

# HTML5

- Novos atributos de formulários, como `number`, `date`, `time`, `search`, e `range`.



- Novos elementos de semântica como `<header>`, `<footer>`, `<section>` e `<nav>`

- Novos elementos gráficos e de multimédia



# HTML5



**HTML**

```
<div id="header">
```

```
    <div id="nav">
```

```
        <div class="article">
```

```
            <div class="section">
```

```
        <div id="footer">
```

**HTML5**

```
<header>
```

```
    <nav>
```

```
        <article>
```

```
            <section>
```

```
        <aside>
```

```
    <footer>
```



# HTML5

HTML5

<header>

<nav>

<article>

<section>

<aside>

<footer>

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <style type="text/css">
    header, nav, div, footer, article {
      border: 1px solid;
      background-color: gray ;
    }

    p {
      text-align: center;
    }
  </style>
</head>

<body>
  <header>
    <h2><p>Html 5 &lt;header&gt;</p></h2>
  </header>
  <nav>
    <ul>
      <li> &lt;nav&gt; </li>
    </ul>
  </nav>
  <article>
    <p>&lt;article&gt;</p>
    <section style="float:left;width:50%;border:1px solid">
      <p>&lt;section&gt;</p>
    </section>
    <aside style="float:right;width:49%;border:1px solid">
      <p>&lt;aside&gt;</p>
    </aside>
  </article>
  <footer>
    <p>&lt;footer&gt;</p>
  </footer>
</body>

</html>
```

# Tags Semântica

Article:	Aside:	Footer:	Header:	Nav:	Section:
Este é um elemento autocontido. Pode ser um post, um artigo entre outros!	Este elemento indica um seção do documento com conteúdo relacionado. Pode ser uma sidebar!	Este elemento representa um rodapé para o elemento pai, ou mais próximo a ele.	Este é um elemento de cabeçalho. Pode ser utilizado para logo, navegação, pesquisa e outros.	Este elemento indica uma navegação. Pode ser utilizado para apontar os principais blocos de navegação do site!.	Este elemento representa uma seção genérica de conteúdo do site. Realizando um agrupamento temático.



# CSS Box Model

O CSS é uma caixa envolvendo todo elemento HTML

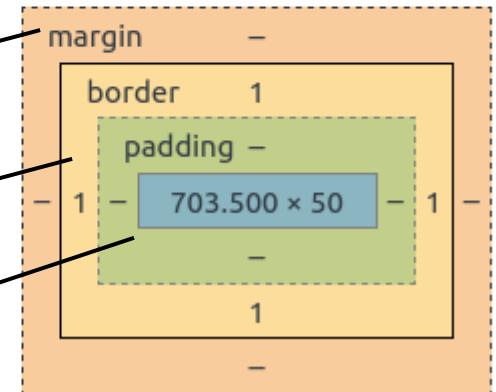
```
<section style="float:left; width:50%; border:1px solid">  
  <p><section></p>  
  </section>  
<aside style="float:right; width:49%; border:1px solid">  
  <p><aside></p>  
  </aside>
```

```
element.style {  
  float: left;  
  width: 50%;  
  border: 1px solid;  
}  
  
article, aside, footer, header, hgroup, main, nav,  
section {  
  display: block;  
}
```

margem limpa área ao redor da borda

borda ao redor do conteúdo e "enchimento"

padding limpa área ao redor do conteúdo é transparente



# CSS Sintaxe

```
<style type="text/css">  
  
header, nav, div, footer, article {  
    border: 1px solid;  
    background-color: gray ;  
}  
  
p {  
    text-align: center;  
    color: blue; font-size: 12px  
}  
  
#id_section {  
    float:left;  
    width:50%;  
    border:1px solid  
}  
  
.class_aside {  
    float:right;  
    width:49%;  
    border:1px solid  
}  
  
</style>
```

O CSS define seletores e aplica valores às propriedades.  
seletor { atributo:valor; ... }

Seletores em grupo, vários elementos Html

Seletor de Elemento um elemento Html

Seletor de id aplicado ao atributo id do elemento Html

```
<section id="id_section">  
    <p>&lt;section&gt;</p>  
</section>
```

Seletor de classe aplicado ao atributo class do elemento  
Html

```
<aside class="class_aside">  
    <p>&lt;aside&gt;</p>  
</aside>
```



# CSS Container e Background Imagem



```
header{  
background-image: url(..../imagens/header-back.jpg);  
clear: both;  
opacity: 0.8;  
  
padding: 20px 0;  
margin-bottom: 10px;
```

```
.container{  
position: relative ;  
margin: 0 auto;  
width: 1200px;
```

```
body{  
background-image: url(..../imagens/background.jpg);
```

```
footer{  
background-image: url(..../imagens/fundo-rodape2.png);
```



# Elementos de Cabeçalho



Header - Define um grupo de títulos ou o cabeçalho de uma determinada seção.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link href="css/reset.css" rel="stylesheet">
<link href="css/estilo.css" rel="stylesheet">

</head>
<body>
<header class="container">
    

    <nav class="menu-opcoes">
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Carrinho</a></li>
            <li><a href="#">Cadastro</a></li>
            <li><a href="#">Login</a></li>
            <li><a href="#">Ajuda</a></li>
        </ul>
    </nav>
</header>

</body>
</html>
```

Nav - Define um grupo ou bloco de links de navegação.



# Elementos de Rodapé

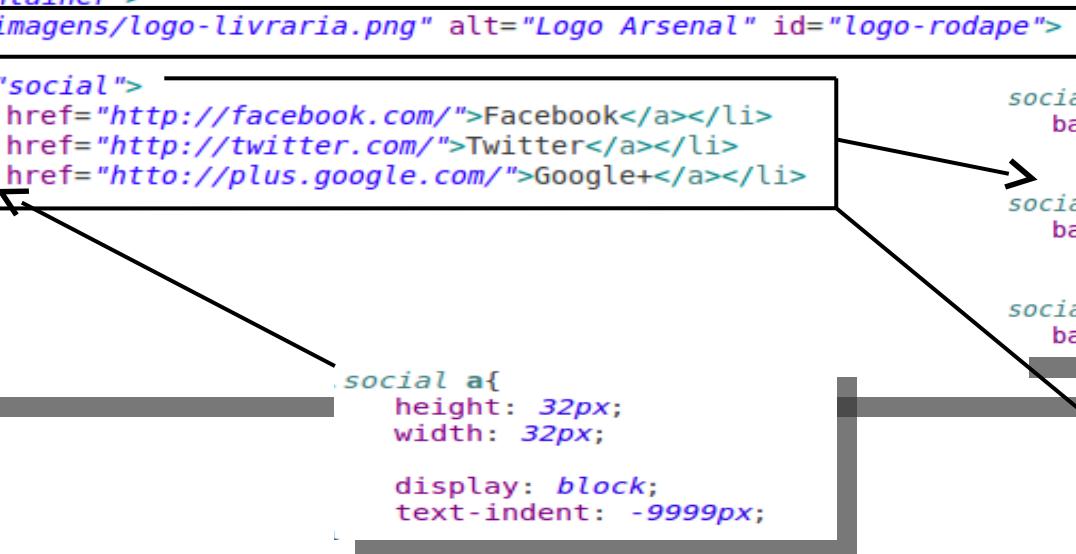
```
<!DOCTYPE html>
@<html>
@<head>
<meta charset="UTF-8">
<link href="css/reset.css" rel="stylesheet">
<link href="css/estilo.css" rel="stylesheet">

</head>
@<body>
@<footer>
@<div class="container">
![Logo Arsenal](imagens/logo-livraria.png)
```

**Footer -**  
Define o  
rodapé das  
seções ou da  
página.



www.3way.com.br



```
social a[href*="facebook.com"]{
    background-image: url(..../imagens/facebook.png);

social a[href*="twitter.com"]{
    background-image: url(..../imagens/twitter.png);

social a[href*="plus.google.com"]{
    background-image: url(..../imagens/googleplus.png);
```



# Barra Navegação



```
<!DOCTYPE html>
<html>
<body>
<div class="container destaque">
    <section class="busca">
        <h2>Busca</h2>
        <form action="Pesquisa" id="form-busca">
            <input type="search" name="titulo" id="q">
            <input type="image" src="imagens/busca.png">
        </form>
    </section>

    <section class="menu-departamentos">
        <h2>Departamentos</h2>
        <nav>
            <ul>
                <li><a href="#">Livros</a>
                    <ul>
                        <li><a href="#">Auto-ajuda</a></li>
                        <li><a href="#">Bibliografias</a></li>
                        <li><a href="#">Teen</a></li>
                        <li><a href="#">Romance</a></li>
                        <li><a href="#">Ficcao</a></li>
                        <li><a href="#">Gibis e HQs</a></li>
                    </ul>
                </li>
                <li><a href="#">Filmes</a></li>
                <li><a href="#">Games</a></li>
                <li><a href="#">Musica</a></li>
                <li><a href="#">Acessorios</a></li>
            </ul>
        </nav>
    </section>
</div>
</body>
</html>
```

Section - Define um grupo ou bloco de por seção de um assunto específico

# Barra Lateral



```
<section class="menu-departamentos">
  <h2>Departamentos</h2>
  <nav>
    <ul>
      <li><a href="#">Livros</a>
        <ul>
          <li><a href="#">Auto-ajuda</a></li>
          <li><a href="#">Bibliografias</a></li>
          <li><a href="#">Teen</a></li>
          <li><a href="#">Romance</a></li>
          <li><a href="#">Ficcao</a></li>
          <li><a href="#">Gibis e HQs</a></li>
        </ul>
      </li>
      <li><a href="#">Filmes</a></li>
      <li><a href="#">Games</a></li>
      <li><a href="#">Musica</a></li>
      <li><a href="#">Acessorios</a></li>
    </ul>
  </nav>
</section>
</div>
</body>
</html>
```

busca,  
menu-departamentos{  
 float: left;

menu-departamentos li ul{  
 display: none;

menu-departamentos li:hover ul{  
 display: block;

Float - O painel flutua ao lado de uma imagem ou texto

Display: none - A segunda lista não irá aparecer

Hover - A lista irá aparecer somente se for invocada  
ao passar o mouse em cima de "Livros".



# Formulário Novos Elementos

```
<form>
email:   <input type="email" multiple required autofocus placeholder="meu@dominio.com" />
url:    <input type="url" />
numero: <input type="number" />
data hora: <input type="date"/>
          <input type="time"/>
busca:  <input type="search"/>
range:   <input type="range" max="100" min="80" step="2" title="Sua pontuacao" name="rscore"/>
          <output name="score">90</output>
color:   <input type="color"/>

spellcheck:<input type="text" spellcheck="false" placeholder="Com Spellcheck">
           <input type="text" spellcheck="true" placeholder="Sem Spellcheck">

datalist: <input type="text" id="somenames" list="fullList" />
          <datalist id="fullList">
            <option value="Brasil">
            <option value="Alemnha">
            <option value="Peru">
          </datalist>
</form>
```

email:

url:

numero:

data hora:

busca:

range:

color:

spellcheck:  Com Spellcheck  Sem Spellcheck

datalist:   
Brasil  
Alemnha



# Bootstrap CSS e Responsividade

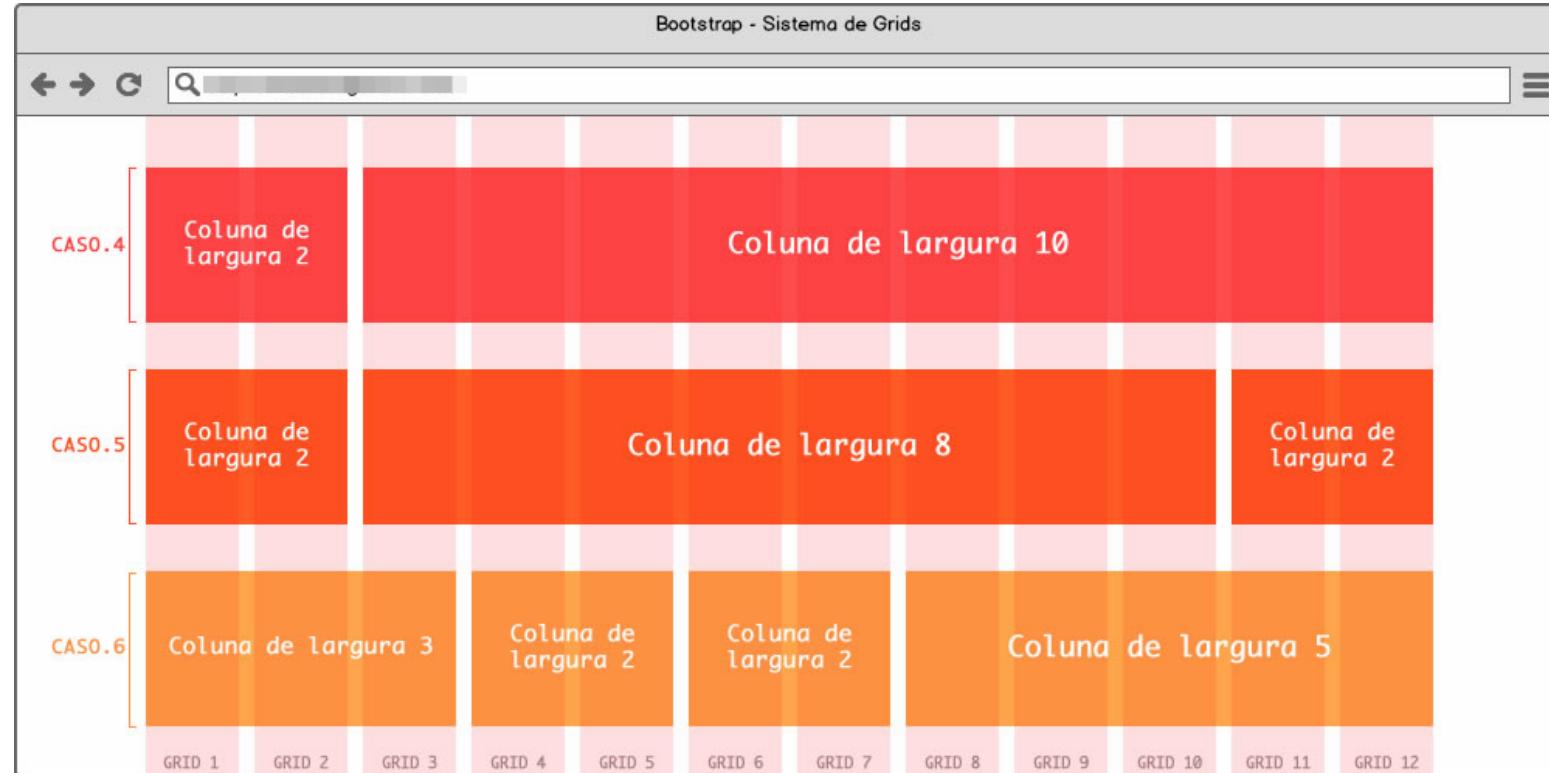
ViewPort - adaptar

área disponível para

dispositivos móveis

```
<meta name="viewport" content="width=device-width" />
```

Bootstrap Grid  
System



# Bootstrap – Estrutura Básica

Cria um container

Define uma  
nova linha

	Dispositivos extra pequenos Telefones (<768px)	Dispositivos pequenos Tablets (≥768px)	Dispositivos médios Desktops (≥992px)	Dispositivos grandes Desktops (≥1200px)
Comportamento do grid	Horizontal em todos os momentos	Colapsa ao princípio, horizontal acima de breakpoints		
Largura do container	nenhum (auto)	750px	970px	1170px
Prefixo de classe	.col-xs-	.col-sm-	.col-md-	.col-lg-
número de colunas	12			
Largura da coluna	Auto	~62px	~81px	~97px

```
<div class="container">  
  <div class="row">  
    <div class="col-*-*"></div>  
  </div>  
  <div class="row">  
    <div class="col-*-*"></div>  
    <div class="col-*-*"></div>  
    <div class="col-*-*"></div>  
  </div>  
  <div class="row">  
    ...  
  </div>  
</div>
```

Define o número

de colunas ocupadas



# Formulário Bootstrap CSS

Cadastro

Nome Completo

Usuário ! Preencha este campo.

Senha

Repita a senha

```
<link href="css/bootstrap.css" rel="stylesheet">

<div class="panel panel-primary painel-cadastro">
<div class="panel-heading">Cadastro</div>
<div class="panel-body">

<form method="post" action="EditarCliente?acao=cadastrar">
<div class="row">

${mensagem}
<fieldset>

<div class="form-group">
<label for="nome">Nome Completo</label>
<input type="text" class="form-control" id="nome" name="nome" autofocus required>
</div>

<div class="form-group">
<label for="login">Usuário</label>
<input type="text" class="form-control" id="login" name="login" required>
</div>

<div class="form-group">
<label for="senha1">Senha</label>
<input type="password" class="form-control" id="senha1" name="senha1" required>
</div>

<div class="form-group">
<label for="senha2">Repita a senha</label>
<input type="password" class="form-control" id="senha2" name="senha2" required>
</div>

</fieldset>
<input type="submit" class="btn btn-primary" value="Confirmar">
</form>
```

# JavaScript - Declaração e Atribuições de Valores

Variáveis não precisam definir tipo de dados é 'tipos dinâmicos'

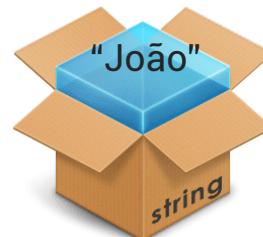
```
<script>
    var nome = "João";
    var altura = 1.80;
    idade = 30; //O que acontece aqui? Por que não tem 'var' nessa linha?
    var maiorDeIdade = true;
</script>
```

Atribuição  
→ `var nome = "João";`

Valor  
→ "João"

Nome da variável  
→ nome

Comando de declaração  
→ var



# Operadores de Comparaço

```
var idade = 30;
if (idade > 18) {
  alert("Maior de idade");
}
```

Operador	Descrição	Comparação	Retorna
==	Igual a	x == 8	false
		x == 5	true
		x == "5"	true
====	Estritamente igual	x === 5	true
		x === "5"	false
!=	diferente	x != 8	true
!==	Estritamente não igual	x !== 5	false
		x !== "5"	true
		x !== 8	true
>	Maior que	x > 8	false
<	Menor que	x < 8	true
>=	Maior ou igual	x >= 8	false
		x <= 8	true



# Tomada de Decisão – else if

```
var idade = prompt('Digite sua idade');
if (idade < 12) {
    alert('Criança.');
}
else if (idade < 14) {
    alert('Pré-adolescente.');
}
else if (idade < 18) {
    alert('Adolescente.');
}
else if (idade < 60) {
    alert('Adulto.');
}
else { alert('Idoso.')}
```

# Tomada de Decisão - switch

```
var fruta = prompt("qual fruta ?");
switch (fruta) {
    case "Laranja":
        document.write("O quilo da laranja está R$0,59.<br>");
        break;
    case "Maçã":
        document.write("O quilo da maçã está R$0,32.<br>");
        break;

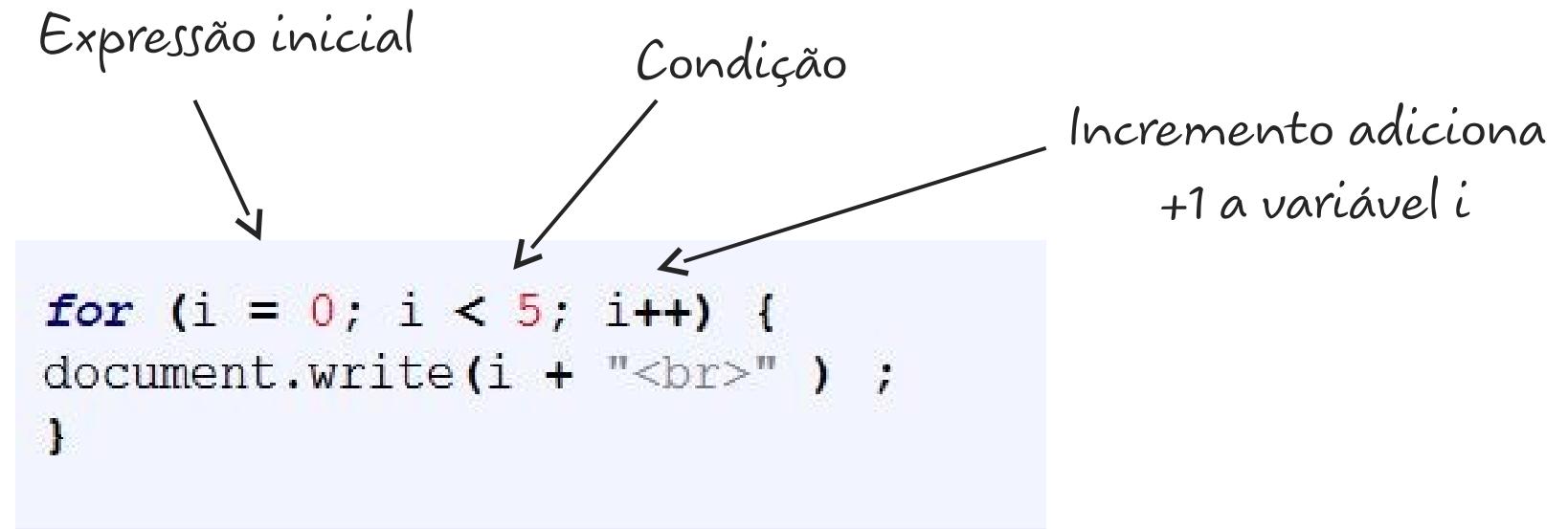
    default:
        // se não for nenhum dos cases acima, então:
        console.log("Desculpe, não temos " + fruta + ".<br>");
}
```

o programa faz a correspondência do valor com case "Banana", "Maçã" ou  
default executa a declaração associada



## Declaração for

Enquanto (condição for verdadeira) { código que será executado}



- Quando o bloco de código é encerrado, o laço for incrementa +1 a variável i.
- Repetimos essas etapas até que a condição  $i < 5$  não seja mais verdadeira.



## Declaração for ... in

Criando objeto do tipo array

Propriedades Do objeto

Valor das propriedades

```
var Moto = {  
    Marca: "Honda",  
    Modelo: "Hornet",  
    Cor: "Preta"  
}  
  
for ( descricao in Moto){  
    document.write(descricao + " = " + Moto[descricao] + "<br>");  
}
```

O laço for in percorre todas as propriedades de um objeto



## Declaração While

- Enquanto (condição for verdadeira) { código que será executado }

The diagram illustrates a while loop structure. It starts with a variable declaration `n = 0;`, followed by the loop header `while (n < 10) {`. Inside the loop, there is a code block `document.write(n + "<br>");` and an increment statement `n++;`. The loop concludes with a closing brace `}`. Handwritten annotations with arrows point to specific parts: an arrow from the word "Condição" points to the condition part of the loop header; another arrow from the phrase "Código que Será executada" points to the code block `document.write(n + "<br>");`.

```
n = 0;  
while (n < 10) {  
    document.write(n + "<br>");  
    n++;  
}
```

Condição

Código que Será executada



## Declaração Do ... While

- A instrução será executada uma vez antes da condição ser verificada
- Caso a condição seja verdadeira, então o laço será executado novamente

```
n = 0;  
do {  
    document.write(n + "<br>");  
    n++;  
}  
while (n < 10)
```

Condição



# Criando um Array e Acessando posições

```
//Criando Array  
var frutas = new Array("Laranja", "Maça", "Banana", "Uva", "Açai");  
var num = [4, 46, 34, 2, 1, 9];  
var mixed = [34, "doce", 11, "azul"];
```

```
//Acessando posições dos Array  
document.write(frutas[0] + "<br>");  
document.write(num[1] + "<br>");  
document.write(mixed[3] + "<br>");
```



# Funções

Definindo  
função Mostrar

Chamando Função  
Mostrar

```
function multiplicar(numero) {  
    document.write(numero * numero);  
}  
  
multiplicar(5)
```



# Funções

Definindo → função verificar

```
var n = prompt("Digite um numero");  
  
function verificar(n) {  
    if(n % 2 == 0){  
        alert("Par");  
    } else {  
        alert("Impar");  
    }  
}  
  
verificar(n);
```

Parâmetros

Chamando Função  
Mostrar



# Criando um Objeto

Instância do Object

Propriedades

Método

```
var carro = new Object();
carro.marca= 'Aston Martin';
carro.modelo= 'Dbs';
carro.velocidade= '250 km/h';
carro.cor= 'Prata';
carro.toString = function(){
    return "marca:"+ this.marca
    +" velocidade:"+this.velocidade;
}
```

Instância do Object

Propriedades

Método

```
var carro = {
    marca: 'BMW',
    modelo: 'x6',
    velocidade: '275 km/h',
    cor: 'Branco',
    toString : function(){
        return "marca:"+ this.marca
        +" velocidade:"+this.velocidade;
    }
}

console.log(carro.toString());
```



# Criando um Objeto Personalizado

Uma "classe" com

construtor

Propriedades

Método

Instância do Object

```
var Carro = function(nome){  
    return {  
        marca: nome,  
        modelo: 'XE',  
        velocidade: '375 km/h',  
        cor: 'Preto',  
        toString : function(){  
            return "marca:"+ this.marca  
            +" velocidade:"+this.velocidade;  
        }  
    };  
};  
  
var carro = new Carro('Jaguar');  
console.log(carro.toString());
```

Javascript é um linguagem baseado em protótipos não tem uma declaração class



# JavaScript – Eventos DOM

```
<section class="busca">
  <h2>Busca</h2>
  <form onSubmit=" return buscarProduto(titulo)">
    <input type="search" name="titulo">
    <input type="image" src="imagens/busca.png">
  </form>
</section>
```

```
<script src="js/busca.js"></script>
```

```
function buscarProduto(titulo){
  var livros = ["Branca de Neve", "A Bela Moça", "O Guia", "Dr. Ozzy", "Pequeno Irmão"];
  var filmes = ["Deadpool", "Godzilla", "Lagoa Azul", "Loucademia de Polícia"];
  var musicas= ["ZZ Top", "Black Sabbath", "Lorranna Santos"];
  var games= ["Mario Kart", "Fifa 17", "Battlefield 4", "Undertale", "SuperHot"];
  if(titulo.value == ""){
    alert("Busca Vazia! Procure entre nossas seleções de LIVROS, FILMES, GAMES e MÚSICAS! ");
  }
  else if(titulo.value=="livros" || titulo.value=="livro"){
    window.alert(livros);
  }
  else if(titulo.value=="filmes"){
    window.alert(filmes);
  }
  else if(titulo.value=="musicas"){
    window.alert(musicas);
  }
  else if(titulo.value=="games"){
    window.alert(games);
  }
  else{
    window.alert("Procure entre nossas seleções de LIVROS, FILMES e MÚSICAS!");
  }
}
```

localhost:8080 diz:

Busca Vazia! Procure entre nossas seleções de LIVROS,  
FILMES, GAMES e MÚSICAS!

OK

localhost:8080 diz:

Branca de Neve,A Bela Moça,O Guia,Dr. Ozzy,Pequeno  
Irmão

OK



# jQuery – Função \$

Garante que  
jQuery só roda  
após carga  
completa da  
página

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").hide(); <----- Seletor de Elemento  
    });  
});  
  
$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide(); <----- Seletor de Id  
    });  
});  
  
$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide(); <----- Seletor de Classe  
    });  
});
```



# jQuery – Outros Exemplos

## Outros Seletores

```
$('div > p:first'); // o primeiro elemento <p> imediatamente filho de  
um <div>  
  
$('input:hidden'); // todos os inputs invisíveis  
  
$('input:selected'); // todas as checkboxes selecionadas  
  
$('input[type=button]'); // todos os inputs com type="button"  
  
$('td, th'); // todas as tds e ths
```

---



# jQuery – Outros Exemplos

## Manipulando Eventos

```
$("#p1").hover(  
    function(){  
        alert("Você está sobre p1!");  
    },  
    function(){  
        alert("Tchau! Você deixou p1!");  
    }  
);  
  
$("input").focus(function(){  
    $(this).css("background-color", "#cccccc");  
});  
  
$("p").on({  
   mouseenter: function(){  
        $(this).css("background-color", "lightgray");  
    },  
   mouseleave: function(){  
        $(this).css("background-color", "lightblue");  
    },  
    click: function(){  
        $(this).css("background-color", "yellow");  
    }  
});
```

