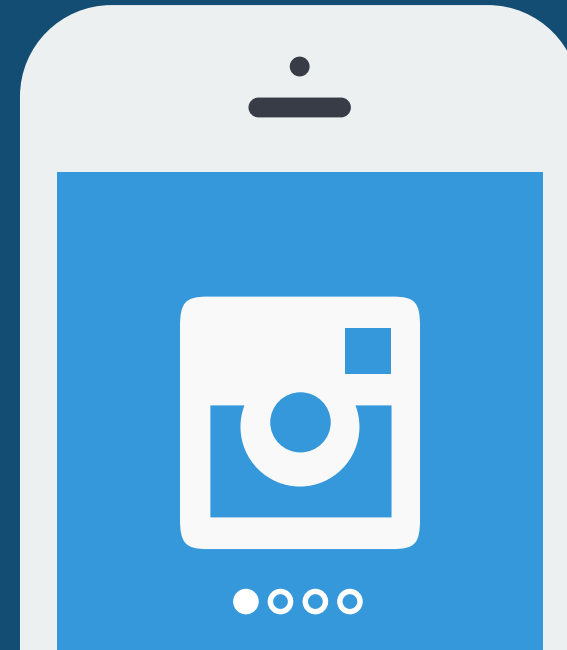
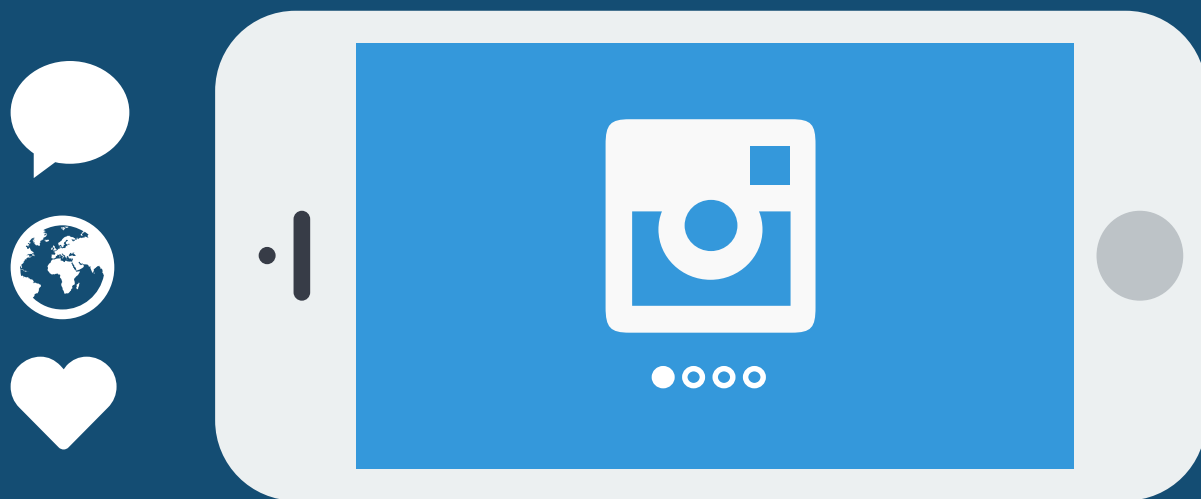


targettrust

treinamento e tecnologia



ORIENTAÇÃO A OBJETOS



Orientação a Objetos

- JavaScript tem fortes capacidades de programação orientação a objetos, apesar de ocorrerem algumas discussões devido às diferenças da orientação a objetos no JavaScript em comparação com outras linguagens.



Orientação a Objetos

- Programação orientada a objetos é um paradigma de programação que usa abstração para criar modelos baseados no mundo real. Ele usa várias técnicas vindo de paradigmas previamente estabelecidos, incluindo modularidade, polimorfismo e encapsulamento.



Orientação a Objetos

- **Classe** - Define as características do objeto.
- **Objeto** - Uma instância de uma classe.
- **Propriedade** - Uma característica de objeto, como cor.
- **Método** - Uma capacidade de objeto, como caminhada.
- **Construtor** - Um método chamado de momento de instanciação.



Orientação a Objetos

- **Herança** - Uma classe pode herdar características de outra classe.
- **Encapsulamento** - A classe define apenas as características do objeto, o método define apenas como o método é executado.
- **Abstração** - A conjunção de herança complexa, os métodos, as propriedades de um objeto deve ser capaz de simular um modelo de realidade.
- **Polimorfismo** - Diferentes Classes pode definir o mesmo método ou propriedade.



Orientação a Objeto

- Programação baseada em protótipo é um estilo de programação orientada a objetos em que as classes não estão presentes, e reutilização comportamento (conhecido como herança em linguagens baseadas em classes) é realizado através de um processo de decorar objetos existentes que servem como protótipos.
- JavaScript tem vários objetos incluídos em seu core; por exemplo, existem objetos como Math, Object, Array, e String.



Orientação a Objeto

- Cada objeto em JavaScript é uma instância do objeto Object e, portanto, herda todas as suas propriedades e métodos.



Orientação a Objeto

Os literais de objeto

- Possibilitam criar e iniciar objetos de uma maneira diferente. A sintaxe é definida por uma lista de pares nome/valor separados por vírgulas entre um par de chaves.
- Cada par nome/valor é definido pelo nome da propriedade seguido de dois pontos e do valor correspondente.



Orientação a Objeto

- A composição é um recurso utilizado para definir uma relação do tipo “tem um” (“has a” relationship), ou seja, um objeto que conta com outros objetos para formar sua estrutura.
- Por exemplo, um objeto do tipo Carro teria em sua estrutura objetos do tipo Roda, Volante, Banco.



Orientação a Objeto

- O encapsulamento tem por objetivo esconder informações que não precisam ser de conhecimento do utilizador da classe.
- Seu uso é uma boa prática quanto à manutenção da classe, pois podemos modificar a parte que é oculta ao utilizador sem alterar sua forma de implementação.



Orientação a Objeto

- Em Javascript a herança ocorre por meio de objetos protótipos e define uma relação do tipo “é um” (“is a” relationship).
- Cada objeto herda propriedades e métodos de seu objeto protótipo que é referenciado pela propriedade prototype.



Orientação a Objeto

- A classe Object é a superclasse de todas as classes definidas em Javascript, ou seja, todos os construtores criados herdam propriedades e métodos definidos no construtor Object() como por exemplo o método **toString()**, que assim como outros pode ser sobrescrito na subclasse.



Orientação a Objeto

Como Acessar Propriedades em um Objeto

- Os dois caminhos primários para acessar as propriedades de um objeto são com a notação com ponto e com a notação com colchete.



Orientação a Objeto

- Para deletar uma propriedade de um objeto, usamos o operador delete. Você não pode deletar propriedades que são herdadas, nem propriedades cujo os atributos estão definidos para configuração.
- Você deve deletar as propriedades herdadas no objeto protótipo (onde as propriedades foram definidas).
- Você também não pode deletar propriedades do objeto global, as quais foram declaradas sem a palavra-chave var.



Orientação a Objeto



Serializar e Desserializar Objetos

- Para transferir seus objetos via HTTP ou convertê-los para uma string, você precisa de serializá-los (converter para string); você pode usar a função `JSON.stringify` para serializar seus objetos.
- Note que quando usado antes da ECMAScript 5, você deve usar a popular biblioteca `json2` (de Douglas Crockford) para obter a função `JSON.stringify`. Ela é padronizada/nativa na ECMAScript 5.



Orientação a Objeto

Serializar e Desserializar Objetos

- Para desserializar seu objeto (convertê-lo para um objeto a partir de uma string), você deve usar a função `JSON.parse` da mesma biblioteca `json2`.
- Esta função também é nativa na ECMAScript 5.

