

# Programação para Internet



INSTITUTO FEDERAL  
SANTA CATARINA  
Câmpus Canoinhas



HTML5



CSS3



JavaScript



O trabalho "Aulas de Programação para Internet - Curso Técnico em Informática" está licenciado com uma Licença Creative Commons - Atribuição 4.0 Internacional.

Programação para Internet

# Variáveis em PHP

No php as variáveis são precedidas pelo marcador \$.

# Operadores

Operadores aritméticos: +, -, \*, / e %. Unários:

- - `-$var`: inverte o sinal;
- `++$var`: incrementa antes;
- `--$var`: decrementa antes;
- `$var++`: incrementa depois; e
- `$var--`: decrementa depois.

Não utilizar in(de)cremento dentro de expressões.  
Diminui a legibilidade.

# Operadores

- Operadores de comparação:

Operador	Descrição
==	Igual (com conversão)
===	Igual (sem conversão)
!=	Diferente (com conversão)
!==	Diferente (sem conversão)
>=	Maior ou igual
<=	Menor ou igual
<>	Diferente
>	Maior
<	Menor

# Operadores

- É possível utilizar um operador aritmético na atribuição. Exemplo:
- `$a = 3;`
- `$a *= 5; // resultado é que $a vale 15`

# Operadores

- Operadores lógicos:
  - &&: operador E;
  - ||: operador OU;
  - !: negação (unário);
  - XOR: verdadeiro se exatamente um elemento for verdadeiro.

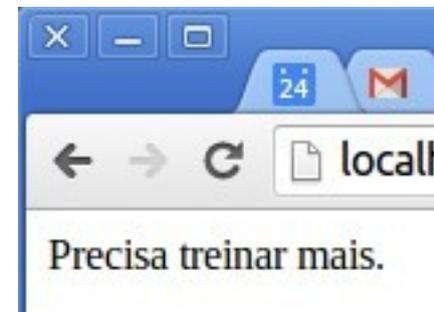
# Operadores

- PHP possui suporte ao operador ternário.
- condição ? Caso verdadeiro : caso falso;
- Exemplo:  
`$situacao = $nota >= 6 ? 'aprovado' : 'reprovado';`

# Estruturas de Controle

- IF, ELSEIF e ELSE: executa um bloco de código dependendo de uma condição.

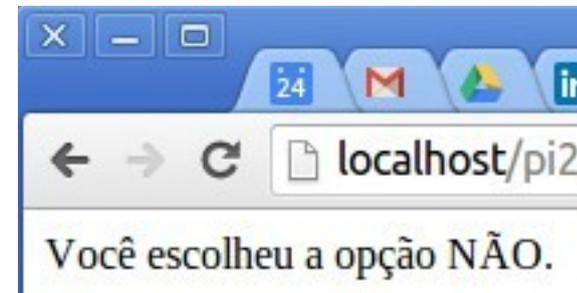
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$pontos = 5;
if ($pontos > 100) {
    echo 'Parabéns.';
} elseif ($pontos > 50) {
    echo 'Quase lá!';
} else {
    echo 'Precisa treinar mais.';
}
?>
</body>
</html>
```



# Estruturas de Controle

- SWITCH: testa vários valores para a mesma variável.

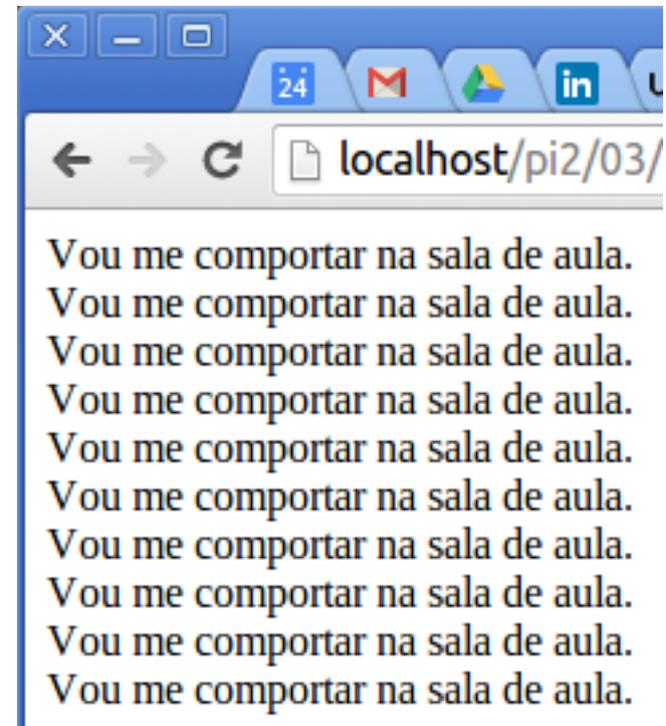
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$opcao = 'n';
switch ($opcao) {
  case 's':
    echo 'Você escolheu a opção SIM.';
    break;
  case 'n':
    echo 'Você escolheu a opção NÃO.';
    break;
  default:
    echo 'Opção inválida.';
}
?>
</body>
</html>
```



# Estruturas de Controle

- WHILE e DO..WHILE: executa bloco o enquanto a condição for verdadeira.

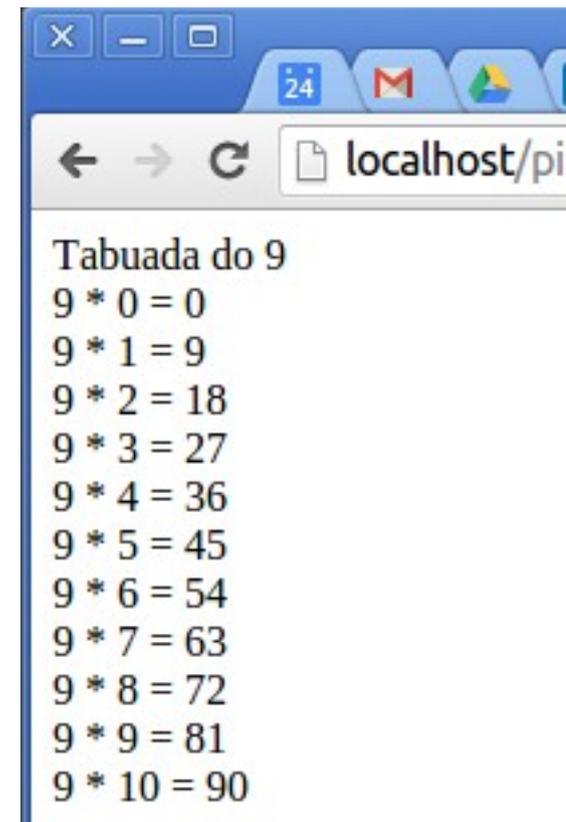
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$contador = 0;
while ($contador < 10) {
  echo 'Vou me comportar na sala de aula.<br />';
  $contador++;
}
?>
</body>
</html>
```



# Estruturas de Controle

- FOR: semelhante ao while, porém possui inicialização e incremento.

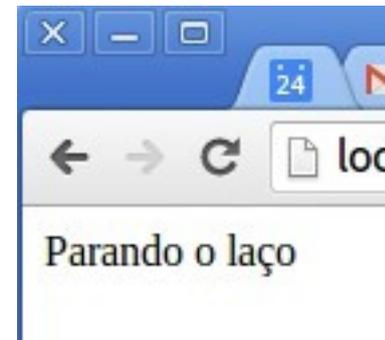
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
Tabuada do 9<br />
<?php
for ($atual = 0; $atual < 11; $atual++) {
  echo '9 * ' . $atual . ' = ';
  echo 9 * $atual;
  echo '<br />';
}
?>
</body>
</html>
```



# Estruturas de Controle

- BREAK: é utilizado para “sair” de um laço ou de um switch.

```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$contador = 0;
while ($contador < 99999) {
  echo 'Parando o laço';
  break;
  $contador++;
}
?>
</body>
</html>
```

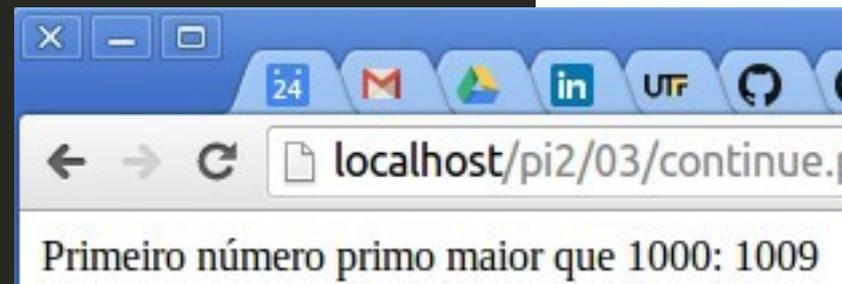


# Estruturas de Controle

- CONTINUE: pula para o ponto em que a condição do laço é testada.

Primeiro número primo maior que 1000:

```
<?php
$numeroAtual = 1000;
while (true) {
    $numeroAtual++;
    $ePrimo = true;
    for ($numTeste = 2; $numTeste < $numeroAtual; $numTeste++) {
        if ($numeroAtual % $numTeste == 0) {
            $ePrimo = false;
            break;
        }
    }
    if (!$ePrimo) {
        continue;
    }
    echo $numeroAtual;
    break;
}
?>
```



# Vetores

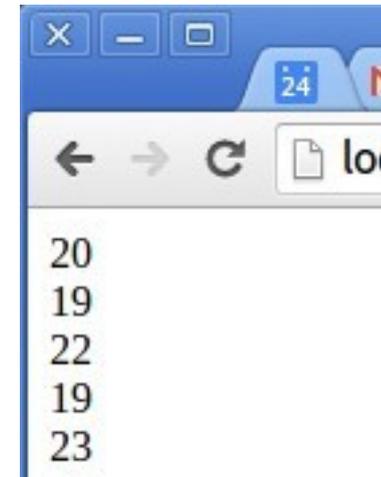
- Os vetores em PHP são bem flexíveis. É possível indexar valores por número ou por string.
- Os valores em um mesmo vetor podem ser de qualquer tipo.
- Exemplo de vetor:

Índice (Chave)	0	1	'nome'	'sobrenome'
Valor	17	5.5	'Guilherme'	'Silva'

# Vetores

- Para criar um vetor podemos utilizar a função `array`.

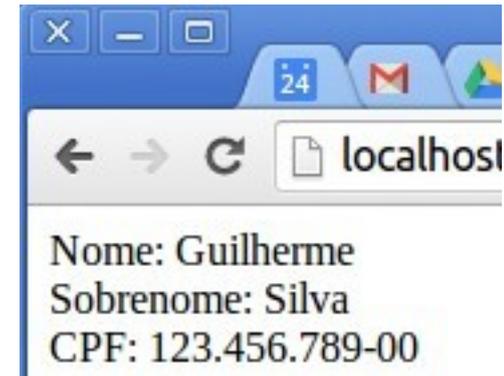
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$vetor = array(20,19,22,19,23);
for ($posicao = 0; $posicao < 5; $posicao++) {
    echo $vetor[$posicao] . '<br />';
}
?>
</body>
</html>
```



# Vetores

- Podemos já inicializar com as chaves.

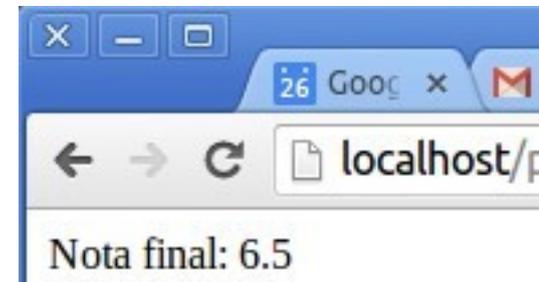
```
<html>
<head>
  <meta charset="UTF-8" />
</head>
<body>
<?php
$vetor = array(
  'nome' => 'Guilherme',
  'sobrenome' => 'Silva',
  'cpf' => '123.456.789-00'
);
echo 'Nome: ' . $vetor['nome'] . '<br />';
echo 'Sobrenome: ' . $vetor['sobrenome'] . '<br />';
echo 'CPF: ' . $vetor['cpf'];
?>
</body>
</html>
```



# Vetores

- Podemos criar um vetor simplesmente definindo um elemento.

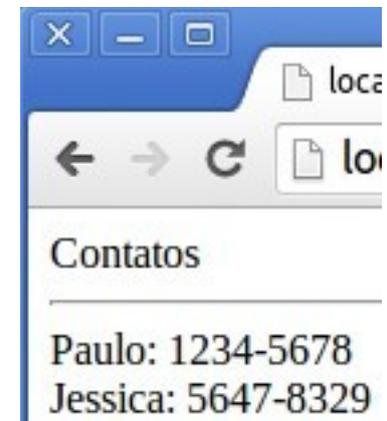
```
<body>
<?php
$vetor[0] = 6.5;
$vetor[1] = 9.5;
$vetor[] = 5.0;
$vetor[] = 5.0;
$media = 0;
for ($posicao = 0; $posicao < 4; $posicao++) {
    $media += $vetor[$posicao];
}
$media /= 4;
echo 'Nota final: ' . $media;
?>
</body>
```



# Vetores

- Para criarmos matrizes, podemos utilizar a notação [ ][ ] ou adicionar elementos que são vetores.

```
<body>
<?php
$contatos[0]['nome'] = 'Paulo';
$contatos[0]['telefone'] = '1234-5678';
$contatos[] = array(
    'nome' => 'Jessica',
    'telefone' => '5647-8329'
);
$quantidade = count($contatos);
echo 'Contatos<hr>';
for ($posicao = 0; $posicao < $quantidade; $posicao++) {
    echo $contatos[$posicao]['nome']
        . ': ' . $contatos[$posicao]['telefone']
        . '<br>';
}
?>
</body>
```

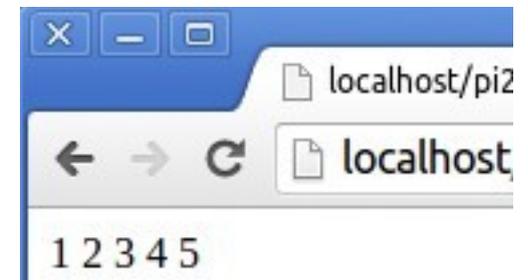


# FOREACH

- PHP possui um laço de repetição especial chamado FOREACH. Ele é utilizado para percorrer um vetor.
- Vamos comparar o FOR e o FOREACH.

```
$vetor1 = array(1,2,3,4,5);  
$quantidade = count($vetor1);  
for ($posicao = 0; $posicao < $quantidade; $posicao++) {  
    echo "$vetor1[$posicao] ";  
}
```

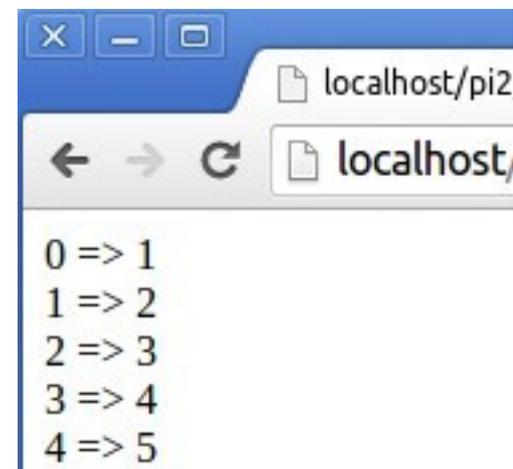
```
$vetor2 = array(1,2,3,4,5);  
foreach ($vetor2 as $elemento) {  
    echo "$elemento ";  
}
```



# FOREACH

- Também é possível obter a chave (índice) do vetor utilizando o FOREACH. Para isso, basta seguir a notação a seguir.

```
$vetor3 = array(1,2,3,4,5);  
foreach ($vetor3 as $chave => $elemento) {  
    echo "$chave => $elemento <br>";  
}
```



# Referências

- NIEDERAUER, Juliano. **Desenvolvendo websites com PHP**. 2. ed. São Paulo: Novatec, 2011. 301 p. ISBN