

Recebendo e manipulando dados em php

Prof. Me. Hélio Esperidião

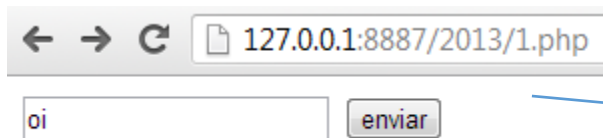
Recebendo dados de uma caixa de texto

Formulario.html

```
<form action="paginaDestino.php" method="get">  
  <input type="text" name="txtNome" id="txtNome" />  
  <input name="btnEnviar" type="submit" value="enviar" />  
</form>
```

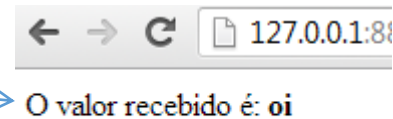
paginaDestino.php :

```
<?php  
    $Nome= $_GET['txtNome'];  
    echo "O valor recebido é: <b>$Nome<b>";  
?>
```



← → ↻ 127.0.0.1:8887/2013/1.php

oi enviar



← → ↻ 127.0.0.1:8887/2013/1.php

O valor recebido é: oi

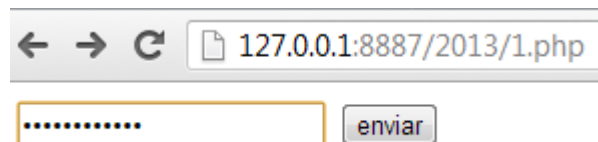
Recebendo dados de uma caixa de senha

Formulario.html

```
<form action="paginaDestino.php" method="get">  
  <input type="password" name="txtSenha" id="txtSenha" />  
  <input name="btnEnviar" type="submit" value="enviar" />  
</form>
```

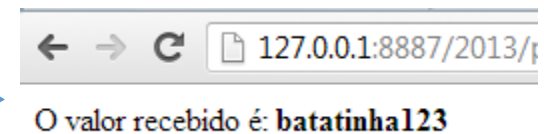
```
<?php  
  $Nome= $_GET['txtSenha'];  
  echo "O valor recebido é: <b>$Nome<b>";  
?>
```

paginaDestino.php :



127.0.0.1:8887/2013/1.php

..... enviar



127.0.0.1:8887/2013/1.php

O valor recebido é: **batatinha123**

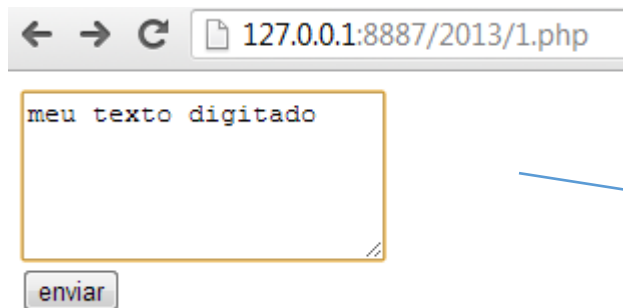
Recebendo dados de uma caixa de área

Formulario.html

```
<form action="paginaDestino.php" method="get">  
  <textarea name="txtComentario" id="txtComentario" cols="20" rows="5"></textarea><br/>  
  <input name="btnEnviar" type="submit" value="enviar" />  
</form>
```

paginaDestino.php :

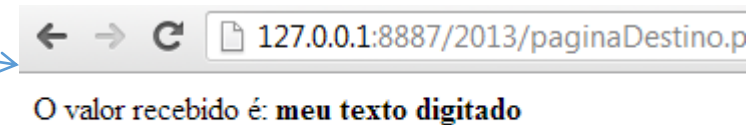
```
<?php  
  $comentario= $_GET['txtComentario'];  
  echo "O valor recebido é: <b>$comentario<b>";  
?>
```



← → ↻ 127.0.0.1:8887/2013/1.php

meu texto digitado

enviar



← → ↻ 127.0.0.1:8887/2013/paginaDestino.p

O valor recebido é: **meu texto digitado**

Recebendo dados de um checkbox

Formulario.html

```
<form action="paginaDestino.php" method="get">  
  <input type="checkbox" name="ckbReceberEmail" value="aceito" id="ckbReceberEmail" /> Aceito receber<br/>  
  <input name="btnEnviar" type="submit" value="enviar" />  
</form>
```

paginaDestino.php :

```
<?php  
  $testeCheckbox= $_GET['ckbReceberEmail'];  
  echo "O valor recebido é: <b>$testeCheckbox<b>";  
?>
```

Valor que será
enviado para o
servidor

← → ↻ 127.0.0.1:8887/2013/1.php

Aceito receber

enviar

← → ↻ 127.0.0.1:8887/2013/paginaDe

O valor recebido é: **aceito**

Recebendo dados de um botão de rádio

Formulario.html

```
<form action="paginaDestino.php" method="get">
  <input type="radio" name="rdoSexo" value="m" id="rdoSexo" checked> Masculino <br/>
  <input type="radio" name="rdoSexo" value="f" id="rdoSexo"> Feminino <br/>
  <input type="submit" name="btnEnviar" value="Enviar" id="btnEnviar">
</form>
```

paginaDestino.php :

```
<?php
    $sexo= $_GET['rdoSexo'];
    echo "O valor recebido é: <b>$sexo<b>";
?>
```

Valor que será
enviado para o
servidor

← → ↻ 127.0.0.1:8887/2013/1.php

Masculino
 Feminino

Enviar

← → ↻ 127.0.0.1:8887/2013/p

O valor recebido é: **m**

Recebendo dados de uma caixa de combo

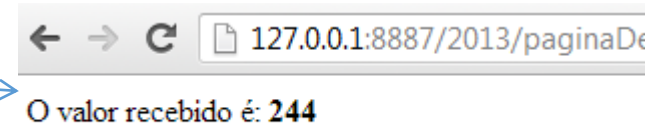
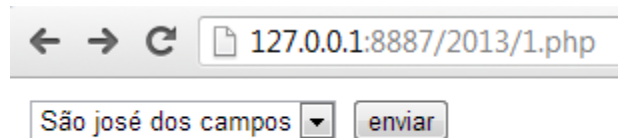
```
<form action="paginaDestino.php" method="get">
  <select name="cboCidades" id="cboCidades" >
    <option value="244">São José dos Campos</option>
    <option value="44">Jacareí</option>
  </select>
  <input name="btnEnviar" type="submit" value="enviar" />
</form>
```

Formulario.html

Valor que será
enviado para o
servidor

```
<?php
  $codigoCidade= $_GET['cboCidades'];
  echo "O valor recebido é: <b>$codigoCidade<b>";
?>
```

paginaDestino.php :



Recebendo dados de uma caixa de lista

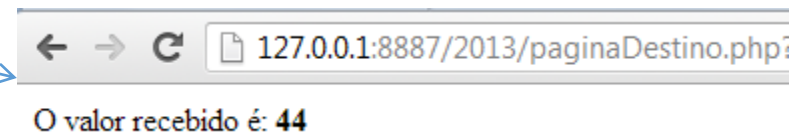
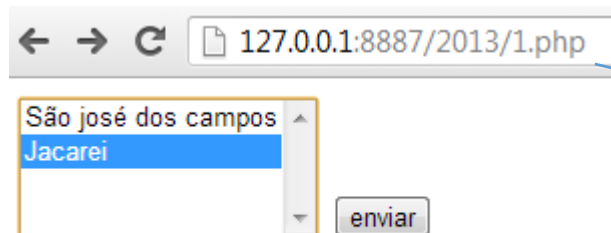
```
<form action="paginaDestino.php" method="get">
  <select name="lstCidades" id="lstCidades" size="3" >
    <option value="244">São José dos Campos</option>
    <option value="44">Jacarei</option>
  </select>
  <input name="btnEnviar" type="submit" value="enviar" />
</form>
```

Formulario.html

Valor que será
enviado para o
servidor

```
<?php
  $codigoCidade= $_GET['lstCidades'];
  echo "O valor recebido é: <b>$codigoCidade<b>";
?>
```

paginaDestino.php



Nomes de variáveis

Toda variável em PHP tem seu nome composto pelo caractere \$

PHP é case sensitive

Nos casos mais comuns não há declaração de tipo.

```
<?php
$ExemploVariavelString="teste";
$ExemploVariavelFloat= 1.5;
$ExemploVariavelDouble=1.565456489;
$ExemploVariavelInteira=1;
$ExemploVariavelVetor[0]=1;
$ExemploVariavelVetor[1]=2;
$Vetor = array('Maria', 'Carlos', 'Edgar');
$Vetor = array(0, 1, 2);
?>
```

Comentários

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
</head>
<body>
Exemplo:
<? echo "teste"; #isto é um teste
    echo "teste"; //este teste é similar ao anterior

    /*Bloco de comentarios*/
?>

</body>
</html>
```

constantes

```
<?php  
define ("pi", 3.1415926536);  
$raio=5;  
$circunf = 2*pi*$raio;  
echo $circunf;  
?>
```

Operadores

+	adição
-	subtração
*	multiplicação
/	divisão
%	módulo

==	igual a
!=	diferente de
<	menor que
>	maior que
<=	menor ou igual a
>=	maior ou igual a

++	incremento
----	------------

--	decremento
----	------------

Blocos de controle

IF e SWITCH

```
<?php
$x=1;
$y=0;
if ($x == $y){
    echo "iguais";
}else{
    echo "Diferentes";
}
?>
```

```
<?php
$x=1;

switch ($x) {
case 0:
    echo "i é igual a zero";
    break;
case 1:
    echo "i é igual a um";
    break;
case 2:
    echo "i é igual a dois";
    break;
}
?>
```

Blocos de controle

IF

Concatenar instruções lógicas condicionais

```
<?php
```

```
$x=1;
```

```
$y=1;
```

```
$z=1;
```

```
if($x==$y && $x==$z) {
```

```
    echo "O valor de x, y e z são iguais a $x";
```

```
}
```

```
?>
```

operador	Ação
&&	.e. lógico
	.ou. lógico

Estrutura de repetição

```
<?php  
  
$i = 1;  
while ($i <=10) {  
    echo $i++;  
}  
?>
```

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

Quebra de fluxo

```
<?php
$x=0;
while ($x < 30) {
    echo $x;
    if ($x >15 ) {
        echo "<br>erro! x > 15";
        break;
    }
    $x++;
}
?>
```


Estrutura de repetição

Calculo Simples

```
<?php  
  
$n=3;  
  
for ($x=1;$x<=10;$x++) {  
  
    echo "$n x $x = ".$x*$n;  
  
}  
  
?>
```

Outras Verificações

`is_bool()`

- Verifica se a variável é um booleano

`is_float()`

- Informa se a variável é do tipo float

`is_string()`

- Informa se a variável é do tipo string

`is_array()`

- Verifica se a variável é um array

Outras Verificações

isset()

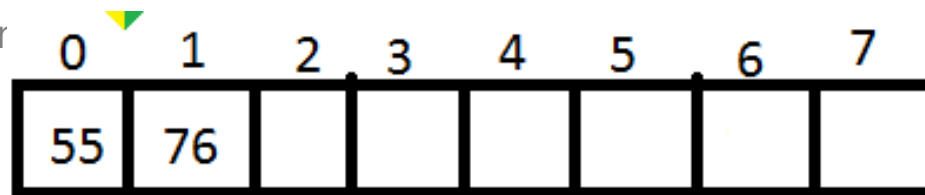
- Verifica se uma determinada variável foi inicializada
- Pode ser utilizada para verificar se campos de formulários
preenchidos.

```
<?php  
  
if(isset($_GET['txtNome'])) {  
    echo "O campo nome foi preenchido";  
} else {  
    echo "O campo nome não foi preenchido";  
}  
  
?>
```

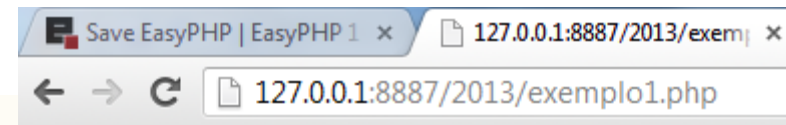
Vetores

Permitem que sejam armazenados diversos valores em uma mesma variável.

Vetores são unidimensionais



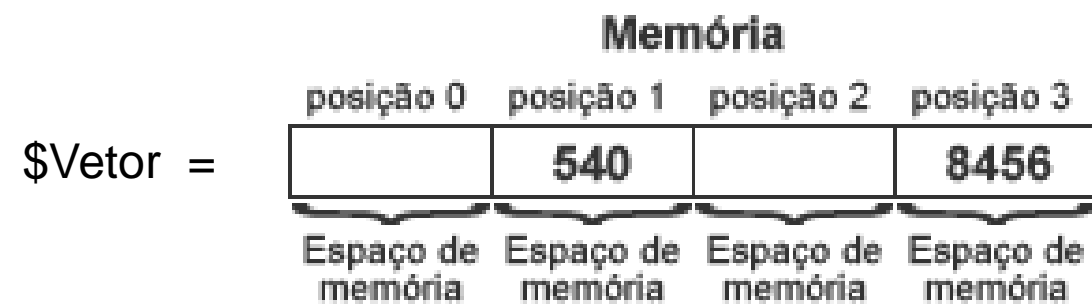
```
<?php
$vetor = array();
$vetor[0] = 55;
$vetor[1] = 76;
echo "O valor no índice 0 é: ".$vetor[0];
echo "<br/>";
echo "O valor no índice 1 é: ".$vetor[1];
?>
```



O valor no índice 0 é: 55
O valor no índice 1 é: 76

Entendendo melhor

```
<?php
    $vetor = array();
    $vetor[1] = 540;
    $vetor[3] = 8456;
?>
```



Varrendo array

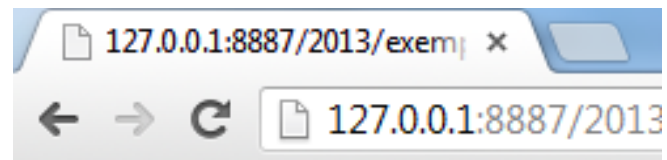
```
<?php
$arr[0]= "one";
$arr[1]= "two";
$arr[2]= "three";

$tamanho= sizeof($arr);

$i=0;
while($i<$tamanho){

    echo "O valor é: ".$arr[$i]."<br/>";
    $i++;
}

?>
```



O valor é: one
O valor é: two
O valor é: three

Varrendo array

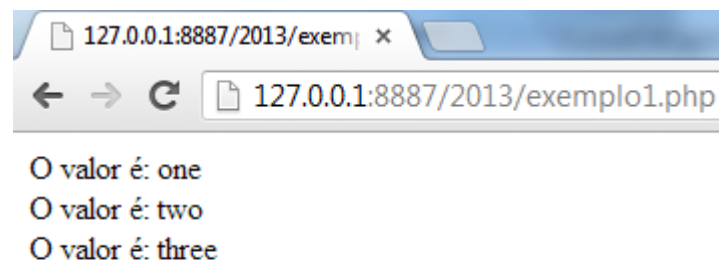
O comando foreach permite varrer o vetor sem saber o seu tamanho.

Cada posição do vetor é armazenada em uma variável que pode ser utilizado

posteriormente

```
<?php
$arr[0]= "one";
$arr[1]= "two";
$arr[2]= "three";

$tamanho= sizeof($arr);
foreach ($arr as $value) {
    echo "O valor é: ".$value."<br/>";
}
?>
```



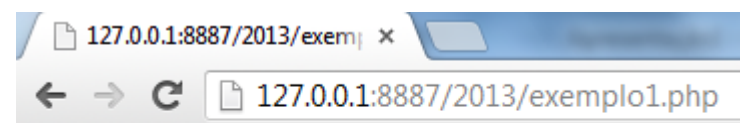
Arrays Associativos

Ao invés de usarmos números como índices, podemos usar strings (nomes).

Fica muito mais simples e intuitivo obter um valor de um array, pois nomes fazem mais sentido do que números.

```
<?php
$doc = array();
$doc['rg'] = "00.000.000-X";
$doc['cpf'] = "000.000.000-00";
$doc['cartao de credito'] = 12345;

echo $doc['rg'];
?>
```



00.000.000-X

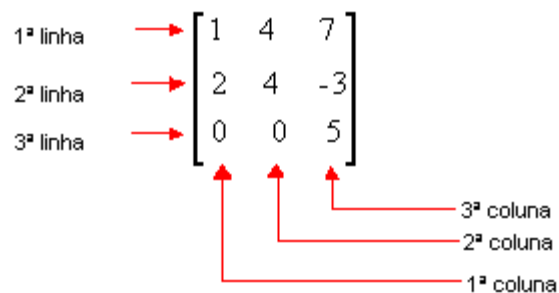
Matrizes

Permitem que sejam armazenados diversos valores em uma mesma variável.

Matrizes são unidimensionais.

Podem ser consideradas um conjunto de Vetores

Cada Linha da matriz é um



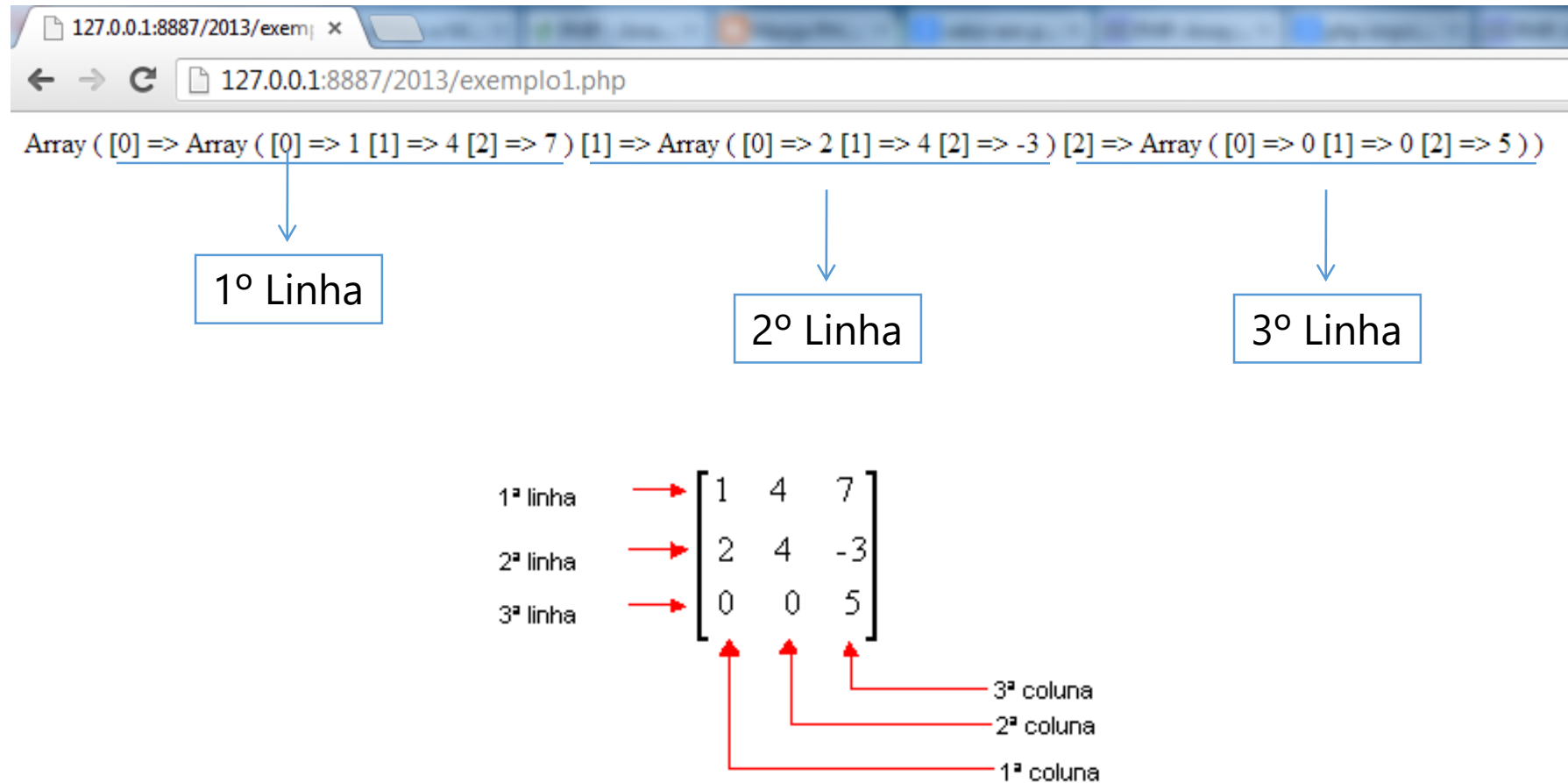
```
<?php
    $M[0][0]=1;
    $M[0][1]=4;
    $M[0][2]=7;

    $M[1][0]=2;
    $M[1][1]=4;
    $M[1][2]=-3;

    $M[2][0]=0;
    $M[2][1]=0;
    $M[2][2]=5;

    print_r($M);
?>
```

Matrizes



Varrendo Matrizes

```

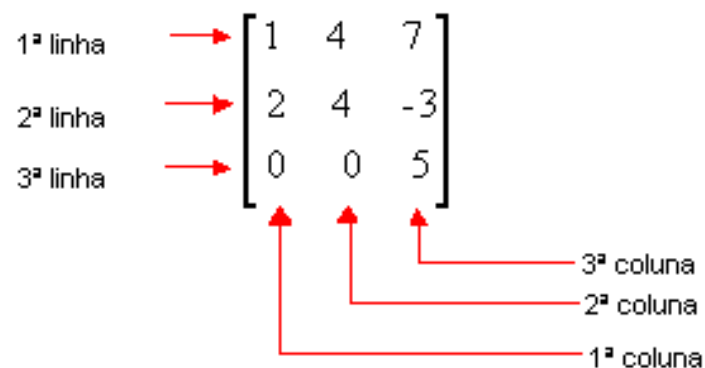
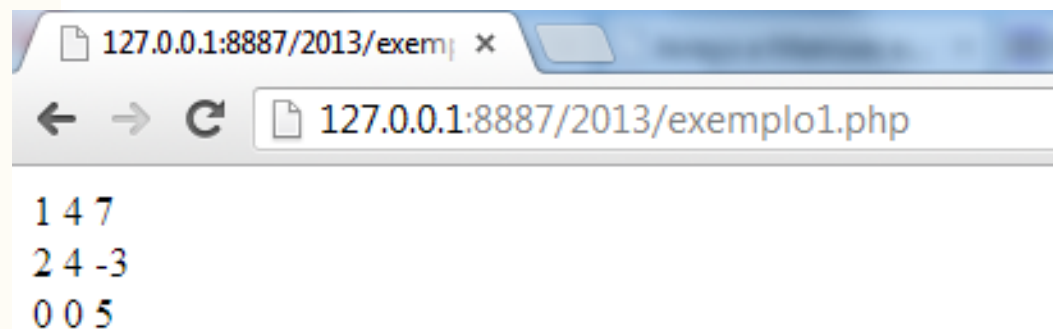
<?php
    $M[0][0]=1;
    $M[0][1]=4;
    $M[0][2]=7;
    $M[1][0]=2;
    $M[1][1]=4;
    $M[1][2]=-3;
    $M[2][0]=0;
    $M[2][1]=0;
    $M[2][2]=5;

    $tamanho = sizeof($M);
    $tamanho;

    $i=0;
    $j=0;

    while($j<$tamanho){
        while($i<$tamanho){
            echo $M[$j][$i]. " ";
            $i++;
        }
        echo "<br/>";
        $i=0;
        $j++;
    }
}
?>

```



Varrendo Matrizes

```
<?php
    $M[0][0]=1;
    $M[0][1]=4;
    $M[0][2]=7;

    $M[1][0]=2;
    $M[1][1]=4;
    $M[1][2]=-3;

    $M[2][0]=0;
    $M[2][1]=0;
    $M[2][2]=5;

    foreach ($M as $i){
        foreach ($i as $j){
            echo $j." ";
        }
        echo "<br/>";
    }
?>
```

