if ... else</a>	Executa um bloco de comandos ou não dependendo da condição.	<pre>if (time &lt; 10) {     greeting = "Good morning"; } else if (time &lt; 20) {     greeting = "Good day"; } else {     greeting = "Good evening"; }</pre>
<a href="#">return</a>	Sai de uma função.	Veja o item "function" dessa tabela.
<a href="#">switch</a>	Executa um bloco de comandos de acordo com a condição.	<pre>switch (new Date().getDay()) {     case 6:         text = "Today is Saturday";         break;     case 0:         text = "Today is Sunday";         break;     default:         text = "Looking forward to the Weekend"; }</pre>
<a href="#">try ... catch</a>	Implementa tratamento de erros para um bloco de código.	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;p id="demo"&gt;&lt;/p&gt;  &lt;script&gt; try {     adddler("Welcome guest!"); } catch(err) {     document.getElementById("demo").innerHTML = err.message; } &lt;/script&gt;  &lt;/body&gt; &lt;/html&gt;</pre> <p>Veja o exemplo:  <a href="http://www.w3schools.com/js/tryit.asp?filename=tryjs_try_catch">http://www.w3schools.com/js/tryit.asp?filename=tryjs_try_catch</a></p>

<a href="#">var</a>	Declara uma variável.	<pre>var pi = 3.14; var person = "John Doe"; var answer = 'Yes I am!';</pre>
---------------------	-----------------------	--

## Variáveis

- Todas as variáveis em Javascript devem ser identificadas com nomes únicos. Esses nomes únicos são chamados de identificadores.
- Você pode declarar variáveis em qualquer parte do seu código Javascript, mas é aconselhado declarar todas as variáveis necessárias no início do script ou função, agrupando todas em um mesmo local.
- O JavaScript move todas as declarações de variáveis para o topo do bloco de código aonde ela é usada. Isso é chamado hoisting.<sup>16</sup>
- Regras para nomenclatura de identificadores:
  - Podem conter letras, número, *underline* e sinal de dolar (\$).
  - Devem começar com uma letra.
  - Também podem começar com "\$" e "\_".
  - Identificadores são case sensitiv, ou seja, "x" e "X" são duas variáveis diferentes.
  - Palavras reservadas não podem ser usadas como nomes de variáveis.

## Operadores<sup>17 18</sup>

### Operadores Aritiméticos

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo da divisão (resto).
++	Incremento
--	Decremento

### Operadores de Atribuição

Operador	Exemplo	Equivalente a
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y

<sup>16</sup> [http://www.w3schools.com/js/js\\_hoisting.asp](http://www.w3schools.com/js/js_hoisting.asp)

<sup>17</sup> [http://www.w3schools.com/js/js\\_operators.asp](http://www.w3schools.com/js/js_operators.asp)

<sup>18</sup> [http://www.w3schools.com/js/js\\_arithmetic.asp](http://www.w3schools.com/js/js_arithmetic.asp)

### Operadores de Comparação

Operador	Descrição
==	Igualdade
===	Igualdade de conteúdo e tipo
!=	Diferente
!==	Diferente em conteúdo ou tipo
>	Maior que
<	Menor que
>=	Maior ou igual que
<=	Menor ou igual que

### Operadores Lógicos

Supondo que x = 6, y = 3.

Operador	Descrição	Exemplo
&&	and	(x < 10 && y > 1) = true
	or	(x == 5    y == 5) = false
!	not	!(x == y) = true

### Operadores de Bit (Bitwise)

- Funcionamento em número de 32 bits.
- Os números são convertidos para binário e depois que a instrução é executada, eles são novamente convertidos para decimal.

Operador	Descrição	Exemplo	Equivalente a	Resultado	Decimal
&	AND	x = 5 & 1	0101 & 0001	0001	1
	OR	x = 5   1	0101   0001	0101	5
~	NOT	x = ~ 5	~0101	1010	10
^	XOR	x = 5 ^ 1	0101 ^ 0001	0100	4
<<	Left shift	x = 5 << 1	0101 << 1	1010	10
>>	Right shift	x = 5 >> 1	0101 >> 1	0010	2

### Precedência de Operadores

Operador	Precedência
( )	Agrupamento de expressões
++ --	Incremento e decremento
* / %	Multiplicação, divisão e módulo da divisão
+ -	Adição e subtração.

### Tipos de Dados



### Jogo: Adivinhe o resultado

Tente adivinhar o resultado de cada um desses comandos abaixo.

1.

```
var x = 16 + "Volvo";
```

2.

```
var x = 16 + 4 + "Volvo";
```

3.

```
var x = "Volvo" + 16 + 4;
```

4.

```
typeof undefined
typeof null
null === undefined
null == undefined
```

- Variáveis em Javascript podem ser de diferentes tipos de dados: Número, String, Objeto, Array, Booleano.
- Em uma expressão, se o segundo parâmetro é uma string, o primeiro será tratado como string também, por isso no jogo acima o exemplo 1 resultará em "16Volvo".
- Expressões são executadas da esquerda para a direita, por isso no jogo acima o exemplo 2 resultará em "20Volvo" e o exemplo 3 resultará em "Volvo164".
- Javascript tem tipos dinâmicos, isso significa que você pode trocar o tipo de uma variável em tempo de execução.

```
var x;           // Agora x é "undefined"
var x = 5;       // Agora x é um "Number"
var x = "John";  // Agora x é uma "String"
```

- Strings podem usar aspas duplas ou simples.
- Valores booleanos são representados por "true" e "false".
- Arrays são definidos usando colchetes e separando os valores por vírgula.

```
var cars = ["Saab", "Volvo", "BMW"];
```

- Objetos são definidos usando chaves e são escritos em um formato de pares de chave e valor.

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

- Você pode usar o comando "typeof" para descobrir o tipo de dados de uma variável.

```
typeof "John"           // Returns string
typeof 3.14              // Returns number
typeof false            // Returns boolean
typeof [1,2,3,4]         // Returns object
typeof {name:'John', age:34} // Returns object
```

- Quando uma variável ainda não foi inicializada, ela terá um tipo chamado "undefined".
- Undefined e Null são diferentes

```

typeof undefined // undefined
typeof null      // object
null === undefined // false
null == undefined // true

```

- Não instancie String com o operador "new", isso torna a execução mais lenta e pode levar a erros como no exemplo abaixo:

```

var x = new String("John");
var y = new String("John");

// (x == y) será FALSO
// Objetos não podem ser comparados em Javascript

```

## Strings

Objetos do tipo String possuem algumas funções pré-definidas. Clique sobre o nome da função para ver um exemplo ou [clique aqui](#)<sup>19</sup> para a lista completa das funções de string.

Propriedade / Método	Descrição
<a href="#">length</a>	Retorna o comprimento da string (quantos caracteres ela tem). Note que é uma propriedade, não um método (não tem []).
<a href="#">charAt()</a>	Retorna o caracter em uma posição.
<a href="#">charCodeAt()</a>	Retorna o Unicode de um caracter em uma posição.
<a href="#">concat()</a>	Junta duas ou mais strings e retorna.
<a href="#">fromCharCode()</a>	Converte valores Unicode para caracteres.
<a href="#">indexOf()</a>	Retorna a posição da primeira ocorrência de um termo em uma string.
<a href="#">lastIndexOf()</a>	Retorna a posição da última ocorrência de um termo em uma string.
<a href="#">localeCompare()</a>	Compara duas string com a localidade corrente.
<a href="#">match()</a>	Faz uma pesquisa na string usando uma expressão regular e retorna as ocorrências.
<a href="#">replace()</a>	Substitui uma parte de uma string por outra.
<a href="#">search()</a>	Retorna a posição de uma expressão dentro de uma string.
<a href="#">slice()</a>	Extrai uma parte de uma string.
<a href="#">split()</a>	Divide a string em um array de strings.
<a href="#">substr()</a>	Extrai uma parte de uma string baseado no índice de início e número de caracteres subsequentes.
<a href="#">substring()</a>	Extrai uma parte de uma string baseado no índice inicial e final.
<a href="#">toLocaleLowerCase()</a>	Transforma a string para minúsculo de acordo com a localidade do host.
<a href="#">toLocaleUpperCase()</a>	Transforma a string para maiúsculo de acordo com a localidade do host.
<a href="#">toLowerCase()</a>	Converte uma string para minúsculo.
<a href="#">toString()</a>	Retorna o valor de um objeto String.

<sup>19</sup> [http://www.w3schools.com/jsref/jsref\\_obj\\_string.asp](http://www.w3schools.com/jsref/jsref_obj_string.asp)



<a href="#">toUpperCase()</a>	Converte uma string para maiúsculo.
<a href="#">trim()</a>	Remove os espaço em branco do início e fim de uma string.
<a href="#">valueOf()</a>	Retorna o valor primitivo de um objeto string.

## Números

- Javascript tem apenas um tipo de dados para número.
- Podem ser escritos com ou sem casas decimais.
- Use "." para definir número com casas decimais.

```
var x1 = 34.00;
var x2 = 34;
```

- Valores muito grandes ou muito pequenos podem ser expressos usando notação científica (exponencial).

```
var y = 123e5;      // 12300000
var z = 123e-5;     // 0.00123
```

- Números em JavaScript são sempre armazenados como double, seguindo o padrão internacional ([IEEE 754](#)<sup>20</sup>)
- Os número são armazenados em 64 bits, de acordo com a definição abaixo:

Valor	Expoente	Sinal
52 bits (0 - 51)	11 bits (52 - 62)	1 bit (63)

- Precisão da parte inteira: 15 dígitos:

```
var x = 999999999999999; // x will be 999999999999999
var y = 999999999999999; // y will be 1000000000000000
```

- Precisão da parte decimal, até 17 casas, mas operações aritméticas com pontos flutuantes nem sempre são 100% precisas.

```
var x = 0.2 + 0.1;      // x will be 0.30000000000000004
```

- Quando um número ultrapassa os valores definidos por esses limites, o Javascript irá atribuir o valor "Infinite".

```
var myNumber = 2;
while (myNumber != Infinity) {      // Execute until Infinity
    myNumber = myNumber * myNumber;
}
```

- Divisão por zero não gera um erro, mas o resultado será "Infinite".

```
var x = 2 / 0;      // x will be Infinity
var y = -2 / 0;     // y will be -Infinity
```

<sup>20</sup> [http://pt.wikipedia.org/wiki/IEEE\\_754](http://pt.wikipedia.org/wiki/IEEE_754)

- Quando uma operação matemática não pode ser realizada, Javascript irá retornar NaN (Not a Number / Não é um número) como resultado.

```
var x = 100 / "Apple"; // x will be NaN (Not a Number)
```

- Você pode usar a função isNaN() para testar se um valor é um número ou não.

```
var x = 100 / "Apple";
isNaN(x); // returns true because x is Not a Number
```

- Não instancie número usando o operador "new". Isso torna a execução mais lenta e pode levar a erros como o exemplo abaixo.

```
var x = new Number(500);
var y = new Number(500);

// (x == y) será FALSO
// objetos não podem ser comparados em Javascript
```

Objetos de números tem algumas funções pré-definidas. Clique sobre o nome da função para ver um exemplo ou [clique aqui](#)<sup>21</sup> e [aqui](#)<sup>22</sup> para a lista completa das funções de string.

Propriedade / Método	Descrição
<a href="#">MAX_VALUE</a>	Retorna o maior valor possível em Javascript.
<a href="#">MIN_VALUE</a>	Retorna o menor valor possível em Javascript.
<a href="#">NEGATIVE_INFINITY</a>	Representa o negativo infinito (retornado em caso de overflow).
<a href="#">NaN</a>	Representa um valor que não é um número.
<a href="#">POSITIVE_INFINITY</a>	Representa o infinito (retornado em caso de overflow).
<a href="#">toExponential(x)</a>	Converte o número em notação exponencial.
<a href="#">toFixed(x)</a>	Formata o número com "x" dígitos após o separador decimal.
<a href="#">toPrecision(x)</a>	Formata um número para "x" dígitos de precisão.
<a href="#">toString()</a>	Converte um número para String.
<a href="#">valueOf()</a>	Retorna o valor primitivo do número.
<b>Funções globais do Javascript</b>	
<a href="#">parseFloat()</a>	Interpreta o argumento e o transforma em um número de ponto flutuante.
<a href="#">parseInt()</a>	Interpreta o argumento e o transforma em um número inteiro.

### Datas

- Javascript permite criar e manipular datas.
- Internamente, Javascript armazena datas como um número inteiro que representa o número de milissegundos desde 01/01/1970 00:00:00.
- Você pode criar datas de 4 maneiras diferentes

<sup>21</sup>[http://www.w3schools.com/js/js\\_number\\_methods.asp](http://www.w3schools.com/js/js_number_methods.asp)

<sup>22</sup> [http://www.w3schools.com/jsref/jsref\\_obj\\_number.asp](http://www.w3schools.com/jsref/jsref_obj_number.asp)

```
// Cria a data atual
new Date()

// Cria uma data com o número de milissegundos desde 01/01/1970 00:00:00
new Date(milissegundos)

// Cria uma data a partir de uma String que representa uma data
new Date(dataEmString)

// Cria uma data com os valores especificados
new Date(ano, mes, dia, hora, minuto, segundo, milissegundo)
```

- O mês é um intervalo entre 0 e 11, onde 0 é Janeiro e 11 Dezembro.
- Para ver exemplos de criação de objetos de data consulte [esse link](#)<sup>23</sup> e [esse link](#)<sup>24</sup>.
- Objetos de data já possuem alguns funções pré-definidas que você pode usar, consulte na tabela abaixo. Clique no nome da função para ver um exemplo. Clique [nesse link](#)<sup>25</sup> para ver a lista completa de funções disponíveis.

Método	Descrição
<a href="#">getDate()</a>	Retorna o dia do mês (entre 1-31)
<a href="#">getDay()</a>	Retorna o dia da semana (entre 0-6)
<a href="#">getFullYear()</a>	Retorna o ano (4 dígitos)
<a href="#">getHours()</a>	Retorna a hora (entre 0-23)
<a href="#">getMilliseconds()</a>	Retorna os milissegundos (entre 0-999)
<a href="#">getMinutes()</a>	Retorna os minutos (entre 0-59)
<a href="#">getMonth()</a>	Retorna o mês (entre 0-11)
<a href="#">getSeconds()</a>	Retorna os segundos (entre 0-59)
<a href="#">getTime()</a>	Retorna o número de milissegundos transcorridos desde 01/01/1970 até a data representada pelo objeto.

### Objetos

Você pode criar objeto em Javascript, definir suas propriedades e atribuir-lhe comportamento através de funções (métodos). Veja um exemplo.

<sup>23</sup> [http://www.w3schools.com/js/js\\_dates.asp](http://www.w3schools.com/js/js_dates.asp)

<sup>24</sup> [http://www.w3schools.com/js/js\\_date\\_formats.asp](http://www.w3schools.com/js/js_date_formats.asp)

<sup>25</sup> [http://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](http://www.w3schools.com/jsref/jsref_obj_date.asp)



```
1 <span id="informacoes"></span>
```

```
1 // Cria um objeto de pessoa
2 var pessoa = {
3
4   // Atributos da pessoa
5   primeiroNome:"João",
6   ultimoNome:"da Silva",
7   nascimento: new Date("1980-10-25"),
8
9   // Nome completo
10  nomeCompleto: function() {
11    return this.primeiroNome + ' ' + this.ultimoNome;
12  },
13
14  // Calcula a idade
15  idade: function() {
16    var dataAtual = new Date();
17    return dataAtual.getFullYear() - this.nascimento.getFullYear();
18  }
19 };
20
21 // Mostra informações da pessoa
22 var info = document.getElementById('informacoes');
23 info.innerHTML = pessoa.nomeCompleto() +
24   " tem " + pessoa.idade() + " anos.";
```

João da Silva tem 35 anos.

<http://jsfiddle.net/diegokeller/j957s2Lr/>

## Arrays

- Arrays são usados para armazenar múltiplos valores em uma única variável.
- Criando arrays

### Sintaxe

```
var array-name = [item1, item2, ...];
```

### Exemplo

```
var cars = ["Saab", "Volvo", "BMW"];
```

- Você também pode criar usando a palavra chave "new", mas isso é desaconselhado pois diminui a performance do programa.

```
var cars = new Array("Saab", "Volvo", "BMW");
```

- Você pode criar um array com um determinado número de elementos. No exemplo abaixo o array terá 40 posições.

```
var points = new Array(40);
```

- Os elementos de um array não precisam ser do mesmo tipo de dados. Dentro de um mesmo array você pode armazenar, números, strings, datas, etc.
- Typeof de um array será "object".
- O que diferencia um array de um objeto é que no array acessamos as propriedades (elementos) através de um índice numérico, e nos objetos acessamos através do nome da propriedade.
- Arrays em Javascript não podem ser indexados por Strings.
- Para saber quantos elementos um array tem use a propriedade "length".
- Arrays são indexados usando número, iniciando em 0. O primeiro elemento do array está na posição 0 (zero), o segundo elemento do array está na posição 1 (um), e assim por diante.

```
var person = [];  
person[0] = "John";  
person[1] = "Doe";  
person[2] = 46;  
var x = person.length;           // person.length retorna 3  
var y = person[0];               // person[0] retorna "John"
```

- Para percorrer os elementos de um array use o comando "for".

```
var index;  
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
for (index = 0; index < fruits.length; index++) {  
    text += fruits[index];  
}
```

Arrays já possuem algumas funções pré-definidas que você pode usar. Consulta a tabela abaixo e clique no nome da função para ver um exemplo. Para ver um lista completa de exemplos clique [nesse link](http://www.w3schools.com/js/js_array_methods.asp)<sup>26</sup>.

Propriedade	Descrição
<a href="#">length</a>	Seta ou obtém o número de elementos de um array
Método	Descrição
<a href="#">concat()</a>	Junta dois ou mais arrays
<a href="#">indexOf()</a>	Retorna a posição de um elemento dentro do array
<a href="#">join()</a>	Junta todos os elementos de um array em uma string
<a href="#">lastIndexOf()</a>	Retorna a última posição de um elemento dentro do array
<a href="#">pop()</a>	Remove o último elemento de um array e retorna esse elemento
<a href="#">push()</a>	Adiciona um novo elemento ao final do array e retorna o novo comprimento do array
<a href="#">reverse()</a>	Reverte a ordem dos elementos de um array
<a href="#">shift()</a>	Remove o primeiro elemento de um array e retorna esse elemento.
<a href="#">slice()</a>	Extraí uma parte de um array e retorna um novo array com essa parte.
<a href="#">sort()</a>	Ordena os elementos de um array

<sup>26</sup> [http://www.w3schools.com/js/js\\_array\\_methods.asp](http://www.w3schools.com/js/js_array_methods.asp)

<a href="#">splice()</a>	Adiciona o remove elementos de um array
<a href="#">toString()</a>	Converte o array uma string e retorna essa string.
<a href="#">unshift()</a>	Adiciona elementos ao início de um array e retorna o seu novo comprimento
<a href="#">valueOf()</a>	Retorna o valor primitivo de um array (equivalente a toString())

## Objeto Math

- Javascript tem um objeto chamado “Math” que lhe permite executar diversas operações matemáticas.
- Com esse objeto você pode gerar número aleatório, efetuar arredondamentos, calcular raiz quadrada, exponencial, etc.
- Veja abaixo um exemplo de como gerar um número aleatório:

```
Math.random();
```

Consulte a lista completa de propriedades e funções disponíveis o objeto Math. Clique no nome da propriedade ou função para ver um exemplo, ou consulte a referência completa [clikando aqui](#)<sup>27</sup>.

Propriedade	Descrição
<a href="#">E</a>	Retorna o número de Euler (aprox. 2.718)
<a href="#">LN2</a>	Retorna o logaritmo natural de 2 (aprox. 0.693)
<a href="#">LN10</a>	Retorna o logaritmo natural de 10 (aprox. 2.302)
<a href="#">LOG2E</a>	Retorna o logaritmo de E na base 2 (aprox. 1.442)
<a href="#">LOG10E</a>	Retorna o logaritmo de E na base 10 (aprox. 0.434)
<a href="#">PI</a>	Retorna o valor de PI (aprox. 3.14)
<a href="#">SQRT1_2</a>	Retorna a raiz quadrada de 1/2 (aprox. 0.707)
<a href="#">SQRT2</a>	Retorna a raiz quadrada de 2 (aprox. 1.414)
Método	Descrição
<a href="#">abs(x)</a>	Retorna o valor absoluto de x
<a href="#">acos(x)</a>	Retorna o arco co-seno de x em radianos
<a href="#">asin(x)</a>	Retorna o arco seno de x em radianos
<a href="#">atan(x)</a>	Retorna a arco tangente de x como um valor numérico entre -PI/2 e PI/2 radianos
<a href="#">atan2(y,x)</a>	Retorna o angulo em radianos no sentido anti horário entre o eixo X positivo e o ponto definido por y e x
<a href="#">ceil(x)</a>	Retorna o número x arredondado para cima
<a href="#">cos(x)</a>	Retorna o co seno de x (x em radianos)
<a href="#">exp(x)</a>	Retorna o valor de E elevado na potência x
<a href="#">floor(x)</a>	Retorna o número x arredondado para baixo
<a href="#">log(x)</a>	Retorna o logaritmo natural de E na base x.
<a href="#">max(x,y,z,...,n)</a>	Retorna o maior valor dentre os argumentos passados para a função

<sup>27</sup> [http://www.w3schools.com/jsref/jsref\\_obj\\_math.asp](http://www.w3schools.com/jsref/jsref_obj_math.asp)



<a href="#">min(x,y,z,...,n)</a>	Retorna o menor valor dentre os argumentos passado para a função
<a href="#">pow(x,y)</a>	Retorna o valor de x elevado na potência y
<a href="#">random()</a>	Gera um número aleatório entre 0 e 1
<a href="#">round(x)</a>	Retorna o número x arredondado para o inteiro mais próximo (para cima se a parte decimal for maior ou igual a 0.5 e para baixo se for menor que 0.5)
<a href="#">sin(x)</a>	Retorna o valor do seno de x (x em radianos)
<a href="#">sqrt(x)</a>	Retorna a raiz quadrada de x
<a href="#">tan(x)</a>	Retorna a tangente um angulo

## Eventos

- Eventos são “coisas” que acontecem aos elementos HTML. Quando usamos Javascript podemos reagir a esses eventos.
- Eventos podem acontecer em função de algo que o navegador fez, ou que o usuário fez.
- Exemplos de eventos seriam:
  - Quando a página termina de carregar
  - Quando o valor de um campo é alterado
  - Quando um botão é clicado
- Quando alguns desses eventos acontecem você pode querer executar alguma ação.
- Eventos são definidos em atributos dos elementos HTML que “disparam” esses eventos.
- Você pode programar um evento diretamente dentro do elemento HTML (não recomendado):

- ```
<button onclick="this.innerHTML=Date()">Que hora é?</button>
```

- Ou você pode chamar funções para serem executadas (aconselhado)

- ```
<button onclick="mostrarHora()">Que hora é?</button>
```

- Alguns eventos mais comuns:

Evento	Descrição
<a href="#">onchange</a>	Quando um elemento HTML é alterado.
<a href="#">onclick</a>	Quando o usuário clica em um elemento.
<a href="#">onmouseover</a>	Quando o usuário passa o mouse sobre um elemento.
<a href="#">onmouseout</a>	Quando o usuário tira o mouse de um elemento.
<a href="#">onkeydown</a>	Quando o usuário pressiona uma tecla.
<a href="#">onload</a>	Quando a página termina de carregar.

Para uma lista completa dos eventos disponíveis consulte:

[http://www.w3schools.com/jsref/dom\\_obj\\_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)

### Exercício: Triângulo



- Monte uma página HTML com três inputs texto e um botão.
- Ao clicar no botão o programa deve mostrar um alerta informando se o triângulo formado pelos três valores é equilátero (todos iguais) isósceles (dois lados iguais) ou escaleno (os três são diferentes).

### Exercício: Fizz Buzz



- Faça uma programa que liste os número de 1 a 100
- Para os múltiplos de 3, ao invés do número, escreva "Fizz".
- Para os múltiplos de 5, ao invés do número escreva "Buzz".
- Para os que são múltiplos de 3 e 5, ao invés do número escreva "FizzBuzz".

### Exercício: Báskara



- Monte uma página que tenha três campos correspondentes aos valores de a, b e c de uma fórmula de báskara.
- Calcule as duas raízes possíveis e mostre o resultado em elemento HTML.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

### Exercício: Fibonacci



- Monte uma página HTML com um campo texto.
- O programa deve gerar e mostrar em um alerta o enésimo termo da série de fibonacci (n será digitado no campo texto).

### Exercício: Retângulo



- Crie um objeto chamado retângulo que tenha dois atributos referentes a altura e largura.
- Crie um método que calcule a área e mostre em um alert.

### Exercício: Fatorial



- Monte uma página HTML com um campo texto.
- Calcule e mostre o fatorial do número informado.

### Exercício: Ordenação



- Monte uma página HTML com um campo texto e um botão chamado "Ordenar".
- O usuário irá digitar dentro desse campo texto n números separados por ",".
- Ao clicar em "Ordenar", ordene os valores do campo texto e atualize o campo mostrando os valores ordenados novamente.
- Você mesmo deve implementar o algoritmo de ordenação. Não usar a função sort().

### Exercício: Idade



- Monte uma página HTML com um campo texto para que o usuário digite a data de nascimento.
- Calcule a idade do usuário e mostra em um alerta a se é maior de idade ou menor, e a sua idade.