Primeiros passos...

O que você precisa saber para ir um pouco além do mundialmente famoso script “Alô Mundo!”

Um programa PHP

- Delimitadores de código

```
<?php
   //código
   //código
   //código
?>
```

- Comentários

Única linha:

```
// echo “a”;
# echo “a”;
```

Muitas linhas

```
/* echo “a”
   echo “b” */
```

Um programa PHP

- Comandos de saída
  - echo ‘a’, ‘b’, ‘c’;
  - print(‘abc’);
  - var_dump
    - Exemplo:

```
$vetor = array(‘Palio’, ‘Gol’, ‘Fiesta’, ‘Corsa’);
var_dump($vetor);
```

Resultado:

```
array(4){
   [0] => string(5) “Palio”
   [1] => string(3) “Gol”
   [2] => string(6) “Fiesta”
   [3] => string(5) “Corsa”
}
```

Um programa PHP

- Comandos de saída
  - print_r
    - Exemplo:

```
$vetor = array(, ‘Gol’, ‘Fiesta’, ‘Corsa’);
print_r($vetor);
```

Resultado:

```
Array{
   [0] => Palio
   [1] => Gol
   [2] => Fiesta
   [3] => Corsa
}
```

Um programa PHP

- Variáveis

```
<?php
   $nome = “Joao”;
   $sobrenome=“da Silva”;
   echo “$sobrenome, $nome”;
?>
```

💻Resultado:

da Silva, João

Um programa PHP

- Variáveis - algumas dicas
  - Nunca inicie a nomenclatura de variáveis com números;
  - Nunca utilize espaços em branco no meio do identificador da variável;
  - Nunca utilize caracteres especiais (! @ # % ^ & / | [] {}) na nomenclatura das variáveis;
  - Evite criar variáveis com mais de 15 caracteres em virtude da clareza do código-fonte;
  - Nomes de variáveis devem ser significativos e transmitir a idéia de seu conteúdo dentro do contexto no qual a variável está inserida;
  - Utilize preferencialmente palavras em minúsculo (separadas pelo caractere “_”) ou somente as primeiras letras em maiúsculo quando da ocorrência de mais palavras.

Um programa PHP

- Variáveis variantes

```
<?php
   //define o nome da variável
   $variavel = ‘nome’;

   //cria a variável identificada pelo conteúdo de $variavel

   $$variavel = ‘maria’;

   //exibe variável $nome na tela

   echo $nome;
?>
```

Um programa PHP

- Referências entre variáveis

```
<?php
   $a = 5;
   $b = &$a;
   $b = 10;
   echo $a;
   echo $b;
?>
```

Um programa PHP

- Tipos de dados
  - Booleano

```
<?php
   $exibir_nome = TRUE;

if ($exibir_nome)
{
   echo 'José da Silva';
}
?>
```

Um programa PHP

- Tipos de dados
  - Booleano

```
<?php
$umidade = 91;
$vai_chover = ($umidade > 90);

if ($vai_chover)
{
   echo 'Está chovendo';
}
?>
```

Um programa PHP

- Tipos de dados
  - Também são considerados valores falsos em comparações booleanas
    - Inteiro 0
    - Ponto flutuante 0.0
    - Uma string vazia “” ou “0”
    - Um array vazio
    - Um objeto sem elementos
    - Tipo NULL

Um programa PHP

- Tipos de dados
  - Numérico

```
<?php
//número decimal
$a = 1234;

//um número negativo
$a = -1234

//número hexadecimal (equivalente a 26 em decimal)
$a = 0x1A;

/número octal (equivalente a 83 em decimal
$a = 0123

//ponto flutuante
$a = 1.234

//notação científica
$a = 4e23;
?>
```

Um programa PHP

- Tipos de dados
  - String

```
<?php
    $variavel = 'Isto é um teste';
    $variavel = "Isto é um teste";
?>
```

  - Objeto

```
<?php
    class Computador
    {
        public $cpu;
        function ligar()
        {
          echo "Ligando computador a {$this -> cpu}....";
        }
    }
$obj = new Computador.
$obj -> cpu = "500Mhz";
$obj -> ligar();
?>
```

Um programa PHP

- Tipos de dados especiais
  - Recurso (resource): é uma variável especial que mantém uma referência de recurso externo. Recursos são criados e utilizados por funções especiais, como uma conexão ao banco de dados.

    resource mysql_connect(...)

  - Misto (mixed): representa múltiplos (não necessariamente todos) tipos de dados em um mesmo parâmetro. Um parâmetro do tipo mixed indica que a função aceita diversos tipos de dados como parâmetro.

    string gettype (mixed var)

  - Callback: Algumas funções como call_user_func() aceitam um parãmetro que significa uma função a ser executada. Esste tipo de dado é chamado de callback.
  - Null: A utilização do valor especial NULL significa que a variável não tem valor. NULL é o único valor possível do tipo NULL.

Um programa PHP

- Constantes

Uma constante é um valor que não sofre modificações durante a execução do programa. Ela é representada por um identificador, assim como as variáveis, com a exceção de que só pode conter valores escalares (boolean, inteiro, ponto flutuante e string).

MAXIMO_CLIENTES;

```
<?php
   define(“MAXIMO_CLIENTES”, 100);
   echo MAXIMO_CLIENTES;
?>
```

Um programa PHP

- Operadores
  - Atribuição

```
<?php
   $var = 0;
   $var += 5;
   $var -= 5;
   $var *= 5;
   $var /= 5;
?>
```

++$a: pré-incremento. Incrementa $a em um e, então, retorna $.

$a++: pós-incremento. Retorna $a e, então, incrementa $a em um.

--$a: Pré-decremento. Decrementa $a em um e, então, retorna $a.

$a--: Pós-decremento. Retorna $a e, então, decrementa $a em um.

Um programa PHP

- Operadores
  - Aritméticos

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

Um programa PHP

- Operadores
  - Relacionais

Operadores	Descrição
==	Igual. Resulta verdadeiro (TRUE) se expressões forem iguais.
===	Idêntico. Resulta verdadeiro (TRUE) se as expressões forem iguais e do mesmo tipo de dados.
!= ou <>	Diferente. Resulta verdadeiro se as variáveis forem diferentes.
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual

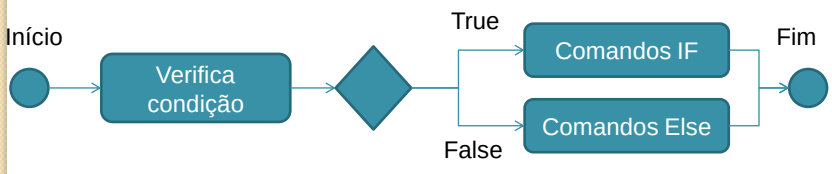
Um programa PHP

- Operadores
  - Lógicos

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 (TRUE) se $a ou $b forem verdadeiros, de forma exclusiva.
(!$a)	NOT: verdadeiro (TRUE) se $a for false.
($a && $b)	E: verdadeiro (TRUE) se tanto $a quanto $b forem verdadeiros.
($a || $b)	OU: verdadeiro (TRUE) se $a ou $b forem verdadeiros.
Observação : or e	and têm precend~encia menor que && ou ||.

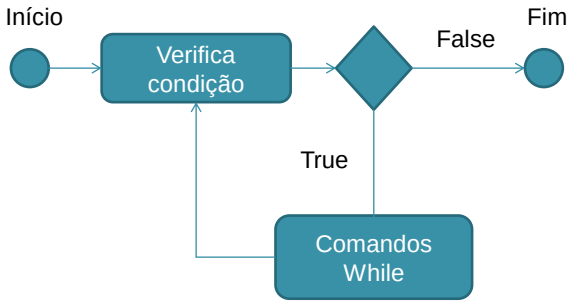
Um programa PHP

- Estruturas de Controle
  - IF



Um programa PHP

- Estruturas de Controle
  - While



Um programa PHP

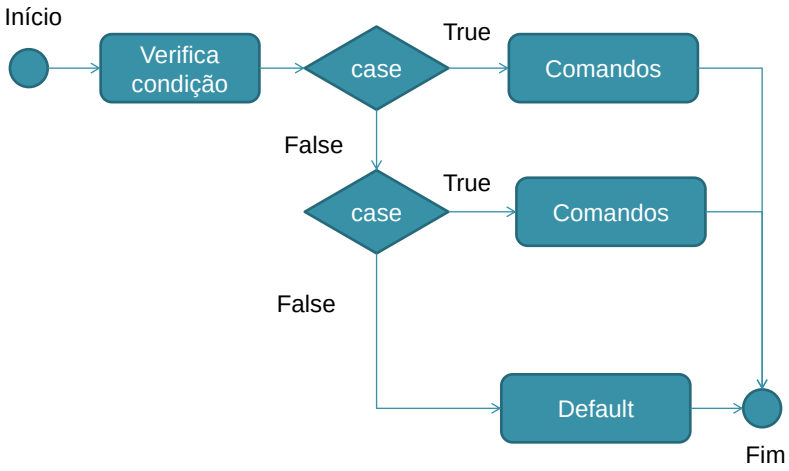
- Estruturas de Controle
  - For

```
for (expr1; expr2; expr3)
{
   comandos;
}
```

Parâmetros	Descrição
expr1	Valor inicial da variável controladora.
expr2	Condição de execução. Enquanto for TRUE, o bloco de comandos será executado.
expr3	Valor a ser incrementado após cada execução.

Um programa PHP

- Estruturas de Controle
  - Switch



Um programa PHP

- Estruturas de Controle
  - Foreach

```
Foreach ($array as $valor)
{
   instruções
}
```

Um programa PHP

- Estruturas de Controle
  - Continue

A instrução continue, quando executada em um bloco de comandos FOR/WHILE, ignora as instruções restantes até o fechamento em "}". Dessa forma o programa segue para a próxima verificação da condição de entrada do laço de repetição.

  - Break

O comando break aborta a execução de blocos de comandos, como o IF, WHILE, FOR. Quando estamos em uma execução com muitos níveis de iteração e desejamos abortar n níveis, a sintaxe é a seguinte:

```
While...
    For ...
        break <quantidade de níveis>
```

Um programa PHP

- Requisição de arquivos
  - include : a instrução include() inclui e avalia o arquivo informado. Seu código (variáveis, objetos e arrays) entra no escopo do programa, tornando-se disponível a partir da linha em que a inclusão ocorre.
  - require: identico ao include. Difere somente na manipulação de erros. Enquanto o include produz uma warning, o require produz uma mensagem de Fatal Error caso o arquivo não exista.
  - include_once: Funciona da mesma maneira que o comando include, a não ser que o arquivo informado já tenha sido incluído, não refazendo a operação (o arquivo é incluído apenas uma vez).
  - require_once: Funciona da mesma forma que o comando require, a não ser que o arquivo informado já tenha sido incluído, não refazendo a operação (o arquivo é incluído apenas uma vez).

Um programa PHP

- Manipulação de funções

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

         return $valor;
   }
?>
```

Um programa PHP

- Variáveis globais

```
<?php
     $total = 0;
function km2mi ($quilometros)
{
   global $total;
   $total += $quilometros;

   return $quilometros * 0.6;
}

echo 'Percorreu: '.km2mi(100). " milhas\n";
echo 'Percorreu: '.km2mi(200). " milhas\n";
echo 'Percorreu no total: '. $total. "quilômetros.";?>
```

Um programa PHP

- Variável estática

```
<?php

function percorre($quilometros)
{
    static $total;
    $total += $quilometros;
    echo "Percorreu mais $quilometros do total de $total <br>";
}

percorre(100);
percorre(200);
percorre(50);
?>
```

Um programa PHP

- Passagem de parâmetros
  - Por valor (by value)

```
<?php
    function Incrementa ($variavel, $valor)
    {
      $variavel += $valor;
    }

$a = 10;
Incrementa($a, 20);
echo $a;
?>
```

  - Por referência (by reference)

```
<?php
    function Incrementa (&$variavel, $valor)
    {
      $variavel += $valor;
    }

$a = 10;
Incrementa($a, 20);
echo $a;
?>
```

Um programa PHP

- Função com argumentos dinâmicos

  O PHP também permite definir uma função com o número de argumentos variáveis, permite obtê-los de forma dinâmica, mesmo sem saber quais são ou quantos são.

```
<?php

function Ola()
{
    $argumentos = func_get_args();
    $quantidade = func_num_args();

    for  ($n=0; $n < quantidade; $n++)
    {
            echo 'Ola '. $argumentos[$n].'\n';
    }
}
```

Um programa PHP

- Recursividade

```
<?php
    function Fatorial($numero)
    {
            if ($numero==1)
            return $numero;
            else
            return $numero = fatorial($numero -1);
    }
    echo Fatorial(5).”\n”;
    echo Fatorial(7).”\n”;
?>
```

Um programa PHP

- Manipulação de strings

Uma string é uma cadeia de caracteres alfanuméricos. Para declarar uma string podemos utilizar as simples (‘ ’) ou dupla (“ “).

```
<?php
    $fruta = “maçã”;
    print “Como $fruta \n”;
    print ‘Como $fruta \n’
?>
```

Podemos também declarar uma string literal como muitas linhas observando a sintaxe a seguir.

```
<?php
$texto = <<<CHAVE
Aqui nesta área
Você poderá escrever
Textos com múltiplas linhas
CHAVE;

echo $texto;
?>
```

Um programa PHP

- Manipulação de strings
  - Caracteres de escape

Caractere	Descrição
\n	Nova linha, proporciona uma quebra de linha
\r	Retorno de carro
\t	Tabulação
\\	Barra invertida “\\” é o mesmo que’\’
\”	Aspas duplas
\$	Símbolo $

Um programa PHP

- Manipulação de arrays
  - A manipulação de arrays no PHP é, sem dúvida, um dos recursos mais poderosos da linguagem. O programador que assimilar bem esta parte terá muito mais produtividade no seu dia-a-dia. Isto porque os arrays no PHP servem como verdadeiros contêiners, servindo para armazenar números, strings, objetos, dentre outros, de forma dinâmica.
  - $cores = array(‘vermelho’, ‘azul’, ‘verde’, ‘amarelo’);
  - $cores = array(0=> ‘vermelho’, 1=>’azul’, 2=>’verde’, 3=>’amarelo’).

Uma outra alternativa:
$nomes[] = ‘maria’;
$nomes[] = ‘joao’;
$nomes[] = ‘carlos’;
$nomes[] = ‘josé’;

E uma outra alternativa:
$pessoa[‘nome’] = ‘Maria da Silva;
$pessoa[‘rua’] = ‘São João’;
$pessoa[‘bairro’]= ‘Cidade Alta’;
$pessoa[‘cidade’] = ‘Porto Alegre’;

Um programa PHP

- Arrays multidimensionais
  - Arrays multidimensionais ou matrizes são arrays nos quais algumas de suas posições podem conter outros arrays de forma recursiva.

```
<?php
$carros = array('Palio' => array('cor' => 'azul', 'potencia' => '1.0',
   'opcionais' => 'Ar condicionado'),
      'Corsa' => array('cor' => 'cinza', 'potencia' => '1.3', 'opcionais'
   => 'MP3'),
      'Gol' => array('cor' => 'preto', 'potencia' => '1.8', 'opcionais' =>
   'Metalica')
   );
   echo $carros['Palio']['opcionais'];
?>
```

Um programa PHP

- Arrays multidimensionais

```
<?php
$carros['Palio']['cor'] = 'azul';
$carros['Palio']['potencia'] = '1.0';
$carros['Palio']['opcionais'] = 'Ar Condicionado';

$carros['Corsa']['cor'] = 'cinza';
$carros['Corsa']['potencia'] = '1.03';
$carros['Corsa']['opcionais'] = 'MP3';

$carros['Gol']['cor'] = 'azul';
$carros['Gol']['potencia'] = '1.8';
$carros['Gol']['opcionais'] = 'Metalica';

echo $carros['Corsa']['potencia'];

?>
```

Cenas do próximo capítulo...

- Manipulação de arquivos
- Funções para manipulação de:
  - Strings
  - Arrays
  - Objetos