

Programação Avançada

Pedro Quaresma

Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

2008/2009

Questões de Organização

- Uma correcta organização dos ficheiros/directórios.
- Uma correcta escolha dos nomes (ficheiros/variáveis/...).
- Documentação Interna e Externa.
- Programas de gestão de de Projectos.

- Uma programação multi-sistemas (Linux; MS-Windows; MacOS; ...).
- Soluções de acesso global (interface Web).
- Internacionalização/Localização na programação.

Bibliografia

- PHP and MySQL Web Development (3rd Edition) Luke Welling, Laura Thomson. Sams, 2004.
- dotProject, <http://www.dotproject.net/>
- Manual de referência do PHP, http://www.php.net/manual/pt_BR/
- Manual de referência do MySQL, <http://dev.mysql.com/doc/refman/4.1/pt/index.html>
- Doxygen manual, <http://www.stack.nl/~dimitri/doxygen/manual.html>
- Literate Programming, <http://www.literateprogramming.com/>

A diferença entre um manual (de estudo), e um manual de referência.

A solução LAMP

Linux+Apache+MySQL+PHP

Apache servidor de páginas capaz de processar o HTML e o PHP (entre outras).

MySQL um SGBD capaz de comunicar com várias linguagens de programação (PHP, ...).

PHP linguagem de programação genérica embutida em ficheiros HTML e capaz de comunicar com SGBDs.

HTML (Hypertext Meta-Language) linguagem para a construção de páginas da rede.

- Solução possível em: Linux; MacOS; MS-Windows; ...
- Solução servidor, isto é, não necessita de nenhuma funcionalidade especial nos navegadores dos “clientes”.

O programa *Apache* é o servidor de páginas “Web”.

- Responde aos pedidos dos clientes.
- Trata das questões de segurança inerentes a um serviço “público”.
- Tem como clientes os Navegadores com os quais comunica.
- Interpreta (se configurado para tal) a linguagem PHP, assim como outras linguagens “externas”.

- Num sistema Linux/Unix cada utilizador pode criar uma página:
 - ▶ `~nomeUtilizador/public.html/inicial.html` ← Sistema de ficheiros Unix.
 - ▶ `http://nomeDoDominio/~nomeUtilizador/inicial.html` ← URL
- Por omissão o ficheiro inicial designa-se por `index.html`
 - ▶ `~nomeUtilizador/public.html/index.html` ← Sistema de ficheiros Unix.
 - ▶ `http://nomeDoDominio/~nomeUtilizador/` ← URL
- Para que num ficheiro contendo código PHP este seja interpretado de forma correcta, a extensão do ficheiro tem de ser `php`

HTML — HyperText Markup Language

Uma colecção de “meta-marcas” (“markup tags”) usadas para definir as várias componentes de um texto da rede.

- “NCSA, A Beginner’s Guide to HTML” —
<http://www.ee.surrey.ac.uk/localcopy/HTMLPrimer.html>
- “Getting started with HTML” Dave Raggett,
<http://www.w3.org/MarkUp/Guide/>.
- Formulários I —
<http://www.w3.org/TR/html4/interact/forms.html>
- Formulários II —
<http://www.mat.uc.pt/~pedro/manuais/overview.html>

HTML, (muito) Breve Introdução

O HTML (HyperText Markup Language) é uma meta-linguagem com capacidade de referência. Isto é, é uma linguagem que descreve uma outra linguagem (a linguagem dos textos da rede) e que é capaz de incluir, num dado texto, referências a outros textos.

- Cabeçalho (“head”) - contém dados gerais sobre o texto que se segue.
- Corpo (“body”) - contém a descrição do texto através de uma série de comandos que lidam com as várias estruturas de um texto.
 - ▶ O espaçamento entre palavras é definido de forma dinâmica. Um espaço no texto fonte, vale tanto como vinte espaços, o resultado final é o mesmo.
 - ▶ Não tem a noção de linha, ou melhor de quebra de linha. As linhas podem ter uma largura variável.
 - ▶ Não tem a noção de página de texto. O comprimento dos textos é, em teoria, infinito.

HTML — Um exemplo

```
<html>
<head>
<TITLE>Um exemplo simples de um texto em HTML</TITLE>
</head>

<body>
<H1>HTML é simples de Aprender</H1>
<P>
Exemplo de um parágrafo em HTML.
Como podem ver tem a marca ‘tag’, ‘p’!
</P>
<P>
Todos (quase todos) os comandos HTML tem uma marca inicial e uma marca
final, sendo que essa só difere da inicial pelo prefixo ‘/’.
</P>
</body>
</html>
```

HTML, Marcas

- HTML** Marca o início (e o fim) do texto HTML. Informa o navegador que o texto contém código HTML (a extensão .html tem o mesmo efeito).
- HEAD** Define o cabeçalho - informação genérica sobre o documento.
- TITLE** Contém o título “exterior” do documento, isto é identifica o texto no contexto global (é uma das informações que o “google” procura).
- BODY** Define o corpo do documento, isto é, o documento que vai ser visível através do navegador. É aqui que se pode definir o texto através de um conjunto de marcas para as várias estruturas em que um texto é constituído.

HTML, (muito) Breve Introdução

- Como numa linguagem de programação “normal” o uso da indentação é opcional.
- Para textos normais (sem PHP) a utilização de editores especializados permite “esquecer” os pormenores da linguagem
- O HTML é uma linguagem sem a noção de estado (“stateless”), isto é não permite, entre outras, a comunicação entre textos através de parâmetros.
- A comunicação entre texto é feita através do mecanismos de formulários (“forms”).

HTML, Marcas (continuação)

- Headings** Cabeçalhos, o HTML tem seis níveis de cabeçalhos: <Hx>, com $x = 1, 2, \dots, 6$.
- Parágrafos** Dado que o HTML não tem a noção de linha (nem de espaçamento fixo), sempre que se quer começar um novo parágrafo é necessário usar a marca <P>.
- Listas** O HTML suporta: listas não numeradas; listas numeradas; e listas de definições.
 - **Listas Não Numeradas:** Marca de lista não numerada ; marca de um elemento (“item”) da lista .
 - **Listas Numeradas:** Marca de lista numerada ; marca de um elemento (“item”) da lista .
 - **Listas de Definições:** Marca de lista de definições <DL>; marca do título para o elemento a definir <DT>; marca de início da definição <DD>;

HTML, Marcas: Tabelas

A marca <TABLE> delimita o espaço de construção de uma tabela, isto é, um texto formatado em linhas, em que cada linha está dividida em colunas. Dentro desse espaço temos acesso às seguintes marcas:

Linhas cada linha da tabela é definida através da marca <TR>.

Cabeçalhos (das colunas) podemos definir uma linha de cabeçalhos (em geral a linha de topo) usando a marca <TH>, uma por coluna.

Colunas dentro de cada linha podemos definir as colunas através da marca <TD>.

HTML, Marcas (continuação)

Texto pre-formatado <PRE> secção de texto em que os espaços e as mudanças de linha são significativas e em que o tipo de letra usado é de largura fixa. Ótimo para incorporar descrições de programas escritos numa dada linguagem de programação.

Mudanças de linha pode-se forçar a mudança de linha utilizando a marca
. Note-se que neste caso não se está perante um par de marcas, só existe a marca descrita.

Linhas Horizontais pode-se introduzir uma linha horizontal (a separar duas secções de texto), através da marca <HR>. Esta é também uma marca isolada.

HTML, Atributos

Muitas das marcas no HTML podem ter a sua acção modificada através da utilização de atributos. Por exemplo <P ALIGN="center"> permite definir um parágrafo centrado.

HTML, Formulários

Sendo que o HTML é uma linguagem sem a noção de estado... como é que é possível comunicar entre páginas, e entre uma página e um programa?

A resposta é dada pelos formulários ("Forms").

```
<FORM ACTION="url"> ... </FORM>
```

Os formulários em HTML podem ser de dois tipos:

GET os dados são passados através do URL.

POST os dados são passados através de uma estrutura do tipo tabela (vector).

Temos então a marca <Form> a marcar o início/fim de um formulário. Neste caso a especificação dos atributos é essencial.

HTML, Formulários

Atributos da marca "Form"

ACTION define o "URL" do programa que é suposto processar a informação recolhida pelo formulário.

METHOD como já disse pode ser "GET", ou "POST". Qual dos métodos a utilizar vai depender da aplicação pretendida, em geral o primeiro reserva-se para quando a quantidade de informação a passar é pequena, ficando o outro método reservado para os outros casos.

No âmbito dos formulários temos acesso às seguintes marcas:

INPUT um elemento simples.

SELECT a escolha de um elemento entre várias opções.

TEXTAREA um elemento multi-linhas.

HTML, Formulários, INPUT

```
<input type='text' name='nomeVar' size='80' maxlength='80'>
```

Os atributos desta marca são:

TYPE text; password; checkbox; radio; submit; reset.

NAME define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.

VALUE pode ser usado para se definir o valor do elemento.

CHECKED especifica se um elemento do tipo checkbox ou radio está activado, por omissão.

SIZE tamanho físico (em caracteres) do campo de entrada tal como ele vai ser formatado.

MAXLENGTH tamanho máximo (em caracteres) do elemento de entrada.

HTML, Formulários, SELECT

```
<SELECT NAME="lista-de-opções">
<OPTION> Primeira opção
<OPTION> Segunda opção
</SELECT>
```

Os atributos desta marca são:

NAME define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.

SIZE 1 - lista de opções; 2 - lista de opções com barra de corrimão; $n > 2$, número de elementos visíveis.

MULTIPLE se presente pode-se escolher vários elementos

Os atributos da marca "OPTION" são:

SELECTED especifica que a opção em causa é seleccionada por omissão.

HTML, Formulários, TEXTAREA

```
<TEXTAREA NAME="nomeVar" ROWS=4 COLS=40>
Este é o conteúdo por omissão
</TEXTAREA>
```

Os atributos desta marca são:

NAME define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.

ROWS o número de linhas da janela de entrada.

COLS o número de colunas (caracteres) da janela de entrada.

PHP — Bibliografia

- Manual do PHP (“on-line”)
http://www.php.net/manual/pt_BR/
- Luke Welling & Laura Thomson, *PHP and MySQL Web Development*, 3rd Edition, Sams Publishing, 2005
- Andy Harris, *PHP 5/MySQL Programming*, Premier Press, 2004.

PHP

Linguagem de programação genérica embutida em ficheiros HTML.

- Um ficheiro de extensão `php` pode conter: HTML; marcas PHP; código PHP (entre um par de marcas PHP);
- O Apache (ou outro programa que faça a gestão do serviço) tem de estar configurado para interpretar o código PHP.
- Os ficheiros que contêm código PHP têm de ter a extensão `php`

Um exemplo de programa (`exemplo.php`).

```
<?php
echo "Olá mundo";
?>
```

O PHP é uma das mais populares linguagens para construção de páginas dinâmicas numa perspectiva de uma solução servidor (todo o processamento é feito no servidor).

O PHP providência uma muito fácil ligação a bases de dados.

PHP — Introdução

A sintaxe, do PHP é semelhante à linguagem C (assim como as potencialidades).

- **Variáveis**
 - ▶ sem declaração explícita, uma dada instrução de atribuição cria a variável, de um dado tipo, com um dado valor;
 - ▶ o valor, assim como o tipo de uma variável pode ser mudado por uma outra atribuição posterior.
 - ▶ identificam-se pelo carácter \$ inicial, por exemplo: `$nome`
- **Tipos**
 - ▶ Inteiros;
 - ▶ Reais;
 - ▶ Sequências de caracteres;
 - ▶ Lógicos (Booleanos);
 - ▶ Tabelas;

A linguagem é “case sensitive”, ou seja maiúsculas e minúsculas são consideradas diferentes.

PHP — Âmbito das Variáveis

O PHP não tem mecanismos (explícitos) de comunicação entre ficheiros, como tal o âmbito de uma variável está sempre ligado ao ficheiro aonde foi definida.

- Variáveis **Super Globais** – sempre visíveis (excepção ao que foi dito acima).
- **Constantes** – sempre visíveis.
- Variáveis **Globais** – em todo o ficheiro excepto dentro das funções.
- Variáveis declaradas dentro de uma função como estáticas, referem-se às variáveis com o mesmo nome, mas mantêm o valor entre chamadas sucessivas.
- Variáveis declaradas dentro de uma função são locais a essa função.

Algumas variáveis Super Globais

- `$_SERVER` - tabela contendo as variáveis de ambiente do servidor;
- `$_SESSION` - tabela contendo as variáveis de sessão;
- `$_GET` - tabela contendo as variáveis de um formulário HTML em modo "get";
- `$_POST` - tabela contendo as variáveis de um formulário HTML em modo "post";
- Os operadores são os usuais da linguagem C
- As precedência dos operadores são também as usuais;
- Os comentários são também os usuais.

Estruturas de Controlo ● composição sequencial;

- atribuição.
- condicionais:
 - ▶ if
 - ▶ if else
 - ▶ elseif
 - ▶ switch
- Ciclos:
 - ▶ while
 - ▶ for
 - ▶ for each
 - ▶ do ...while
- Funções.

PHP — Programação Modular

Programação modular - O PHP tem uma dupla personalidade: programação imperativa clássica; programação orientada aos objectos. Nestas breves notas vou tratar só da primeira aproximação.

Funções a declaração de funções segue a sintaxe da linguagem C;

```
function nome_da_funcao($arg1,$arg2,...,$argN);
```

Os argumentos são passados por valor, se se pretender obter um valor de saída é necessário recorrer à referência do identificador (&\$argK);

Comunicação entre funções a chamada de uma função é também idêntica à da linguagem C;

```
nome_da_funcao(val1,val2,...,valN);
```

Além dos argumentos a comunicação pode também ser feita através da utilização de variáveis globais.

Comunicação entre módulos por módulos entenda-se ficheiros. Esta é fácil de responder: não é possível.

PHP — Comunicação Entre Ficheiros

Comunicação entre ficheiros: O PHP não possui nenhum mecanismos de comunicação entre ficheiros, isto dado que o protocolo HTML não tem a noção de estado.

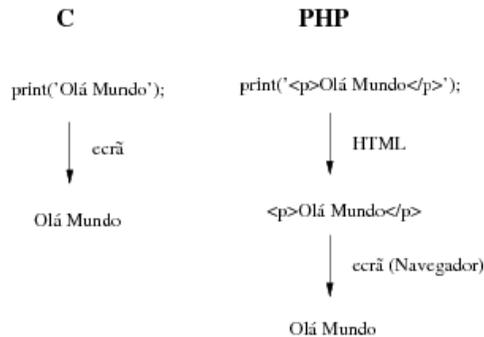
Soluções?

- Formulários (POST ou GET).
- Sessões (com as respectivas variáveis de sessão).
- Ficheiros auxiliares: é possível ler e escrever ficheiros

PHP & HTML

A *entradas* e *saídas* na linguagem PHP são sempre feitas através da linguagem HTML:

Saídas Os comandos usuais de visualização de uma linguagem de programação estão presentes no PHP (sintaxe similar ao C), no entanto elas têm de ser pensadas como tendo um passo intermédio que é o HTML



PHP & HTML

Entradas a leitura de valores é feita somente através dos formulários HTML

- especificação do ficheiro PHP que vai processar os dados do formulário (no formulário HTML):
`<form action='processa.php' method='post'>`
- Especificação dos nomes dos campos que vão receber os valores do formulário (no formulário HTML):
`<input type='text' name='nomedocampo'>`
- a “leitura” dos valores no PHP é feita através da matriz associativa `$_POST` (ou `$_GET`) (no ficheiro PHP):
 `$nomeVariavel = $_POST['nomedocampo'];`

MySQL — Grupos de Trabalhos 2006/2007

Cada grupo tem uma conta em “‘rena2.mat.uc.pt’” e uma base de dados.

```
pedro@hilbert: mysql -u bdGrupo1 -p -h rena2.mat.uc.pt
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 781 to server version: 4.1.13
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show databases;
+-----+
| Database |
+-----+
| bdGrupo1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

MySQL

Referências — MySQL, páginas oficiais.

- <http://www.mysql.com/> — Página geral
- <http://dev.mysql.com/doc/refman/5.1/en/> — Manual 5.1 (inglês)
- <http://dev.mysql.com/doc/refman/4.1/pt/> — Manual 4.1 (português)

phpMyAdmin — ferramenta de administração.

- http://www.phpmyadmin.net/home_page/index.php

Alguns dos comandos mais úteis na linha de comando:

- `use` — seleccionar uma base de dados;
- `show databases/tables` — mostrar as bases de dados/tabelas acessíveis;
- `describe <nome_da_tabela>` — mostra a informação respeitante aos campos de uma dada tabela.

PHP & MySQL

A ligação entre o PHP e o MySQL é feita através de um conjunto de funções própria do PHP.

- **Ligação ao Servidor MySQL** função `mysql_connect`, argumentos: nomes do servidor, utilizador e senha de acesso.

```
@ $ligacao = mysql_connect($servidor,$utilizador,$senha);
if (mysql_errno()) { // verifica a ligação ao servidor
    echo "<p>Erro: ligação aos servidor não possível</p>";
    exit;
}
```

- **Ligação à Base de Dados do Grupo:** função `mysql_select_db`, argumentos: nomes da base de dados.

```
if (!mysql_select_db($nomeBD)) { // verifica a ligação à BD
    $msg=mysql_error();
    echo $msg;
    echo '<p>Erro: ligação à Base de Dados não possível</p>';
    exit;
}
```

PHP & MySQL

- **Comandos SQL:** coloca-se a pesquisa que se quer efectuar numa variável do tipo "string", e após isso usa-se a função `mysql_query`, tendo como argumentos os comandos SQL e a ligação.

- ▶ Obter elementos

```
$sql = "SELECT correio_electronico,utilizador,senha FROM pessoa";
$resultado = mysql_query($sql,$ligacao);
```

- ▶ Inserir elementos

```
$sql = "INSERT INTO pessoa (correio_electronico,utilizador,senha)
VALUES ('ana@ana.pt','Ana','xpto')";
$resultado = mysql_query($sql,$ligacao);
```

- **Obtenção dos Resultados:** os resultados são obtidos através de várias funções, entre elas temos `mysql_fetch_assoc` que permite recolher os resultados obtidos numa tabela associativa;

```
$linha = mysql_fetch_assoc($resultado);
$utilizador = $linha['utilizador'];
$senha = $linha['senha'];
```