



- **AnsiLowerCase** → Converte todos os caracteres de uma *string* para minúsculo;

Sintaxe:

AnsiLowerCase(const S: string): string;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
Var
  X:String;
Begin
  X:=Label1.caption;
  Label1.caption:=AnsiLowerCase(X); { Label1 ficará tudo em letras minúsculas }
End;
```

- **AnsiUpperCase** → Converte todos os caracteres de uma *string* para maiúsculo;

Sintaxe:

AnsiUpperCase(const S: string): string;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
Var
  X:String;
Begin
  X:=Label1.caption;
  Label1.caption:=AnsiUpperCase(X); { Label1 ficará tudo em letras maiúsculas }
End;
```

- **Break** → Interrompe totalmente o loop das condições *for*, *while*, ou *repeat*;

Sintaxe:

Break;

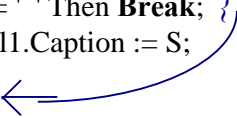
Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```

var
  S: String;
begin
  While true Do
  begin
    S := Edit1.text;
    If S = ' ' Then Break; { Interrompe o loop }
    Label1.Caption := S;
  end;
end;

```



- **CloseFile** → Fecha um arquivo aberto;

Sintaxe:

```
CloseFile(var F);
```

Exemplo:

```

Procedure TForm1.Button1Click(Sender: TObject);
var
  F: TextFile;
begin
  if OpenFileDialog1.Execute then
  begin
    AssignFile(F, OpenFileDialog1.FileName);
    Reset(F);
    Edit1.Text := Str(FileSize(F));
    CloseFile(F); { Fecha o arquivo }
  end;
end;

```

- **Continue** → Continua na próxima condição *for*, *while*, ou *repeat*;

Sintaxe:


```
Continue;
```

Exemplo:

```

Procedure TForm1.Button1Click(Sender: TObject);
var
  S: String;
begin
  While true Do
  begin
    S := Edit1.text;
    If S = ' ' Then Continue; { Voltará para o próximo loop }
    Label1.Caption := S;
  end;
end;

```



- **Eof** → Retorna um valor lógico se fim de arquivo;

Sintaxe:

Eof(var F): Boolean;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
var
  F1, F2: TextFile;
  Ch: Char;
begin
  if OpenFileDialog1.Execute then begin
    AssignFile(F1, OpenFileDialog1.FileName);
    Reset(F1);
  if SaveDialog1.Execute then begin
    AssignFile(F2, OpenFileDialog1.FileName);
    Rewrite(F2);
    While not Eof(F1) Do { Executa enquanto não for fim do arquivo F1 }
    begin
      Read(F1, Ch);
      Write(F2, Ch);
    end;
    CloseFile(F2);
  end;
  CloseFile(F1);
end;
end;
```

- **Exit** → Sai imediatamente de um bloco corrente;

Sintaxe:

Exit;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
begin
  repeat
    if KeyPressed then Exit; { Se alguma tecla pressionada sairá do bloco repeat }
    Memo1.Lines := 'Xx';
  until False;
end;
```

- **FileOpen** → Abre um arquivo específico;

Sintaxe:

FileOpen(const FileName: string; Mode: Word): Integer;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
MyFileHandle := FileOpen(EXISTS.TXT); { Abre o arquivo EXISTS.TXT }  
end;
```

- **GotoXY** → Move o cursor para as coordenadas dadas dentro da tela;

Sintaxe:

```
GotoXY(X, Y: Byte);
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
begin  
  GotoXY(10,25); { Posiciona o ponteiro do mouse no pixel 10 , 25 }  
end;
```

- **Int** → Retorna a parte inteira de um argumento;

Sintaxe:

```
Int(X: Real): Real;
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  R: Real;  
begin  
  R := Int(123.456); { Mostra "123.0" }  
  R := Int(-123.456); { Mostra "-123.0" }  
end;
```

- **KeyPressed** → Determina se uma tecla foi pressionada no teclado;

Sintaxe:

```
KeyPressed: Boolean;
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  A,B:integer;  
begin  
  B:=1;  
  repeat  
    A:=A+B;  
    Label1.Caption := str(A);  
  until KeyPressed; { Repetirá até alguma tecla ser pressionada }  
end;
```

- **Now** → Retorna a data e hora corrente;

Sintaxe: *Formato da data e hora }*

Now: TDateTime;

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
  Label1.Caption := 'A data e hora é ' + DateTimeToStr(Now); { Mostra a hora e data }
```

```
end;
```

- **Sqr** → Retorna um número ao quadrado;

Sintaxe:

Sqr(X: Real): (Real);

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  S, Temp: String;
```

```
begin
```

```
  Str(Sqr(5.0):2:1, Temp); { Retorna 25.0 }
```

```
  S := '5 ao quadrado é ' + Temp;
```

```
  MessageDlg(S, mtInformation, [mbOk], 0);
```

```
end;
```

- **Str** → Converte um valor numérico para uma *string*;

Sintaxe:

Str(X [: Width [: Decimals ]]); var S);

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  X, Temp: String;
```

```
  A: Integer;
```

```
begin
```

```
  A:=A+1
```

```
  Str(A:2:1, Temp); { Transforma A em string de tamanho dois e uma casa decimal }
```

```
  X := 'O número é ' + Temp;
```

```
  MessageDlg(X, mtInformação, [mbOk], 0);
```

```
end;
```

- **Val** → Converte um valor *string* para sua representação numérica;

Sintaxe:

```
Val(S; var V; var Code: Integer);
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  I, Code: Integer;  
begin  
  Val(Edit1.Text, I, Code); { Converte o conteúdo de Edit1 para um valor }  
end;
```

- If ... Then ... Else → If, then, e else especifica a condicional sobre a qual uma declaração será executado.

Sintaxe:

```
If <Expressão1> then <expressão2> else <expressão3>;
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  idade:integer;  
begin  
  idade := Edit1.text;  
  If (idade < 1) or (idade > 120) then  
    Label2.caption := ‘Você não está dizendo a verdade.’ {Executa se idade<1 ou idade>120}  
  Else  
    Label2.caption := ‘Ok!’; { Executa se idade não for < 1 ou > 120 }  
end;
```

- Case ... of → O procedimento *Case* é usado para ramificar um código dependendo do resultado ou valor encontrado neste código.

Sintaxe:

```
Case <expressão> of  
  <valores>  
else  
  <valor>
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  tecla:char;  
begin  
  tecla := Edit1.text;  
  Case tecla of  
    ‘A’..‘Z’, ‘a’..‘z’:Label2.caption:=‘Você digitou uma letra’;{Executa se digitado uma letra}  
    ‘0’..‘9’:Label2.caption:=‘Você digitou um número’ { Executa se digitado um número }  
  else
```

```
Label2.caption:='Você digitou um caracter especial'; { Executa se digitado outra tecla }  
end;
```

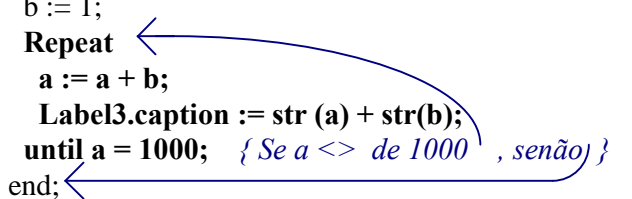
- **Repeat ... until** → Os procedimentos entre *Repeat* e *until* é executado na seqüência enquanto uma condição seja satisfeita.

Sintaxe:

```
Repeat  
  <procedimentos>  
until <expressão>;
```

Exemplo:

```
Procedure TForm1.Button1.Click(Sender: TObject);  
var  
  a, b : integer;  
begin  
  b := 1;  
  Repeat  
    a := a + b;  
    Label3.caption := str (a) + str(b);  
  until a = 1000; { Se a <> de 1000 , senão }  
end;
```



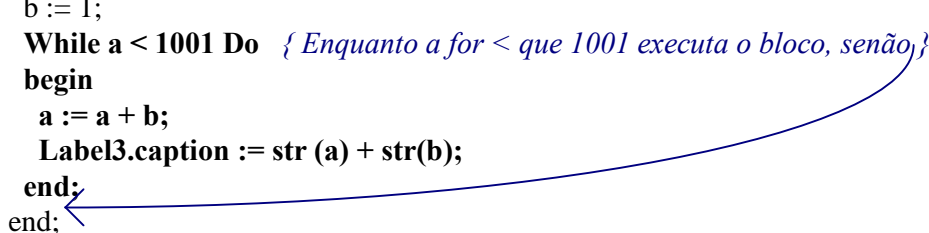
- **While .. Do** → A declaração *While* controla a execução repetida de uma declaração simples ou composta.

Sintaxe:

```
While <expressão> do <procedimento>
```

Exemplo:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
  a, b : integer;  
begin  
  b := 1;  
  While a < 1001 Do { Enquanto a for < que 1001 executa o bloco, senão }  
  begin  
    a := a + b;  
    Label3.caption := str (a) + str(b);  
  end;  
end;
```



- **For .. Do** → O loop *For .. Do* é uma declaração que será executada até que a variável de controle seja satisfeita.

Sintaxe:

```
For <variável de controle> := <valor inicial> To ou Downto <valor final> Do <procedimentos>
```

Exemplo:

```

Procedure TForm1.Button1Click(Sender: TObject);
var
  cont, x : integer;
begin
  For cont := 1 To 100 Do { Repete o bloco 100 vezes }
  begin
    x := x + cont;
    Label4.caption := str(x);
  end;
end;

```

- **Goto / Label:** → A instrução *Goto* permite que você salte de onde está no momento para um linha rotulada específica. Um rótulo é definido usando-se a palavra *label*.

Sintaxe:

```

Goto <rótulo>;
Label <rótulo>;

```

Exemplo:

```

Procedure TForm1.Button1Click(Sender: TObject);
var
  resp : char;
  a, b : integer
begin
  b := 1;
  Label inicio: { A linha é rotulada como inicio }
  a := a + b;
  Label2.caption := str(a);
  Label3.caption := 'Deseja somar mais um';
  Edit1.visible := True;
  resp := Edit1.text;
  If (resp = 'S' or 's' or 'Y' or 'y') then Goto inicio; { Volta para o rótulo }
end;

```

- **RunError** → Este comando emite um número de erro que você introduz como motivo de falha de programa.

Sintaxe:

```

RunError [ ( Errorcode: Byte ) ];

```

Exemplo:

```

Procedure TForm1.Form1Load(Sender: TObject);
var
  I : Char;
  Cont : Integer;
begin
  repeat
    I := Edit1.text;
    If I := 'Q' then RunError (230); { Emite ao usuário o erro 230 }

```



```
until FALSE;  
end;
```

- [ShowWindow](#) → A função *ShowWindow* modifica o estado de visibilidade da janela.

Sintaxe:

```
ShowWindow(Wnd: HWND; CmdShow: Integer): Bool;
```

Parâmetro    Descrição

Wnd            Identifica a janela.

CmdShow        Especifica como a janela será mostrada. Este parâmetro pode ser um dos seguintes valores:

Valor

Significado

SW\_HIDE        Esconde a janela atual e ativa uma outra janela.

SW\_MINIMIZE    Minimiza a janela específica.

SW\_RESTORE    Ativa e mostra uma janela. Se a janela está minimizada ou maximizada, o Windows restaura ela para o tamanho e posição original.

SW\_SHOW        Ativa uma janela e mostra ela na posição e tamanho corrente.

SW\_SHOWMAXIMIZED Ativa uma janela e mostra ela como uma janela maximizada.

SW\_SHOWMINIMIZED Ativa uma janela e mostra ela como um ícone.

SW\_SHOWMINNOACTIVE    Mostra uma janela como um ícone. A janela que está atualmente ativada permanece ativa.

SW\_SHOWNORMAL    Ativa e mostra uma janela. Se a janela estiver minimizada ou maximizada o Windows restaura ela para a posição e tamanho original.

Exemplo:

```
Procedure TForm1.Form1Load(Sender: TObject);  
ShowWindows (Tform2 : HWND ; SW_SHOW : integer)
```