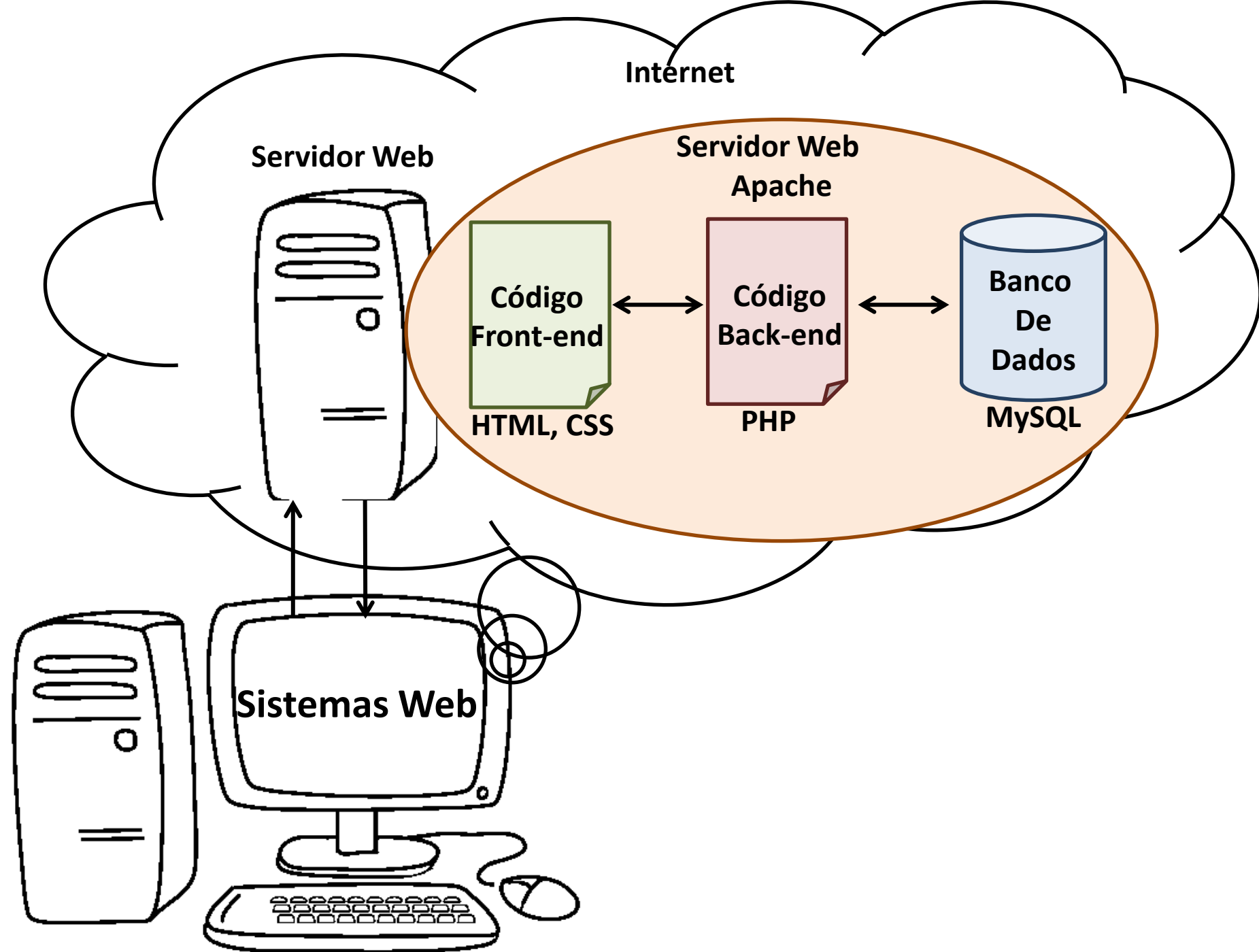


Programação



[1]





Código



PHP: *Hypertext Preprocessor* – Processador de hipertexto

Surgiu em 1994 por Rasmus Lerdorf, com o nome *Personal Home Page Tools*.

Portabilidade - independência de plataforma.

Estruturado e orientado a objetos.

Software gratuito.

Código é executado no servidor.



Sublime Text


Editor de texto (código-fonte)




Prática

Para Baixar XAMPP

 Visual Studio Code

 Sublime Text 3

 Notepad++

→ Notepad++ , Sublime e Visual são editores de texto e de código fonte

<?php

?>

Indica que o código é em PHP
Obrigatório

<?php

echo 'Oi Mundo!';

?>

echo – escrever na tela – (ecoar na tela)

Salvar

Local: Windows: C:\xampp\htdocs ou
Linux: /opt/xampp/htdocs

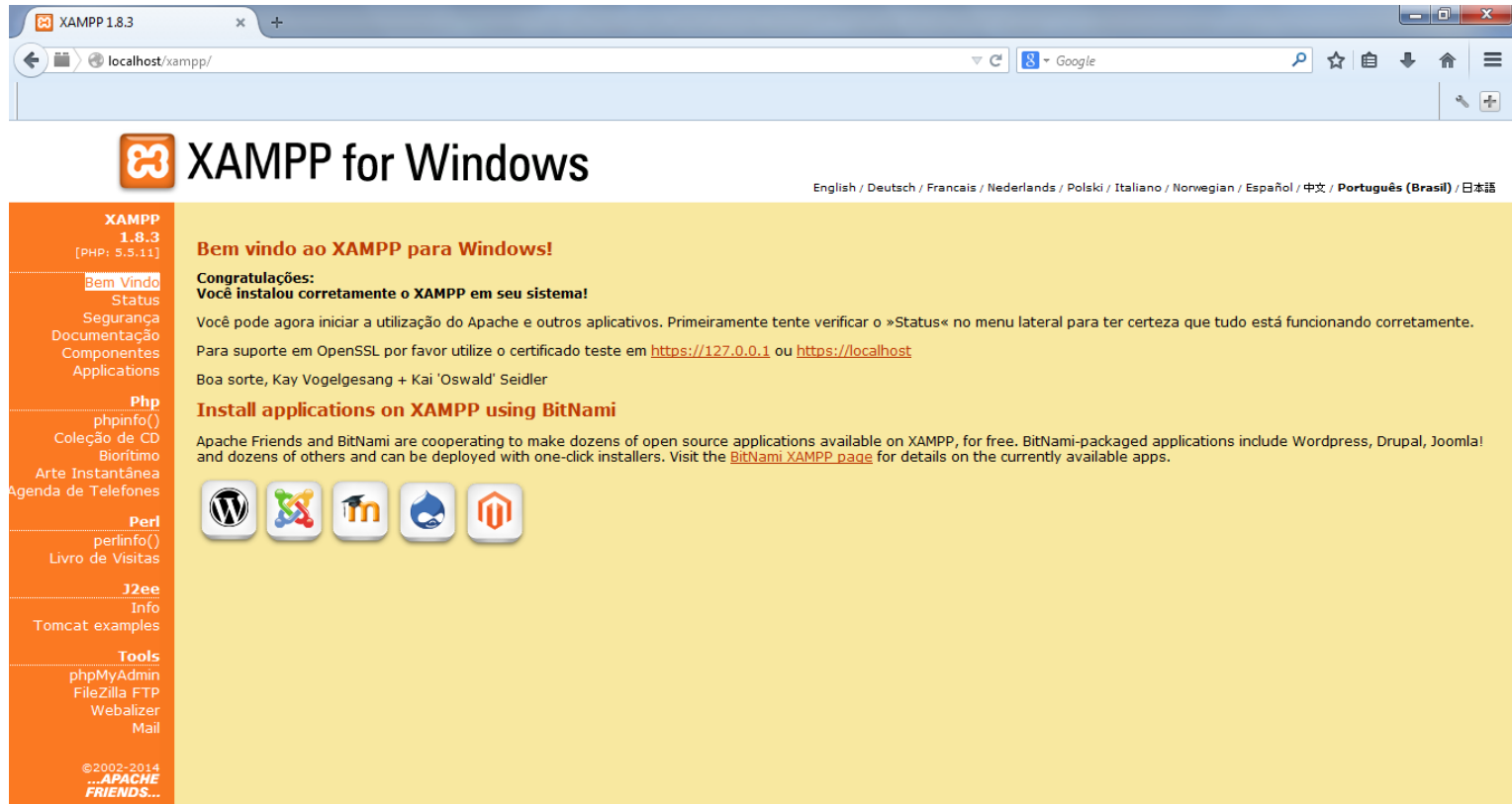
Cria uma pasta: variaveis

Nome do arquivo: index.php

Tipo: Todos os arquivos

Prática

Acessar



localhost/nomePasta/nomeArquivo.php

Prática

<?php

echo 'Oi Mundo!'; // echo para exibir

/*

Comentário


com

mais

Linhas

*/

?>



// comentário
/* */ comentários com mais linhas

Prática

Variáveis

```
<?php
```

```
    $var = 'Oi Mundo!';
```

```
    echo $var;
```

```
?>
```



Nome da variável ou identificador.

\$ símbolo para definir variáveis.

**Não precisa declarar variáveis,
basta atribuir um valor a ela.**

Prática

Variáveis


```
<?php
```

```
$var = 'Oi Mundo!';
```

```
$Var = 'Oi';
```

```
echo $var, $Var;
```

```
?>
```



Case sensitive – sensível entre letras maiúsculas e minúsculas.

Prática

Variáveis

\$4 – errado

Dicas para definir nome de variáveis - identificadores.

```
<?php
```

```
$var = 'Oi Mundo!';
```

```
$Var = 'Oi';
```

```
$4 = 123;
```

```
echo $var, $Var, $4;
```

```
?>
```

Nunca inicie com números;

Nunca utilize espaços em branco no meio do identificador;

Nunca utilize caracteres especiais (!@#%^&*/[[]{});

Evite ter mais do que 15 caracteres;

Nome de variáveis devem ser significativos e transmitir a ideia do conteúdo;

Utilize preferencialmente palavras em minúsculo (separadas pelo caractere “_” ou somente as primeiras letras em maiúsculo, quando da ocorrência e mais palavras).

Um nome de variável válido se inicia com uma letra ou sublinhado, seguido de qualquer número de letras, algarismos ou sublinhados.

Prática

Variáveis

<?php

```
$var = 'Oi Mundo!';  
$Var = 'Oi';
```

```
$_4 = 123;
```

```
echo $var, $Var, $_4;
```

?>

Variáveis tipo string

Pode utilizar aspas simples "
ou aspas duplas ""

Variável tipo numérico

<?php

\$var = 1;
\$var2 = 3;

echo \$var + \$var2;

?>

“\$var + \$var2” – resultado: 1 + 3

\$var + \$var2 – resultado: 4

'\$var + \$var1' – resultado: \$var + \$var2 (não processa)

<?php

```
$var = 1;  
$var2 = 3;
```

```
echo 'Resultado ', $var + $var2;
```

?>

‘exibe’, processa o cálculo

<?php

```
$var = 1;  
$var2 = 3;
```

```
echo 'Resultado <br>', $var + $var2;  
//echo 'Resultado', '<br>', $var + $var2;
```

?>

“
” ou ‘
’ é o enter
pode ser com aspas simples ou duplas

Prática

Exercício

Defina uma variável nome e uma variável sobrenome.

Insira algum nome e sobrenome.

Exiba primeiro o sobrenome e depois o nome.

Salve o arquivo com o nome: nome.php

Prática

Exercício

Defina uma variável nome e uma variável sobrenome.

Insira algum nome e sobrenome.

Exiba primeiro o sobrenome e depois o nome.

```
<?php
```

```
$nome = "Marcos Alexandre";
```

```
$sobrenome = "Silva";
```

```
echo "$sobrenome, $nome";
```

```
?>
```

Se tirar as "" do echo não aparecem a "," e o espaço

```
<?php
```

```
echo strtoupper('convertendo em maiusculo');
```

```
echo mb_strtoupper('convertendo em maiúsculo');
```



Strtoupper – converte para
maiúsculo

Strtolower – converte para
minúsculo

```
?>
```

```
<?php
```

```
$var = 'teste';
```

```
echo strtoupper($var);
```

```
?>
```


Prática

Operadores Aritméticos

Operadores	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão)

```
<?php
```

```
$n1 = 2;
```

```
$n2 = 2;
```

```
echo $n1+$n2;
```



ou

```
echo 2 + 2;
```

```
?>
```

Prática

Operadores Aritméticos

```
<?php
```

```
$n1 = 2;
```

```
$n2 = 2;
```

```
echo $n1+$n2*5;
```

```
?>
```

1º - Multiplicação e divisão mesma prioridade

2º - Soma e Subtração mesma prioridade

Em caso de mesma prioridade, o cálculo é feito da esquerda para direita

Prática

Operadores Aritméticos

Prioridades

$$6/2*(1+2)$$

Resultado 1 ou 9

1?

linha 1) $6 \div 2 \times (1 + 2) =$

linha 2) $6 \div 2 \times (3) =$

linha 3) $6 \div 2 \times 3 =$

linha 4) $6 \div 6 = 1$

9?

linha 1) $6 \div 2 \times (1 + 2) =$

linha 2) $6 \div 2 \times (3) =$

linha 3) $6 \div 2 \times 3 =$

linha 4) $3 \times 3 = 9$

Prática

Operadores Aritméticos

Prioridades

$$6/2*(1+2)$$

```
<?php  
    echo 6/2*(1+2);  
?>
```

Resultado = **9**

O cálculo será executado primeiro multiplicando e dividindo os valores e depois somando e subtraindo.

**Defina a ordem do cálculo utilizando parênteses.
Parênteses têm prioridade.**

```
<?php  
    echo 6/(2*(1+2));  
?>
```



Resultado = **1**

Prática

Operadores Aritméticos

Conversão

```
<?php
```

```
$a = '10';
```

```
echo $a+10;
```

```
?>
```

O PHP realiza automaticamente a conversão de tipos em operações.

Prática

Exercício

Defina uma variável valorA e uma variável valorB.

Insira o número 5 no valorA e 3 no valorB.

Calcule valorA menos valorB.

Exiba o resultado.

Resultado:

2

Prática

Exercício

Defina uma variável valorA e uma variável valorB.

Insira o número 5 no valorA e 3 no valorB.

Calcule valorA menos valorB.

Exiba o resultado.

```
<?php
```

```
    $valorA = 5;
```

```
    $valorB = 3;
```

```
    echo $valorA - $valorB;
```

```
?>
```

Prática

Exercício

Defina quatro variáveis valorA, valorB, valorC, valorD.

Insira o numero 10 para valorA, numero 5 para valorB, numero 8 para valorC, numero 1 para valorD.

Calcule (valorA vezes 3) menos valorB mais (valorC dividido valorD)

Exiba a frase 'O resultado eh' antes do resultado.

Resultado:

O resultado eh 33

Prática

Exercício

Defina quatro variáveis valorA, valorB, valorC, valorD.
Insira o numero 10 para valorA, numero 5 para valorB, numero 8 para valorC, numero 1 para valorD.
Calcule (valorA vezes 3) menos valorB mais (valorC dividido valorD)
Exiba a frase 'O resultado eh' antes do resultado.

```
<?php
```

```
$valorA = 10;
```

```
$valorB = 5;
```

```
$valorC = 8;
```

```
$valorD = 1;
```

```
echo 'O resultado eh ';
```

```
echo ($valorA*3)-$valorB+($valorC/$valorD);
```



```
//echo 'O resultado eh ', ($valorA*3)-$valorB+($valorC/$valorD);
```

```
?>
```

Array – Lista de valores armazenados na memória.

Podem ser de tipos diferentes, como strings, números, etc.

Acessados a qualquer momento – cada valor é relacionado a uma chave.

<?php	<div data-bbox="618 578 937 621">0 1 2</div> <div data-bbox="241 628 975 735">\$carros = array ('Palio', 'Corsa', 'Gol'); echo \$carros[1];</div>		resultado - corsa
?>			
<?php	<div data-bbox="241 999 975 1106">\$carros = array ('Palio', 'Corsa', 123); echo \$carros[2];</div>		tipos diferentes: strings, números
?>			

Array – Lista de valores armazenados na memória.

Podem ser de tipos diferentes, como strings, números, etc.

Acessados a qualquer momento – cada valor é relacionado a uma chave.

```
<?php
```

```
$pessoa['nome'] = 'Maria da Silva';
```

```
$pessoa['rua'] = '7 de Setembro';
```

```
$pessoa['bairro'] = 'Camobi';
```

```
$pessoa['cidade'] = 'Santa Maria';
```

```
echo $pessoa['nome'];
```

```
?>
```

A **chave** pode ser string ou inteiro não negativo.
Aspas "" ou "

<?php

```
$pessoa['nome'] = 'Maria da Silva';  
$pessoa['rua'] = '7 de Setembro';  
$pessoa['bairro'] = 'Camobi';  
$pessoa['cidade'] = 'Santa Maria';
```

```
// echo $pessoa; ERRADO
```

?>

```
print_r($pessoa); —————> resultado - Array ( [nome] => Maria da Silva [rua] => 7  
de Setembro [bairro] => Camobi [cidade] => Santa  
Maria )
```

print_r - Imprime informação.

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 'Resultado' , 10, 25.

Some os valores 5 e 25, em seguida multiplique o resultado por 10

(Utilize os valores do array).

Exiba a frase 'Resultado' do array com o resultado do cálculo.

Resposta:

Resultado 300

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 'Resultado' , 10, 25.

Some os valores 5 e 25, em seguida multiplique o resultado por 10

(Utilize os valores do array).

Exiba a frase “Resultado” do array com o resultado do cálculo.

Resposta:

Resultado 300

<?php

\$valores = array (5, 'Resultado ', 10, 25);

echo \$valores[1], (\$valores[0]+\$valores[3])*\$valores[2];

?>

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: Carlos, 9, 7, Bianca, 7, 10.

Calcule a média aritmética de Carlos e Bianca, e exiba o resultado.

Lembrando que a média aritmética é a soma dos números dividida pela quantidade.

Resposta:

Media de Bianca eh: 8.5

Media de Carlos eh: 8



Os nomes (Bianca, Carlos) devem ser obtidos pelo array.

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: Carlos, 9, 7, Bianca, 7, 10.

Calcule a média aritmética de Carlos e Bianca, e exiba o resultado.

Lembrando que a média aritmética é a soma dos números dividida pela quantidade.

Resposta:

Media de Bianca eh: 8.5

Media de Carlos eh: 8



Os nomes (Bianca, Carlos) devem ser obtidos pelo array.

<?php

```
$media = array ('Carlos', 9, 7, 'Bianca', 7, 10);  
echo 'Media de ', $media[3], ' eh: ', ($media[4]+$media[5])/2;  
echo '<br>Media de ', $media[0], ' eh: ', ($media[1]+$media[2])/2;
```

?>

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 20.

Multiplique os valores 5 e 20. O resultado dessa multiplicação deve substituir o numero 5 inserido no array.

Exiba o conteúdo da posição que estava o número 5.

Resposta:

100

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 20.

Multiplique os valores 5 e 20. O resultado dessa multiplicação deve substituir o numero 5 inserido no array.

Exiba o conteúdo da posição que estava o número 5.

Resposta:

100

```
<?php
```

```
    $valores = array (5, 20);  
    $valores[0] = $valores[0]*$valores[1];
```

```
    echo $valores[0];
```

```
?>
```

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 20.

Multiplique os valores 5 e 20. O resultado dessa multiplicação deve substituir o numero 5 inserido no array.

Exiba o conteúdo da posição que estava o número 5.

Resposta:

100

Prática

Exercício

(Não precisa utilizar chave para identificar os elementos do array)

Defina um array com os valores: 5, 20.

Multiplique os valores 5 e 20. O resultado dessa multiplicação deve substituir o numero 5 inserido no array.

Exiba o conteúdo da posição que estava o número 5.

Resposta:

100

```
<?php
```

```
    $valores = array (5, 20);  
    $valores[0] = $valores[0]*$valores[1];
```

```
    echo $valores[0];
```

```
?>
```

Operadores Atribuição

Operadores	Descrição
=	Atribui
+=5	Soma 5
-=5	Subtrai 5
*=5	Multiplica por 5
/=5	Divide por 5

Operadores Atribuição

Operadores	Descrição
=	Atribui
+=5	Soma 5
-=5	Subtrai 5
*=5	Multiplica por 5
/=5	Divide por 5

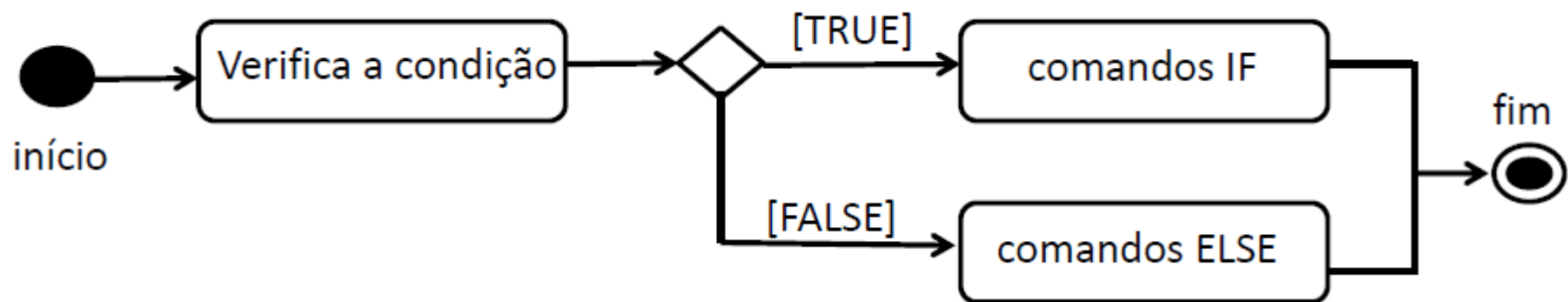
<?php

\$var = 0; —————> Precisa declarar antes de utilizar os operadores
\$var += 10;

echo \$var;

?>

Estruturas de controle - IF



Operadores Relacionais

Operadores	Descrição
==	Igual. Resulta verdadeiro [TRUE] se expressões forem iguais
!= ou <>	Diferente. Resulta verdadeiro se expressões forem diferentes
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual

Estruturas de controle - IF

```
<?php
```

```
    $var = 0;
```

```
    $var += 10;
```

```
    if ($var==10){
```

```
        echo 'igual';
```

```
    }
```

```
    else{
```

```
        echo 'diferente';
```

```
    }
```

```
?>
```

elseif ou else if

(junto ou separado)

Para testar a próxima condição, se a condição anterior não for TRUE

Exercício

Defina um array e insira os numeros 6, 10, 3, 5

Some 15 ao numero 3 inserido no array

O total deve substituir o numero 3 inserido no array

Após essa soma,

Se o total for igual a 18

Some os números 6 e 10 inseridos no array mais o valor 20.

O total deve substituir o numero 6 inserido no array.

Exiba o total

Senão

Some os números 5 e 10 inseridos no array e depois multiplique pelo valor 2.

O total deve substituir o numero 5 inserido no array.

Exiba o total

Resultado: **36**

Exercício

<?php

```
$numero = array (6, 10, 3, 5);
```

```
// $numero[2] = $numero[2] + 15;
```

```
$numero[2] += 15; //some 15 ao numero 3 inserido no array
```

```
if ($numero[2] == 18){
```

```
    $numero[0] = $numero[0] + $numero[1] + 20;
```

```
    echo $numero[0];
```

```
}
```

```
else {
```

```
    $numero[3] = ($numero[3] + $numero[1]) * 2;
```

```
    echo $numero[3];
```

```
}
```

?>

Operadores Lógicos

Utilizados para combinar expressões lógicas, agrupando testes condicionais

Exemplos de expressões

\$a poderia ser **\$numero1 == 11**

\$b poderia ser **\$numero2 == 20**

Operadores	Descrição
(\$a and \$b)	E: Verdadeiro [TRUE] se tanto \$a quanto \$b forem verdadeiros
(\$a or \$b)	OU: Verdadeiro [TRUE] se \$a ou \$b forem verdadeiros
(\$a xor \$b)	XOR: Verdadeiro se \$a ou \$b são verdadeiros, <i>mas não ambos</i> . XOR: Verdadeiro [TRUE] se \$a ou \$b forem verdadeiros, de forma exclusiva.
(\$a && \$b)	E: Verdadeiro [TRUE] se tanto \$a quanto \$b forem verdadeiros
(\$a \$b)	OU: Verdadeiro [TRUE] se \$a ou \$b forem verdadeiros

&& e || têm prioridade sobre and e or.

Prática

```
<?php
```

```
    $valor1 = 10;
```

```
    $valor2 = 20;
```

```
    $valor3= $valor1 + $valor2;
```

```
    if($valor1 == 10 and $valor3 == 20){
```

```
        echo 'Entrou no and';
```

```
    }
```

```
    elseif ($valor1 == 10 or $valor3 == 20){
```

```
        echo 'Entrou no or';
```

```
    }
```

```
?>
```

Exercício

Defina um array e insira: 8, “Aumentou.”, 50, “Diminuiu.”

Defina: a variável salário e insira o numero 1000;

a variável meses de serviço e insira o valor 12 e;

a variável quantidade de reclamações e insira o valor 3.

Faça as seguintes condições:

Se a quantidade de meses de serviço for maior igual a 12 e a quantidade de reclamações menor que 2

o salário deve ser multiplicado pelo valor 8 inserido do array.

após o cálculo, o sistema deve exibir a mensagem de “Aumentou” inserida no array, seguida da frase “Valor a pagar: ”, com a exibição do valor total do salário após a multiplicação.

Senão se a quantidade de meses de serviço for maior igual a 12 e a quantidade de reclamações maior ou igual a 2

o salário deve ser subtraído pelo valor que está na posição 2 do array.

após o cálculo, o sistema deve exibir a mensagem de “Diminuiu” inserida no array, seguida da frase “Valor a pagar: ”, com a exibição do valor total do salário após a subtração.

Senão exiba a mensagem “Pagamento normal”.

Resultado: Diminuiu. Valor a pagar: 950

Exercício

<?php

```
$aumento_mensagem = array (8, 'Aumentou.', 50, 'Diminuiu.');
```

```
$salario = 1000;
```

```
$meses_servico = 12;
```

```
$quantidade_reclamacoes = 3;
```

```
if ($meses_servico >= 12 and $quantidade_reclamacoes < 2){
```

```
    $salario *= $aumento_mensagem[0];
```

```
    echo $aumento_mensagem[1], ' Valor a pagar: ', $salario;
```

```
}
```

```
elseif ($meses_servico >= 12 and $quantidade_reclamacoes >= 2){
```

```
    $salario -= $aumento_mensagem[2];
```

```
    echo $aumento_mensagem[3], ' Valor a pagar: ', $salario;
```

```
}
```

```
else {
```

```
    echo 'Pagamento normal';
```

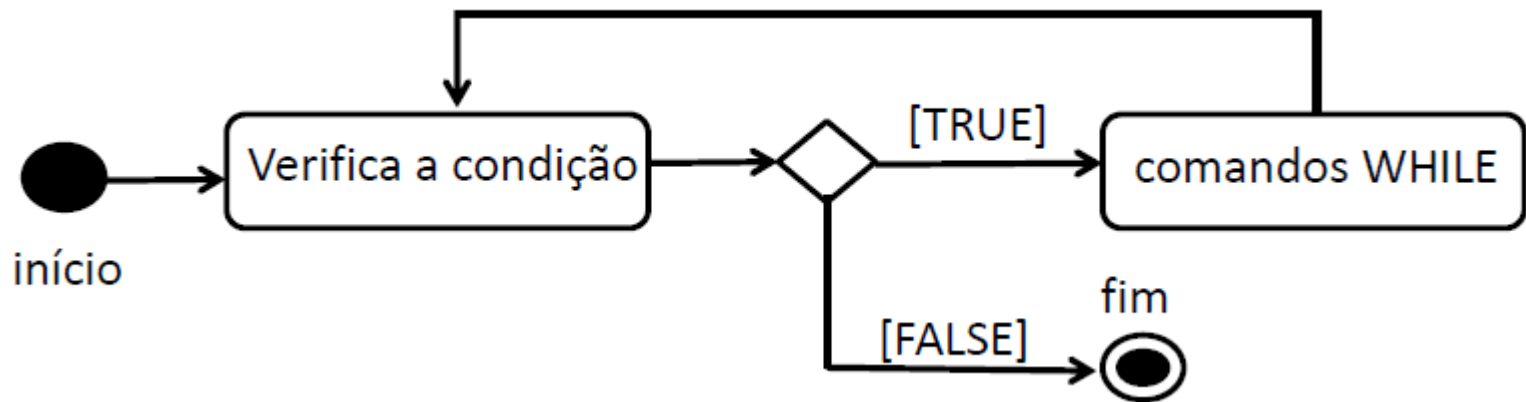
```
}
```

?>

WHILE

WHILE é uma estrutura similar ao IF – condição para executar bloco de comando.

Principal diferença – WHILE estabelece um laço de repetição enquanto a condição de entrada dada pela expressão for verdadeira.




```
<?php
```

```
    $a = 1;  
    while ($a < 5){  
        echo $a;  
        $a++;  
    }
```

```
?>
```



executa até \$a for menor que 5.

resultado - 1234

while

a expressão de avaliação é verificada no começo de cada iteração

```
<?php
    $a = 5;
    while ($a < 5){
        echo $a;
        $a++;
    }
?>
```

do e while

a expressão de avaliação é verificada ao final de cada iteração em vez de no começo
Vai fazer pelo menos uma iteração

```
<?php
    $a = 5;
    do{
        echo $a;
        $a++;
    } while ($a < 5);
?>
```

while

a expressão de avaliação é verificada no começo de cada iteração

```
<?php
    $a = 5;
    while ($a < 5){
        echo $a;
        $a++;
    }
?>
```

Resposta : Não aparece nada.

do e while

a expressão de avaliação é verificada ao final de cada iteração em vez de no começo
Vai fazer pelo menos uma iteração

```
<?php
    $a = 5;
    do{
        echo $a;
        $a++;
    } while ($a < 5);
?>
```

Resposta : 5

Exercício

Defina uma variável chamada numero e atribua o valor 20.

Enquanto o conteúdo de número for maior que 0

Exiba o conteúdo de número

Em seguida, subtraia o número por 5

Resposta:

20

15

10

5

Exercício

Defina uma variável chamada numero e atribua o valor 20.

Enquanto o conteúdo de número for maior que 0

Exiba o conteúdo de número

Em seguida, subtraia o número por 5

```
<?php
    $numero = 20;

    while($numero>0){
        echo $numero, '<br>';
        $numero -= 5;
    }

?>
```

FOR

FOR é muito similar ao WHILE – condição para executar bloco de comando.

Principal diferença – controlado por um bloco de três comandos que estabelecem uma contagem, ou seja, o bloco de comandos será executado um certo número de vezes.

```
<?php
    for ($i = 0; $i <= 10; $i++) {
        echo $i;
    }
?>
```

Variáveis sempre com nomes sugestivos, mas em contadores, como o \$i, é permitido utilizar variáveis de uma letra.

Parâmetros	Descrição
expr 1	Valor inicial da variável contadora
expr 2	Condição de execução. Enquanto for TRUE, o bloco de comando será executado
expr 3	Valor a ser incrementado após cada execução

Defina uma variável chamada numero e atribua o valor 10.

Faça a sequência de ações, descritas a seguir, 2 vezes.

Multiplique a variável numero com 5 e,
em seguida exiba o conteúdo da variável.

Resposta:

50

250

Defina uma variável chamada numero e atribua o valor 10.

Faça a sequência de ações, descritas a seguir, 2 vezes.

Multiplique a variável numero com 5 e,
em seguida exiba o conteúdo da variável.

Resposta:

50

250

```
<?php
    $numero = 10;

    for ($i=0; $i<=1; $i++){
        $numero *= 5;
        echo $numero, '<br>';
    }
?>
```


Exercício

Defina uma variável chamada valor e atribua o valor 30.

Faça a sequência de ações, descritas a seguir, 4 vezes.

Exiba o conteúdo da variável valor e,
em seguida some a essa variável o valor 10.

Resposta:

30

40

50

60

Exercício

Defina uma variável chamada valor e atribua o valor 30.

Faça a sequência de ações, descritas a seguir, 4 vezes.

Exiba o conteúdo da variável valor e,
em seguida some a essa variável o valor 10.


```
<?php
    $valor = 30;

    for ($i=0; $i<=3; $i++){
        echo $valor, '<br>';
        $valor += 10;
    }
?>
```

FOREACH

Laço de repetição para iterações em arrays. É um for simplificado que decompõe um vetor em cada um de seus elementos por meio de sua cláusula AS.

```
<?php
    $fruta = array('Uva', 'Laranja', 'Manga', 'Morango');
    //echo $fruta[1];
    //print_r($fruta);
    foreach ($fruta as $valor){
        echo " $valor ";
    }
?>
```




resultado –
Uva Laranja Manga Morango

FOREACH - **FOR**

É possível fazer com for, mas neste caso é necessário saber a quantidade de elementos.

```
<?php
    $fruta = array('Uva', 'Laranja', 'Manga', 'Morango');

    for ($i=0; $i<=3; $i++){
        echo $fruta[$i];
    }
?>
```



resultado –
Uva Laranja Manga Morango

FOREACH – COUNT


Conta a quantidade de elementos.

Cuidado: São 4 elementos, que vão até a posição 3, por isso `$i<$qtde`

`echo $qtde;` → `$qtde` vai ser igual a 4.

```
<?php
    $fruta = array('Uva', 'Laranja', 'Manga', 'Morango');

    $qtde = count($fruta);
    //echo $qtde;
    for ($i=0; $i<$qtde; $i++){
        echo $fruta[$i];
    }
?>
```



resultado –
Uva Laranja Manga Morango

SWITCH

```
<?php
```

```
    $i = 1;
```

```
    switch ($i){
```

```
        case 0:
```

```
            echo 'Valor i = 0';
```

```
        break;
```

```
        case 1:
```

```
            echo 'Valor i = 1';
```

```
        break;
```

```
        case 2:
```

```
            echo 'Valor i = 2';
```

```
        break;
```

```
        default:
```

```
            echo 'Valor i diferente de 0, 1 ou 2';
```

```
    }
```

```
?>
```

Executa linha por linha até encontrar a ocorrência de **break**. Importante para evitar que os blocos de comando seguintes sejam executados.

Essa cláusula “default” será executada caso nenhuma das expressões anteriores tenha sido verificada.

SWITCH

```
<?php
    $valor = 'teste';

    switch($valor){
        case 'teste':
            echo 'Ok';
            break;

    }

?>
```

Case ' ':
Aspas simples ou aspas
duplas

Exercício

Defina uma variável chamada numero e atribua o valor 10.

Utilizando o switch faça as seguintes condições.

Caso a variável numero tenha 15

Exiba 'Ok 15'

Caso a variável numero tenha 10

Exiba 'Ok 10'

Exercício

Defina uma variável chamada numero e atribua o valor 10.

Utilizando o switch faça as seguintes condições.

Caso a variável numero tenha 15

Exiba 'Ok 15'

Caso a variável numero tenha 10

Exiba 'Ok 10'

```
<?php
    $numero = 10;

    switch($numero){
        case 15:
            echo 'Ok 15';
            break;
        case 10:
            echo 'Ok 10';
            break;
    }
?>
```

Hora extra

Faça um código em PHP para controlar os salários de três funcionários da empresa “Trabalhe Sem Parar”.

Na empresa há os funcionários Mateus, Maria e Carlos. O salário inicial do Mateus é 2000, Maria é 1500 e Carlos é 1000.

Para incentivar a concorrência entre os funcionários, a empresa acrescenta 100 no salário do funcionário para cada hora extra que ele faz e, automaticamente reduz 10 dos salários dos outros dois funcionários que não fizeram.

Reduz 10 para cada hora que o funcionário, que fez hora extra, trabalhou; sendo assim, quando um funcionário trabalha 2 horas, ele tem o acréscimo de 200 e, os outros dois o decréscimo de 20.

Apenas um funcionário pode fazer hora extra por vez, por isso, quando é inserido o nome de um funcionário e a quantidade de hora extra, o código exibe o nome da empresa, os nomes de todos os funcionários, com os seus respectivos salários (com os acréscimos ou decréscimos) ao lado de cada nome.

Resultado: Nome do Funcionário: Carlos
Horas Extras: 10



Trabalhe Sem Parar
Mateus R\$ 1900
Maria R\$ 1400
Carlos R\$ 2000

Nome do Funcionário: Maria
Horas Extras: 6



Trabalhe Sem Parar
Mateus R\$ 1940
Maria R\$ 2100
Carlos R\$ 940

<?php

Com switch

```
$salarioMateus = 2000;
```

```
$salarioMaria = 1500;
```

```
$salarioCarlos = 1000;
```

```
$nomeFuncionario = 'Carlos';
```

```
$horaExtra = 10;
```

```
switch ($nomeFuncionario){
```

```
    case 'Mateus':
```

```
        for ($i=0; $i<$horaExtra; $i++){
```

```
            $salarioMateus += 100;
```

```
            $salarioMaria -= 10;
```

```
            $salarioCarlos -= 10;
```

```
        }
```

```
    break;
```

```
    case 'Maria':
```

```
        for ($i=0; $i<$horaExtra; $i++){
```

```
            $salarioMateus -= 10;
```

```
            $salarioMaria += 100;
```

```
            $salarioCarlos -= 10;
```

```
        }
```

```
    break;
```

...continua no próximo slide

```
case 'Carlos':  
    for ($i=0; $i<$horaExtra; $i++){  
        $salarioMateus -= 10;  
        $salarioMaria -= 10;  
        $salarioCarlos += 100;  
    }  
    break;  
}
```

```
echo 'Trabalhe Sem Parar <br><br>';  
echo 'Mateus R$ ', $salarioMateus, '<br>';  
echo 'Maria R$ ', $salarioMaria, '<br>';  
echo 'Carlos R$ ', $salarioCarlos, '<br>';
```

?>

<?php

Com if

```
$salarioMateus = 2000;
$salarioMaria = 1500;
$salarioCarlos = 1000;

$nomeFuncionario = 'Carlos';
$horaExtra = 10;

if ($nomeFuncionario == 'Mateus'){
    for ($i=0; $i<$horaExtra; $i++){
        $salarioMateus += 100;
        $salarioMaria -= 10;
        $salarioCarlos -= 10;
    }
}
elseif ($nomeFuncionario == 'Maria'){
    for ($i=0; $i<$horaExtra; $i++){
        $salarioMateus -= 10;
        $salarioMaria += 100;
        $salarioCarlos -= 10;
    }
}
```

...continua no próximo slide

```
elseif ($nomeFuncionario == 'Carlos'){  
    for ($i=0; $i<$horaExtra; $i++){  
        $salarioMateus -= 10;  
        $salarioMaria -= 10;  
        $salarioCarlos += 100;  
    }  
}  
echo 'Trabalhe Sem Parar <br><br>';  
echo 'Mateus R$ ', $salarioMateus, '<br>';  
echo 'Maria R$ ', $salarioMaria, '<br>';  
echo 'Carlos R$ ', $salarioCarlos, '<br>';
```

//tratar alguns erros nos próximos códigos

//pode ser nome de outro funcionário

//há a possibilidade de melhorar o código – evitar repetição

?>

Funções

Funções

Pedaço/parte do código com um objetivo específico, encapsulado sob uma estrutura única que recebe um conjunto de parâmetros e retorna um dado.

Uma função é declarada uma única vez, mas pode ser utilizada diversas vezes.

REUSO.

```
<?php
    function nome_da_funcao ($arg1, $arg2, $argN){
        $valor = $arg1 + $arg2 + $argN;
        return $valor;
    }
?>
```

operador **function** seguido do nome da função
lista de argumentos (**parâmetros**) que a função irá receber, separados por vírgula
em seguida entre chaves **{}** vem o código da função
return para retornar o resultado da função (inteiro, string, array, etc.)

Funções

Função que dobra (multiplica por 2) qualquer valor de entrada.

Com parâmetro e retorno

```
<?php
    function dobra ($valor){
        return $valor * 2;
    }
    echo dobra(120);
?>
```

Sem parâmetro e com retorno

```
<?php
    function dobra (){
        $valor = 120;
        return $valor * 2;
    }
    echo dobra();
?>
```

Sem parâmetro e sem retorno

```
<?php
    function dobra (){
        $valor = 120;
        echo $valor * 2;
    }
    dobra();
?>
```

Funções

(Com parâmetro e com retorno)

Faça uma função para calcular o Índice de Massa Corporal (IMC) a partir da entrada de peso e altura.

Cálculo IMC: $\text{Peso} / (\text{Altura} * \text{Altura})$

Peso: **70**

Altura: **1.85**

Resultado:

20.45288531775

Exercício

Funções

Faça uma função para calcular o Índice de Massa Corporal (IMC) a partir da entrada de peso e altura.

Cálculo IMC: $\text{Peso} / (\text{Altura} * \text{Altura})$

```
<?php
```

```
function calcularMassaCorporal ($peso, $altura){  
    return $peso / ($altura * $altura);  
    //return number_format($peso / ($altura * $altura), 2); - retornar com duas casas decimais.  
    //Pode ser com variável, por exemplo, echo number_format($numero, 2);  
}
```

```
echo calcularMassaCorporal (70, 1.85);
```

```
?>
```

Variáveis globais

Todas as **variáveis dentro do escopo de uma função** são **locais**.

Para **acessar uma variável** externa ao contexto de um função **sem passá-la como parâmetro**, é necessário declará-la como **global**.

```
<?php
function calcularMassaCorporal ($peso, $altura){
    $totalConsulta = 1;
    return $peso / ($altura * $altura);
}

echo calcularMassaCorporal (70, 1.85);
echo 'Total de pessoas que consultaram o IMC: ', $totalConsulta; //Erro
?>
```

Variáveis globais

Todas as **variáveis dentro do escopo de uma função** são **locais**.

Para **acessar uma variável** externa ao contexto de um função **sem passá-la como parâmetro**, é necessário declará-la como **global**.

```
<?php
```

```
$totalConsulta = 0;
```

```
function calcularMassaCorporal ($peso, $altura){  
    global $totalConsulta;  
    $totalConsulta += 1;  
    return $peso / ($altura * $altura);  
}
```

```
echo calcularMassaCorporal (70, 1.85), '<br>';  
echo calcularMassaCorporal (60, 1.95), '<br>';  
echo calcularMassaCorporal (90, 1.65), '<br>';
```

```
echo 'Total de pessoas que consultaram o IMC: ', $totalConsulta;
```

```
?>
```

Variáveis globais podem ser alteradas a partir de qualquer parte da aplicação.

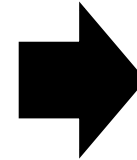
Requisição de arquivos

Incluir dentro de programas outros arquivos com definições de funções, constantes, configurações, etc.
comando: include 'arquivo';

```
<?php
```

```
function calcularMassaCorporal ($peso, $altura){  
    global $totalConsulta;  
    $totalConsulta += 1;  
    return $peso / ($altura * $altura);  
}
```

```
?>
```



calcularMassa.php

```
<?php
```

```
include 'calcularMassa.php';
```

```
$totalConsulta = 0;
```

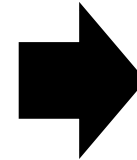
```
echo calcularMassaCorporal (70, 1.85), '<br>';
```

```
echo calcularMassaCorporal (60, 1.95), '<br>';
```

```
echo calcularMassaCorporal (90, 1.65), '<br>';
```

```
echo 'Total de pessoas que consultaram o IMC: ', $totalConsulta;
```

```
?>
```



imc.php

Requisição de arquivos

Incluir dentro de programas outros arquivos com definições de funções, constantes, configurações, etc.
comando: include 'arquivo';

```
<?php
```

```
function calcularMassaCorporal ($peso, $altura){  
    global $totalConsulta;  
    $totalConsulta += 1;  
    return $peso / ($altura * $altura);  
}
```

```
?>
```

```
<?php
```

```
include 'calcularMassa.php';
```

```
$totalConsulta = 0;
```

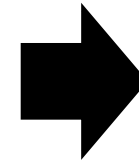
```
$resultado = calcularMassaCorporal (70, 1.85);  
echo 'Resultado + 10: ', $resultado + 10, '<br>';
```

```
echo calcularMassaCorporal (60, 1.95), '<br>';
```

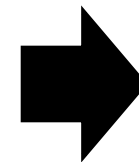
```
echo calcularMassaCorporal (90, 1.65), '<br>';
```

```
echo 'Total de pessoas que consultaram o IMC: ', $totalConsulta;
```

```
?>
```



calcularMassa.php



imc.php

```
<?php
```

```
    function calcularSoma ($n1, $n2){  
        return ($n1 + $n2);  
    }
```

```
    function calcularSubtracao ($n1, $n2){  
        return ($n1 - $n2);  
    }
```

```
?>
```



```
<?php
    include 'calculadora.php';

    $resultado = calcularSoma(10, 9);

    echo 'Resultado Soma: ', $resultado;

    //echo 'Resultado Soma: ', calcularSoma(10, 9);

    echo '<br>Resultado Subtracao: ', calcularSubtracao(8, 4);

?>
```

Exercício

```
calcularSoma(10, 9);
```

```
calcularSubtracao(8, 4);
```

O que fazer no código para não precisar de um método para cada operador aritmético?

Por exemplo: `calcular(10, 9, '+')`

Exercício

if**ou****switch**

<?php

```
function calcular ($n1, $n2, $n3){  
    if ($n3 == '+'){  
        return $n1+$n2;  
    }  
    else{  
        return $n1-$n2;  
    }  
}
```

?>

<?php

```
function calcular ($n1, $n2, $n3){  
    switch($n3){  
        case '+':  
            return $n1+$n2;  
            break;  
        default:  
            return $n1-$n2;  
    }  
}
```

?>

Exercício

```
<?php
    include 'calculadora.php';

    $resultado = calcular(10, 9, '+');

    echo 'Resultado Soma: ', $resultado;

?>
```

Referências

- [1] http://www.google.com.br/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&docid=0MNoxjocrNovwM&tbnid=NPOfC0EOY5OIZM:&ved=0CAUQjRw&url=http%3A%2F%2Fwww.rgbcomunicacao.com.br%2Fferramentas-web.html&ei=ZhjmU-W7L4XLsATuxoHwDg&bvm=bv.72676100,d.cWc&psig=AFQjCNFRR3q4R8HI0f45insErjZ_zao4Ag&ust=1407674844836901
- [2] https://www.apachefriends.org/pt_br/index.html
- [3] <http://www.pontotutorial.com/wp-content/uploads/2012/11/apache-logo.png>
- [4] <http://db-engines.com/en/ranking>
- [5] <http://www.profcardy.com/cardicas/ordem-das-operacoes-aritmeticas-e-algebricas.php>