

Rotinas / Funções

Delphi

- Ceil
- Concat
- Copy
- CurrToStrF
- Date
- DayOfWeek
- EncodeDate
- FormatDateTime
- Lenght
- Max / Min
- Now
- SQRT
- Time
- Trim
- Trunc

Ceil

Arredonda um determinado número para o inteiro que seja maior ou igual a ele. No exemplo ao lado, os valores 12.0 e 12.34 serão respectivamente arredondados para 12 e 13.

Exemplo:

```
var
  numero: Double;
begin
  numero := Ceil(12.0);
  ShowMessage(' Ceil(12.0) = {0}', numero.ToString);
  numero := Ceil(12.34);
  ShowMessage(' Ceil(12.34) = {0}', numero.ToString);
end;
```

Concat

Essa função é utilizada para fazer a concatenação(junção) de várias Strings em uma só. No exemplo, o texto exibido será "Olá, mundo!". Pois o mesmo está sendo concatenado pela função Concat.

Exemplo:

```
var
  result : string;
begin
  result := Concat('Olá', ' mundo', '!');
  ShowMessage(result);
end;
```

Copy

Copia parte do texto de uma variável para outra. Recebe 3 parâmetros, sendo eles, em ordem:

- Fonte de onde os dados serão copiados;
- Posição do Caracter de onde iniciará a cópia;
- Contador de quantos caracteres serão copiados;

No exemplo, é copiado o conteúdo da variável Source, iniciando pelo terceiro carácter (3).

Após contar 4 caracteres, irá terminar a cópia do conteúdo, no número 6.

Mostrará o texto: 3456

```
var
  Source, Target : string;
begin
  Source := '12345678';
  Target := Copy(Source, 3, 4);
  ShowMessage('Target : ' + Target);
end;
```

CurrToStrF

Converte um valor Currency para uma String seguindo determinada formatação.

No exemplo, indicamos que a variável valor receberá o número 1234.567 e nas mensagens, mostraremos esse número com 4 dígitos após a vírgula, e logo em seguida com 2 dígitos após a vírgula.

Exemplo:

```
var
  valor: Currency;
begin
  valor := 1234.567;
  ShowMessage(' Usando 4 dígitos = ' + CurrToStrF(valor, ffCurrency, 4));
  ShowMessage(' Usando 2 dígitos = ' + CurrToStrF(valor, ffCurrency, 2));
end;
```

O resultado será, respectivamente: 1234.5670; 1234.56

Date

Retorna a data atual do sistema. No exemplo, a descrição do componente Label1 assumirá a data atual do sistema assim que o Button1 receber um clique.

Exemplo:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  Label1.Caption := 'Hoje é: ' + DateToStr(Date);  
end;
```

DayOfWeek

Retorna o dia da semana correspondente a uma data.

No exemplo, se o dia da semana extraído da informação no Edit1 corresponder a 1, então mostrará a mensagem 'Dia da semana: Domingo', pois Domingo é o primeiro dia da semana.

Exemplo:

```
if (DayOfWeek(StrToDate(edit1.text))) = 1 then  
  Showmessage('Dia da semana: Domingo');
```

EncodeDate

Junta valores de ano, mês e dia formando uma data. No exemplo, estamos informando um valor às variáveis dia, mês e ano, e, a partir da função EncodeDate() iremos concatenar esses valores no formato de data, que será armazenada na variável data.

Exemplo:

```
procedure TForm2.Button1Click(Sender: TObject);
var
  dia, mes, ano: Word; data: TDateTime;
begin
  ano := 2015;
  mes := 2;
  dia := 15;
  data := EncodeDate(ano, mes, dia);
  lblData.Caption := 'Data: ' + DateToStr(data);
end;
```

FormatDateTime

Formata data e hora usando um formato especificado. Ao lado, estamos definindo que o formato de data que queremos que seja exibido seja em Data (dois dígitos) / Mês (dois dígitos) / Ano (dois dígitos).

Nesse exemplo, a mensagem que seria exibida ficaria parecida com a seguinte:

'dd/mm/yy = 06/11/17'.

Exemplo:

```
procedure TForm2.Button1Click(Sender: TObject);
var
  dataAtual: TDateTime;
begin
  ShowMessage(' dd/mm/yy = ' + FormatDateTime(' dd/mm/yy' , dataAtual));
end;
```

Length

Retorna tanto a quantidade de caracteres de uma String, quanto o número de elementos em um array. É muito utilizado para estruturas de loop, percorrendo um conjunto enquanto houver dados no mesmo.

Desse modo, o primeiro texto a ser mostrado será: "Tamanho da shortStr = 8", pois essa é a quantidade de caracteres encontradas na variável.

Agora, o segundo texto será exibido dividido em linhas, da seguinte maneira:

"Letra 1 = A

Letra 2 = B

Letra 3 = C (...)" até chegar no último registro encontrado, no caso, H.

Exemplo:

```
var
    shortStr : String;
    i : Integer;
begin
    shortStr := 'ABCDEFGH';
    ShowMessage(' Tamanho da shortStr = ' + IntToStr(Length(shortStr)));

    //Utilização em loops
    for i := 1 to Length(shortStr) do
        ShowMessage(' Letra: ' + IntToStr(i) + ' = ' + shortStr[i]);
    end;
```

Max / Min

Retorna o maior/menor valor entre dois números.

Exemplo:

```
begin
  Label1.Caption := Max(1, 2);
  Label2.Caption := Min(1, 2);
end;
```

Now

Retorna a data e hora correntes.

Exemplo:

```
begin  
  ShowMessage('Data e hora atuais: ', DateTimeToStr(Now));  
end;
```

SQRT

Encontra a raiz quadrada de determinado número. No exemplo utilizado, estamos falando que a variável `raizQuadrada` recebe a raiz quadrada de 225 (`numero`), e, após isso, mostramos o seu valor em uma mensagem.

Exemplo:

```
var
  numero, raizQuadrada: Extended;
begin
  numero := 225;
  raizQuadrada := Sqrt(numero);
  ShowMessage('Raiz quadrada de ', numero, ' é: ', raizQuadrada);
end;
```

O resultado será: 15

Time

Retorna a hora corrente.

Exemplo:

```
begin  
  ShowMessage(' Hora atual: ', TimeToStr(Time));  
end;
```

Trim

A função Trim é utilizada para que sejam removidos espaços em branco de determinados textos. Existem duas variações da função Trim, sendo elas:

- **TrimLeft**, que remove os espaços à esquerda; e
- **TrimRight**, que remove os espaços à direita.

No exemplo, os espaços removidos seriam, consecutivamente:

[Teste]

[Teste]

[Teste]

Exemplo:

```
const
  StringTeste = '  Teste  ';
begin
  ShowMessage([' + TrimLeft(StringTeste) + ']);
  ShowMessage([' + TrimRight(StringTeste) + ']);
  ShowMessage([' + Trim(StringTeste) + ']);
end;
```

Trunc

Trunca um valor do tipo Real para um do tipo Inteiro, ignorando completamente as casas decimais.

Exemplo:

```
begin
  ShowMessage(' Trunc(12.75) = ' +IntToStr(trunc(12.75)));
end;
```

O resultado será: 12.