

PROGRAMAR

A REVISTA DE PROGRAMAÇÃO PORTUGUESA

Revista nº2 - Maio de 2006

www.portugal-a-programar.org

Começar a Programar

Visual Basic

Comece já a programar para Windows com este guia completo

Introdução ao PHP

Aprenda a fazer sites Dinâmicos

Visual Basic.NET

Coding conventions, porquê?

XGL

A arquitectura que está a revolucionar os desktops environments do mundo GNU/Linux

- <03> Notícias
- <04> Tema de capa
- <09> A programar
- <15> GNU/Linux
- <16> Internet
- <17> Em análise
- <18> Dúvidas
- <19> Blue screen
- <20> Comunidade P@P



<Equipa PROGRAMAR>

<Administração>

deathseeker25 (Rui Maia)
d_pintassilgo (David Pintassilgo)
melSpeedl.ine (Márcio Lima)
satanuke (Hugo Violante)

<Coordenador>

neon_prannock (Sérgio Santos)

<Redacção>

vbmaster (Miguel Pais)
marceluh (Marcelo Martins)
revX (Ricardo Ribeiro)
Ped@ntilva (Pedro Silva)
perdido_e_sozinho (Sérgio Matias)

<Colaboradores>

AriOps (Daniel Correia)

<Contacto>

revistaprogramar@portugal-a-programar.org

A segunda vaga

Após o lançamento da primeira edição da revista PROGRAMAR, foi possível receber o *feedback* de muitos leitores que conheciam, ou passaram a conhecer a comunidade **Portugal-a-Programar**. Em geral, a revista foi muito bem recebida e foi possível recolher muitas críticas construtivas. Com base nessas críticas e na nossa auto-avaliação, procurámos tornar esta edição da revista como um exemplo da nossa dedicação em partilhar todos os conhecimentos que fomos adquirindo. Assim, tentámos também elevar o nível deste nosso projecto, para que seja reconhecido por todos os programadores. Esse é o nosso objectivo.

Para o desenvolvimento desta edição, podemos contar com o apoio de mais voluntários, aos quais desde já agradecemos. A sua presença e participação foram importantes para manter a motivação e dar novas ideias à nossa equipa. Com o lançamento da segunda edição da revista, determinámos o período de desenvolvimentos desta. A partir de agora uma nova edição será lançada a cada dois meses. Esta edição marca também o aniversário da comunidade **Portugal-a-Programar**, assunto a ser abordado na secção Comunidade.

</neon_prannock>

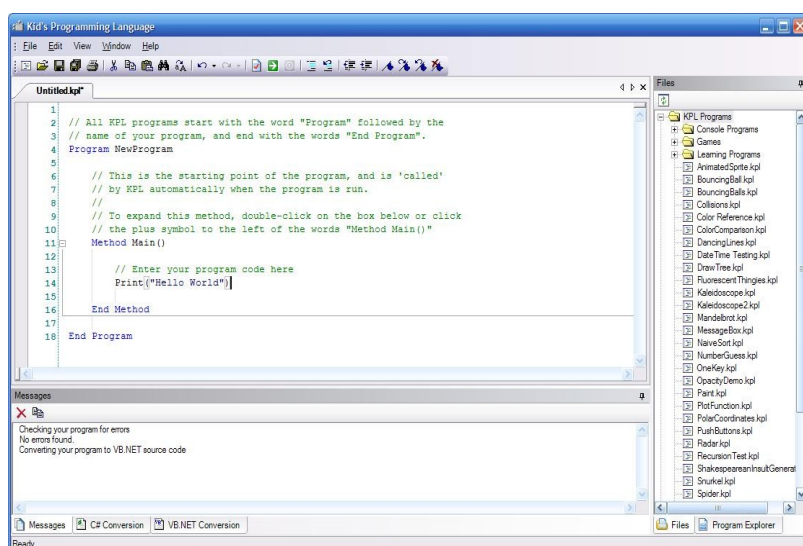
Kids Programming Language

O KPL, Linguagem de Programação para Crianças, é uma linguagem muito recente, criada exclusivamente para atrair as crianças ao mundo da programação. Morrison Schwartz, criador desta nova linguagem, orientou esta linguagem para o desenvolvimento de jogos, pois talvez será a melhor forma de incentivar as crianças a utilizarem a mesma. Para programarem em KPL, será apenas necessário a leitura, a matemática e a digitação, necessidades estas que serão desenvolvidas mais rapidamente com a ajuda da programação.

Com o IDE disponível para download, é possível fazer a conversão dos nossos programas em KLP para VB.NET ou C#.

Site Oficial:

<http://www.kidsprogramminglanguage.com/>



Google em grande



O Google obteve um aumento de 60% nos lucros do primeiro trimestre de 2006. Esse valor equivale a lucro líquido de 592,5 milhões de dólares, segundo o anúncio da empresa. Na origem dos bons resultados está o aumento das vendas de publicidade. Entre Janeiro e Março, as receitas da empresa que gere o motor de busca mais popular do mundo cresceram 79%, ultrapassando pela primeira vez a barreira dos dois mil milhões de dólares (2,25 mil milhões). No primeiro trimestre de 2005 o Google havia registado receitas de 1,26 mil milhões de dólares.

No período em análise, o Google viu a sua liderança entre os motores de busca mais utilizados reforçada, aumentando a quota de pesquisas online de 50 por cento para 60 por cento, à frente do Yahoo! e do MSN.

O presidente do Google, Eric Schmidt não esconde a sua satisfação: "Estamos obviamente muito felizes com os nossos resultados do primeiro trimestre. Tivemos boas notícias em todas as divisões".

Ganhe dinheiro convertendo utilizadores para o Firefox

Foi criado um novo projecto para ajudar a converter utilizadores do Internet Explorer. A ideia consiste num script que os webmasters podem adicionar ao site, e que detecta que browser os visitantes estão a usar. O script pode simplesmente alertá-los para as vantagens de usar o *browser* da Mozilla, colocando um link para obterem o Firefox com a Google Toolbar, ou até mesmo bloquear os *users* do *browser* da Microsoft. A parte mais atractiva é que a Google irá dar \$1 por cada pessoa convertida. Para isso basta ter uma conta Google AdSense.

Mais informações em: <http://www.explorerdestroyer.com>

Algo em que a nossa comunidade tem vindo a querer intervir, e que tem sido um dos objectivos principais desde a sua criação, foi o objectivo de alargar a programação a todos os utilizadores de computador. Esta, ao contrário do que muitos pensam, não é aquele *bicho de sete cabeças* que só os mais inteligentes conseguem aprender. Pelo contrário, a sua iniciação é bastante fácil, só dificultada pelos programas que se possa eventualmente precisar para programar na linguagem em questão.

Este pretende ser um tutorial que apoie os utilizadores novatos nos seus primeiros passos com a linguagem de programação Visual Basic, para que comecem a sentir-se à vontade, e para que percebam realmente que qualquer um pode programar.

Requisitos:

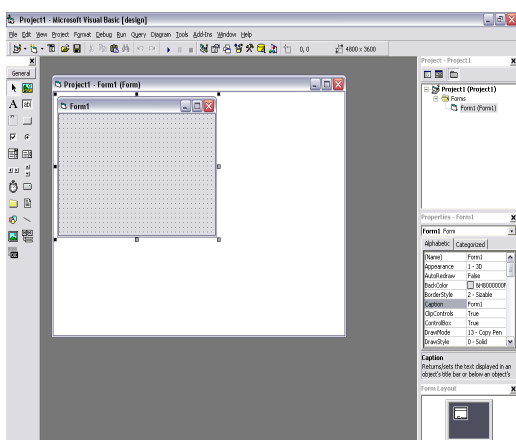
- IDE Microsoft Visual Studio 6 ou Microsoft Visual Studio.net 2002/2003 ou ainda Microsoft Visual Studio.net 2005 (à venda em lojas de software);
- Disponibilidade para aprender;
- Paciência (qualidade que terá de vos acompanhar ao longo da vossa cruzada da programação);
- Curiosidade (nem imaginam o montante de coisas que se descobrem bastando apenas possuir curiosidade);

Apertem os cintos, pois sairão daqui como um peixe na água, com o Visual Basic.

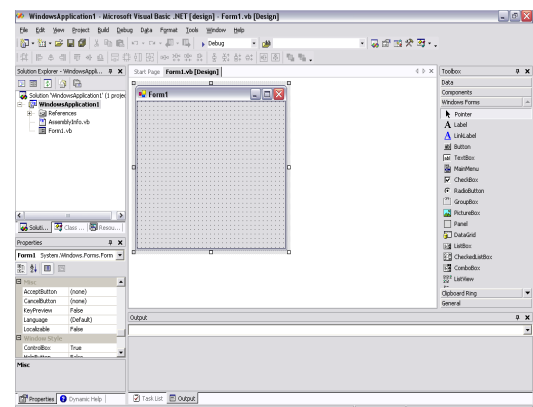
Antes de pormos as mãos ao trabalho, convém ambientarmos com o ambiente em que vamos produzir e desenvolver os nossos programas de iniciação. Como puderam constatar na caixa Requisitos, há várias versões disponíveis do Visual Studio, pacote que permite o desenvolvimento de linguagens de programação desde o c++ ao vb, passando pelo c# e asp (algumas delas só a partir da versão .net 2002/2003). Em primeiro lugar, vamos analisar as três versões do Visual Studio, nas quais poderão desenvolver os vossos programas.

A primeira, e mais antiga (aproximadamente de 1998) é o Visual Studio 6. Apesar de já estar bastante desactualizada e não conter as facilidades e funções importantes das outras duas, ainda se comporta na perfeição para um iniciante em vb poder começar a programar, proporcionando-lhe um IDE rápido e eficaz.

Por volta de 2002 a Microsoft decidiu fazer uma mini revolução na forma como se programava e para isso lançou uma nova versão do seu pacote de IDE's, desta vez denominada Visual Studio.net 2002, ou Visual Studio 7. Integrando a versão posterior de 2003, com a de 2002, podemos caracterizá-la por uma série de inovações que deixaram todo o mundo da programação pasmado. Apesar de tudo, os programadores da Microsoft souberam, nalguns aspectos, complicar o que era simples no Visual Studio 6, pelo que o upgrade de conhecimentos tornou-se um pouco atribulado.

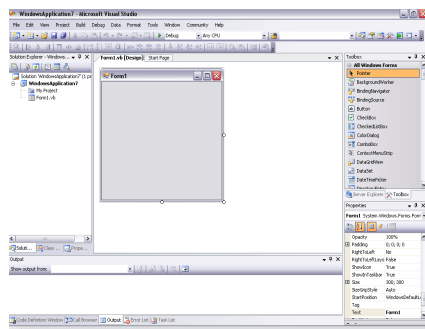


O ambiente de desenvolvimento integrado do Visual Basic 6



O ambiente de desenvolvimento integrado do Visual Basic.net 2003

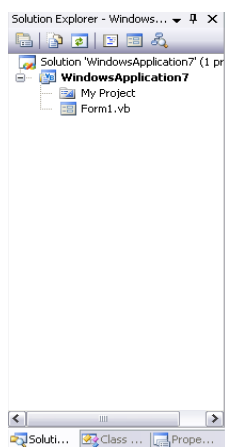
Quando todos pensavam que já não se podia inovar mais, eis que os programadores da Microsoft dão um safanão ao mundo e o põem-no de novo surpreendido. A nova versão que tinham acabado de lançar era simplesmente estupenda. Integrava toda as vantagens a que tínhamos ficado habituados no .net 2002/2003, com a simplicidade que reinava no vb6. A classe My. foi umas das grandes vantagens. É a versão que eu recomendo para os iniciantes e para todos os programadores desta grande linguagem.



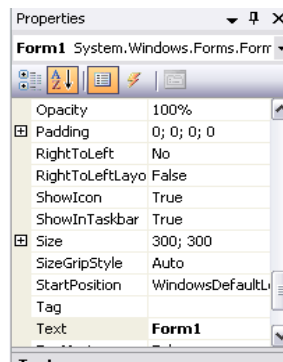
Aspecto do ambiente de desenvolvimento integrado do Visual Basic.net 2005

Agora que já demos uma olhadela pelos vários ambientes nos quais podemos desenvolver os nossos programas, convém dar ao utilizador uma ideia do porquê das janelas que lá se encontram e para que servem. Tentarei ao máximo manter a compatibilidade com as restantes versões, mas os exemplos dados serão da .net 2005.

Logo à esquerda (na maior parte dos IDE's), encontramos a Solution Explorer. Esta é a janela que nos dá informações sobre a quantidade e conteúdo de ficheiros que compõe o nosso projecto. É lá que por exemplo podemos adicionar mais *Forms* (comum janela do Windows standard que aparece ao fazer um novo projecto de Visual Basic) ao nosso projecto, bem como todo o tipo de itens.



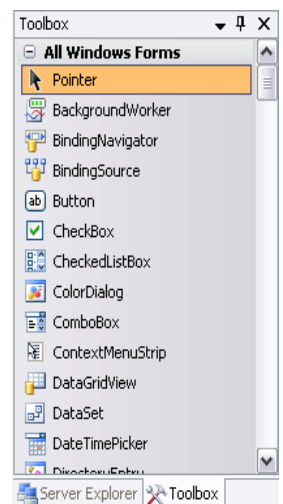
A Solution Explorer Window



A Proprieties Window

Outras das janelas mais importantes é a *Proprieties Window*. É nela que podemos editar todas as propriedades dos objectos inseridos na *Form*, bem como as suas próprias propriedades. De referir que tudo o que lá se faz pode ser feito por código.

Por fim, um utilizador novato apenas precisa de tomar conhecimento da barra mais importante do IDE, a *Toolbox*, esta é onde se encontram todos os componentes que se podem "desenhar" na *Form*, e posteriormente adicionar código. Incluí desde botões a caixas de texto, passando por menus, *optionboxes*, *comboboxes*, etc, etc.



A Toolbox

Mas no meio de tudo isto, é natural que o utilizador iniciante se sinta confundido, sem sequer saber como chegar às janelas mostradas nas imagens dos IDE's.

Bem, para começarmos a *codar* precisamos apenas de:

- Em VB6: Abrir o Visual Basic 6, visto que nesta versão ele encontrava-se separado do resto dos IDE's, e não integrado como nas .net. Em seguida aparece uma janela que nos pergunta o tipo de projecto que queremos. O *Standard* dá perfeitamente para a maior parte dos casos;
- Em vb.net 2002/2003 e vb.net 2005, visto que todos os IDE's se encontram aglutinados num programa, há que fazer, *New Project*, seleccionar *Visual Basic Project* e por fim *Windows Application*, não esquecendo de dar o nome respectivo.

Vamos então entrar um pouco mais no âmbito desta linguagem.

Mais algumas informações...

Antes de começarmos propriamente a programar convém saber algumas noções sobre programação. Vamos começar por falar de variáveis.

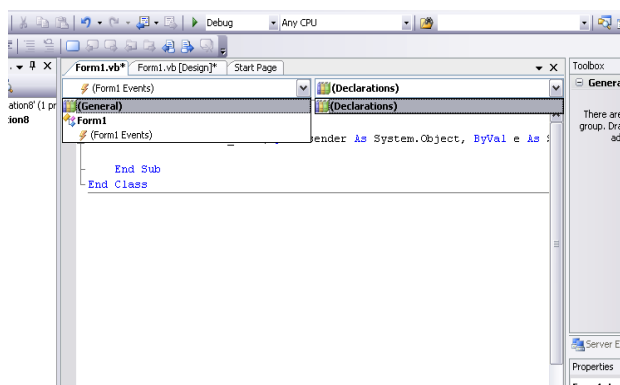
Variáveis são locais reservados na memória do computador para receber um determinado valor. Estas têm obrigatoriamente de ter um nome e um tipo especificado. Por exemplo, podemos reservar na memória espaço com o nome de número, que fica, durante a execução do nosso programa, à espera que lhe seja passado um valor numérico a guardar.

Em *Visual Basic*, estas declaram-se com o seguinte código:

```
Dim nome_da_variavel As Tipo
```

O tipo pode ser inteiro, real, string (texto), etc. Convém também ver que o nome destas nunca deve conter números ou espaços.

Deve-se ter sempre atenção ao número de variáveis que o nosso programa vai utilizar, de maneira a declará-las logo o mais cedo possível, e tornar a posterior leitura do código muito mais simples. Geralmente podemos declará-las em qualquer área, mas por vezes isso pode trazer inconvenientes. Como tal, quando se tem dúvidas, declara-se sempre as variáveis no *General Declarations*. Esta é uma instância que as mantém públicas para toda a *Form*, sem nunca trazer inconvenientes, a não ser na leitura do código, se o user não for cuidadoso.



O *General Declarations*

Nota: Pode-se aceder ao *General Declarations* fazendo duplo click na nossa *Form*, para abrir a janela de código e, em seguida, seleccionar no primeiro drop-down que se observa no topo dessa janela, *General*, e no segundo dropdown, seleccionar *Declarations*.

Como de pequenino é que se torce o pepino, convém também passar já neste momento ao utilizador alguns truques que o poderão ajudar a organizar o seu código. Se tem um espírito curioso, certamente já se fartou de adicionar e tirar componentes da *Form*, e de certo já reparou que ao adicionar, por exemplo, uma *textbox*, ela toma o valor de *Textbox1* (*Text1* em vb6). Se adicionar outra, tomará o valor *TextBox2* (*Text2* em vb6). Ao fim de pouco tempo, num projecto de grandes dimensões, estaria confuso sobre qual é qual e o que faz cada um, e então, para quem possivelmente fosse analisar o código do seu programa mais confuso pareceria.

É para isso que a **notação Hungara** surgiu: ao adicionar componentes, lhes dê um nome que tanto diga que espécie de objecto são, bem como a função que têm no programa.

Ex: uma *textbox* de um programa calculadora que resulta a soma entre dois valores, poderá chamar-se *txtResultado*. Repare as primeiras três letras referem que o objecto é uma *TextBox* e as restantes o que é que ela executa no programa. Pode mudar o nome dos objectos ao seleccioná-los quando já "desenhados" na *Form*, e alterar a Propriedade *Name* na *Properties Window*.

Mais à frente nesta edição encontrará um tutorial sobre este mesmo assunto, de nome *Coding Conventions, Porquê?*

Programa – Hello World

Este é um programa bastante simples, que apenas vai introduzir a palavra 'olá' na caixa de texto *txtMensagem* quando carregado o botão *btnMensagem*.

Precisamos de:

- Uma *Label* de nome *lblTitulo*;
- Uma *TextBox* de nome *txtMensagem*;
- Um *Button* (ou *CommandButton* em vb6) de nome *btnMensagem*;

Para fazer tal coisa, basta desenhar os controlos na *Form*, assim como eles aparecem na imagem (todas as propriedades dos controlos estão acessíveis na *Properties Window*, entre elas a propriedade *Text* das *Labels* (*Caption* em vb6), que é a que define o texto que apresentam, bem como a *Text* dos botões (*Caption* em vb6)).

Para fazer com que, ao carregar no botão, o texto apareça na *TextBox* é muito simples: basta fazer um duplo clique sobre o botão. Novamente aparece a janela de código, e lá em cima no dropdown deve estar seleccionado, no da esquerda, *btnMensagem*, e no da direita, o evento *Click*. ou seja, este será o código a ser executado quando fazemos um clique sobre o botão.

O código é o seguinte (e deve ser colocado entre o *Private Sub* e o *End Sub*):

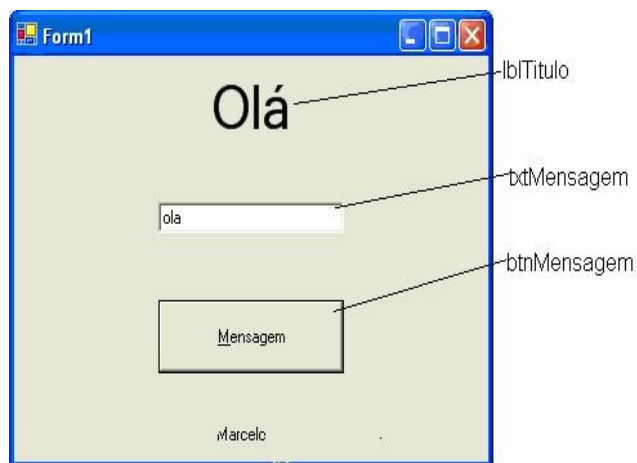
```
txtMensagem.Text = "Olá"
```

Ao carregar no botão, iremos portanto, editar a propriedade *Text* da *txtMensagem* (que é a que determina o texto inserido nesta), e mudá-lo para a palavra "Olá". Ao adicionar uma *TextBox* em vb6 e vb.net 2002/2003 o texto inserido por defeito é o seu próprio nome, como já deve ter percebido podemos alterar esse valor na *Properties Window*.

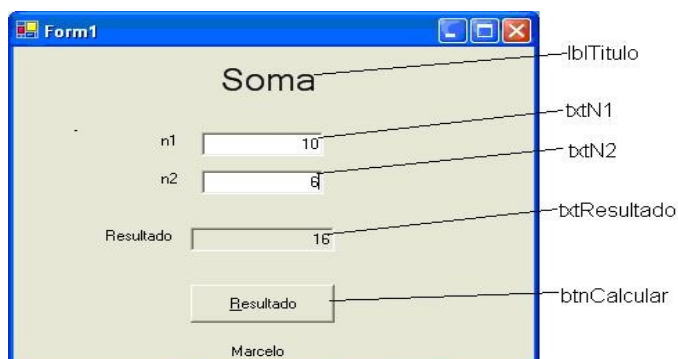
Para testar o seu programa basta fazer F5, ou carregar no símbolo semelhante a um botão de *Play* no seu IDE.

Para um iniciante o fazer o seu primeiro programa é uma sensação de alegria que não se consegue exprimir por palavras. Esperemos que tenha ficado motivado com a programação em vb, e prepare-se para as próximas lições.

Nota: Para distribuir o seu programa a amigos basta, nas versões .net, ir ao menu *Build* e fazer *Rebuild All*. Em vb6, basta ir ao menu *File* e fazer *Make programa.exe*. Na primeira hipótese o executável há-de ficar guardado no directório do projecto, na pasta *Debug*. Na segunda hipótese, o próprio programa pergunta onde quer guardar o .exe.



O aspecto do nosso primeiro programa



O aspecto do nosso segundo programa

Programa - Soma

Este programa vai nos permitir inserir dois valores numéricos, e em seguida apresentar numa *TextBox* o resultado.

Eu sei que soube a pouco, mas prometemos continuar a ensinar Visual Basic nas próximas edições. Até lá, não tenham medo do vosso próprio computador.... o vosso computador é que deve ter medo de vocês. :)

</vbmaster & marceluh>

Precisaremos de:

- Uma *Label lblTitulo*;
- Uma *TextBox txtN1*, uma *TextBox txtN2* e uma *txtResultado*;
- Um botão *btnCalcular*;

Para que o programa funcione basta apenas adicionar o seguinte código no evento *Click* do *btnCalcular*:

```
txtResultado.Text = Val(txtN1.Text) +  
Val(txtN2.Text)
```

Visto que as *TextBox* são, como o próprio nome indica, caixas de texto, é necessário incluir a preposição *Val()* antes de somar os dois valores. Desta maneira, o programa pegará no valor introduzido na primeira *TextBox* e no valor Introduzido na segunda *TextBox* e somá-los-á, apresentando-os na *txtResultado* ao editar a sua propriedade *Text*.

Procuram-se

Programadores dispostos ajudar na criação de artigos para revista de programação única no mercado português. Mais informações em:

www.revista-programar.info



“Uma linguagem dinâmica server-side”

PHP é a sigla de “Hypertext Preprocessor” que em português significa processador de Hipertexto. É uma linguagem open-source (o código é disponibilizado aos utilizadores) e é usada principalmente no desenvolvimento de aplicações server-side (as operações ocorrem do lado do servidor). Ou seja, o PHP é usado no servidor para tornar o código PHP em HTML legível aos browsers de Internet.

Podem até perguntar “E qual é a vantagem disso?”. A vantagem é que podemos usar o PHP para fazer cálculos, manipular dados, ligar a Bases de Dados, fazer com que 10 páginas HTML fiquem em apenas uma PHP, ter uma página com *login* de utilizador para tornar conteúdos privados, entre muitas outras coisas que irão aprender nestes tutoriais. A linguagem PHP é uma linguagem baseada noutras linguagens, nomeadamente C, Java e Perl. Com o tempo tem vindo a atingir uma grande dinâmica e está cada vez mais poderosa, com novas funcionalidades.

Antes de começarem a trabalhar em PHP é aconselhável que tenham alguma noção de HTML, pois o *output* do PHP é feito para HTML.

Esta linguagem interage com uma enorme quantidade de sistemas de base de dados entre as quais se podem destacar:

- MySQL;
- Oracle;
- IBM DB2;
- Microsoft SQL Server;
- PostgreSQL;
- SQLite.

O *download* do ficheiro de instalação PHP pode ser feito em <http://www.php.net/downloads.php>

Como funciona um ficheiro PHP?

O ficheiro tem como extensão “.php”, por exemplo “pagina.php” e este ficheiro funciona como um HTML limpo aos olhos do Navegador. O código do ficheiro PHP é executado antes e cria o *output* HTML, simples e directo sendo interpretado em conjunto com o restante código HTML do ficheiro.

“Olá mundo”

Como com todos os programadores quando interagem com uma nova linguagem de programação pela primeira vez, esta interacção começa pelo simples “Olá mundo!” e consiste em simplesmente imprimir o texto “Olá mundo” nessa linguagem.

Aqui podemos observar que no fundo é um ficheiro html onde adicionamos as tags `<?php ?>` e lá dentro executamos o comando PHP neste caso o `echo` que é o comando PHP que imprime texto no ecrã.

```
<?php
    echo "Olá mundo!";
?>
```

Resultado:

Olá mundo!

Aqui podemos observar que no fundo é um ficheiro HTML onde adicionamos as *tags* “`<?php`” e “`?>`” e lá dentro executamos o comando PHP neste caso o `echo` que é o comando PHP que imprime texto no ecrã.

PHP é uma linguagem de programação livre e muito usada para gerar conteúdos dinâmicos para Web.

Utilizar Variáveis

Em PHP podemos definir variáveis. As variáveis consistem essencialmente num modo de diminuir o tamanho do código e de o simplificar. Para definir variáveis utilizamos o símbolo `$`. As variáveis podem ser letras, números, ou conjuntos de objectos.

```
<?php
    $nome_da_variavel = "Olá mundo!";
    echo $nome_da_variavel;
?>
```

Resultado:

Olá mundo!

Aqui definimos a variável que se chama "nome_da_variavel" e o seu conteúdo que é o texto "Olá mundo!". De seguida fazemos o echo e imprimimos o conteúdo dessa variável para o ecrã do utilizador.

Para imprimir 2 variáveis seguidas usamos um ponto (.). Como por exemplo, para escrever-mos as variáveis *\$ola="ola"* e *\$mundo="mundo"* de modo a obtermos : "ola mundo" fazemos:

```
echo "".$ola." ".$mundo;
```

Operadores**Aritméticos:**

- + (sinal de adição Ex. x+y)
- - (sinal de subtração Ex. x-y)
- * (sinal de multiplicação Ex. x*y)
- / (sinal de divisão Ex. x/y)
- % (obtém o número de casas decimais da divisão entre 2 números Ex. 5%2 que dá 1 c.d.)

Definição:

- = (define que o primeiro número é igual ao segundo Ex. x=y)
- += (define que x é a soma de x com y Ex. x+=y é igual a x=x+y)
- -= (define que x é a subtração de x por y Ex. x-=y é igual a x=x-y)
- *= (define que x é igual à multiplicação de x com y Ex. x*=y é igual a x=x*y)
- /= (define que x é igual à divisão de x por y Ex. x/=y é igual a x=x/y)
- %= (define que x é o n.º de casas decimais da divisão de x por y Ex. x%=y é igual a x=x%y)

Comparação:

- == ("igual a" Ex. 7==9 dá falso)
- != ("não é igual a" Ex. 7!=9 dá verdadeiro)
- > ("maior que" Ex. 7>9 dá falso)
- < ("menor que" Ex. 7<9 dá verdadeiro)
- >= ("maior ou igual que" Ex. 7>=9 dá falso)
- <= ("menor ou igual que" Ex. 7<=9 dá verdadeiro)

Lógica:

- && ("e" Ex. consideremos x=9 e y=7 então " x < 10 && y > 3 " dá verdadeiro
- || ("ou" Ex. consideremos x=9 e y=7 então " x == 10 || y == 3 " dá falso
- ! ("não é" Ex. consideremos x=9 e y=7 então " !(x == y) dá verdadeiro

Um pequeno exemplo:

```
<?php
    $x = 7;
    $y = 7;
    $z = 2;

    if ($x == $y){ // Se x é igual a y
        echo "x = y = " . $x . "<br>";
    }

    $x += $z;
    echo "x = " . $x . "<br>";

    if ($x > $y){ // Se x maior que y
        echo "x é maior que y <br>";
    }
    else { // Senão
        echo "x não é maior que y";
    }
?>
```

Resultado:

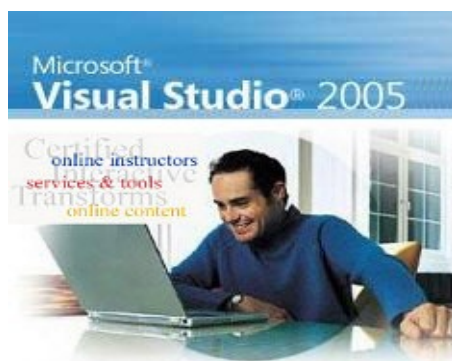
```
x = y = 7
x = 9
x é maior que y
```

A tag "
" é interpretada pelo HTML como mudança de linha. Para colocarem comentários no vosso PHP utiliza-se "//", isto faz com que o PHP ignore tudo o que está para além desse símbolo na linha. Para se fazer um comentário em várias linhas pode-se utilizar "/*" e no final do comentário "*/". É muito importante termos o nosso código devidamente comentado para que seja de fácil leitura para nós e para os outros. Quando se tem centenas de linhas de código pela frente se estiver comentado é fácil de perceber a sua funcionalidade.

Conclusão

E aqui fica a iniciação ao PHP. Esperamos que este artigo seja do vosso agrado e que compreendam que um único artigo nunca chegaria para incluir tudo, deste grande universo que é a linguagem PHP.

</revX & Ped@ntilva>



A importância das Coding Conventions

Nem todos os programadores concordam com as convenções de código, o facto é que estas são muito importantes, uma vez que permitem uniformizar o código, permitindo que vários departamentos programem da mesma maneira de forma a unir mais tarde todo um projecto. Imaginem o caos de programar um Windows ou Linux com milhões de linhas de código onde cada programador programava com as suas próprias regras.

Quando nunca se utilizou qualquer regra na definição de controlos e variáveis, é difícil começar a usar regras, uma vez que se perde por vezes mais tempo, mas o facto é que em grandes programas esse tempo é ganho mais tarde. Além quantos de vocês não voltaram a pegar num projecto e não perceberam nada do que haviam desenvolvido à algum tempo atrás.

As Melhores Práticas para VB

Option Strict On

Esta opção obriga a que todas as variáveis sejam declaradas o facto é que evitam erros desenvolvidos sem razão melhorando a qualidade do código.

Camel Casing

Camel Casing consiste em colocar em minúscula a primeira letra, num formato idêntico ao seguinte: listaUtilizadores, dadosRegisto...

No VB deve ser utilizado em variáveis locais, privadas(private) ou protegidas (protected) que não estão acessíveis externamente.

Pascal Casing

Pascal Casing consiste em colocar em maiúscula a primeira letra de cada palavra num formato idêntico ao seguinte: ListaUtilizadores, DadosRegistos... No VB deve ser utilizado sempre que não se aplique o Camel Casing e em objectos.

Comentários

Todos os procedimentos devem ser correctamente comentados, de forma a possuírem informação que permita definir a sua finalidade e funcionamento.

Deve-se também comentar as variáveis e blocos lógicos. O comentário deve reflectir o que faz o bloco de código e não repetir o código:

ERRADO

```
'Abrir um ficheiro de texto
Dim sr as new StreamReader("c:\texto.txt")
'Declarar um Inteiro
Dim i as Integer
'Repetir a leitura de cada linha
While sr.ReadLine()
    'Acrescentar 1
    i = i + 1
End While
```

CORRECTO

```
'Conta o número de linhas no ficheiro
Dim sr as new StreamReader("c:\texto.txt")
Dim i as Integer
While sr.ReadLine()
    i = i + 1
End While
```

A ter em Conta

Muitas pessoas acham que usar funções do *namespace Microsoft.VisualBasic* é incorrecto. Não podiam estar mais erradas, o facto é que estas funções fazem parte do próprio *framework*, e como tal não têm qualquer incompatibilidade com o *framework*. O que pode causar problemas e deve ser evitado ao máximo são as funções do namespace *Microsoft.VisualBasic.Compactility* que surge normalmente na conversão de projectos de VB6 para VB.NET. Este *namespace* é apenas temporário e tem de ser incluído nos projectos, causando muitos problemas.

Devem ser sempre usados nomes consistentes e reveladores do conteúdo das variáveis. Variáveis i,j,k podem ser usados em loop's triviais, mas se necessita de uma variável para guardar dados deve usar um nome consistente como por exemplo sNomeAluno.

Hungarian Notation

A Hungarian Notation foi criada com o objectivo de identificar o tipo de variável através do seu nome, o facto é que quando trabalhamos com muitos controlos ou em grupo, isto pode ser muito útil. De facto a Hungarian Notation é universal e é usada em diversas linguagens.

Para utilizar a Hungarian Notation basta adicionar um prefixo em minúsculas ao nome da variável. Cada Prefixo tem um significado. Ex.: frmMenu no VB é uma form chamada Menu.

Controlos Standart

Button	btn
CheckBox	chk
CheckedListBox	clst
ColorDialog	cdlg
ColumnHeader	chdr
ComboBox	cbo
ContextMenu	cmnu
CrystalReportViewer	crv
DataGrid	grd
DateTimePicker	dtp
DomainUpDown	dud
ErrorProvider	erp
FontDialog	fdlg
Form	frm
GroupBox	grp
HelpProvider	hlp
HScrollBar	hsb
ImageList	img
Label	lbl
LinkLabel	lnk
ListBox	lst
ListView	lwv
Menu	mnu
MonthCalendar	cal
NotifyIcon	nico
NumericUpDown	nud
OpenFileDialog	odlg
PageSetupDialog	psd
PictureBox	pic
PrintDialog	pdlg
PrintDocument	pdoc
PrintPreviewControl	ppc
PrintPreviewDialog	ppd
ProgressBar	pbr
RadioButton	rad
RadioButtonList	rbl
RichTextBox	rtf
SaveFileDialog	sdlg
Splitter	spl
StatusBar	sbr
StatusBarPanel	sbp
TabControl	tab
TabPage	pge
TextBox	txt
Timer	tmr
ToolBar	tbr

ToolBarButton	tbb
ToolTip	tip
TrackBar	trk
TreeNode	nod
TreeView	twv
VScrollBar	vsb

Data Objects

Connection	cnn
Command	cmd
CommandBuilder	cb
DataAdapter	da
DataColumn	dcl
DataReader	dr
DataRow	drw
DataSet	ds
DataTable	dt
DataRowView	dv

As coisas que sempre foram más, são más e sempre serão más!

O Goto, Option Explicit Off e On Error Resume Next nunca foram muito boas ideias. E ainda não são. Para controlo de erros deve-se utilizar a estrutura Try.

```
Try
    Dim x as Integer
    x = x / 0
Catch ex as Exception
    MsgBox(ex.Message)
Finally
    ' Este código corre sempre
End Try
```

Esta forma de controlo de erros evita que o programa tenha problemas e ainda permite determinar com precisão o tipo de erro, assim como correr algum código de correcção.

Deve-se ainda evitar o uso de caracteres fora do padrão A-Z, 0-9.

User Interface

As aplicações deverão ter sempre uma form de Sobre a aplicação e um menu de Ajuda para que o utilizador saiba onde pode encontrar suporte.

Cada utilizador deve utilizar estas Coding Conventions como entender e devem facilitar o seu trabalho, não complicá-lo.

</perdido_e_sozinho>



Liberte a sua criatividade

O desenvolvimento de jogos é um tópico que tem captado cada vez mais o interesse na área da informática. Este facto deve-se à crescente massificação dos videojogos na sociedade actual. Os jogos são agora uma das principais funções dos computadores, sendo por vezes, a própria razão porque estes são adquiridos.

No entanto, o desenvolvimento de jogos é muitas vezes tido como um passatempo de jovens, que não possui qualquer tipo de futuro, e que não requer muitos conhecimentos. Mas, pelo contrário, a criação de videojogos já deu origem a grandes empresas, sendo o mercado de videojogos, um negócio que movimenta demasiado dinheiro para ser deixado de parte. No entanto, são poucos os que conseguem entrar neste mundo, devido ao elevado grau de dificuldade, quer pelos conhecimentos necessários como pelo constante desenvolvimento das tecnologias. O desenvolvimento de jogos é uma área em expansão, mesmo no nosso país em que, embora pouco divulgados, existem bastantes projectos e várias empresas em funcionamento.

A principal e mais importante característica do desenvolvimento de jogos é que engloba diversas áreas como a programação, artes gráficas, som, e outras que variam de jogo para jogo. Todas estas áreas têm que ser bem coordenadas e associadas, para o jogo se tornar atractivo para o jogador. Um jogo pode ser muito bom graficamente, mas se possuir demasiados bugs, irá perder rapidamente o interesse do jogador, assim como um jogo com gráficos fracos ou um som irritante. Existem também um variado número de actividades externas ao desenvolvimento, mas que também são importantes e contribuem para o sucesso deste. Aqui englobam-se vários pormenores como a distribuição deste por parte de uma

editora, ou por conta própria, a campanha de marketing efectuada, as relações estabelecidas com a imprensa, entre outros factores. Neste artigo irei focar a área da programação, que foi a principal área que deu origem aos videojogos.

Para os programadores, existem dois caminhos principais para se iniciar no desenvolvimento de jogos, cada um com as suas vantagens e desvantagens. É possível dedicar-se ao estudo de uma ou várias linguagens de programação, ou dedicar-se à criação de MODs de jogos que disponibilizem ferramentas para este efeito. É também possível dedicar-se a ambos, para beneficiar da experiência dos dois, se tiverem disponibilidade para isso.

Os jogos que mais se destacam pela qualidade de ferramentas e pelo elevado número de MODs são o Half-Life(2), Doom3, Neverwinter Nights, e os jogos baseados no motor Unreal, como o Unreal Tournament, Deus Ex, Postal 2, entre outros jogos conhecidos. Existem também muitos outros jogos que suportam MODs, com o Max Payne 2, TES3: Morrowind, Command and Conquer. Para ter uma melhor ideia sobre o universo da criação de MODs, dêem uma vista de olhos a <http://moddb.com>. As ferramentas normalmente disponibilizadas são um editor de níveis e outros aspectos do jogo, ou até uma linguagem de scripting, como o UnrealScript?, ou o próprio código-fonte do jogo. As principais vantagens do desenvolvimento de MODs são os conhecimentos adquiridos sobre o funcionamento de um jogo comercial e reconhecido, como trabalham as partes principais do jogo, como os seus ficheiros são distribuídos, etc. Também permite a criação, de forma relativamente mais fácil, de um produto mais apelativo aos jogadores, embora sempre semelhantes ao jogo principal. Essa é mesmo a principal desvantagem, o facto de estarmos limitados às ferramentas disponibilizadas, e também por serem raros os casos MODs comerciais (Counter-Strike, Day Of Defeat, ...).

As vantagens de aprender uma linguagem de programação são várias, mas a principal razão é o facto de obtermos todo o controlo do nosso jogo e podermos criá-lo da forma que entendemos. Mas para isso é preciso dominar uma linguagem de programação, o que pode levar algum tempo. Em relação à criação de MODs, o período e a dificuldade de aprendizagem de uma linguagem de programação são mais elevados. Isto pode levar alguns iniciantes a desistirem, mas a recompensa de dominar uma linguagem de programação é grande. Outra das vantagens de saber trabalhar com uma determinada linguagem, é que esta também pode ser aplicada em outros trabalhos, diferentes de jogos, como software empresarial, entre outros.

O primeiro dilema que os futuros programadores se deparam, é com a decisão de qual linguagem aprender. Uma das opções mais recomendada é C++. É uma linguagem que é vastamente utilizada na indústria de videojogos. Mais de 90% dos jogos que costumamos ver nas prateleiras das lojas, são programados nesta linguagem. É também uma das que possui maior documentação e referências, assim como ferramentas. As suas características, como a velocidade e controlo, também a favorecem em relação às restantes linguagens. Esta linguagem peca apenas pelas dificuldades que introduz a alguém novo à programação. Mas também são usadas outras linguagens, com características bastante variadas.

Java é uma linguagem que se destaca pela sua portabilidade. Os programas feitos em Java podem ser usados em variadas plataformas, como o computador, telemóveis, PDAs, e mesmo websites, apenas com reduzidas modificações. Esta portabilidade leva a algum custo no desempenho, mas não deixa de ser um opção a ter em conta. Temos também Visual BASIC, que se destaca pela sua facilidade de aprendizagem e utilização, embora seja mais utilizada na criação de outro tipo de software. Existem também vários jogos feitos em Delphi, uma linguagem derivada de Pascal, alguns deles comerciais. Os jogos desenvolvidos sobre a plataforma Flash também têm ganho bastante popularidade, graças à sua integração nos sites, que permitem um rápido acesso, e simplicidade. Estes jogos são predominantemente baseados no aspecto gráfico, mas necessitam de uma base de programação, que é feita em ActionScript?

Existem variadas fontes de sobre desenvolvimento de jogos. A mais importante e fiável são os livros. A cada novo tema ou linguagem de programação, que se decide aprender e dominar, convém ter um livro, tanto para auxiliar na aprendizagem, como para ser utilizado como referência. No entanto, os livros de programação, que se encontram à venda em Portugal, possuem poucas referências ao desenvolvimento de jogos. Por isso, é necessário, muitas vezes, mandar vir os livros através de uma loja online, como a Amazon (<http://www.amazon.com>), o que encarece um bocado o preço, mas apresenta uma maior variedade de escolha.

A internet é o local onde é possível encontrar praticamente todo o tipo de informação, sobre qualquer assunto. É possível encontrar um grande número de artigos e tutoriais sobre qualquer linguagem e qualquer aspecto do desenvolvimento de jogos. Mas, essa variedade leva às vezes a alguma desinformação em alguns artigos menos trabalhados e

revistos. Mas não deixam de ser uma das principais fontes, a que irão recorrer muitas vezes. Outro factor importante são as comunidades que se formam à volta deste tema. Estas comunidades são formadas através de um site, fórum, canal de IRC, entre outros. As comunidades são importantes para a partilha de artigos, mas especialmente para tirar dúvidas, partilhar experiências e pedir conselhos a pessoas mais experientes, tendo em consideração as regras estabelecidas. Finalmente, é possível recorrer também ao código-fonte de jogos, que sejam lançados sobre a licença "open-source". Estes já requerem alguns conhecimentos sobre a linguagem em questão, mas são uma fonte de informação muito valiosa, já que apresentam a resolução de problemas com que o programador se deparou e os métodos que devem ser usados em cada situação.

Para conseguir desenvolver o jogo dos nossos sonhos, é necessário muito esforço e dedicação, para além de paciência e persistência para todas as dificuldades que iram surgir. O meu conselho, e o da maioria dos programadores, é começar por jogos pequenos e simples, como por exemplo Tetris, Snake, Pacman, um Shooter 2D, cada um introduzindo uma nova funcionalidade característica, que vai aumentando a dificuldade. Esta é a melhor maneira para um principiante não se perder entre as inúmeras capacidades de programação, que o desenvolvimento de jogos exige.

Grande parte dos que desenvolvem jogos são amadores, já que só o fazem nos tempos livres, por serem estudantes ou terem um emprego noutra área. Não é fácil, mas é o objectivo de praticamente todos, entrar na indústria de videojogos, quer trabalhando para uma produtora, ou criando a sua própria produtora. Este é um caminho longe de ser fácil, mas muito recompensador.

Alguns links:

Comunidades internacionais:

<http://www.gamedev.net>

<http://www.devmaster.net>

Comunidades portuguesas:

<http://www.gamedev-pt.net>

<http://www.igda.org/lisbon>

<neon_prannock>



Nesta edição aproveitamos para falar um pouco sobre a arquitetura que têm dado muito que falar no seio da comunidade opensource, e que se têm vindo a tornar um *must have* de qualquer *desktop* que se preze. O que é afinal o Xgl? Para que serve? Porquê incluí-lo no *desktop environment*? São tudo perguntas que vamos responder neste artigo.

O seu anúncio surgiu no dia 1 de Janeiro de 2005, por David Reveman, e toda a comunidade GNU/Linux ficou de boca aberta com o que poderão ser os próximos *desktop environments*.

Xgl é uma arquitectura do Xserver que aproveita todas as potencialidades do OpenGL, de modo a acelerar todo o *desktop environment* da distribuição em questão. Aliado a um *compositing windows manager* como por exemplo o Compiz, o Xgl fornece ainda uma panóplia de efeitos visuais atraentes ao *desktop environment* que têm sido, no fundo, a razão de todo o alarido que tem surgido.

No entanto, como tecnologia recente, tem tido alguns problemas no que toca a compatibilidade com os modelos de placas gráficas, só suportando as placas mais recentes. As placas da *Nvidia* são as mais aconselháveis para correr esta tecnologia.

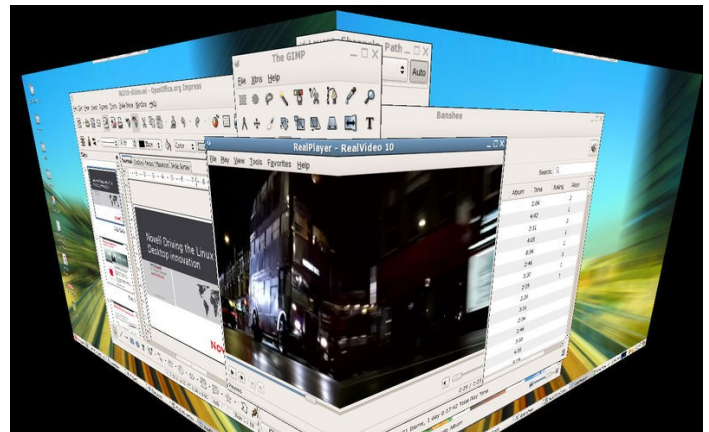
Os efeitos em si não são propriamente novos: um utilizador de *MacOSX* sentirá, muito provavelmente, um pouco de *deja vu*, ao ver os efeitos que estão a chegar ao mundo opensource. As transparências serão uma constante, mas não nos ficamos por aqui: o famoso efeito do cubo, que nos permite de uma maneira 3d "saltar" de área de trabalho em área de trabalho

também foi incorporado, bem como efeitos de janelas gelatinosas ao serem arrastadas, suavização de menus, e muito mais.

E ao contrário do que possamos pensar, esta tecnologia é modesta o suficiente a níveis de recursos, para correr nos pc's menos favorecidos, visto que tudo se baseia no GPU da placa gráfica, e não no CPU do computador.

Resta apenas, através da nossa própria utilização, constatar se tanto efeito não prejudica a funcionalidade do *desktop environment*. Da nossa parte, que já testámos o Xgl, podemos dizer que não atrapalha em nada, e só ganhamos em rapidez com esta tecnologia.

</vbmaster>



Aqui estão alguns links úteis a qualquer programador:

→ **sqlZoo** - <http://sqlzoo.net>

Se estás a aprender SQL e queres um local onde possas praticar convenientemente as tuas consultas, então este site é ideal para ti. Praticas as tuas consultas á base de dados de forma simples e ficas a saber como utilizar consultas em SQL.

→ **Hackaholic.org** - <http://hackaholic.org>

Um site cheio de explicações ligadas ao hacking na pura concepção da palavra. Muito interessante para programadores com alguma experiência em Python, C e C++. Algo a não perder.

→ **FreeProgrammingResources** - <http://www.freeprogrammingresources.com>

Um site com centenas de recursos uteis a programadores. Desde links para e-books até links para download, este site tem uma variedade de ligações a páginas de conteúdo relacionado com a programação. Tem conteúdo relativo a muitas linguagens de programação, como por exemplo Assembly, Ada, Cobol, C,C++,Pascal, Perl,PHP,Lua,Ruby-on-Rails, entre muitas outras.

→ **Linoleum: Linux Programming Resources** - <http://linoleum.leapster.org>

Mais um site cheio de recursos de programação. Desta vez são recursos de programação em GNU/Linux. Algo a não perder.

→ **Python.org** - www.python.org

O site oficial da linguagem de programação Python. Um local onde os programadores podem ler toda a documentação, fazer download de um interpretador e participar nas actividades e discussões da comunidade.

→ **HotScripts** - <http://www.hotscripts.com>

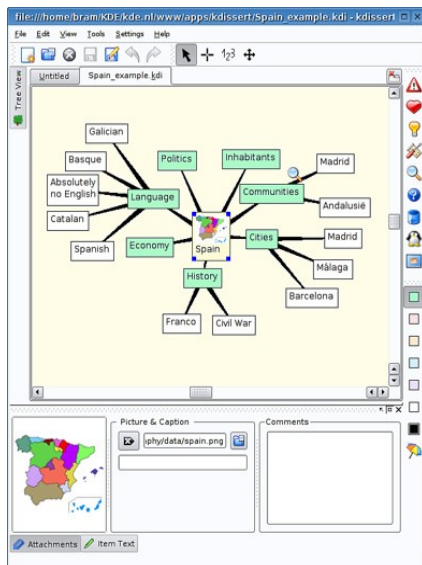
Neste site podem fazer download de milhares de scripts de diversas linguagens. Estão presentes mais de 12 mil scripts de PHP, mais de 3 mil scripts de ASP, mais de 1500 scripts de Java e podem encontrar scripts de linguagens como Python, XML, CGI/Perl, JavaScript, C,C++, entre outras. Um site obrigatório para todos os programadores.

→ **PEAR PHP** - <http://pear.php.net>

Este é o maior repositório mundial de extensões e aplicações PHP. Podem encontrar pacotes de encriptação, bases de dados, redes, estruturas, XML, sistemas de ficheiros, entre muitos outros. O site possui uma interface de fácil navegação, o que facilita em muito o trabalho do visitante.

Kdissert

Para quem precisa de organizar as ideias.



Enquanto um jornalista escreve um artigo, surgem inúmeras ideias e pensamentos na sua cabeça. A tarefa mais complicada é ligar essas ideias de forma a que os leitores compreendam qual o verdadeiro significado das frases construídas. Para ajudar os jornalistas, os bloggers e os escritores a organizar essas ideias, surge o kdissert.

O kdissert, projecto liderado por Thomas Nagy é uma aplicação de mind mapping que tem como objectivo ajudar quem escreve a organizar as ideias. Poderá ser útil na escrita de teses, dissertações, apresentações e até de livros. O programa permite-nos criar um mapa de ideias, que ajuda a estruturar os pensamentos que vamos tendo enquanto escrevemos ou, se o escritor preferir, antes de escrevermos. Com a criação destas estruturas, as probabilidades de escrevermos um texto mais ordenado e com mais qualidade são maiores.

Com o kdissert, organizamos os pensamentos numa página dinâmica, criamos links entre ideias, ordenamos os conceitos, organizamos, reorganizamos,

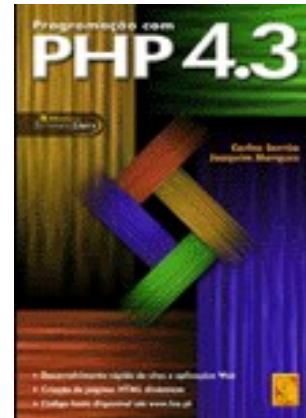
adicionamos links ou imagens, etc. Os documentos resultantes podem ser ficheiros PDF, imagens, documentos LaTeX, documentos HTML para páginas web, ou ficheiros OpenOffice.org Impress. A palavra kdissert em kdissert faz-nos pensar que a aplicação tem como público-alvo estudantes universitários, no entanto este pacote é realmente fantástico para quem quer elaborar apresentações, por exemplo. Assim sendo, o público-alvo são todos aqueles que pretendem organizar as suas ideias de modo a construírem um texto estruturado e de qualidade elevada. A última versão da aplicação está disponível no site oficial do kdissert (<http://freehackers.org/%7Etnagy/kdissert>).

O programa tem uma interface muito amigável ao utilizador, logo é de fácil aprendizagem. Recomendamos este programa a todos os bloggers que pretendem escrever artigos de qualidade, assim como aos estudantes universitários que têm teses para escrever.

</deathseeker25>

Programação com PHP 4.3

Como criar portais dinâmicos.



Programação com PHP 4.3 de Carlos Serrão e Joaquim Marques é um livro que aborda temas como a instalação do PHP 4.3 em diversas plataformas, explicações sobre conceitos PHP 4.3 e respectiva sintaxe, explicações sobre gestão de cookies e sessões, sistemas de comércio electrónico, segurança, desenvolvimento de aplicações com XML, truques e técnicas na utilização do PHP 4.3 e de algumas ferramentas associadas.

Escrito por Carlos Serrão e Joaquim Marques, ambos com mestrados na área da Gestão de Sistemas de Informação pelo ISCTE e com cargos na ADETTI em acções de Investigação e Desenvolvimento (I&D), este é um livro que tem como público alvo os jovens programadores que queiram aprender a programar em PHP, concentrando toda a acção á volta do desenvolvimento de um site dinâmico. O livro baseia-se essencialmente na demonstração de acções através de scripts acompanhados por uma explicação dos mesmos. Assim sendo, temos acesso a uma panóplia de scripts que o leitor pode adaptar consoante as suas necessidades.

O livro aborda temas importantes e comuns na programação de um site dinâmico como instruções SQL, armazenamento em bases de dados, armazenamento de dados e ficheiros XML, utilização de sessões ou de cookies, definição de permissões, ficheiros, entre outros. É, portanto, um livro muito completo, com um enorme número de scripts disponíveis, do qual o utilizador pode tirar todas as informações necessárias para desenvolver as suas primeiras aplicações em PHP.

É um livro recomendado a todos os novos programadores que se queiram iniciar em PHP e a estudantes universitários que abordem a linguagem, já que este tem uma vertente prática e oferece uma explicação sucinta dos conteúdos fulcrais da iniciação na mesma. O Portugal-a-Programar recomenda a leitura deste livro.

</deathseeker25>

<Dúvidas>

Q:

Olá a todos, estou aqui com um problema, pois coloquei a função `clrscr()` num programa, coloquei o `#include <conio.h>` no início mas, mesmo assim, dá-me o erro:

```
[Linker error] undefined reference to 'clrscr'
```

O que se passará?

Nota: estou a programar com o Dev-C++

</JoBlack>

R:

Se não me engano acho que tens que utilizar a biblioteca:

```
#include <stdio.h>
```

</vbtipo>

Q:

Como é que ponho uma pagina a descobrir o ip da pessoa que liga??

tipo o `www.whatismyip.org`

</joninho>

R:

O ip do utilizador está na variável de ambiente `$REMOTE_ADDR`

Se quiseres antes o `host`, podes usar o seguinte código:

```
<?
    $host = GetHostByName($REMOTE_ADDR);
?>
```

</pedrotuga>

Q:

Boas!

Pessoal, é o seguinte, tenho uma *checkbox* e queria inserir o resultado do *checked* numa base de dados. Do tipo, tenho um *form* com um *checkbox* e um botão, imaginemos que o ponho o visto na *checkbox* e carregava no botão, quando carregasse no botão, queria conseguir guardar numa variável se o *checked* está *true* ou *false*:

```
dim variavel as ?
if checkbox1.checked=true then
    variavel =?
end if
```

Nota: O programa ta a ser feito em Microsoft Visual Studio .NET 2003.

Espero respostas...

</ProT>

R:

Poderás fazer da seguinte maneira por exemplo:

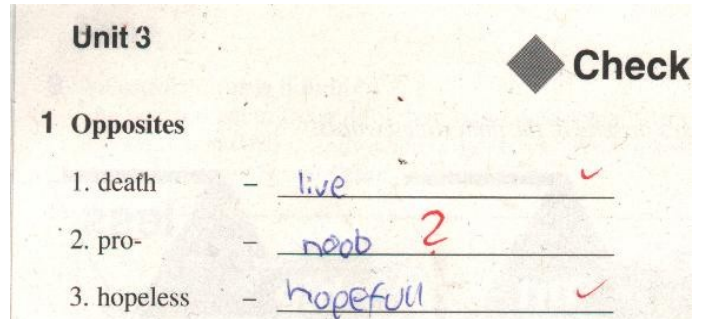
```
Dim variavel as Boolean
If checkbox1.Checked = True Then
    variavel = True
Else
    variavel = False
End If
```

Depois é só guardares o estado da variável na Base de Dados.

</Tiago Salgado>



À procura do seu carro...



Aluno incompreendido



Oficina de reprodução – parte 2



Comprar gato por lebre

A Comunidade P@P

Como todo o leitor que frequenta o fórum tem conhecimento, a nossa comunidade tem crescido significativamente desde o início do novo ano, principalmente nos meses de Março e Abril. Temos cada vez mais utilizadores registados, temos cada vez mais utilizadores activos, mais pessoas a querer contribuir nos projectos da comunidade, mais *posts* no fórum, mais visitas diárias, etc. Somos um fórum que tem feito sucesso a nível nacional, mas temos consciência de que podemos fazer muito mais e melhor no que toca a programação em Portugal.



Têm havido remodelações no que toca ao *staff* e à organização do fórum, projectos de organização interna que ainda se encontram em estudo, secções privadas para utilizadores com um determinado estatuto no fórum, projecto de correcção dos posts com erros (resultantes da passagem de phpBB para SMF), revisão de objectivos, planeamento de concursos, entre outros. Estamos até a planear abrir uma secção sobre segurança, mas esta terá de ser muito bem pensada e terá regras muito rígidas.

O Portugal-a-Programar completa um ano de idade no dia 28 de Maio de 2006. Assim sendo, o leitor está convidado a participar na "festa de aniversário". Para tal, basta registar-se no fórum e começar a contribuir para o crescimento da comunidade. Serão apresentados relatórios de estatística anuais aos utilizadores, bem como um relatório de objectivos a cumprir no ano seguinte.

Parcerias actuais do P@P



TechzonePT – www.techzonept.com



RedCodeStudio-<http://forum.redcodestudio.com>



PCTuga – www.pctuga.net



Foruns.org – www.foruns.org



PTServers – www.ptservers.org