



JAVASCRIPT.



MATERIAL DE APOIO

Disciplina: JavaScript

Nome da aula: Classes

Professor(a): Marcelo Estruc

Introdução

As classes foram introduzidas no ECMAScript 2015 e são simplificações da linguagem para as heranças baseadas nos protótipos. Não utiliza um novo modelo de herança de orientação a objetos em JavaScript. São formas mais simplificadas de criar objetos e lidar com herança.

Objetivos da aula

- Definir os conceitos de classes e objetos.
- Reconhecer classe e objetos na linguagem Javascript
- Demonstrar a implementação de classes e objetos na linguagem Javascript.

Resumo

Declaração de classe

A declaração de classe cria uma classe com um determinado nome usando herança baseada em **protótipo**.

```
2  class Poligono {
3    constructor (altura, largura) {
4      this.area = altura * largura;
5    }
6  }
7
8  console.log (new Poligono (4, 3) .area);
9  // saída esperada: 12
```

Fonte: Autoral

Definição de Classes

Class Expression

A **class expression** é uma maneira de definir uma classe no ECMAScript 2015. Semelhante às expressões de função, as expressões de classe podem ser nomeadas ou não. Se nomeado, o nome da classe é local apenas para o corpo da classe.

```
2  const Retangulo = class {
3      constructor (altura, largura) {
4          this.altura = altura;
5          this.largura = largura;
6      }
7      area() {
8          return this.altura * this.largura;
9      }
10 };
11
12 console.log (new Retangulo (5, 8).area ());
13 // saída esperada: 40
```

Fonte: Autoral

Sintaxe

```
2  const MyClass = class [className] [extends otherClassName] {
3      // corpo da classe
4  }
```

Fonte: Autoral

Descrição

Uma expressão de classe tem uma sintaxe semelhante a uma declaração de classe (instrução). Tal como acontece com as instruções de classe, o corpo de uma expressão de classe é executado no modo estrito.

Existem várias diferenças entre expressões de classe e instruções de classe, no entanto:

- As expressões de classe podem omitir o nome da classe ("identificador de ligação"), o que não é possível com instruções de classe.

- Expressões de classe permitem que você redefina (declare novamente) classes sem lançar um `SyntaxError`. Este não é o caso com declarações de classe.

O método do construtor é opcional. As classes geradas com expressões de classe sempre responderão a `typeof` com o valor "função".

Exemplos

Uma expressão de classe simples

Esta é apenas uma expressão de classe anônima simples à qual você pode se referir usando a variável `Foo`.

```
2  const Foo = class {
3      constructor() {}
4      bar() {
5          return 'Alô mundo!';
6      }
7  };
8
9      const instance = new Foo();
10     instance.bar(); // "Alô mundo!"
11     Foo.name;      // "Foo"
```

Fonte: Autorial

Expressões de classe nomeada

Se quiser fazer referência à classe atual dentro do corpo da classe, você pode criar uma expressão de classe nomeada. O nome só é visível dentro do escopo da própria expressão de classe.

```
2  const Foo = class NamedFoo {  
3      constructor() {}  
4      whoIsThere() {  
5          return NamedFoo.name;  
6      }  
7  }  
8  const bar = new Foo();  
9  bar.whoIsThere(); // "NamedFoo"  
10 NamedFoo.name;    // ReferenceError: NamedFoo não definido  
11 Foo.name;         // "NamedFoo"
```

Fonte: Autoral

Referência Bibliográfica

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!:** programação JavaScript. 1ª Ed. São Paulo: Alta Books, 2016.

Exercícios

1) As classes em JavaScript utilizam herança baseada em...

Assinale o item correto:

- a) funções
- b) protótipos
- c) construtor
- d) métodos
- e) variáveis

2) Para inicializar uma classe em JavaScript utilizamos... Assinale o item correto:

- a) uma função construtora
- b) um método construtor
- c) uma propriedade construtora
- d) a função create()
- e) a função const

3) Observe a porção de código abaixo:

```
2  class Carro {  
3      constructor (nome, ano) {  
4          this.nome = nome;  
5          this.ano = ano;  
6      }  
7      periodo() {  
8          let data = new Date ();  
9          return data.getFullYear () - this.ano;  
10     }  
11 }
```

Fonte: Autoral

Assinale o item correto:

- a) nome e ano são propriedades da classe.
- b) nome e ano são métodos da classe.

- c) período é um objeto da classe
- d) em período é criado o objeto nome
- e) retorna o nome

4) De acordo com o que foi estudado sobre Classes, e observando a figura da classe Pessoa, assinale nos itens abaixo, aqueles que podem ser considerados como propriedades:

Pessoa
- Nome - Telefone - Endereço - Email - Sexo - Data de nascimento
+ Andar + Falar + Dormir + Pensar

Fonte: Autoral

- a) andar e falar
- b) sexo e dormir
- c) telefone e email
- d) telefone e andar
- e) nome e pensar

Gabarito

1. A **Letra B** é a alternativa correta, pois o JavaScript é uma linguagem baseada em protótipo, e cada objeto no JavaScript tem uma propriedade interna escondida chamada `[[Prototype]]`, que pode ser usada para estender as propriedades e métodos de objetos.
2. A **Letra A** é a alternativa correta, pois em JavaScript utilizamos funções construtoras para inicializar as classes.
3. A **Letra A** é a alternativa correta, pois as propriedades também chamadas de características, forma ou atributo. São características do objeto e identificam o objeto em si. Elas podem mudar com o tempo. As propriedades formam a parte estrutural do objeto.
4. A **Letra A** é a alternativa correta, pois as propriedades são as “características” de um objeto.