

Programação Orientada a Objetos

Classes e Objetos em PHP:

Classes é uma estrutura que abstrai um conjunto de objetos contendo características similares, ela determina as ações dos objetos usando os métodos e os modificando.

As classes em PHP são bem semelhante com a linguagem Java, que também é Orientada a Objetos, porém possui suas distinções.

Para criar uma classe em **PHP** usa-se:

```
<?php
class NomeDaClasse{

}
?>
```

Objetos são instâncias de classes, que determinam quais informações ele possui e como ele pode manipulá-las

Exemplos de Objetos em **PHP**:

```
$variavel = new NomeDaClasse();
```

Construtores em PHP:

Os construtores são responsáveis por inicializar um objeto da classe.

Exemplos de Construtores em **PHP**:

```
<?php
class NomeDaClasse{

function __construct() {
}

}
?>
```

Modificadores de acesso em PHP:

Os modificadores de acessos em PHP são parecidos com os modificadores da linguagem Java, tendo:

- **public** – para atributos e métodos que podem ser vistos e alterados por qualquer lugar;
- **private** – para atributos e métodos que só podem ser acessados dentro da classe que os invocaram.
- **Protected** – para atributos e métodos que só podem ser acessados pela própria classe ou classes que herdaram eles.

Getter e Setter em PHP:

Os métodos especiais **get** e **set** em PHP funcionam da mesma forma que acontece em Java, o **get**(pegar) serve para acessar determinado atributo, já o **set** (colocar) serve para inserir valores em atributos/variáveis.

Para utilizar get e set em **PHP** usa-se:

```
<?
class Estudante {
public $nome;

public function getNome(){
    return $this -> nome;
}

public function setNome($nome){
    this.nome -> $nome;
}
}
```

Métodos mágicos para getter e setter em PHP:

__get e __set:

Esses métodos mágicos facilitam a implementação em PHP, basta implementá-los uma única vez e eles poderão ser utilizados para todos os atributos que forem da classe ou até mesmo todas as classes do sistema.

Exemplo retirado do site disponibilizado para pesquisa:

```
<?php
class Pessoa {
    private $nome;
    private $sobrenome;

    public function __set($atrib, $value){
        $this->$atrib = $value;
    }

    public function __get($atrib){
        return $this->$atrib;
    }
}
?>
```

Funções *include*, *require*, *include_once* e *require_once* em PHP.

include: Essa função possibilita incluir arquivos dentro de outros quando acessados, na presença de erros será mostrado o erro *Warning*, que não foi possível incluir o arquivo.

Exemplo:

```
<?php
include('pagina/main.php')
?>
```

require: Funciona de forma semelhante a função **include**, no entanto se o arquivo passado como parâmetro não existir ou não ser encontrado ocasionará um *Fatal Error* que não executará mais nenhuma linha de código abaixo.

include_once() e require_once(): Essas duas funções funcionam bem semelhantes com as funções *require* e *include*, no entanto elas só aceitam passagem de arquivos como parâmetros uma única vez, se ela for chamada pela segunda vez será ignorada.

```
<?php
include('pagina/main.php')
include('pagina/main.php') // Será ignorada...
?>
```

Função isset: Essa função nos permite verificar se uma variável foi definida ou não, se ela estiver definida retornará TRUE ou FALSE caso ela esteja indefinida;

Para utilizar *isset* em **PHP** usa-se:

```
<?php
$var = 'Hello, World!';
if (isset($var)) {
    echo "Definida";
}
>
```

Referências:

Classes e Objetos:

[PHP: Classes/Objetos - Manual](#)

Construtores:

[PHP: Construtores e Destrutores – Manual](#)

Modificadores de acesso:

[Modificadores de acesso no PHP: Programação Orientada a Objetos](#)

Get e Set:

[Programação Orientada a Objetos com PHP — Métodos Getters e Setters](#)

Métodos Mágicos:

[Artigo: Métodos mágicos get e set do PHP 5 - Gigasystems](#)

Include, require, include_once e require_once:

[Artigo: Diferença entre include e require, include_once e require_once - Gigasystems](#)

isset:

[PHP: isset - Manual](#)