

10:53

Quê - Orientação a Objetos em JavaScript

Dados as afirmações abaixo

I. O método `super()` pode ser utilizado em classes que herdam propriedades de outras

II. Getters e setters são utilizados como modificadores de um valor

III. Getters e setters foram o conceito de encapsulamento

São verdadeiras

Nenhuma das alternativas

I e II

Todas as alternativas

I e III

II e III

PRÓXIMA PERGUNTA

10:11

Quê - Orientação a Objetos em JavaScript

Quais são os quatro pilares da Orientação a Objetos?

Abstração, Herança, Encapsulamento e Polimorfismo

Abstração, Herança, Encapsulamento e Polimorfismo

Abstração, Herança, Encapsulamento e Tipos

Abstração, Polimorfismo, Encapsulamento e Polimorfismo

Abstração, Herança, Encapsulamento e Polimorfismo

PRÓXIMA PERGUNTA

11:00

Quê - Orientação a Objetos em JavaScript

Dados as afirmações abaixo

I. A sintaxe de Classes em JavaScript é apenas um "açúcar sintético".

II. JavaScript utiliza Herança Prototípica

III. Funções dentro de um objeto são chamadas de "métodos"

São verdadeiras

I

Todas as alternativas

I e II

Nenhuma das alternativas

II e III

PRÓXIMA PERGUNTA

10:17

Quê - Orientação a Objetos em JavaScript

Sobre protótipos, é correto afirmar que:

São responsáveis pelo conceito de Polimorfismo em JavaScript

São responsáveis pelo conceito de Orientação em JavaScript

São responsáveis pelo conceito de Herança em JavaScript

São responsáveis pelo conceito de Encapsulamento em JavaScript

São responsáveis pelo conceito de Abstração em JavaScript

PRÓXIMA PERGUNTA

11:09

Quê - Orientação a Objetos em JavaScript

Caso um parâmetro tenha getters e setters, por convenção, como ele deve ser nomeado?

Com o sinal - como prefixo (ex: `salario`)

Com o sinal \$ como prefixo (ex: `salario`)

Com o sinal _ como prefixo (ex: `salario`)

Com o sinal # como prefixo (ex: `salario`)

Com o sinal @ como prefixo (ex: `salario`)

PRÓXIMA PERGUNTA

10:31

Quê - Orientação a Objetos em JavaScript

Sobre o método `super()` é correto afirmar que:

Ele permite que sejam acessados parâmetros na hora de instanciar um novo objeto

Permite que seja feito um clone da classe ascendente

Ele passa no argumento enviado para a classe ascendente

É equivalente ao operador `smart`

É equivalente ao operador `extends`

PRÓXIMA PERGUNTA

11:16

Quê - Orientação a Objetos em JavaScript

Qual a melhor definição de Herança?

Classes irmãs compartilham métodos e propriedades entre si

Classes ascendentes herdam características de suas classes descendentes

Classes irmãs herdam características de suas classes primas

Classes descendentes herdam características de suas classes ascendentes

Classes descendentes sempre sobrescrevem características de suas classes ascendentes

PRÓXIMA PERGUNTA

10:37

Quê - Orientação a Objetos em JavaScript

Qual a melhor definição de Encapsulamento?

Propriedades e métodos de uma classe dizem respeito a ela e a suas classes ascendentes

Propriedades e métodos de uma classe são globais

Propriedades e métodos de uma classe dizem respeito apenas a ela

Propriedades e métodos de uma classe dizem respeito apenas às classes descendentes

Propriedades e métodos de uma classe dizem respeito a ela e a suas classes irmãs

PRÓXIMA PERGUNTA

11:30

Quê - Orientação a Objetos em JavaScript

Qual a melhor definição de Abstração?

Remover métodos

Adicionar métodos

Adicionar complexidade

Remover complexidade

Sobrescrever atributos

PRÓXIMA PERGUNTA

10:44

Quê - Orientação a Objetos em JavaScript

Qual a melhor definição de Polimorfismo?

O conceito de que classes só podem ter um tipo

O conceito de que classes mais abstratas herdam métodos de classes mais complexas e os executam à sua maneira

O conceito de classe de protótipos

O conceito de que classes mais complexas herdam métodos de classes mais abstratas e os executam sem modificações

O conceito de que classes mais complexas herdam métodos de classes mais abstratas e os executam à sua maneira

PRÓXIMA PERGUNTA