23/07/17

Classes e objetos

Criar uma classe

class NomeClasse {

}

↳ recomendo-se CamelCase

Propriedades da Classe

public $nomePropriedade

Usando objeto e criando

$objeto = new NomeClasse() — criar (instanciar)

$objeto->nomePropriedade = "valor" — acessar uma propriedade

Passar um objeto em uma função

function NomeFuncao ($conexao, Produto $produto) {

}

só quando a função está fora do documento de criação instanciada

3/8/17

Método

Comportamento de uma classe

Como criar um método?

```
class NomeClasse {
    public $propriedade

    public function metodo(parametros) {
        ...
    }
}
```

metodo

Como chamar um método

```
$objeto->metodo(parametros);
```

Como referenciar uma propriedade do próprio objeto em um método

```
$this->propriedade
```

S T Q [redacted] S S D

3/8/17

Encapsulamento

Evitar acesso a metodos e estados (propriedades) de fora da classe

Como encapsular?

Propriedade

private $propriedade

Metodo

private function Metodo {

}

Qual vantagem?

Evitar que uma propriedade seja alterada sem devida validação.

Ao colocar uma propriedade privada, deve-se criar um metodo na classe para lê-la ou seta-la. (getter setter)

Não crie getter e setter sem necessidade

03/08/17

S T Q S S S D

Comparação de objetos

$objeto1 == $objeto2

Irá comparar os valores e atributos dos objetos para checar se são iguais

$objeto1 === $objeto2

Irá comparar, valores, atributos e instancia assim garante que nesta variáveis a objetos de instancias diferentes.

Magic Methods

São metodos padrões que pertencem a todos os objetos.

Ao criar

function __construct() {

...

}

Pode ser usado para obrigar a(s) ~~produtos~~ propriedades

$this->propriedade = $valor

Ao imprimir na tela

function __toString() { ... }

Ao destruir um objeto

function __destruct() { ... }

No PHP 5 a declaração de construtores era com a criação de uma função com o mesmo nome da classe