

# HTML, CSS e JavaScript (Parte 2)

Esdras Lins Bispo Jr.  
bispojr@ufg.br

Física para Ciência da Computação  
Bacharelado em Ciência da Computação

28 de agosto de 2019

# Plano de Aula

- 1 Revisão
  - HTML5
  - JavaScript
- 2 Continuação...
- 3 Herança, Construtores e Protótipos
  - Herança
  - Construtores
  - Protótipo

# Sumário

- 1 Revisão
  - HTML5
  - JavaScript
- 2 Continuação...
- 3 Herança, Construtores e Protótipos
  - Herança
  - Construtores
  - Protótipo

# HTML5

## O que é?

HTML5 é a nova “encarnação” do padrão HTML.  
Ele traz novas capacidades para o navegador.

## Um exemplo mínimo em HTML5

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Um documento minimo em HTML5</title>
6   </head>
7   <body>
8     <h1>Boa noite , HTML5!</h1>
9   </body>
10 </html>
```

# HTML5

## Elemento Canvas

Permite renderizar gráficos e animação no navegador sem necessitar de uso de plugins externos (como *FlashPlayer*).

...basta incluir esta linha no corpo do documento

```
1 <canvas id="canvas" width="700" height="500">  
2 </canvas>
```

# HTML5

## Exemplo

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Um documento minimo em HTML5</title>
6   </head>
7   <body>
8     <canvas id="canvas" width="700" height="500">
9     </canvas>
10  </body>
11 </html>
```

# HTML5

## Pôr estilo no Canvas

Como qualquer elemento HTML. Você pode, por exemplo, ligar o arquivo HTML a um arquivo CSS externo.

...basta incluir esta linha no head do documento

```
1 <link rel="stylesheet" href="style.css">
```

# HTML5

## Exemplo

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Um documento minimo em HTML5</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <canvas id="canvas" width="700" height="500">
10    </canvas>
11  </body>
12 </html>
```



# Adicionando JavaScript

## Ligando o arquivo .js ao corpo do HTML

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Um documento minimo em HTML5</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <canvas id="canvas" width="700" height="500">
10    </canvas>
11    <script src="bolaQuicando.js"></script>
12  </body>
13 </html>
```

# Depurando no JavaScript

Via Google Chrome...

Tecle `Ctrl + Shift + J`.

Teste...

- `2 + 3`
- `console.log("Hello World")`
- `a=2; b=3; console.log(a*b);`

# Objetos em JavaScript

## Considerações iniciais...

Em JavaScript, os objetos são as unidades básicas.

## O que é um objeto?

Uma coleção de propriedades

## O que é uma propriedade?

É uma associação entre um nome e um valor.

## O que é um valor?

Pode ser um número inteiro, uma cadeia, uma função, entre outros.



# Objetos predefinidos

## Alguns Objetos predefinidos...

- String
- Array
- Date
- Math...

## Criando novos objetos...

- `obj = new Object();`  
ou
- `obj = {};`



# Atribuindo propriedades

## Exemplo 1: Utilizando ponto...

```
1 obj = new Object();  
2 obj.nome = "Primeiro objeto";  
3 obj.length = 20;  
4 console.log(obj.nome, obj.length);
```

## Exemplo 2: Utilizando colchetes...

```
1 obj = {};  
2 obj["nome"] = "Primeiro objeto";  
3 obj["length"] = 20;
```

# Funções e Métodos

## O que é uma função?

É um bloco de código que é executado quando chamado pelo seu nome.

## Sintaxe para a definição de função sem argumentos

```
1 function nomeDaFuncao() {  
2     //bloco de codigo  
3 }
```

# Funções e Métodos

## Sintaxe para a definição de função com argumentos

```
1 function nomeDaFuncao(arg1 , arg2){  
2     //bloco de codigo  
3 }
```

## A função pode retornar valores

```
1 function multiplicar(x , y){  
2     return x*y;  
3 }
```

# Sumário

- 1 Revisão
  - HTML5
  - JavaScript
- 2 Continuação...
- 3 Herança, Construtores e Protótipos
  - Herança
  - Construtores
  - Protótipo



# Funções e Métodos

Atribuindo uma função como propriedade de um objeto...

```
1 nomeDoObjeto.nomeDaPropriedade = nomeDaFuncao;
```

# Funções e Métodos

Atribuindo uma função como propriedade de um objeto...

```
1 nomeDoObjeto.nomeDaPropriedade = nomeDaFuncao;
```

Exemplo

```
1 obj.multiplicar = multiplicar;
```

# Sumário

- 1 Revisão
  - HTML5
  - JavaScript
- 2 Continuação...
- 3 Herança, Construtores e Protótipos
  - Herança
  - Construtores
  - Protótipo

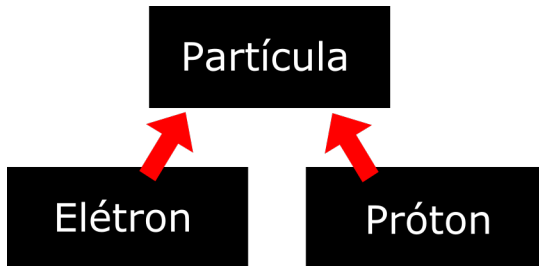
# Herança

Partícula

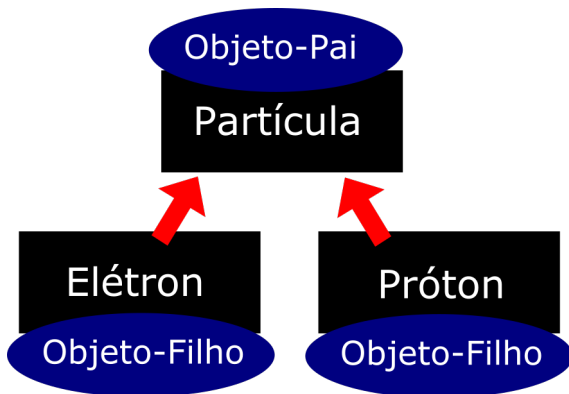
Elétron

Próton

# Herança



# Herança



# Construtor

## Objeto a partir de uma função

É possível construir um objeto a partir de uma “função-modelo”.

```
1 function Particula(pnome){  
2     this.nome = pnome;  
3     this.mover = function(){  
4         console.log(this.nome + " em movimento!");  
5     };  
6 }
```

# Construtor

## Objeto a partir de uma função

```
1 partícula1 = new Particula("eletron");  
2 console.log(partícula1.nome); //exibe "eletron"  
3 partícula1.mover();  
4     //exibe "eletron em movimento"
```



# Construtor

## Objeto a partir de uma função

```
1 partícula1 = new Particula("eletron");  
2 console.log(partícula1.nome); //exibe "eletron"  
3 partícula1.mover();  
4 //exibe "eletron em movimento"
```

## Construtor...

Se uma função é utilizada intencionalmente para criar novos objetos, a chamamos de **construtor**.



# Protótipo

## Adicionando propriedades...

Após declarado, é possível adicionar propriedades a um construtor.

```
1 Particula.prototype.massa = 1;  
2 Particula.prototype.parar =  
3   function(){console.log("Eu parei.");};
```

# Protótipo

## Adicionando propriedades...

Após declarado, é possível adicionar propriedades a um construtor.

```
1 Particula.prototype.massa = 1;  
2 Particula.prototype.parar =  
3     function(){console.log("Eu parei.");};
```

## Para novos objetos...

```
1 particula2 = new Particula("proton");  
2 console.log(particula2.massa); //exibe 1
```

# Protótipo

## Entretanto...

```
1 partícula3 = new Particula("neutron");  
2 partícula3.massa = 2; //propriedade com valor 2  
3 console.log(Particula.prototype.massa); //exibe 1
```

# Protótipo

## Entretanto...

```
1 partícula3 = new Particula("neutron");  
2 partícula3.massa = 2; //propriedade com valor 2  
3 console.log(Particula.prototype.massa); //exibe 1
```

## Necessário ter em mente...

A propriedade prototype modifica o objeto-pai.

# Protótipo

## Algo interessante...

É possível adicionar propriedades apenas para o objeto-filho.

```
1 partícula4 = new Partícula("neutrino");  
2 partícula4.spin = 0; //exibe 0
```

# Exemplo...

## Objeto Bola

```
1 function Bola (raio , cor) {  
2   this.raio = raio;  
3   this.cor = cor;  
4   this.x = 0;  
5   this.y = 0;  
6   this.vx = 0;  
7   this.vy = 0;  
8 }
```

# Exemplo...

## Objeto Bola

```
1 Bola.prototype.desenhar = function (contexto) {  
2     contexto.fillStyle = this.cor;  
3     contexto.beginPath();  
4     contexto.arc(this.x, this.y,  
5                 this.raio, 0, 2*Math.PI, true);  
6     contexto.closePath();  
7     contexto.fill();  
8 };
```



# Exemplo...

## HTML Canvas arc

A propriedade `arc()` cria um arco (utilizado para criar círculos ou parte de círculos).

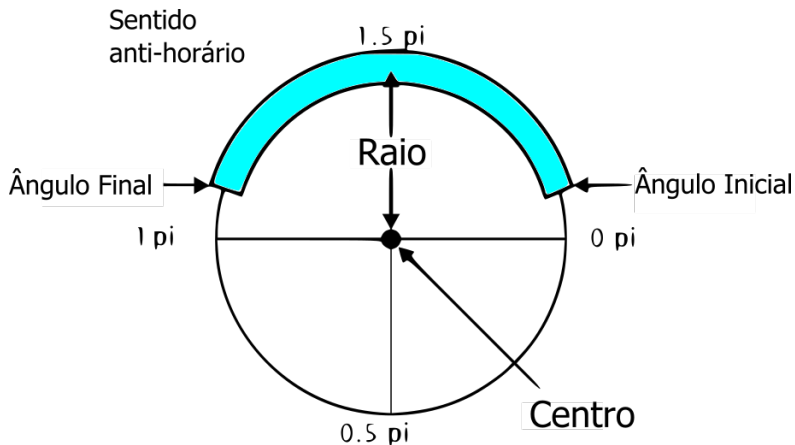
## Sintaxe

```
arc(x, y, raio, angIni, angFin, antiHor);
```

## Argumentos

- `x` e `y`: coordenadas do arco;
- `raio`: raio do arco;
- `angIni` e `angFin`: ângulos inicial e final do arco (em radianos);
- `antiHor`: valor booleano para o sentido anti-horário.

# Exemplo...



# Exemplo...

## Objeto Bola

```
1 var canvas = document.getElementById('canvas');  
2 var contexto = canvas.getContext('2d');  
3 var bola = new Bola(50, '#0000ff');  
4 bola.x = 100;  
5 bola.y = 100;  
6 bola.desenhar(contexto);
```

# HTML, CSS e JavaScript (Parte 2)

Esdras Lins Bispo Jr.  
bispojr@ufg.br

Física para Ciência da Computação  
Bacharelado em Ciência da Computação

28 de agosto de 2019