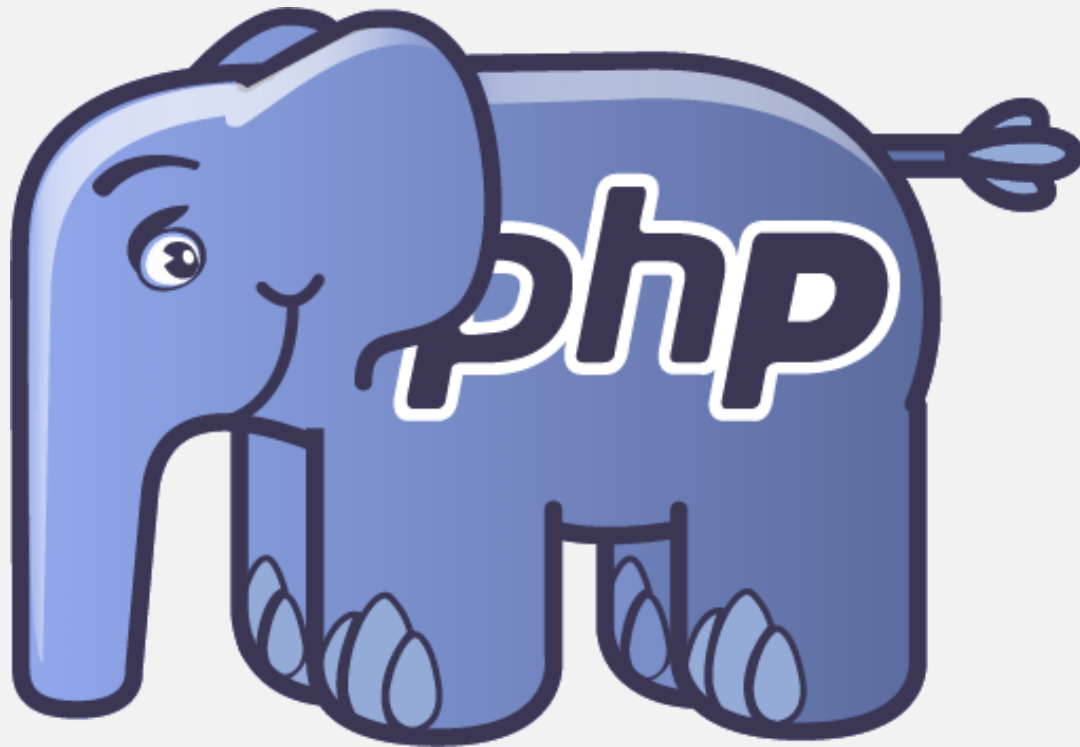


VARIÁVEIS EM PHP

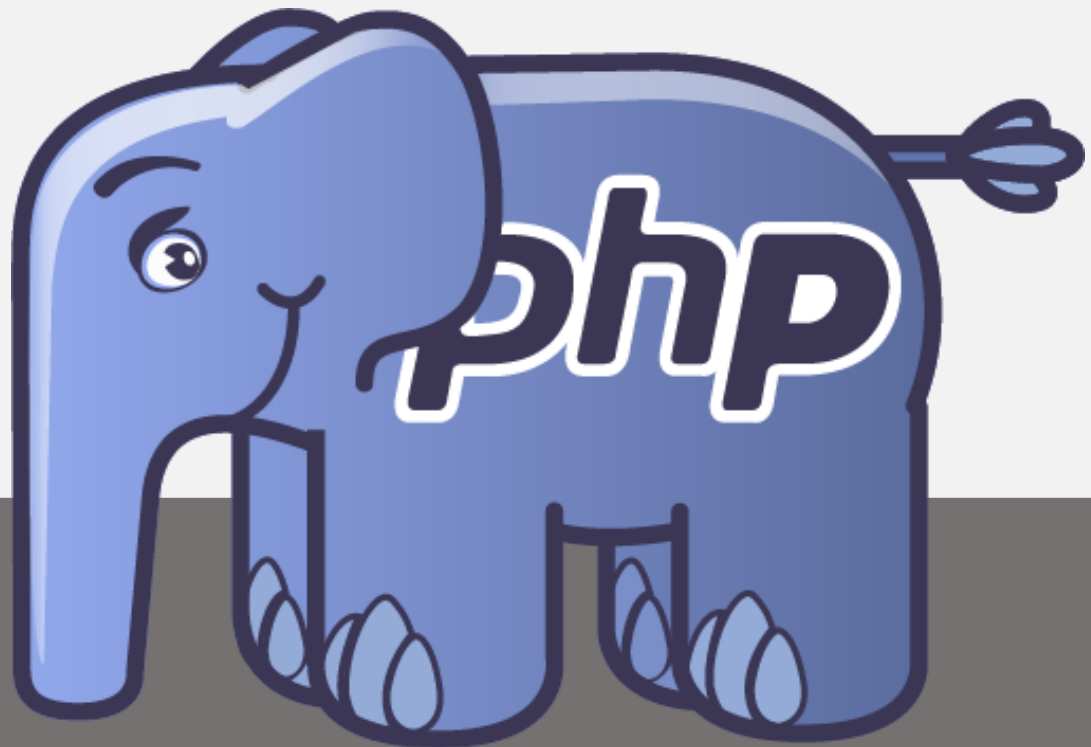


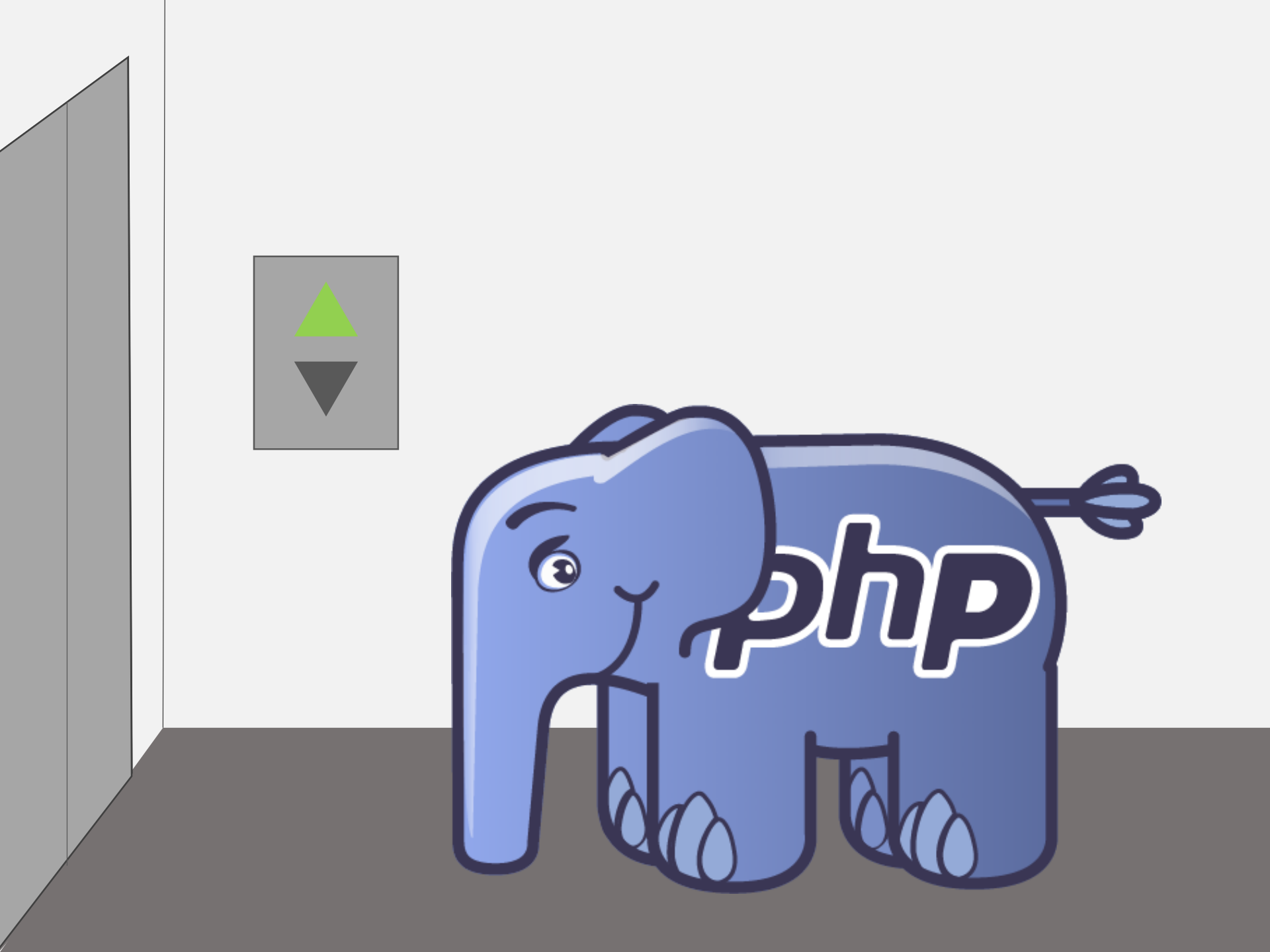
Hoje retomaremos alguns conceitos das aulas de algoritmos e veremos quais as mudanças que ocorrem quando trabalhamos variáveis em PHP.

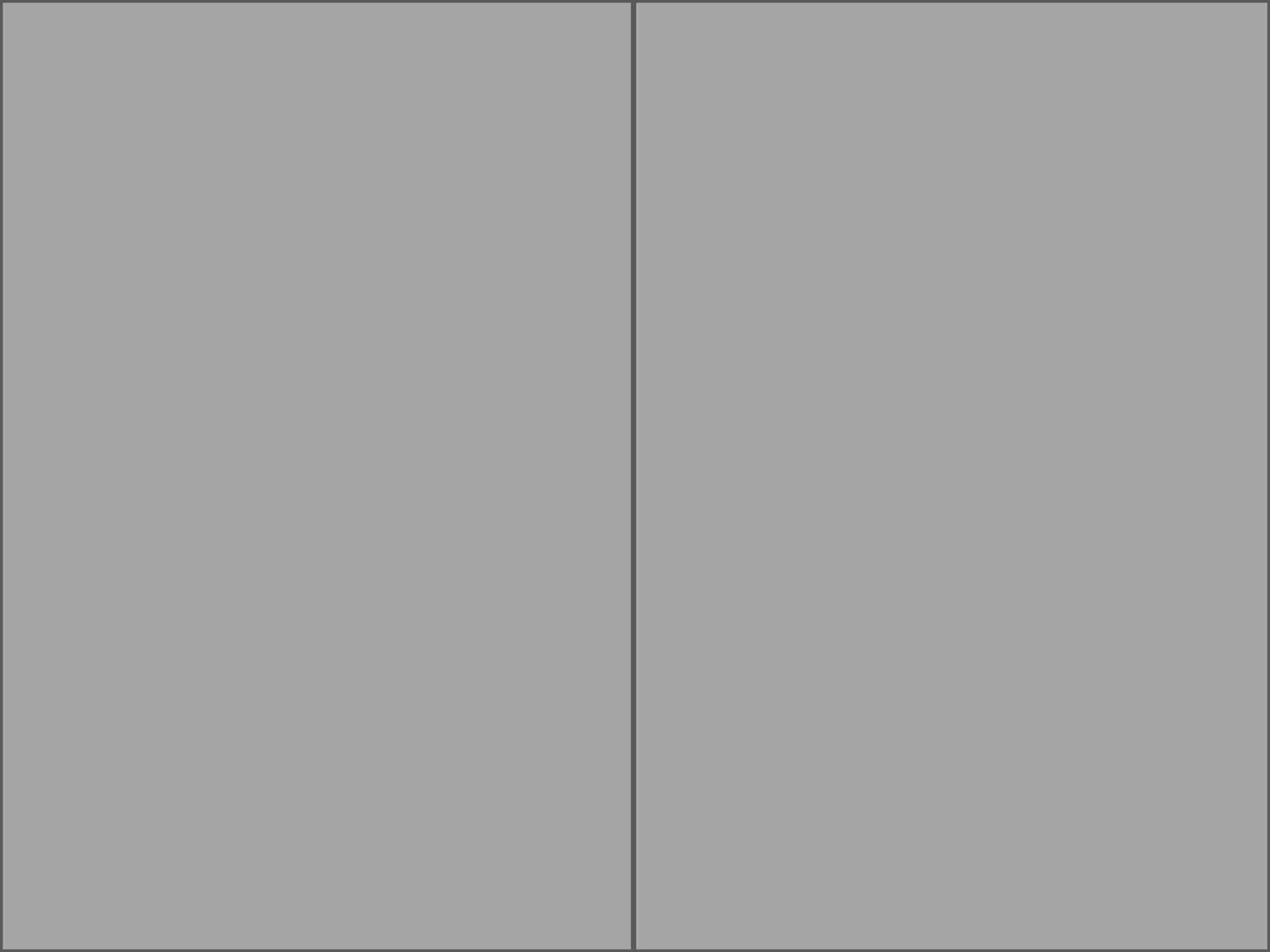


Hoje iremos fazer algumas
comparações dos comandos
do algoritmo com o do PHP.

Venham Comigo!

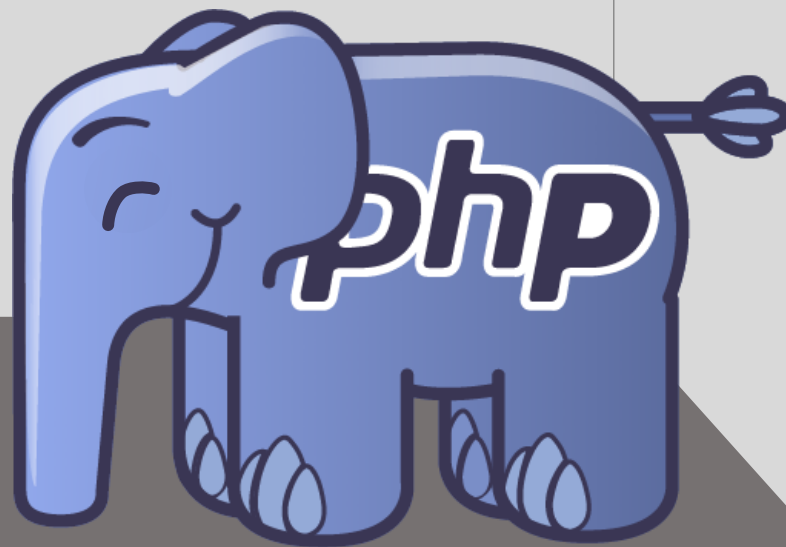






`escreva` (“Uau, como você
fala diferente PHP!”)

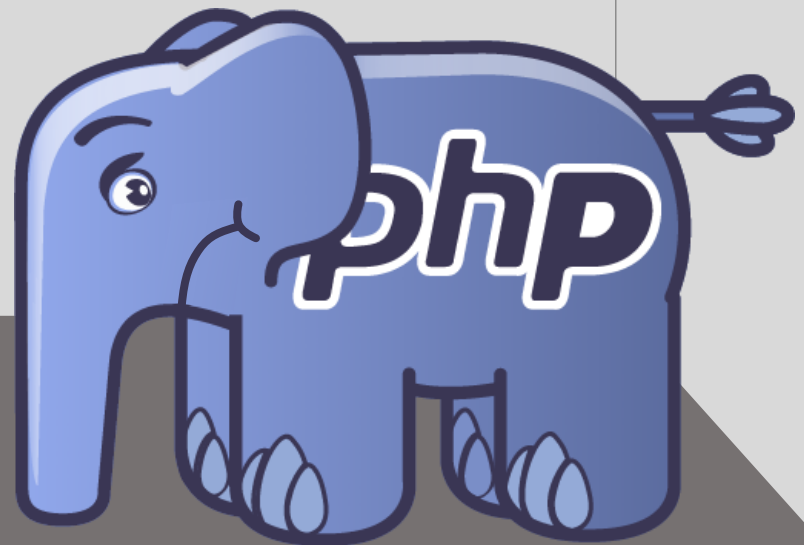
`echo` “Pois é, mas `escreva` e `echo`
significam a mesma coisa! É como
Obrigado e Thanks!”;



`escreval` (“Verdade!”)

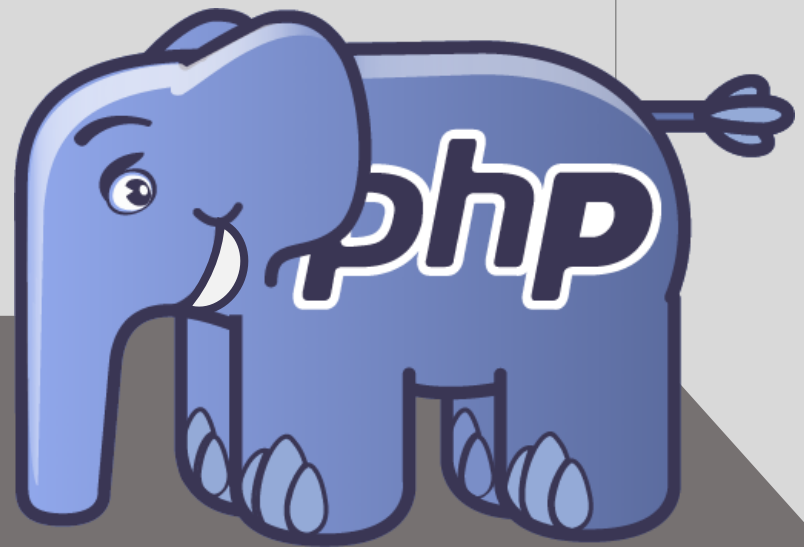
`escreval`(“Com escreval eu posso falar e pular uma linha, você tem um comando assim?”)

`echo` “Não! Não tenho nenhum parecido...”;



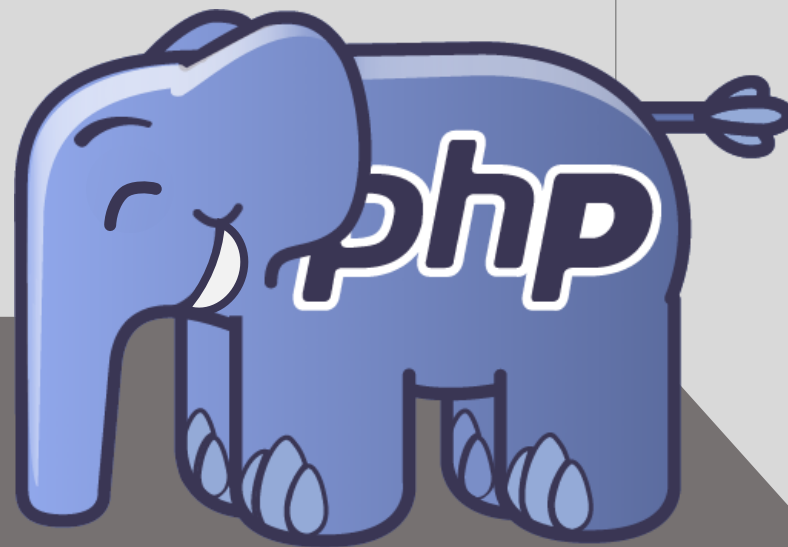


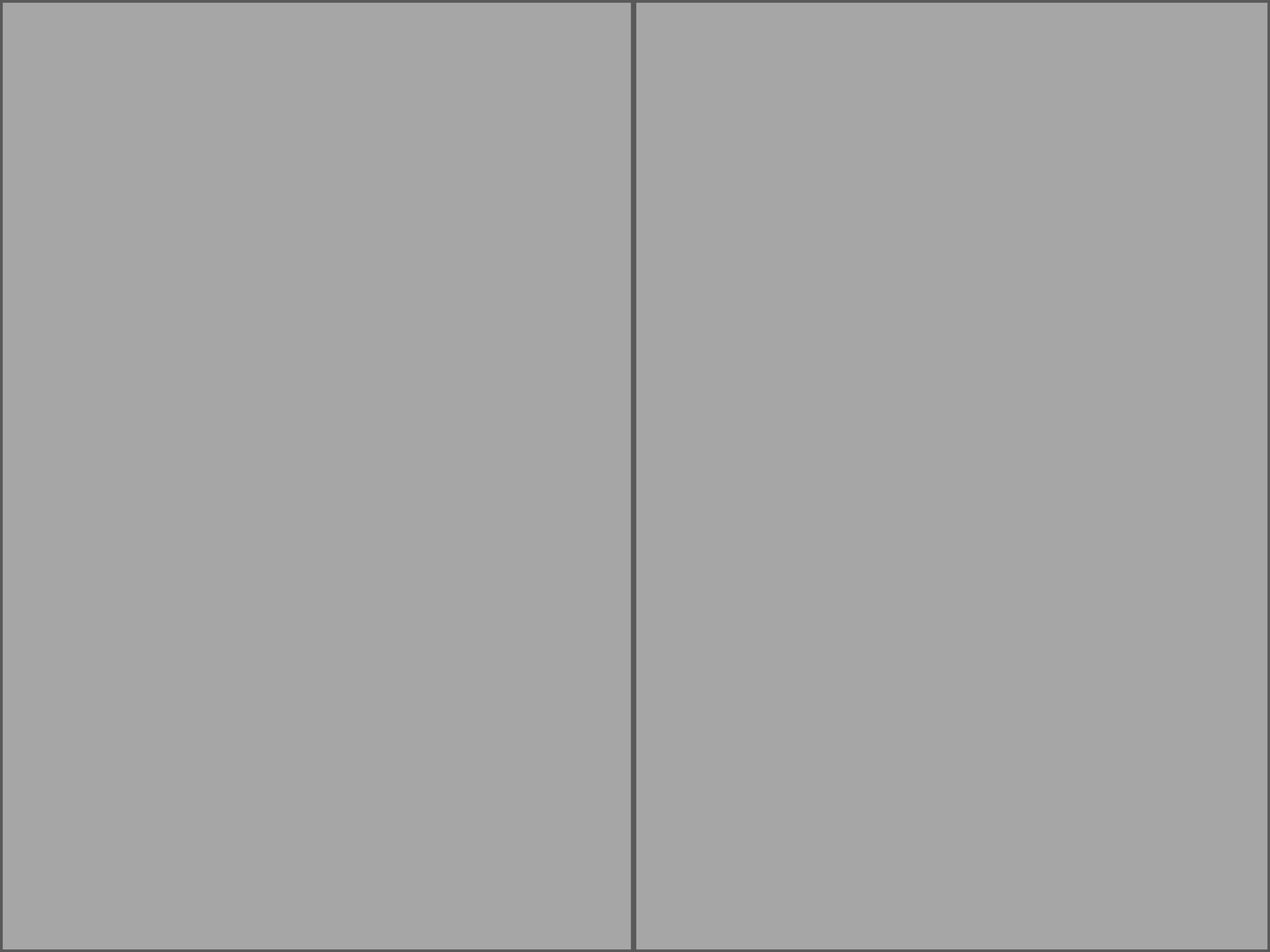
`echo` “Ah! Mas dentro das aspas
do echo posso colocar qualquer
palavra ou tag HTML!”;

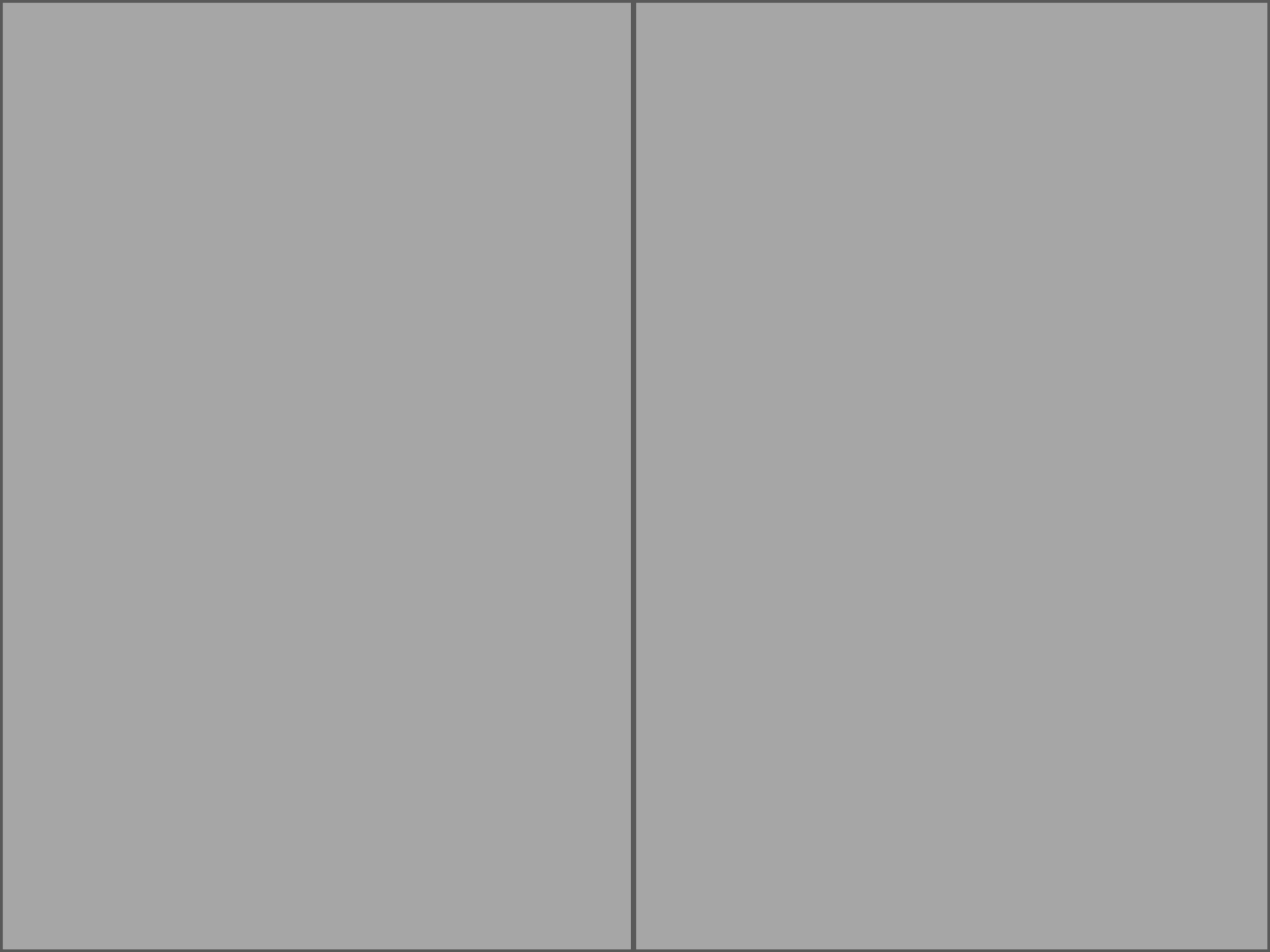




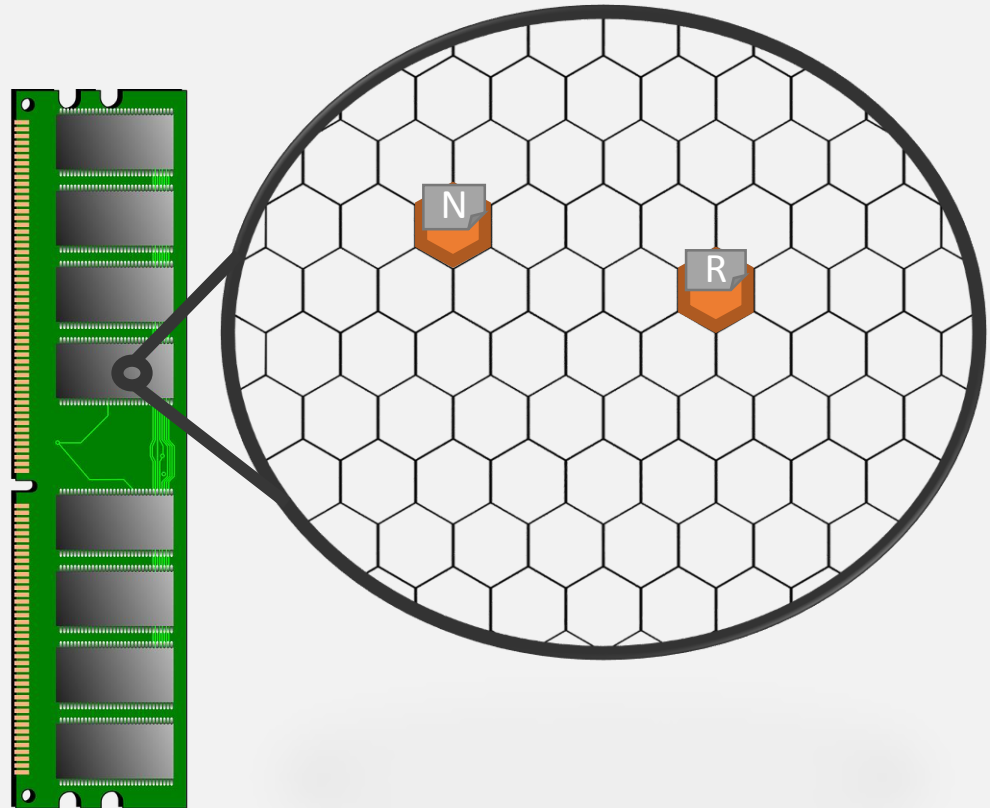
`echo` “Para isso só preciso usar a tag `
`
Consegui, viu?”;







Em Algoritmos e Lógica vimos que variáveis são locais nos quais guardamos algum tipo de informação. No computador, por exemplo, elas ficam na memória RAM.



Time A

BasqueteMasters

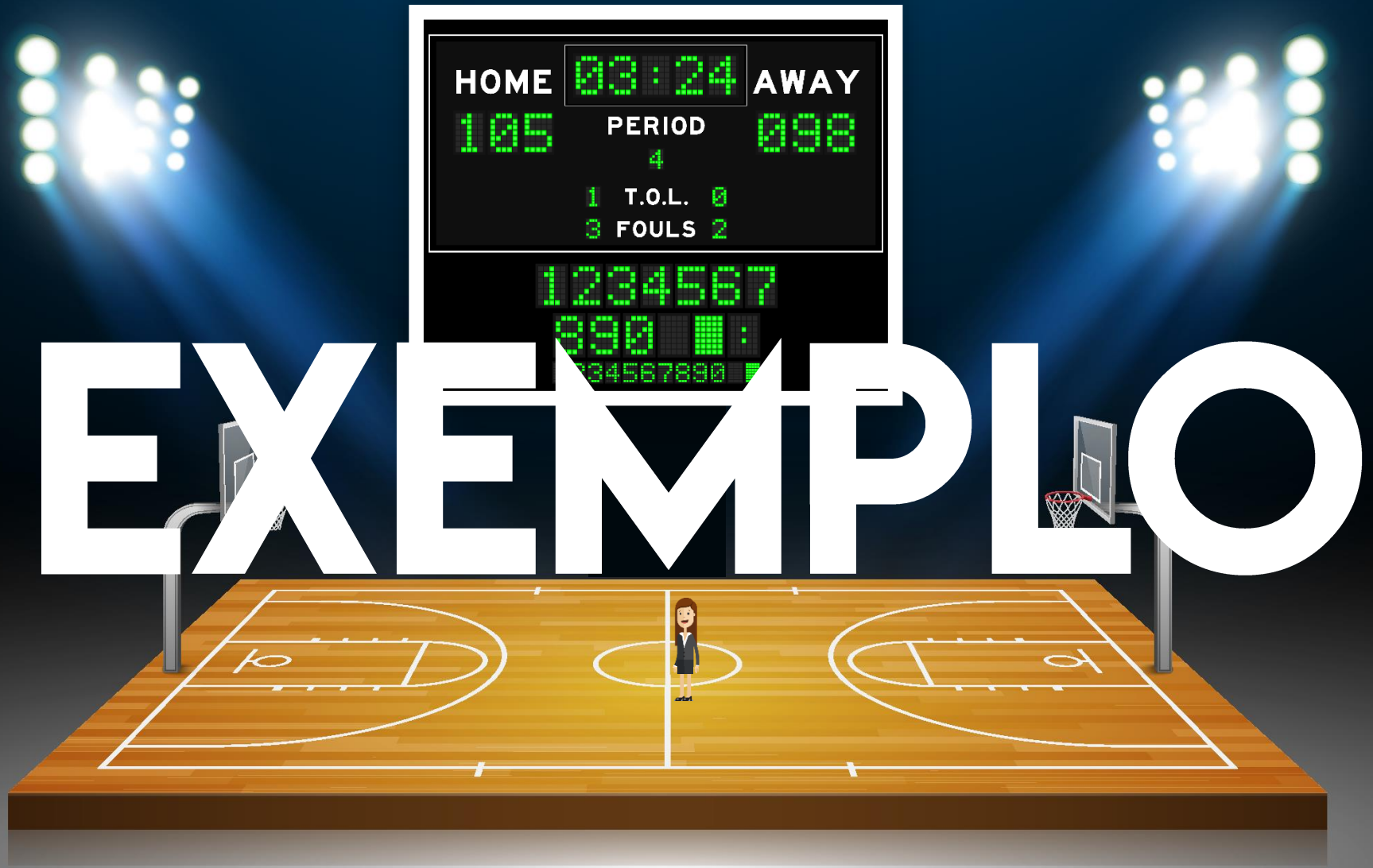
Time B

SuperBasquete



Na ocasião, usamos um jogo de basquete como exemplo. Nele há duas variáveis: TimeA e TimeB.

Elas guardam a pontuação das equipes que estão se enfrentando em uma partida.



SE COLOCARMOS EM ALGORITMO:

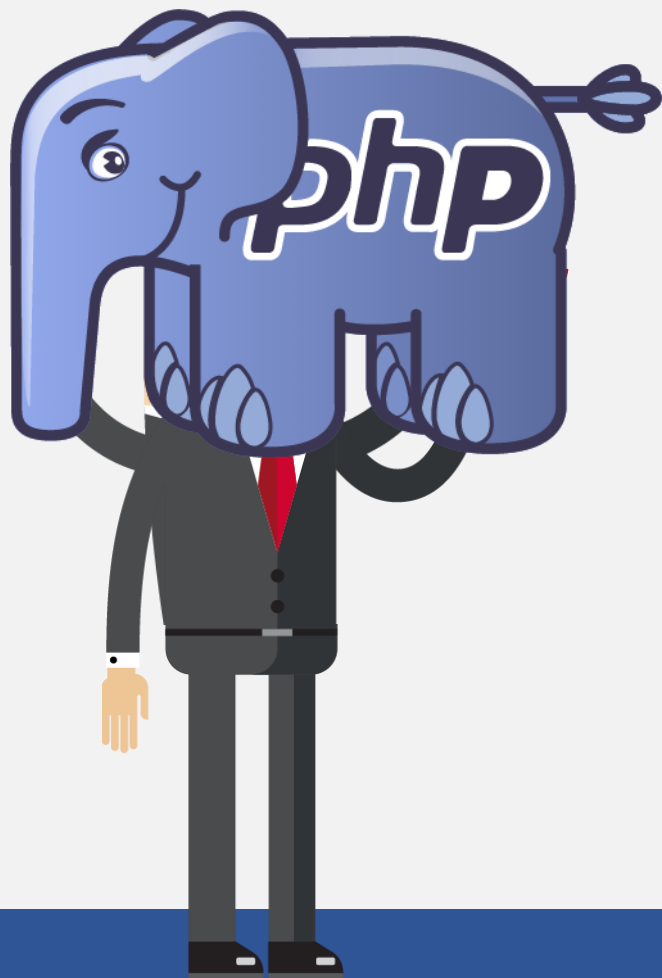


TimeA <- 52

SIGNIFICA QUE TEREMOS:

- uma variável **TimeA**;
- do tipo **inteiro**;
- e de valor **52**.

EM CADA LINHA DE CÓDIGO DA SEGUINTE FORMA:



```
$timeA <= 52;
```

SIGNIFICA QUE TEREMOS:

- uma variável **TimeA**;
- do tipo **inteiro**;
- e de valor **52**.



Perceberam a diferença? Em PHP a variável sempre iniciará com um cifrão e terminará com ponto e vírgula; o nome dela será com letras minúsculas; e o símbolo de atribuição é um sinal de igual.

Algoritmo

TimeA <- 3

PHP

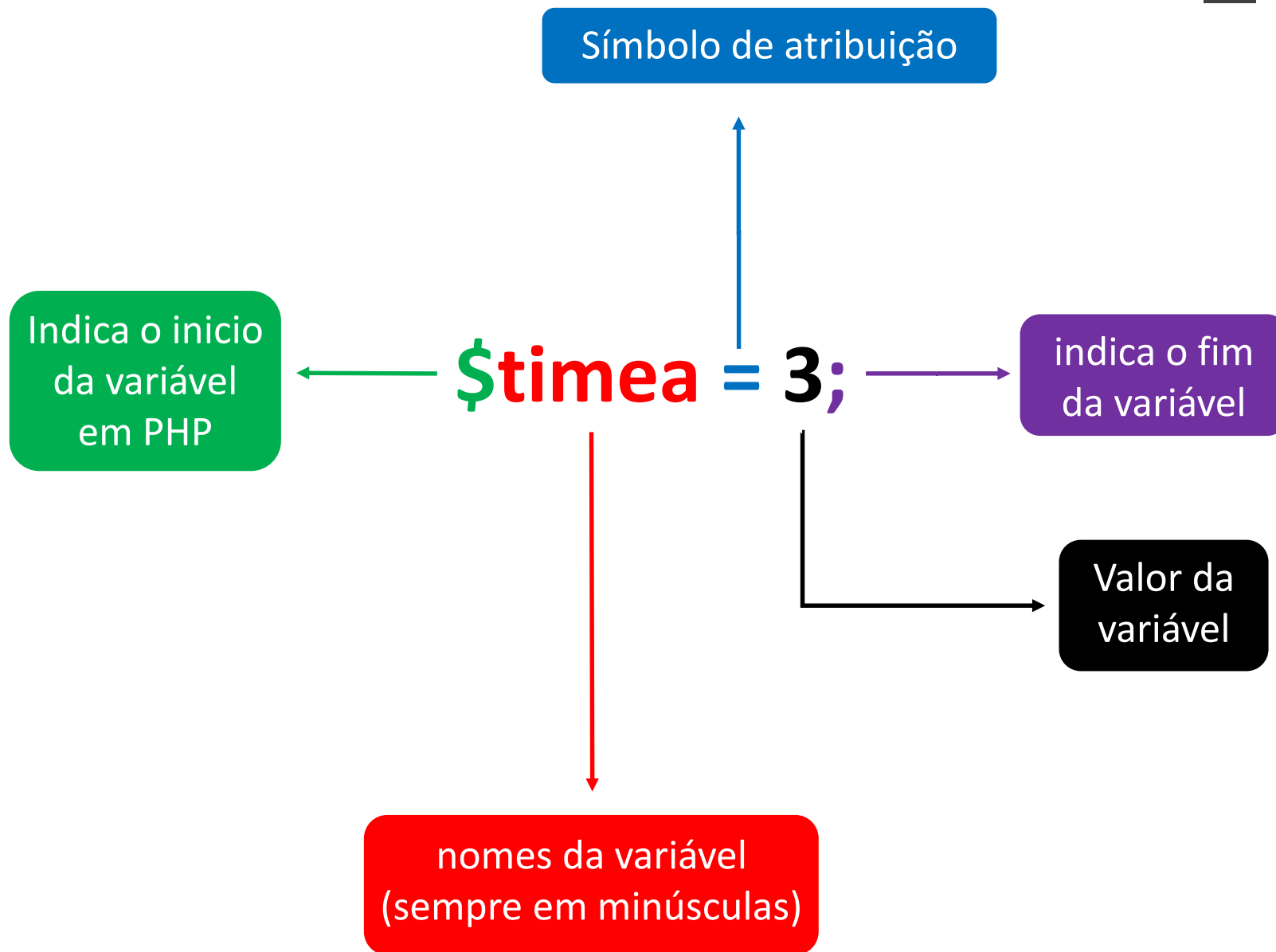
\$timea = 3;



Perceberam a diferença? Já não é mais uma expressão
iniciará com um cifrão, não tem espaço para o ponto e
vírgula; o nome dela é variável, não é uma constante, e
o símbolo de atribuição é o igual, não o seta.

Sem cifrão, sem
variável!







Perceberam a diferença? Já não é mais uma expressão
iniciará com um cifrão, não tem espaço para o ponto e
vírgula; o nome dela é variável, não é uma constante, e
o símbolo de atribuição é o igual, não o seta.

Sem cifrão, sem
variável!



MOMENTO: COISAS DO COTIDIANO

Com Algoritmo e PHP



MOMENTO: COISAS DO COTIDIANO

Com Algoritmo e PHP



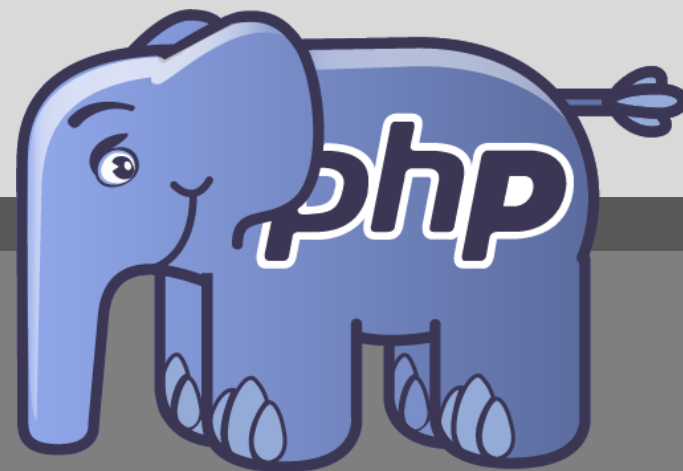
ALGORITMO

COMO O ALGORITMO E O
PHP DIRIAM QUE
ESSA MULHER TEM 22 ANOS?

PHP

`idade <- 22`

`$idade = 22;`



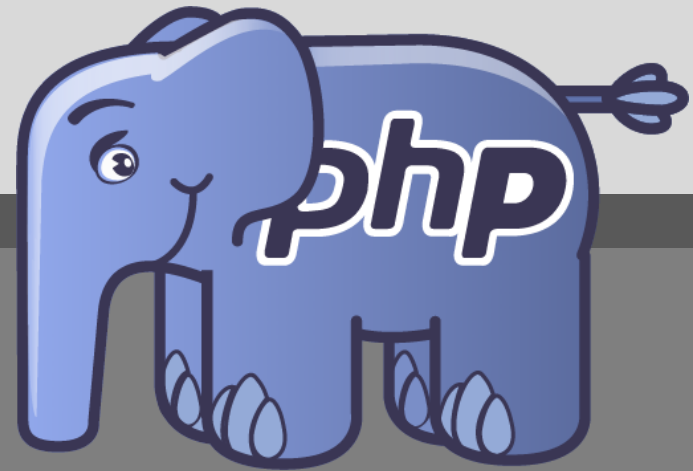
**E SE TIVESSEM
QUE DIZER QUE O
NOME DA MULHER
É KAROL ?**

ALGORITMO

PHP

`nome <- "Karol"`

`$nome = "Karol";`



AGORA, COMO ELES FALARIAM QUE O SALARIO DE KAROL É DE R\$: 5783,64?

ALGORITMO

```
sal <- 5783.64
```

PHP

```
$sal = 5783.64;
```



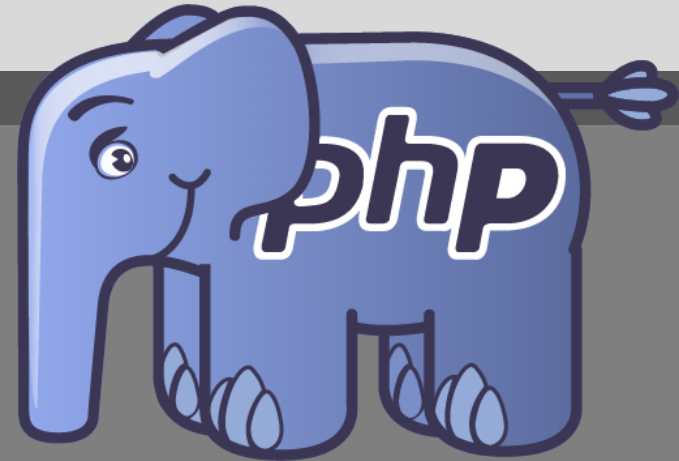
E PARA FINALIZAR, COMO ELES FALARIAM O PASSOU NO CURSO OU?

ALGORITMO

PHP

`curso <- verdadeiro`

`$curso = true;`



Cada uma das variáveis do exemplo anterior é de um tipo diferente. Portanto, em algoritmos nós precisaríamos fazer a declaração da seguinte forma:

idade: inteiro
sal: real
nome: caractere
curso: logico



E em PHP não precisamos declarar os tipos das variáveis.

EXEMPLO DE ALGORITMO

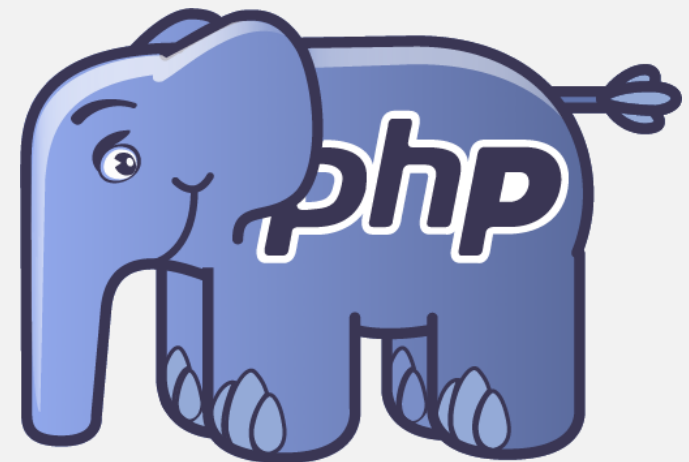
Nome Variável



NOME CARACTERE



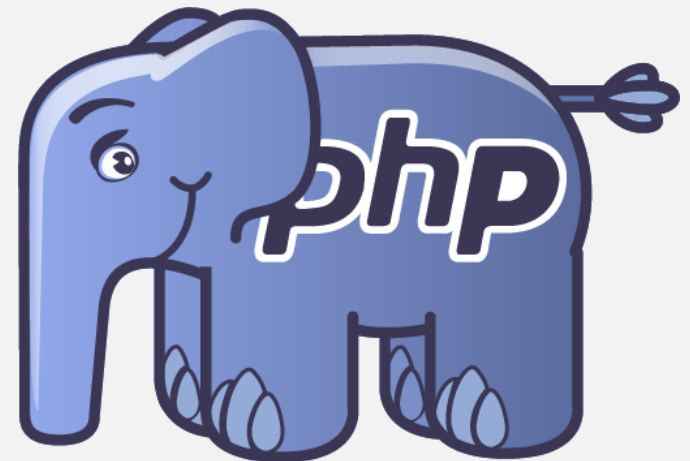
Tipo Variável



Se criamos a variável **timea** e definimos o valor **52** para ela, a variável automaticamente será do tipo **inteira**.

```
$timea = 52;
```

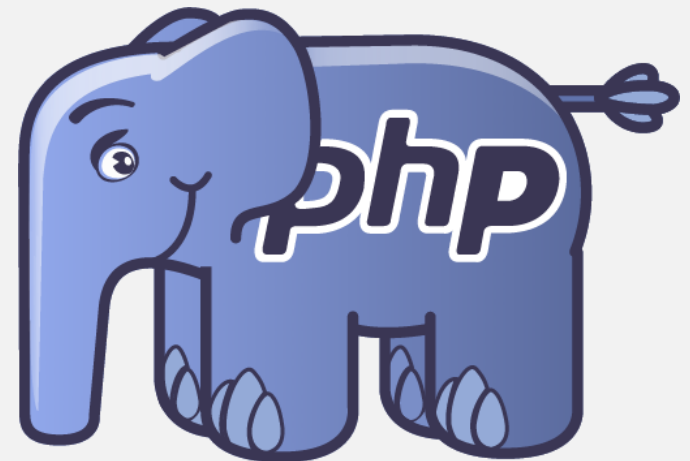
Já que o numero 52 é inteiro,
automaticamente defini o
tipo da variável para inteira.



Se, por algum motivo, mais tarde a variável **timea** receber o valor **52,5** ela se tornará uma variável **real**. Esse processo é feito automaticamente pelo PHP.

```
$timea = 52.5;
```

Isso é chamado de atribuição
é feita por coerção.



Também podemos forçar uma conversão de tipo de dado. Ou seja, podemos fazer com que uma variável seja de um determinado tipo por meio de comando chamado **typecast**.



```
$timea = (string) 52.5;
```


É possível, por exemplo, forçar que uma variável seja do tipo **inteiro** utilizando o typecast **int** ou **integer**; ou que ela seja do tipo **caractere** com o typecast **string**.



```
$timea = ( string ) 52.5;
```

TYPESCAST DE CADA TIPO DE DADO

Inteiro: (int), (integer)

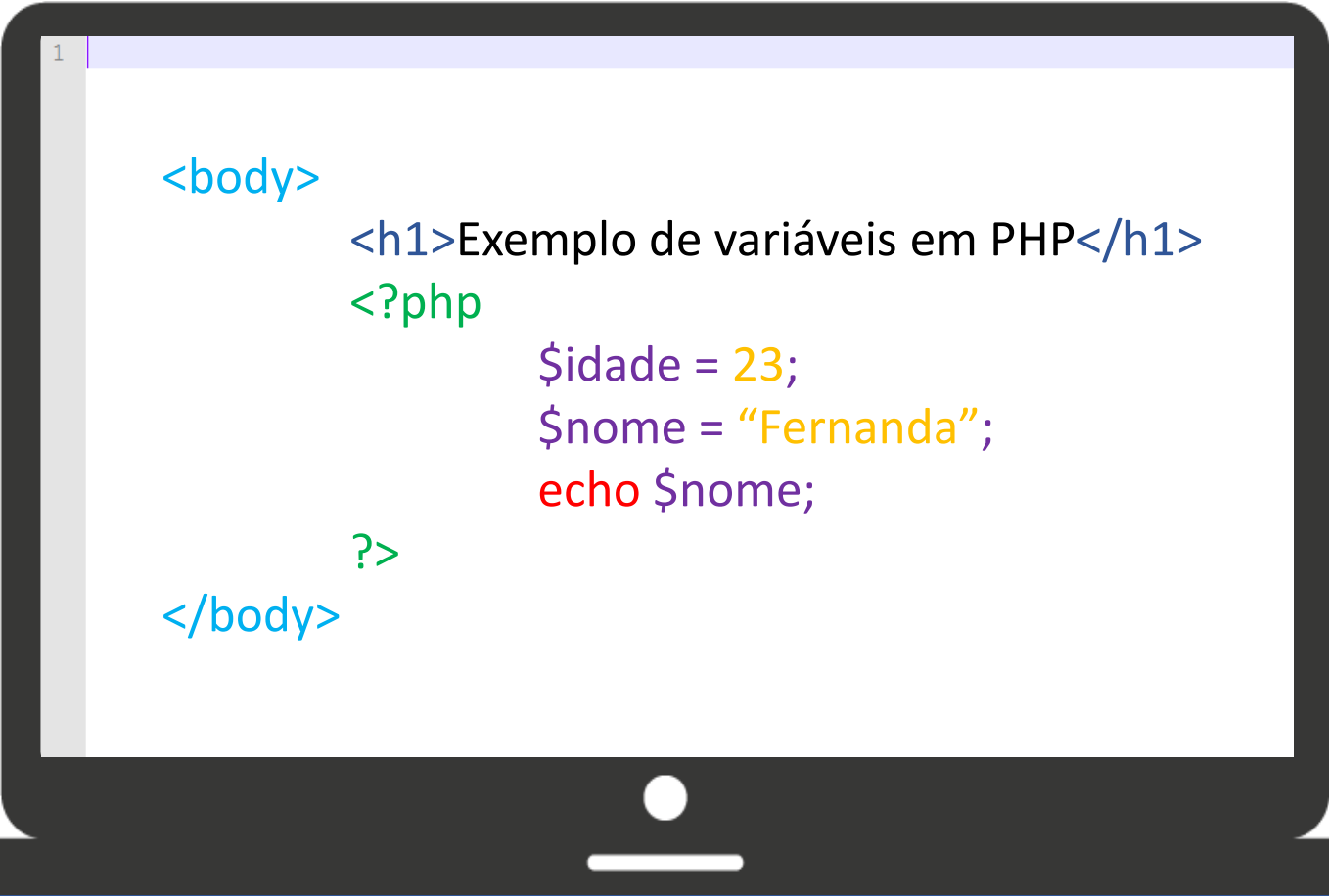
Real: (real), (float), (double)

Caractere: (string)

Lógico: (não existe)

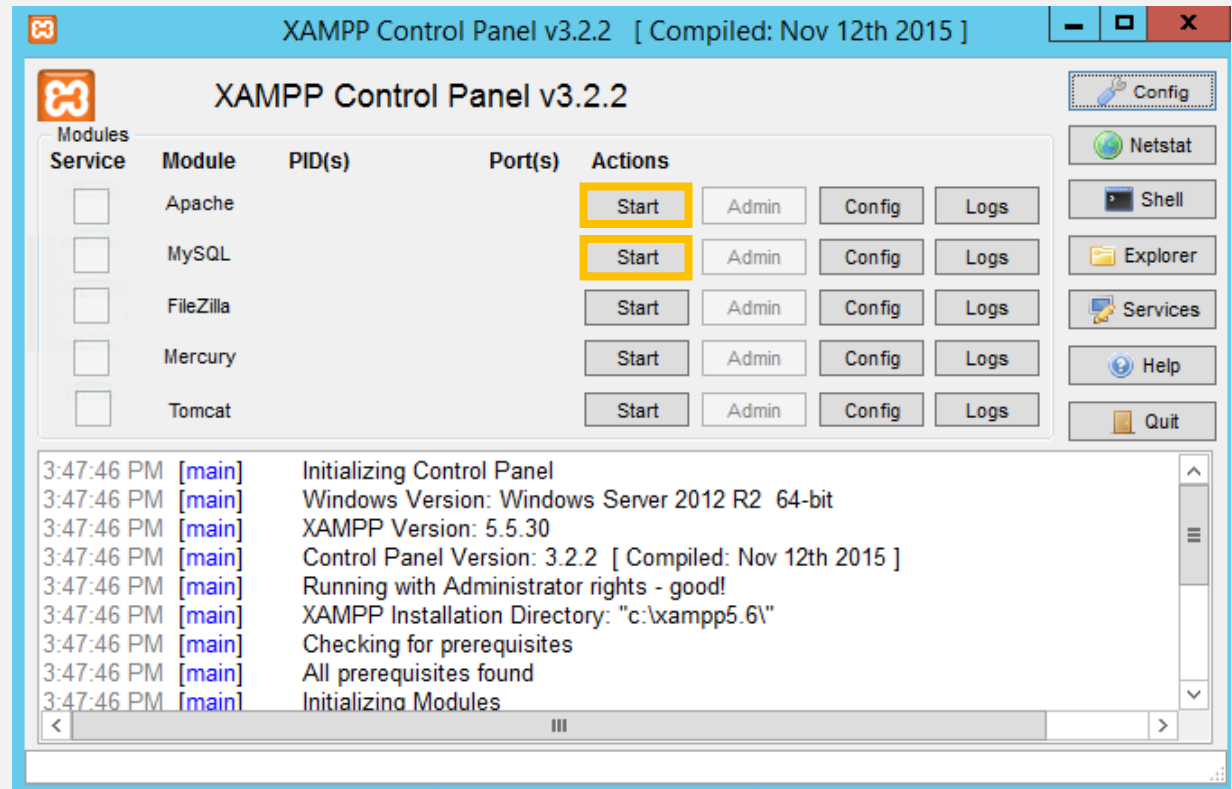
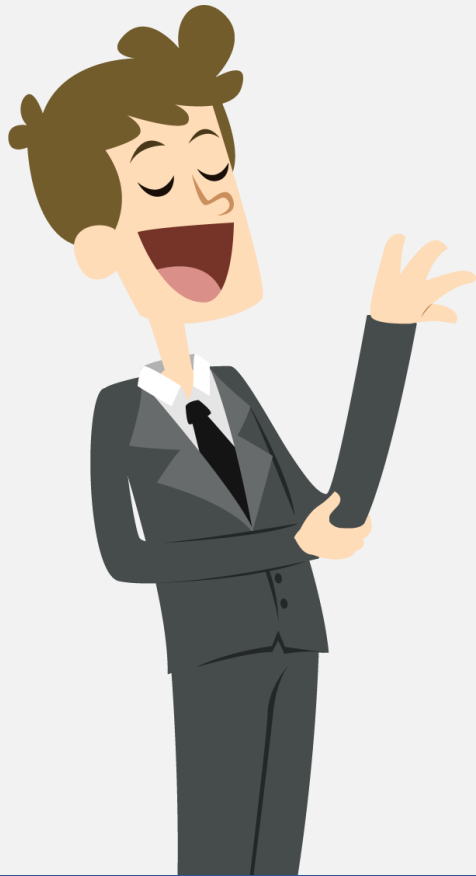


EXEMPLO

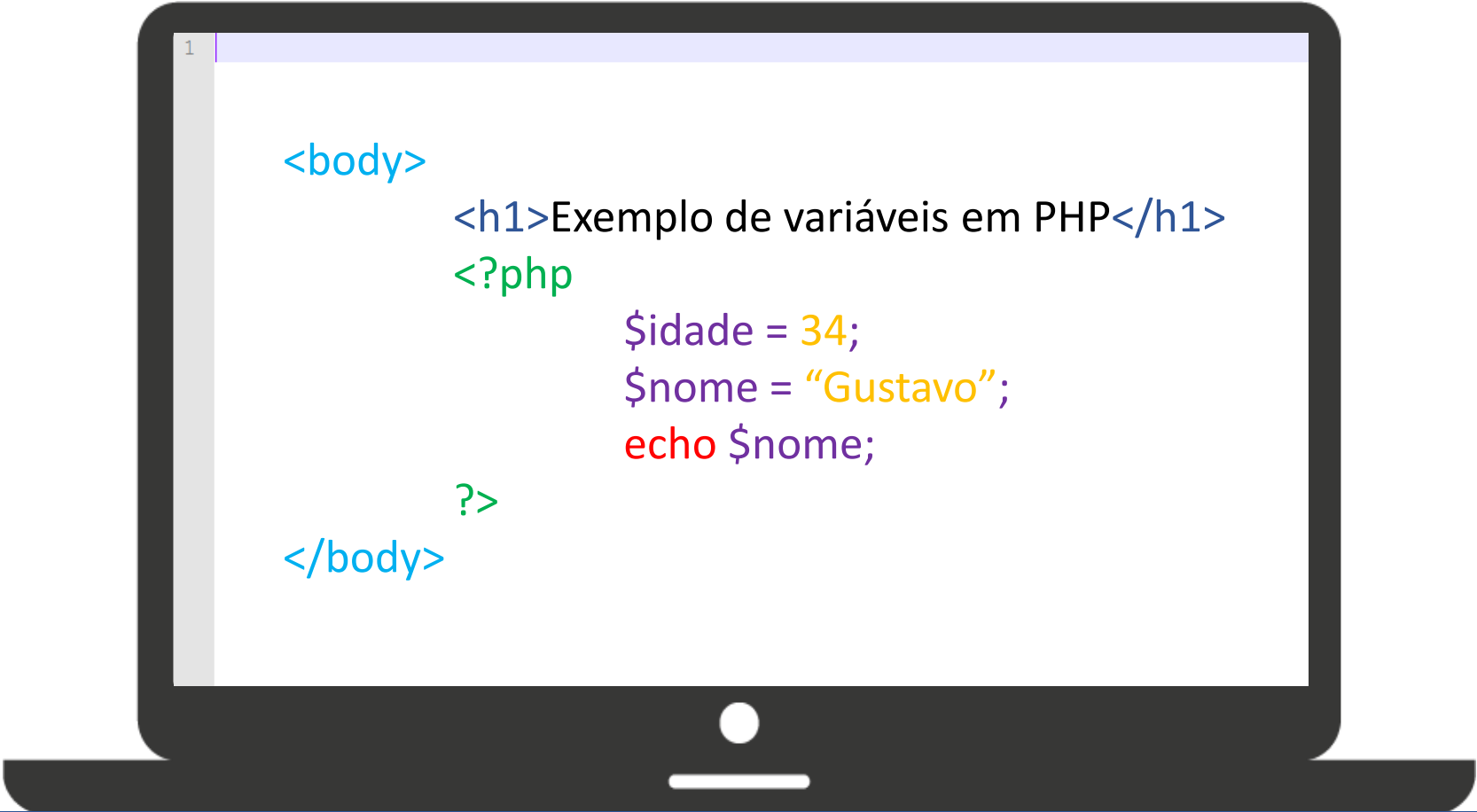


```
1
<body>
  <h1>Exemplo de variáveis em PHP</h1>
  <?php
    $idade = 23;
    $nome = "Fernanda";
    echo $nome;
  ?>
</body>
```

Vamos criar agora um exemplo simples de uso de variáveis em PHP. Antes de começar, no entanto, lembrem-se de abrir o XAMPP e dar o start nos servidores.



EXEMPLO



```
1
<body>
  <h1>Exemplo de variáveis em PHP</h1>
  <?php
    $idade = 34;
    $nome = "Gustavo";
    echo $nome;
  ?>
</body>
```

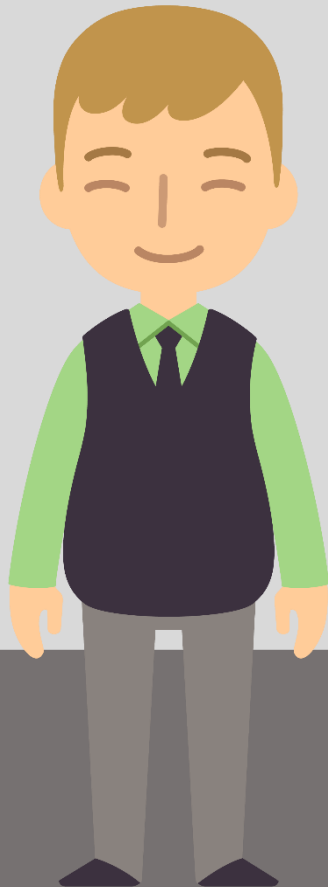
CONCATENAÇÕES EM PHP

Assim como em Algoritmos, em PHP nós também fazemos concatenações de variáveis.

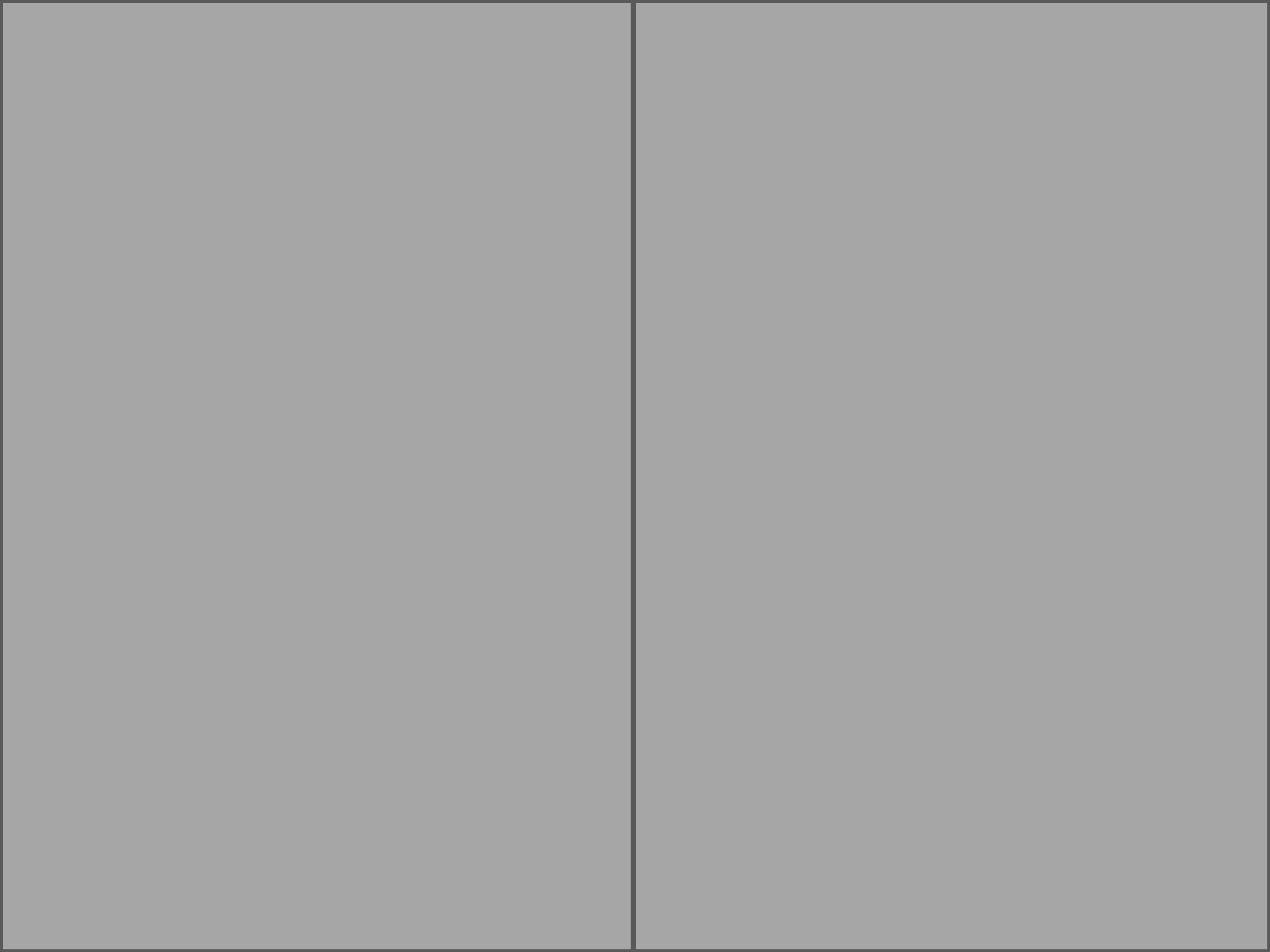
ALGORITMO

```
nome <- "Gustavo"  
idade <- 34
```

```
escreva (nome, "tem", idade, "anos")
```



Ao invés da vírgula, no entanto,
utilizamos o ponto para isso.



PAP

`Nome = "Gustavo";`
`Idade = 34;`

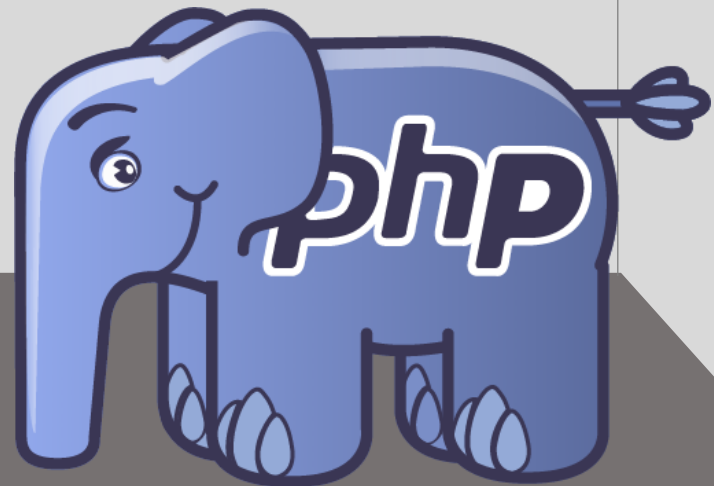
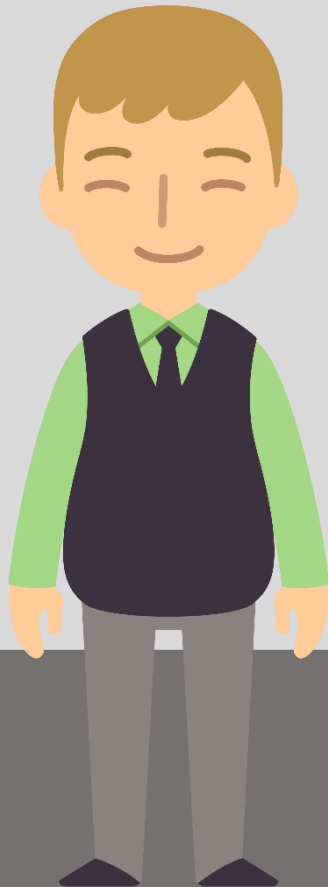
Outra forma de fazermos isso é colocando as aspas na extremidades do comando e eliminando os pontos. Obs: as aspas precisa ser duplas, caso contrário o comando não será interpretado.



PHP

```
$nome = "Gustavo";  
$idade = 34;
```

```
echo "$nome tem $idade anos";
```



A cartoon illustration of a man with brown hair, wearing a dark suit, white shirt, and red tie. He is pointing his right index finger towards a large smartphone. The smartphone screen is white and displays the word "ATIVIDADE" in bold, dark teal capital letters. The phone has a dark blue bezel and a home button at the bottom. There are four orange circles representing fingers at the bottom of the phone, and one orange circle on the left side. The background is a solid blue color.

ATIVIDADE