



Introdução



Classe



Construtor



Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Programação Orientada a Objetos

Introdução

- Abstração que cria modelos baseados no mundo real.
- Coleção de objetos em cooperação
- Oposição à execução de série de funções
- Cada objeto é capaz de receber mensagens, processar dados e envio de mensagens para outros objetos.
- Destina a promover uma maior flexibilidade e facilidade de manutenção





Programação Orientada a Objetos

Introdução

- Comportamento das instâncias é definido em classes através de uma coleção de métodos e atributos.
- A adição de novas funcionalidades a uma classe é feito através da extensão de uma classe já existentes
- Exige que para cada objeto exista uma classe com a sua definição.
- Comportamento → definido pela classe
- Estado → representado pelas instâncias





Programação Baseada em Protótipo

Introdução

- Estilo de programação orientada a objeto onde não existem classes.
- Não existem uma descrição que abstrai um conjunto de objetos com características, similares nem tipos
- Existem apenas objetos, que seguem um protótipo (um molde), que pode ser alterado em tempo de execução.
- Cada novo objeto é criado clonando-se objetos existentes chamados de protótipos.





Programação Baseada em Protótipo

Introdução

- Um objeto é um mapa de chaves e valores, que associa o nome (atributo) para o valor (estado).
- Funções são objetos e podem ser atribuídas a chaves nos objetos, criando métodos
- Alteração da estrutura dos objetos são realizadas em tempo de execução
- Interpretadas com tipagem dinâmica





Introdução



Classe



Construtor



Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano

ECMAScript 6 (ES6)

Classe

- As Classes são, de fato, "funções especiais" onde no seu escopo pode ser definido membros métodos ou construtores.
- Declarações de Classe

```
class Poligono{ }
```

- Expressões de Classe sem nome

```
var Poligono = class { };
```

- Expressões de Classe nomeada

```
var Poligono = class Poligono{ };
```





ECMAScript 5 (ES5)



Classe

→ Declarações de Classe

```
function Poligono() { }
```

→ Expressões de Classe sem nome = classe anônima (único uso)

```
var Poligono = function() { };
```

→ Expressões de Classe nomeada

```
var Poligono = function Poligono() { };
```





Introdução



Classe



Construtor



Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano

Construtor

Construtor

- Usado pelo operador new para criar um objeto clone criado pela classe.
- Só pode existir **um método** com o nome "constructor" dentro da classe que pode ter parâmetros com valores padrões.
- Um construtor pode usar a palavra-chave super para chamar o construtor da classe pai.

```
class Poligono {  
  constructor() { console.log("obj clonado"); } }  
}
```

ES6

```
function Poligono() { console.log("obj clonado"); }  
}
```

ES5

Construtor

Construtor

- Usado pelo operador new para criar um objeto clone criado pela classe.
- Só pode existir **um método** com o nome "constructor" dentro da classe que pode ter parâmetros com valores padrões.
- Um construtor pode usar a palavra-chave super para chamar o construtor da classe pai.

```
class Poligono {  
  constructor() { console.log("obj clonado"); } }  
}
```

ES6

```
function Poligono() { console.log("obj clonado"); }  
}
```

ES5

Construtor

operador new

- Cria um novo objeto → 'clonado'
- Define a propriedade 'prototype' e vincula ao clone do objeto.
- Retorna o objeto recém-criado
- Exceto quando função construtora retorna valor
- Quando uma função não incluir a declaração de retorno, o compilador inserirá 'return this;'
- O objeto clone retornado é um mapa de chaves e valores, que associa nome (atributo) → valor (estado).



operador new

Construtor

- Chamadas construtoras de funções em ES6

```
class Poligono{ }  
var p = new Poligono()
```

ES6

- Chamadas construtoras de classes em ES5

```
function Poligono() { }  
var p = new Poligono()
```

ES5

- O objeto recém-criado é atribuído como valor à variável





Introdução



Classe



Construtor



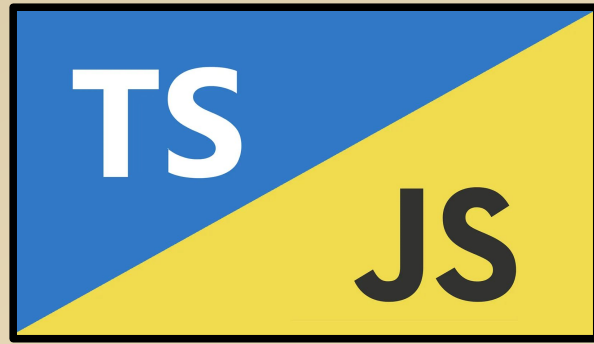
Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano

Propriedades e prototype

Propriedade

- Propriedade de um objeto pode ser explicada como uma variável que é ligada ao objeto e define suas características
- Acessa as propriedades de um objeto com uma simples notação de ponto.
- Qualquer objeto recebe todas as propriedades do objeto referido pela propriedade prototype.
- Não existe obrigatoriedade todos objetos terem as mesmas características, mas sempre serão clonadas no operador new
- Todas instâncias adquirem nova propriedade ao designar uma propriedade e valor





Propriedade

Propriedades ES6

```
class Ponto {  
  constructor(){  
    this.x=10;  
    this.y=20; } }  
var p1 = new Ponto();  
console.log(`p1 x = ${p1.x}`);  
console.log(`p1 y = ${p1.y}`);  
//undefined  
console.log(`p1 z = ${p1.z}`);  
var p2 = new Ponto();  
p2.z = 50;  
console.log(`p2 x = ${p2.x}`);  
console.log(`p2 y = ${p2.y}`);  
console.log(`p2 z = ${p2.z}`);
```



Propriedades ES5

```
function Ponto() {  
    this.x=10;  
    this.y=20;  
}  
var p1 = new Ponto();  
console.log("p1 x = " + p1.x);  
console.log("p1 y = " + p1.y);  
//undefined  
console.log("p1 z = " + p1.z);  
var p2 = new Ponto();  
p2.z = 50;  
console.log("p2 x = " + p2.x);  
console.log("p2 y = " + p2.y);  
console.log("p2 z = " + p2.z)
```

Prototype ES6

Propriedade

```
class Poligono{
  constructor(){
    this.x = 0;
    this.y = 0; } }
var obj2d = new Poligono();
Poligono.prototype.z = 0;
var obj3d = new Poligono();
console.log(`obj2d x = ${obj2d.x}`);
console.log(`obj2d y = ${obj2d.y}`);
console.log(`obj2d z = ${obj2d.z}`);
console.log(`obj3d x = ${obj3d.x}`);
console.log(`obj3d y = ${obj3d.y}`);
console.log(`obj3d z = ${obj3d.z}`);
```

Prototype ES5

Propriedade

```
function Poligono(){  
    this.x = 0;  
    this.y = 0; }  
var obj2d = new Poligono();  
Poligono.prototype.z = 0;  
var obj3d = new Poligono();  
console.log("obj2d x = " + obj2d.x);  
console.log("obj2d y = " + obj2d.y);  
console.log("obj2d z = " + obj2d.z);  
console.log("obj3d x = " + obj3d.x);  
console.log("obj3d y = " + obj3d.y);  
console.log("obj3d z = " + obj3d.z);
```

Atributos operador delete

- Exclui propriedades do objeto
- Não exclui variáveis comuns

```
class Poligono{  
  constructor(){  
    this.x = 0;  
    this.y = 0;  
    this.z = 0;}}  
var obj2d =  
  new Poligono();  
delete obj2d.z;  
console.log(  
  obj2d.z);
```

ES6

```
function Poligono(){  
  this.x = 0;  
  this.y = 0;  
  this.z = 0;}  
var obj2d =  
  new Poligono();  
delete obj2d.z;  
console.log(  
  obj2d.z);
```

ES5





Introdução



Classe



Construtor



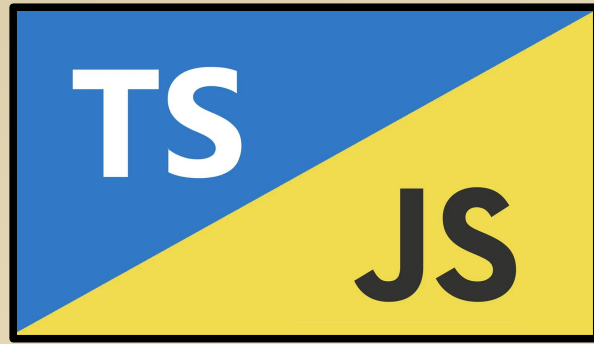
Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Getter/Setter

Getter e Setter

- São funções ou métodos usados para obter e definir os valores das variáveis
- Ajudam a encapsular, proteger, isolar propriedades
- Utilizado para fazer validações ou tratamentos antes de salvar um dados
- Operadores em ECMAScript 6 get e set
- Comportamento de variáveis
- Podem ter o mesmo nome





Getter/Setter



ES6

```
class Poligono {  
  constructor() {  
    this.x = 0;  
    this.y = 0;  
  }  
  get getX() {  
    return x;  
  }  
  set setX(x) {  
    this.x = x;  
  }  
  get getY() {  
    return y;  
  }  
  set setY(y) {  
    this.y = y;  
  }  
}
```

```
var p =  
  new Poligono();  
p.x=5;  
console.log(p.x);
```

ES5

```
function Poligono() {  
  this.x = 0;  
  this.y = 0;  
  this.getX=function() {  
    return x; }  
  this.setX=function(x) {  
    this.x=x; }  
  this.getY=function() {  
    return y; }  
  this.setY=function(y) {  
    this.y=y; } }  
}
```

```
var p =  
  new Poligono();  
p.getX(5);  
console.log(p.getX());
```



Introdução



Classe



Construtor



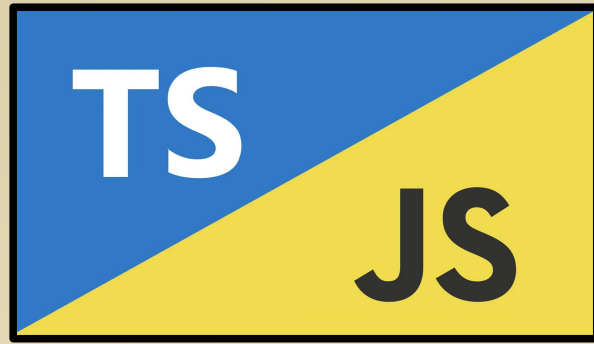
Propriedade



Getter / Setter



Método



JavaScript - Parte 2

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Métodos

Método

- Função associada a um objeto
- Manipulam valores de propriedades do objeto
- Podem possuir parâmetros
- Podem retornar valor usando palavra return



Métodos

```
class Poligono{  
  constructor(){  
    this.x = 0;  
    this.y = 0;  
  }  
  deslocar(x, y){  
    this.x+=x;  
    this.y+=y;  
  }  
}
```

ES6

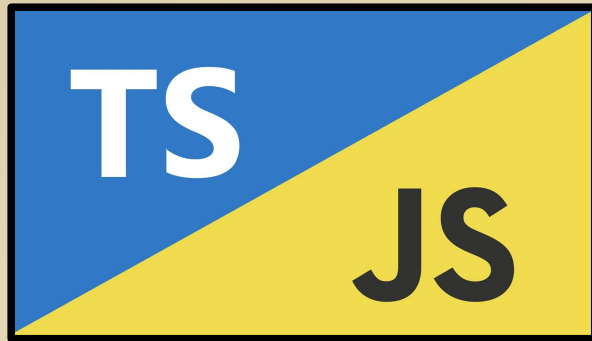
```
function Poligono(){  
  this.x = 0;  
  this.y = 0;  
  this.deslocar=  
    function(x, y){  
      this.x+=x;  
      this.y+=y;  
    }  
}
```

ES5

**Prof. Enzo
Seraphim**

**Profa. Bárbara
Pimenta Caetano**

Os logotipos, marcas comerciais e nomes de produtos citados nesta publicação tem apenas o propósito de identificação e podem ser marcas registradas de suas respectivas companhias.



JavaScript - Parte 2