

## Software Necessário

Accede a <http://rsergio.pt>

Descarrega o Software para a Pen:

- IDE – **Edit Plus** pode ser **colocado** em **qualquer localização** da Pen
- Servidor **UsbServer** - tem de ser **extraído** para uma **pasta** com o **mesmo nome** do ZIP para a raiz da Pen

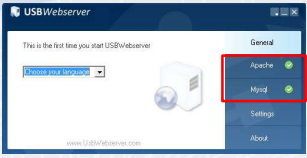
Programação de alto nível

## Software Necessário

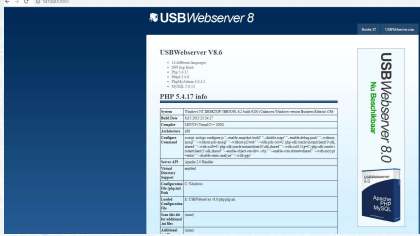
O **servidor Web** é **portátil** por isso **não necessita** por isso de **instalação** ou **privilégios de administrador** para correr

Para **verificar** se está a **funcionar** :

- **Check a verde** no **Apache** e **MySQL**



- **Abrir o browser** e digitar **127.0.0.1:8080**



Programação de alto nível

## Declaração

Estilo XML:

```
<?php
(...)
?>
```

Estilo abreviado:

```
<?
(...)
?>
```

Estilo Script:

```
<script language="php">
(...)
</script>
```

← O mais utilizado

Programação de alto nível

## Imprimir no ecrã

Para imprimir usa-se o comando **echo**

**Exemplo:**

```
echo "Olá turmaX!";
```

**Ou**

```
echo 'Olá turmaY!';
```

**Testar:**

Guardar ficheiro como **olamundo.php** na **pasta root** do **usbserver**

Aceder no **browser** a:

**127.0.0.1:8080/olamundo.php**

**Ver código fonte da página!**

*Programação de alto nível*

## Imprimir no ecrã

Com o **echo** podemos **imprimir HTML**

**Altera** o **código** do teu exemplo anterior para

```
echo "<font color='yellow'>Olá turmaX!</font>";
```

*// yellow tem de estar entre plicas e não aspas. Explicação noutra aula.*

Aceder no **browser** a:

**127.0.0.1:8080/olamundo.php**

**Ver código fonte da página!**

*Programação de alto nível*



## Comentários

Para inserir um **comentário** em **php** podemos fazê-lo de **3 formas distintas**

Comentar uma linha ou parte dela		Comentar várias linhas
// Comentário aqui	# Comentário aqui	/* Comentário aqui */

**<?PHP**

**echo "Olá turma X!"; //o echo serve para imprimir no ecrã**

**//echo "Olá turma11i!";**

**#echo "Olá turma10i!";**

**/\*echo "Olá turma10j!";**

**echo "Olá turma11j!";**

**echo "Olá turma12j!";\*/**

**?>**

os **comentários** são **úteis** como **memorando**, indicando o que cada parte do código é responsável por fazer, ou para **comentar** parte do **código** que **não** seja **necessário** a dado momento.

Programação de alto nível

## Elementos Básicos em PHP

### Operadores aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão inteira

### Operadores de incremento e decremento

Operador	Descrição
++	Incremento
--	Decremento

### Operadores relacionais ou comparativos

Operador	Descrição
==	Igual a
!=	Diferente de
<	Menor que
>	Maior que
<=	Menor ou igual que
>=	Maior ou igual que

### Operadores aritméticos de atribuição

Operador	Descrição
+=	Adição seguida de atribuição
-=	Subtração seguida de atribuição
*=	Multiplicação seguida de atribuição
/=	Divisão seguida de atribuição
%=	Resto da divisão inteira seguida de atribuição

### Operadores lógicos

Operador	Descrição
!	Not – Negação
&&	And – e
	Or – ou

### Operadores de indicação de tipo

Operador	Descrição
gettype()	Indica o tipo (string, number, etc)

Programação de alto nível

## Declaração de Variáveis

As variáveis em PHP **não necessitam** de ser **declaradas previamente**

Passam a existir desde o momento em que aparecem no código

Todas as variáveis são **precedidas** do símbolo **\$**

**Exemplo:**

```
$x=10;
```

```
$y="Olá turma!";
```

Achas que consegues imprimir estas variáveis no ecrã?

Regras:

As **variáveis** começam **obrigatoriamente** por uma **letra** ou **underscore**;

Linguagem **PHP** é **case-sensitive** ex: variáveis

```
$teste != $Teste != $TeStE
```

Programação de alto nível

## Concatenar

**Concatenar** significa **juntar elementos diferentes**: imprimir **texto** juntamente com **variáveis**

**Exemplo:**

```
$idade=10;
```

```
echo "O Dinis tem ".$idade." anos.";
```

Sempre que há uma **transição** entre elementos diferentes **texto→variável** ou **variável←texto** é necessário **separá-los** por um **ponto .**

Desafio: Imprime na próxima linha a seguinte frase

```
O teu_nome tem a_tua_idade anos.
```

Programação de alto nível

## Desafio I

Resolve o seguinte desafio:

1. Cria um **ficheiro** de nome **string.php** (dentro da pasta **root**)

a) Sabendo que as **variáveis**  $\$x = \text{"A turma do "}$  e  $\$y = \text{"12°X"}$  **imprime** no ecrã a seguinte **frase utilizando** as **variáveis anteriores**.

A turma do 12°X domina PHP.

b) **Imprime de novo** a **frase anterior**, noutra **linha**, mas a **bold** e a **cor azul**.

Programação de alto nível

## Estruturas de Controlo - Decisão

IF

(...)	if (condição) {bloco de instruções;} elseif (condição) {bloco de instruções;} else {bloco de instruções;} (...)	(...)	se (variável é isto) {faz isto; sai fora} mas se (variável é aquilo) {faz aquilo; sai fora} senão {faz outra coisa qualquer;} (...)
-------	-----------------------------------------------------------------------------------------------------------------------------------	-------	-------------------------------------------------------------------------------------------------------------------------------------------------------



A condição **if** **não implica** que tenha de **existir** sempre um **elseif** ou sequer um **else**. Podemos ter **apenas if** ou **if e elseif** ou **if e else** ou **if, elseif e else**.

Programação de alto nível

## Estruturas de Controlo - Decisão

Exemplo: Pretende-se **imprimir** no **ecrã** o **valor** de uma **variável** consoante ela tem o valor **0**, **1** ou outro qualquer (neste caso deve aparecer a mensagem "**Número Errado!**").

```
if ($x==0)
  { echo "Número " .$x; }
elseif ($x==1)
  { echo "Número " .$x; }
else
  { echo "Número errado! "; }
```

**Atenção** deves começar por **definir** o valor de **\$x** antes do if

Programação de alto nível

## Desafio 2

Cria um ficheiro de nome **iftipo.php**

Pretende-se saber que **tipo** de **utilizador** está a **entrar** um **Website**. Podem ser de **3** tipos.

- **Aluno;**
- **Professor;**
- **Administrador.**

**Testa** uma **variável**, de nome à tua escolha, **com** uma estrutura de decisão do tipo **IF**.

Deves **começar** por **atribuir** um **valor** à tua **variável** (tipo de **utilizador**) e **conforme** o **valor** da variável deve **aparecer** no ecrã "**Bem vindo X**" onde **X** pode ser **aluno**, **professor** ou **administrador**.

Programação de alto nível

## Desafio 3

Resolve o seguinte desafio:

1. Cria um ficheiro de nome `ifhora.php`

**Pretende-se** imprimir no ecrã **bom dia** (até 12h), **boa tarde** (entre 13h e 20h) ou **boa noite** (entre 21h e 11h) **utilizando** uma estrutura de decisão IF.

**Utiliza** a função `date("G")` para **obteres** a **hora atual** (atenção por defeito dá **menos uma hora** pelo que deves **resolver** esta **situação antes** de **fazer** a **comparação** no IF)

*Ajuda: imprime a função `date("G")` no ecrã para veres o seu formato e como deves fazer a comparação no IF.*

Programação de alto nível

## Estruturas de Controlo - Decisão

A estrutura de decisão **Switch-case** pode ser **aplicada** exatamente aos **mesmos casos** que a estrutura de decisão **IF**

**Switch-case melhor** quando existem **muitas condições**

*Switch...case*

(...)

```
switch (expressão) {
  case valor1: bloco de instruções; break;
  case valor2: bloco de instruções; break;
  (...)
  case valorN: bloco de instruções; break;
  default: bloco de instruções;}
```

(...)

Cada Instrução termina com a palavra **Break** que impõe **saltar fora** da **estrutura** Switch-case

**Default** será a instrução por **defeito**, a que não se encontra discriminada em nenhum case

Módulo V – Desenvolvimento de Páginas Web Dinâmicas



## Estruturas de Controlo - Decisão

*Switch...case*

(...)

```

Seleciona (variável a verificar valor) {
caso tenha valor a: faz isto; salta fora;
caso tenha valor b: faz aquilo; salta fora;
(...)
caso tenha valor z: faz isto e aquilo; salta fora;
por defeito: faz outra coisa qualquer;
}

```

(...)

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

## Estruturas de Controlo - Decisão

Exemplo: Pretende-se **imprimir** no **ecrã** o **valor** de uma **variável** consoante ela tem o valor **0**, **1** ou outro qualquer (neste caso deve aparecer a mensagem "**Número Errado!**") mas agora com uma estrutura de decisão do tipo **Switch-case**.

```

switch ($x) {
  case 0: echo "Número " . $x; break;
  case 1: echo "Número " . $x; break;
  default: echo "Número errado! ";
}

```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

## Desafio 2

Cria um ficheiro de nome **switchtipo.php**

Pretende-se saber que **tipo** de **utilizador** está a **entrar** um **Website**. Podem ser de **3** tipos.

- **Aluno;**
- **Professor;**
- **Administrador.**

**Testa** uma **variável**, de nome à tua escolha, **com** uma estrutura de decisão do tipo **Switch**.

**Aproveita** o **código** do ficheiro **iftipo.php** e **altera** a estrutura de decisão IF para a estrutura de decisão **Switch**.

Programação de alto nível

## Estruturas de Controlo - Ciclos

*For*

```
(...)
for (início;condição_de_paragem;incremento)
{
    bloco de instruções;
}
(...)
(...)
Desde (início; até que queremos que pare; incremento)
{
    O que queremos que se faça várias vezes.
}
(...)
```



Programação de alto nível

## Estruturas de Controlo - Ciclos

Exemplo: Pretende-se **imprimir** no **ecrã** por linha os números de **1** a **16**.

```
for($i=1; $i<=16; $i++)  
{  
    echo "Número " . $i. "<br>";  
}
```

Programação de alto nível

## Desafio 4

Resolve o seguinte desafio:

**Cria** um novo ficheiro **fornum.php**

**Pretende-se** que **imprimas**, por **linha**, os **números** de **23 (inclusive)** a **36 (exclusive)** utilizando um **ciclo FOR**.

Programação de alto nível

## Desafio

Cria um ficheiro de nome **for.php**

Altera o código seguinte para que seja possível construir dinamicamente uma tabela de 16 linhas e 2 colunas com o número de combinações possíveis por número de bits (16).

Nº de bits	Nº de Combinações
1	2
2	4
3	8
(...)	(...)

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <table border="1">
6 <tr> <td><b> Número <td><b> Quadrado
7 <?php
8 for ($i=1; $i <= 5; $i++) {
9     $q = $i*$i;
10    echo "<tr> <td> $i <td> $q";
11 }
12 ?>
13 </table>
14 </body>
15 </html>

```

Ajuda: Número de combinações  $NC = 2^{\text{nº de bits}}$

Em PHP para fazer elevado utiliza-se a função **pow(valor, elevado)**

Programação de alto nível

## Estruturas de Controlo - Ciclos

*While*

```

(...)
inicializar incremento fora do ciclo;
While (condição)
{
bloco de instruções a realizar repetidamente;
incremento;
}
(...)

```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas



## Estruturas de Controlo - Ciclos

*While*

(...)

**Enquanto** (a variável for/não for isto)...

```
{
```

...fazes o que estiver dentro das chavetas  
quantas vezes forem necessárias;

```
}
```

(...)

Módulo V - Desenvolvimento de Páginas Web Dinâmicas

## Estruturas de Controlo - Ciclos

Exemplo: Pretende-se **imprimir** no **ecrã** por linha os números de **1** a **16**, mas agora com uma estrutura de controlo **While**

```
$i=1; # Necessário atribuir valor inicial à variável fora do ciclo
while ($i<=16) {
    echo "Número=" . $i. "<br>";
    $i++; /* Necessário incrementar valor da
    variável no final de cada iteração */
}
```

Módulo V - Desenvolvimento de Páginas Web Dinâmicas

## Estruturas de Controlo - Ciclos

Exatamente igual à estrutura **While** mas **garante** que pelo **menos uma vez** o **ciclo** é **executado**

*Do-while*

```
(...)
    inicializar incremento fora do ciclo;
Do
{
    bloco de instruções a realizar repetidamente;
    incremento;
}
While (condição);
(...)
```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

## Estruturas de Controlo - Ciclos

Exatamente igual à estrutura **While** mas **garante** que pelo **menos uma vez** o **ciclo** é **executado**

*Do-while*

```
(...)
    Faz...
    {
        ...o que estiver dentro destas chavetas as vezes
        que forem necessárias...
    }
    ...Enquanto (variável for/não for isto);
(...)
```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

## Estruturas de Controlo - Ciclos

Exemplo: Pretende-se **imprimir** no **ecrã** por linha os números de **1** a **16**, mas agora com uma estrutura de controlo **Do-While**

```
$i=1; // Necessário atribuir valor inicial à variável fora do ciclo
do {
  echo "Número=" . $i. "<br>";
  $i++; /* Necessário incrementar valor da variável no final
  de cada iteração */
}
while ($i<=16);
```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas


## funções

As **funções** servem para que **não** seja **necessário repetir** o mesmo **código várias vezes** numa página.

*Function*

**Ou sem argumentos**

```
(...)
function nome_da_funcao ($argumento1,$argumento2,...)
{
  bloco de instruções;
}
(...)
```



Programação de alto nível

## funções

*Function*

### As funções podem ainda retornar valores

(...)

```
function nome_da_funcao ($argumento1,$argumento2,...)
{
bloco de instruções;
return variável ou expressão;
}
```

(...)

Programação de alto nível

## funções

Exemplo: Pretende-se **imprimir** no **ecrã** uma **mensagem "Dados incorrectos!"** mas **através** da chamada a uma **função**

```
function alerta(){
    echo "Dados incorrectos!";
}

alerta();
```

Programação de alto nível



## Desafio

Cria um ficheiro de nome **funcaonome.php**

**Copia o seguinte código para lá.**

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <?php
6 function escreverNome()
7 {
8     echo "Ricardo Sérgio";
9 }
10 echo "O meu nome é ";
11 escreverNome();
12 ?>
13 </body>
14 </html>

```

Cria uma nova função de nome **escreveridade** que imprima a tua idade.  
A frase final deve ser "O meu nome é Ricardo Sérgio e tenho X anos."

Programação de alto nível

## funções

Exemplo: Pretende-se **imprimir** no ecrã uma mensagem "**A soma dos golos do FCP à 2ª jornada são X**" onde **X** é o **valor** da **soma** dos **golos** das duas jornadas, **devolvidos** por uma **função**

```

function soma ($valor1, $valor2)
{
    $soma_val = $valor1 + $valor2; /*ao receber os golos da jornada 1 e 2
através dos argumentos da função soma-os*/
    return $soma_val; /*devolve o valor de golos somados para a variável
    $golos_FCP*/
}

$golos_jornada1=2;
$golos_jornada2=5;
$golos_FCP=soma($golos_jornada1,$golos_jornada2); /*chama a função
e como argumentos envia os golos da jornada 1 e 2*/
echo "A soma dos golos do FCP à 2ª jornada são ".$golos_FCP;

```

Programação de alto nível

## Classes e Objetos

A linguagem **PHP** é do tipo **POO – Programação Orientada ao Objeto**

Uma **classe** representa um **conjunto** de **funções**

Um **objeto** é uma **instância** ou **concretização** da **estrutura** definida numa **classe**

### Exemplo:

```

1 <?php
2 class ver → Criação da classe
3 {
4     function primeira()
5     {
6         echo "Esta é a primeira função da class!<br>";
7     }
8     function segunda()
9     {
10        echo "Esta é a segunda função da class!<br>";
11    }
12 }
13 $x = new ver; → Criação do objeto através do operador new
14 $x->primeira(); → Resultado: Esta é a primeira função da classe!
15 $x->segunda(); → Resultado: Esta é a segunda função da classe!
16 ?> → Forma de aceder ao conteúdo de uma classe através do objeto
  
```

Classe composta por duas funções

Programação de alto nível

## Classes e Objetos

### Exercício

**Substitui** as funções **primeiro()** e **segundo()** pelas funções **alerta()** e **soma()** que realizaste nos slides anteriores relativos a funções. Pretende-se que o **resultado** seja o **mesmo** que obtido apenas com as funções.

Programação de alto nível

## Variáveis locais e Globais

As **variáveis** em PHP podem ser **locais** ou **globais**

Quando definidas **localmente** apenas existem e têm valor onde foram definidas

### Exemplo:

```

1 <?php
2 $a=3;
3 function ver(){
4     $b=$a+2;
5     echo $b;
6 }
7 ver(); → Resultado: 2
8 echo $a; → Resultado: 3
9 ?>
```

Programação de alto nível

## Variáveis locais e Globais

Para o mesmo exemplo usar a variável **\$a** como **global** seria necessário acrescentar uma nova linha de código

### Exemplo:

```

1 <?php
2 $a=3;
3 function ver(){
4     GLOBAL $a; → nova linha que transforma $a em
5     $b=$a+2;     variável global
6     echo $b;
7 }
8 ver(); → Resultado: 5
9 echo $a; → Resultado: 3
10 ?>
```

Programação de alto nível

## Constantes

Podemos definir valores para **constantes** em PHP

No exemplo anterior era possível fixar o valor de **\$a** numa **constante**

### Exemplo:

```

1 <?php
2 define("A", 3);
3 define("ST", "Olá turma!");
4 function ver(){
5     $b=A+2;
6     echo $b;
7 }
8 ver();
9 echo A;
10 echo ST;
11 ?>
```

Definição da constante A com o valor 3  
 Definição da constante ST com a String Olá turma!  
 Resultado: 5  
 Resultado: 3  
 Resultado: Olá turma!

Por **convenção** deve usar-se todas as **letras capitais** para definir **constantes**.  
 Desta forma é mais **simples** a sua **identificação** no código.

Programação de alto nível

