



1

formulários

Uma pequena **revisão**:

```

1 <html>
2   <head>
3     <title> Formulário </title>
4   </head>
5   <body>
6     <form>
7       Nome:
8       <input type="text" name="nome" value="Escreva o seu nome" > <br>
9       <input type="submit" value="Confirmar">
10      <input type="reset" value="Limpar">
11    </form>
12  </body>
13 </html>

```

Resultado:

Nome:

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

2

Form Action

Num formulário tipo **username/password** como é possível captar os dados que o utilizador introduziu e sabe se **estão corretos**?

Só com linguagem HTML isso é impossível fazer. É **necessário indicar** o **nome** do **ficheiro PHP** que se irá encarregar de captar os dados.

Exemplo:
`<form action="fx_que_capta_os_dados.php">`

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

3

Form Action

Conclusão:

Obrigatórios **2 ficheiros**

Ficheiro com o formulário ex: **formulario.html**

Ficheiro que **capta** os dados do formulário ex: **capta_dados.php**

No **ficheiro formulario.html** o atributo **Action** da etiqueta **<form>** tem de ter o **nome** do **ficheiro php** que irá **captar** os **dados**, neste caso **capta_dados.php**

`<form action="capta_dados.php">`

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

4

Form Method

Falta ainda **indicar** de que **forma** se vão **captar** os **dados**

Exemplo:

```
<form method="POST" action="fx_que_capta_os_dados.php">
```

ou

```
<form method="GET" action="fx_que_capta_os_dados.php">
```

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

5

Get e Post – Qual usar?

O **GET** e o **POST** são em tudo similares. Contudo existem algumas **diferenças**.

GET

- **Não** existe **confidencialidade** – os dados passados por este método **aparecem** na barra de endereços (**URL**) do browser, pelo que **passwords** ou **dados confidenciais não** devem ser **passados** por este método
- **Limite** de **carateres** – os dados a passar têm um **limite** de **100** carateres

POST

- **Garante** **confidencialidade** – os dados passados por este método **não aparecem** na barra de endereços (**URL**) do browser
- **Não** há **limite** de carateres

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

6

Exercício

Adiciona uma nova caixa de texto ao formulário para ser preenchida com a tua idade.

Altera o ficheiro PHP para que seja possível também captar a tua idade e escrevê-la no ecrã.

9

GET e POST - Arrays Associativos

Como vimos anteriormente, os **arrays associativos** associam uma **chave** a cada posição do **array**

Os arrays `$_GET[]` e `$_POST[]` são associativos pelo que guardam toda a informação de um formulário através do **par chave** (**nome** dos **elementos** do **formulário**) **valor** (**valores preenchidos** pelo utilizador)

Exemplo:

Nome: <input type="text" value="Joana"/>	Nome: <code><input type="text" name="nome"></code>
Sobrenome: <input type="text" value="Lopes"/>	Sobrenome: <code><input type="text" name="sobrenome:"></code>
Idade: <input type="text" value="27"/>	Idade: <code><input type="text" name="idade" size="3"></code>
<input type="button" value="Enviar dados"/>	

Para este formulário os arrays `$_GET[]` ou `$_POST[]`, dependendo do método, **armazenam** automaticamente os **3 campos** do **formulário**

Array associativo

`$_GET[]` ou `$_POST[]` →

[“nome”]	Joana
[“sobrenome”]	Lopes
[“idade”]	27

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

10

GET e POST - foreach

Acrescenta a caixa de texto para o sobrenome e realiza o exercício:

Para o exemplo anterior,

Nome:

Sobrenome:

Idade:

(...)

```
echo "Nome:" . $_POST["nome"];
e o resto???
```

(...)

Deverá ser mostrado no ecrã (1 resultado por linha)

Nome: conteúdo_da_caixa_de_texto_Nome
Sobrenome: conteúdo_da_caixa_de_texto_Sobrenome
Idade: conteúdo_da_caixa_de_texto_Idade

Módulo V – Desenvolvimento de Páginas Web Dinâmicas

11



12