

```
<?php
```

```
    //__get:
```

//Quando usamos esse método mágico implementado as nossas classes possibilitamos que qualquer método seja acessado por uma instância somente referenciando ao nome do atributo diretamente (assim como existe o __get existe também o __set para selecionar atributos e atribuir valores), esse é um método mais prático de criar nossos getters e setters, vejamos como...

```
//Temos uma classe que implementa o nome e sobrenome de uma pessoa...
```

```
class Pessoa {
```

```
    private $nome; //Note que ambos atributos são privados, deveria ser impossível acessá-los somente por referência
    private $sobrenome; //em uma instância...
```

```
    public function __set($atrib, $value){ //Veja que só usamos um único método mágico que vai receber o nome do
                                            //atributo e um valor, e irá atribuir o valor sobre o atributo escolhido.
        $this->$atrib = $value; //O $this torna possível a comparação com o valor passado por referência com os
                                //atributos existentes na classe...
    }
```

```
    public function __get($atrib){ //Enquanto isso, o get irá possibilitar que possamos acessar o valor do método...
        return $this->$atrib; //Mais uma vez o $this torna possível que o valor passado na referência seja comparado
    }                                //com algum atributo pré-existente na classe...
}
```

```
$Pessoa = new Pessoa();
```

```
$Pessoa->nome = 'Fulano'; //Aqui o __set está sendo aplicado...
```

```
$Pessoa->sobrenome = ' de Tal';
```

```
echo 'Nome: ' . $Pessoa->nome . PHP_EOL; //Aqui o __get está sendo aplicado...
```

```
echo 'Sobrenome: ' . $Pessoa->sobrenome . PHP_EOL;
```

```
var_dump($Pessoa) . PHP_EOL; //Note que tanto o set como get estavam se referindo a atributos reais...
```

```
//USANDO O MÉTODO MESMO EM CLASSES ONDE NÃO HÁ ATRIBUTOS...
```

```
class Gritar { //Essa classe tem o objetivo de pegar um atributo criado temporariamente e transformar o valor dele
                //numa string com todas as letras em uppercase...
}
```

```
function __get(string $palavraQualquer):string //Essa função get deve só pegar o nome do atributo transformá-lo
{
    //em UpperCase...
    return strtoupper($palavraQualquer) . PHP_EOL;
}
```

```
}
```

```
$palavra = new Gritar(); //Veja que bastou instanciar a variável...
echo $palavra->valdir; //Qualquer atributo, mesmo que não exista, é transformado em UpperCase e retornado...
echo $palavra->gabriel;
echo $palavra->fencer;
echo $palavra->nobredecoder;
```

```
?>
```

[Running] php "c:\Users\Almoxarifado\Documents\php\arquivos_das_aulas\42-__get__ACESSANDO_QUALQUER_ATRIBUTO_DE_UMA_CLASSE.php"

Nome: Fulano

Sobrenome: de Tal

```
object(Pessoa)#1 (2) {
    ["nome":"Pessoa":private]=>
    string(6) "Fulano"
    ["sobrenome":"Pessoa":private]=>
    string(7) " de Tal"
}
```

VALDIR

GABRIEL

FENCER

NOBRECODER

[Done] exited with code=0 in 0.1 seconds