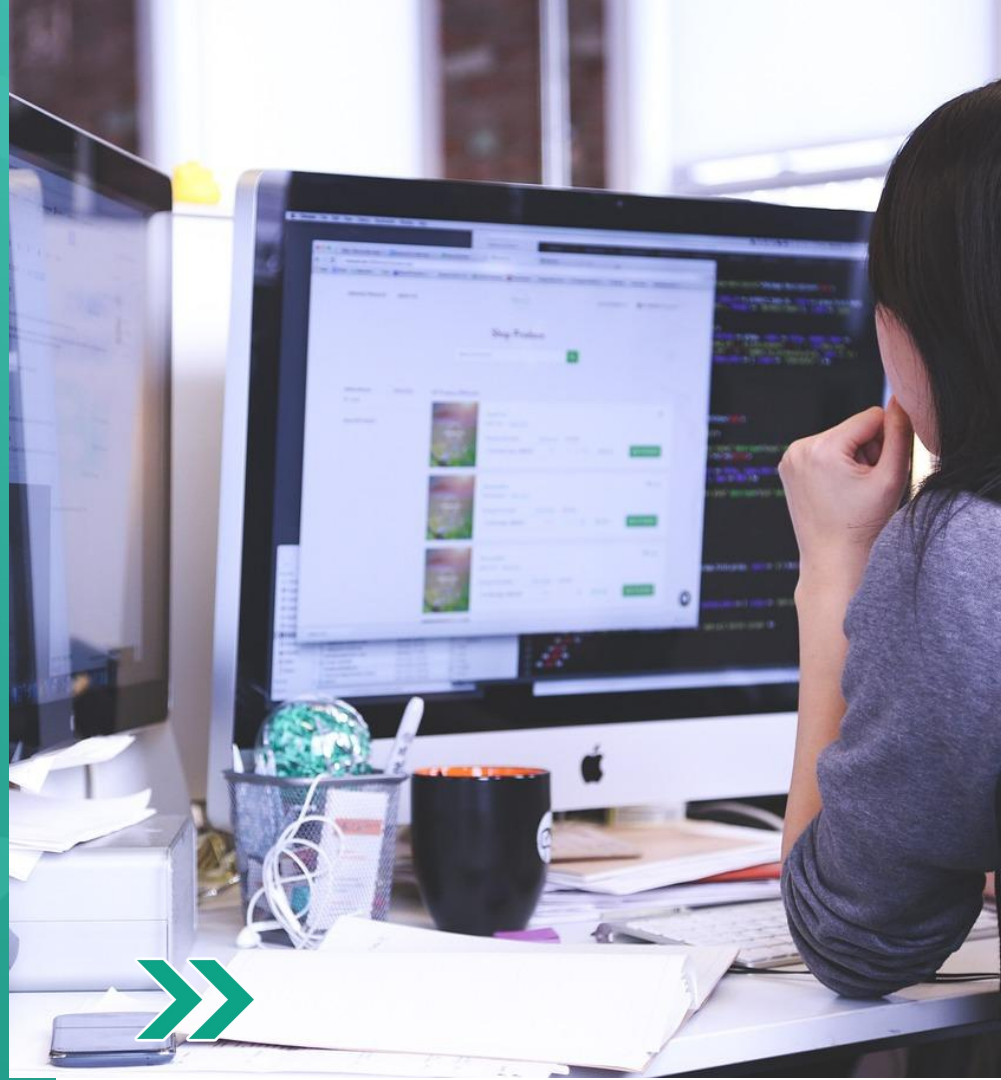
escola britânica de artes criativas & tecnologia

Profissão: Engenheiro Front-End

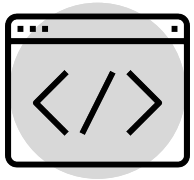
escola britânica de artes criativas & tecnologia

BOAS PRÁTICAS



Orientação a Objetos com JavaScript

- **Crie objetos**
- **Encapsulamento**
- **Polimorfismo**



Confira boas práticas da comunidade de Front-End por assunto relacionado às aulas.

Crie objetos

Funções construtoras
As funções construtoras são uma parte importante da programação orientada a objetos em JavaScript e podem ser usadas para criar objetos personalizados com propriedades e métodos específicos. Acompanhe algumas dicas úteis para usar as funções construtoras no JavaScript:

- **Utilize a convenção de nomenclatura:** Comece o nome da função construtora com uma letra maiúscula. Isso ajuda a diferenciar a função construtora de outras funções regulares.

- **Use a palavra-chave new:** Ao criar uma instância de um objeto usando uma função construtora, utilize a palavra reservada new antes do nome da função. Isso indica ao JavaScript que uma nova instância do objeto deve ser criada.

- **Defina propriedades e métodos usando this:** Dentro da função construtora, atribua propriedades e métodos ao objeto que está sendo construído utilizando a palavra-chave this. Isso garante que essas propriedades e métodos sejam associados a cada instância criada.

Crie objetos

Funções construtoras
As funções construtoras são uma parte importante da programação orientada a objetos em JavaScript e podem ser usadas para criar objetos personalizados com propriedades e métodos específicos. Acompanhe algumas dicas úteis para usar as funções construtoras no JavaScript:

- **Evite definir métodos diretamente na função construtora:** Definir métodos diretamente dentro da função construtora fará com que cada instância tenha uma cópia separada desse método. Isso pode levar a um consumo excessivo de memória. Em vez disso, é recomendado adicionar métodos ao protótipo do objeto. Isso permitirá que todas as instâncias compartilhem o mesmo método, economizando memória.

- **Lembre-se de chamar a função construtora com new:** Se esquecer de usar a palavra reservada new ao chamar a função construtora, a função será executada como uma função regular em vez de uma função construtora. Isso pode levar a erros ou resultados inesperados.

- **Use o operador instanceof para verificar a instância:** O operador instanceof pode ser usado para verificar se um objeto é uma instância de uma função construtora específica. Por exemplo: objeto instanceof FuncaoConstrutora.

Encapsulamento

Acompanhe algumas dicas para aplicar o encapsulamento e controlar o acesso a atributos e métodos, melhorando a organização e a segurança do seu código:

- **Uso de closures**: As closures permitem criar funções internas que têm acesso às variáveis locais de sua função externa. Isso pode ser usado para criar atributos privados e métodos com escopo limitado.

- **Convenção de nomenclatura**: Use uma convenção de nomenclatura para indicar que um atributo ou método deve ser considerado privado. Uma prática comum é adicionar um prefixo _ ao nome do atributo ou método para indicar que eles são internos e devem ser tratados como privados.

- **Getters e Setters**: Use métodos getter e setter para acessar e modificar atributos internos, permitindo controle adicional sobre a manipulação dos dados.

Polimorfismo

- **Polimorfismo de sobrescrita de métodos**
  Você pode criar uma hierarquia de classes e definir métodos com o mesmo nome em classes derivadas (filhas) que substituam a implementação do método da classe base (pai). Isso permite que você chame o método de forma polimórfica e obtenha o comportamento específico da classe concreta.

- **Polimorfismo de sobrecarga de métodos**
  Como o JavaScript não suporta sobrecarga de métodos com base em tipos de parâmetros, você pode simular o polimorfismo de sobrecarga verificando os tipos de parâmetros em tempo de execução.

escola britânica de artes criativas & tecnologia

Bons estudos!