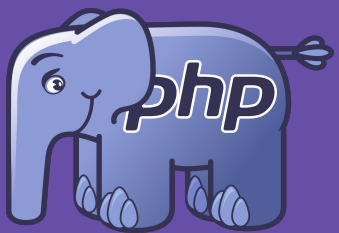


# Curso PHP e MySQL

## \_P00 em PHP

Programação Orientada a Objetos.

Adriel Sales

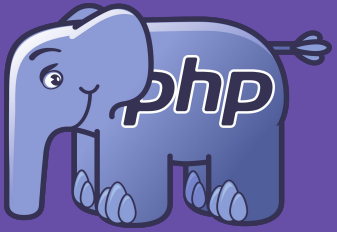


## \_P00 Programação Orientada a Objetos

- Arrays são legais, mas...
  - Como garantir que ao criar um aluno sejam preenchidos todos os campos solicitados pelo sistema?
  - Como definir funções específicas para determinados índices do array aluno, como por exemplo, validar um cpf sempre que criar um novo aluno?
  - E se um programador novato na equipe errar os identificadores dos índices?

```
$aluno = [  
    'nome'      => 'João',  
    'matricula' => '123456',  
    'cpf'       => '123.456.789.10',  
];
```

```
$aluno = [  
    'name'      => 'João',  
    'Matricula' => '123456',  
    'CPF'       => '123.456.789.10',  
];
```



## \_paradigmas de programação

Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa.

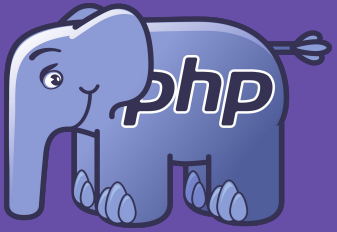
Na **programação orientada a objetos (POO)**, os programadores podem abstrair um programa como uma coleção de objetos que interagem entre si, enquanto em **programação funcional** os programadores abstraem o programa como uma sequência de funções executadas de modo empilhado.

Fonte:

[https://pt.wikipedia.org/wiki/Paradigma\\_de\\_programação](https://pt.wikipedia.org/wiki/Paradigma_de_programação)

Nas linguagens orientadas a objetos, os dados e as rotinas para manipulá-los são mantidos numa unidade chamada objeto. O utilizador só pode acessar os dados através das sub-rotinas disponíveis, **chamadas métodos**, o que permite alterar o funcionamento interno do objeto sem afetar o código que o consome.

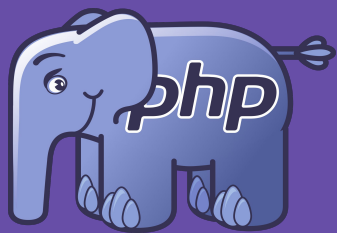




## \_POO Programação Orientada a Objetos

- **A Programação Orientação a Objetos (POO)** é um padrão de programação em que um software não é composto por um grande bloco de códigos específicos, e sim de vários blocos de códigos independentes, que juntos montam um sistema.
- **O PHP** faz parte de um grupo de linguagens que possuem suporte a POO, mas não é preso a ela.
- **Objetivos da POO:**
  - Reutilização de código (tempo economizado);
  - Escalabilidade (código adicionado mais facilmente);
  - Manutenibilidade (manutenção mais fácil);
  - Desenvolvimento mais rápido.



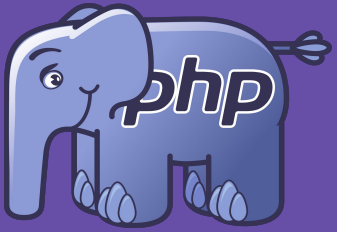


## **\_POO** Conceitos fundamentais

### Conceitos fundamentais para seu desenvolvimento OO:

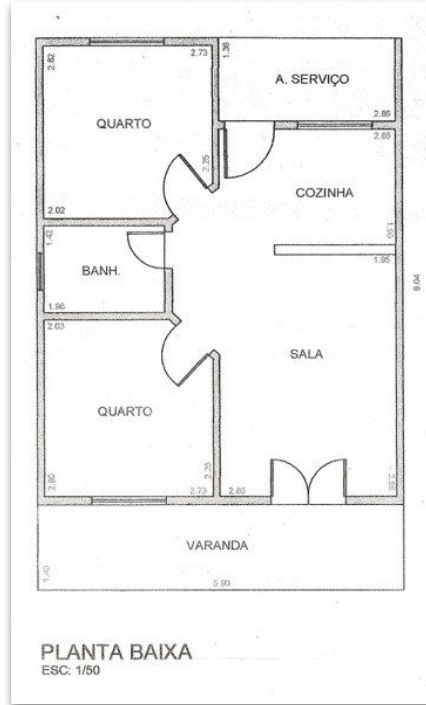
- **Abstração:**
  - É separar mentalmente, considerar isoladamente(...). Para construir o sistema orientado a objetos não devemos projetar o sistema como sendo uma grande peça monolítica, mas devemos separar em partes concentrando-nos nas peças mais importantes(...) em um primeiro momento, para podermos construir peças bem definidas e que possam ser reaproveitadas mais tarde, formando uma estrutura hierárquica (Pablo, 2009, p. 86).
- **Classes:**
  - É a representação de vários objetos que possuem as mesmas características e comportamentos no mundo real;
- **Objetos:**
  - É uma estrutura dinâmica originada com base em uma classe sendo aplicável ao sistema;
- **Atributos / Propriedades:**
  - são as características de um determinado objeto.
- **Métodos:**
  - São funções declaradas dentro de uma classe, que trabalham diretamente com as propriedades do objeto e as suas lógicas específicas. Os métodos diferenciam as propriedades dos objetos de variáveis locais .

# Classes → Objetos



## \_P00 Conceitos fundamentais

**Classe** → Casa.php



**new** Casa()



**Objeto Casa 1**

**new** Casa()

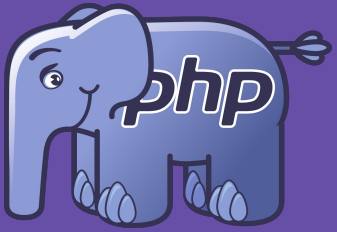


**Objeto Casa 2**

**new** Casa()



**Objeto Casa 3**



## \_POO Conceitos fundamentais

```
<?php
```

```
class Turma {
```

```
    public $nome;
```

```
    public function setNome(int $nome)
```

```
    {  
        $this->nome = $nome;  
    }
```

```
    public function getNome()
```

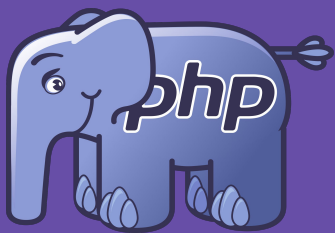
```
    {  
        return $this->nome;  
    }
```

```
}
```

Nome

Atributos / propriedades

Métodos / comportamentos



\_POO

Conceitos  
fundamentais

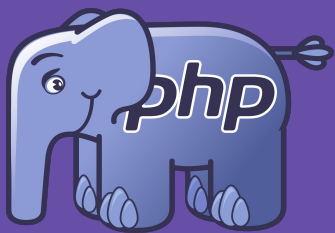
class

- A definição de uma classe começa com a palavra chave **class**, seguida do nome da classe, seguido de um par de colchetes que englobam as definições de propriedades e métodos pertencentes à classe.
- O nome de uma classe tem de ser válido, que não seja uma palavra reservada do PHP.
- Um nome de classe válido começa com uma letra ou sublinhado, seguido de qualquer sequência de letras, números e sublinhados.
- Definição Simples de uma Classe:

```
<?php
class SimpleClass
{
    // declaração de propriedade
    public $var = 'um valor padrão';

    // declaração de método
    public function displayVar() {
        echo $this->var;
    }
}
```



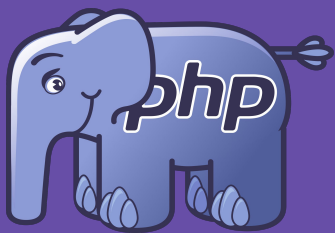


## \_POO Conceitos fundamentais

### this

- A pseudo-variável **\$this** está disponível quando um método é chamado a partir de um contexto de um objeto. **\$this** é o valor do objeto chamado.
- Alguns exemplos da pseudo variável **\$this**:

```
<?php
class A
{
    function foo()
    {
        if (isset($this)) {
            echo '$this está definida (';
            echo get_class($this);
            echo ")\n";
        } else {
            echo "\$this não está definida.\n";
        }
    }
}
```

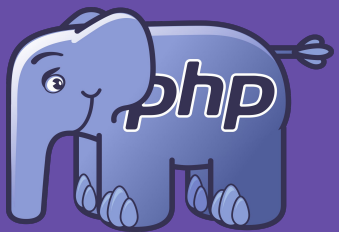


## \_POO Conceitos fundamentais

new

- Para criar uma **instância/objeto** de uma classe, a instrução **new** deve ser utilizada. Um objeto sempre será criado a não ser que a classe tenha um construtor definido que dispare uma exceção em caso de erro.
- Nota:
  - Se não há argumentos a serem passados para o construtor da classe, os parênteses após o nome da classe podem ser omitidos.
- Criando uma instância

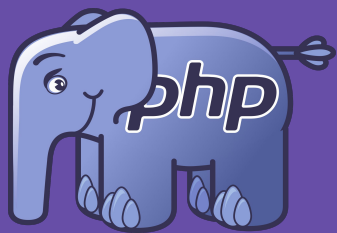
```
<?php  
$instance = new SimpleClass();  
  
// Também pode ser feito com uma variável:  
$className = 'SimpleClass';  
$instance = new $className(); // new SimpleClass()
```



### \_atividades de fixação

1. Crie uma classe 'Pessoa' em PHP com as seguintes características:
  - a. Contenha os seguintes atributos/propriedades públicas: id, nome, cpf, data\_nascimento e endereço;
  - b. Caso necessário, crie os métodos getters e setters para os atributos da classe;
2. Crie um outro arquivo PHP chamado index.php. Neste novo arquivo, crie alguns objetos do tipo Pessoa (utilizando a classe criada no exercício anterior), popule esses objetos e, com o auxílio de um foreach, liste as informações adicionadas em cada instância.
3. **DESAFIO:**  
Aproveitando do conhecimento adquirido com arrays e os conteúdos vistos nas aulas anteriores, como poderíamos desenvolver um programa em PHP, utilizando POO, para cadastro de clientes via linha de comando?





## \_POO

### Conceitos fundamentais

### Construtores

O PHP permite aos desenvolvedores declararem métodos construtores para as classes. Classes que tem um método construtor chamam o método a cada objeto recém criado.

```
<?php
class Point {
    protected int $x;
    protected int $y;

    public function __construct(int $x, int $y = 0) {
        $this->x = $x;
        $this->y = $y;
    }
}

// Passagem de ambos os argumentos.
$p1 = new Point(4, 5);
// Passar somente o argumento obrigatório, $y terá o valor padrão zero.
$p2 = new Point(4);
// Com parâmetros nomeados (a partir do PHP 8.0):
$p3 = new Point(y: 5, x: 4);
?>
```



Professor:  
Adriel Sales

# Fonte? Vá na fonte!

A screenshot of the official PHP website. The header includes navigation links for "php", "Downloads", "Documentation", "Get Involved", "Help", and "php.net", along with a search bar. The main content area features the "php" logo, a description of PHP as a general-purpose scripting language, and buttons for "What's new in 8.1" and "Download". Below this, there are links for "8.1.9", "8.0.22", and "7.4.30", each followed by "Changelog" and "Upgrading". A prominent white box contains an announcement for "PHP 8.2.0 Beta 3 available for testing" dated "18 Aug 2022". The text in the box states that the PHP team is pleased to announce the third beta release of PHP 8.2.0, Beta 3, and provides links for source downloads, bug reporting, and upgrading notes. It also includes a warning not to use this version in production. On the right side of the page, there are sections for "Upcoming conferences" (listing International PHP Conference Munich 2022, SymfonyWorld Online 2022 Winter Edition, SymfonyCon Disneyland Paris 2022, and CakeFest 2022), "User Group Events", "Special Thanks", and "Social media" (with a link to @official\_php).

<https://www.php.net>

<https://github.com/adrielacademico>