

PROFESSOR MARCOS NIELSEN

TIPOS DE DADOS ACEITOS PELO DELPHI

<http://www.delphibasics.co.uk/ByFunction.asp?Main=Types>

Tipo	Nome	Descrição
Type	AnsiChar	A character type guaranteed to be 8 bits in size
Type	AnsiString	A data type that holds a string of AnsiChars
Keyword	Array	A data type holding indexable collections of data
Type	Boolean	Allows just True and False values
Type	Byte	An integer type supporting values 0 to 255
Type	Cardinal	The basic unsigned integer type
Type	Char	Variable type holding a single character
Keyword	Class	Starts the declaration of a type of object class
Type	Comp	A 64 bit signed integer
Type	Currency	A floating point type with 4 decimals used for financial values
Type	Double	A floating point type supporting about 15 digits of precision
Type	Extended	The floating point type with the highest capacity and precision
Keyword	File	Defines a typed or untyped file
Type	Int64	A 64 bit sized integer - the largest in Delphi
Type	Integer	The basic Integer type
Type	LongInt	An Integer whose size is guaranteed to be 32 bits
Type	LongWord	A 32 bit unsigned integer
Type	PAnsiChar	A pointer to an AnsiChar value
Type	PAnsiString	Pointer to an AnsiString value
Type	PChar	A pointer to an Char value
Type	PCurrency	Pointer to a Currency value
Type	PDateTime	Pointer to a TDateTime value
Type	PExtended	Pointer to a Extended floating point value
Type	PInt64	Pointer to an Int64 value
Type	Pointer	Defines a general use Pointer to any memory based data
Type	PShortString	A pointer to an ShortString value
Type	PString	Pointer to a String value
Type	PVariant	Pointer to a Variant value
Type	PWideChar	Pointer to a WideChar
Type	PWideString	Pointer to a WideString value
Type	Real	A floating point type supporting about 15 digits of precision
Type	Real48	The floating point type with the highest capacity and precision
Keyword	Record	A structured data type - holding fields of data
Type	ShortInt	An integer type supporting values -128 to 127

PROFESSOR MARCOS NIELSEN
TIPOS DE DADOS ACEITOS PELO DELPHI

Tipo	Nome	Descrição
Type	Single	The smallest capacity and precision floating point type
Type	SmallInt	An Integer type supporting values from -32768 to 32767
Type	String	A data type that holds a string of characters
Variable	TConvFamily	Defines a family of measurement types as used by Convert
Type	TConvType	Defines a measurement type as used by Convert
Type	TDateTime	Data type holding a date and time value
Type	Text	Defines a file as a text file
Type	TextFile	Declares a file type for storing lines of text
Type	TFloatFormat	Formats for use in floating point number display functions
Type	TFormatSettings	A record for holding locale values for thread-safe functions
Type	TObject	The base class type that is ancestor to all other classes
Type	TPoint	Holds X and Y integer values
Type	TRect	Holds rectangle coordinate values
Type	TReplaceFlags	Defines options for the StringReplace routine
Type	TSearchRec	Record used to hold data for FindFirst and FindNext
Type	TSysCharSet	Characters used by supplied string parsing functions
Type	TThreadFunc	Defines the function to be called by BeginThread
Type	Variant	A variable type that can hold changing data types
Type	WideChar	Variable type holding a single International character
Type	WideString	A data type that holds a string of WideChars
Type	Word	An integer type supporting values 0 to 65535

PROFESSOR MARCOS NIELSEN

TIPOS DE DADOS ACEITOS PELO DELPHI

Boolean

Tipo

Permite apenas **True** e **False** valores

```
tipo Boolean = (False, True);
```

Descrição

As **Boolean** tipo fornece uma enumeração de lógica **True** e **False** valores.

Diferente de outras linguagens, não é um número - ele só irá permitir que estes valores. Isso torna o código muito mais confiável.

Exemplo de código: Boolean tarefas e testes

```
var
  same : Boolean;
  small, large, i : Integer;

begin
  small := 23;
  large := 455555;

  // Test directly to see if these numbers are the same size
  if small = large
  then ShowMessage('small = large')
  else ShowMessage('small <> large');

  // Use a Boolean to hold and test this outcome
  same := (small = large);
  if same
  then ShowMessage('small = large')
  else ShowMessage('small <> large');

  // Assign a direct logical value to this Boolean
  same := True;
  if same
  then ShowMessage('same is True')
  else ShowMessage('same is False');
end;
```

PROFESSOR MARCOS NIELSEN

TIPOS DE DADOS ACEITOS PELO DELPHI

Delphi Basics

Inteiro

O tipo Integer básico

Sistema de unidade

Tipo

```
tipo Integer = -2147483648 .. 2147483647;
```

Descrição

A **Integer** tipo é um número inteiro cujo tamanho não é garantida. É o tipo inteiro básicos em Delphi, e atualmente tem a mesma capacidade que o **LongInt** tipo - 1 bit de sinal, valor e 31 bits.

Para realizar grandes inteiros muito, use o **Int64** tipo..

Example code : Showing the capacity of Integer

```
var  
  min, max : Integer;  
begin  
  // Set the minimum and maximum values of this data type  
  min := Low(Integer);  
  max := High(Integer);  
  ShowMessage('Min integer value = '+IntToStr(min));  
  ShowMessage('Max integer value = '+IntToStr(max));  
end;
```

PROFESSOR MARCOS NIELSEN TIPOS DE DADOS ACEITOS PELO DELPHI

Currency

Tipo

Um tipo de ponto flutuante com quatro casas decimais usadas para valores financeiros

Sistema
de unidade

tipo de moeda ;

Descrição

A **moeda** tipo é projetado para uso em aplicações financeiras. Ele suporta quatro casas decimais com precisão de pelo menos 53 bits *.

casas decimais além das quatro suportados são arredondados para cima ou para baixo, conforme o caso. Veja o código de exemplo para um exemplo.

Notas

* Depende do processador flutuante de precisão ponto.

Muito grande **Moeda** valores irão perder precisão com alguns dos **StrUtils** funções.

Exemplo de código: arredondamento para cima e para baixo valores em moeda

```
var
  Account1, account2, account3: Moeda ;
começar
  Account1: = 123,456749;    / muitas casas decimais Too - será
arredondado por defeito
  account2: = 123,456750;    / muitas casas decimais Too - será
arredondada para cima
  account3: = Account1 + account2;

  ShowMessage ('Account1 =' + CurrToStr (Account1));
  ShowMessage ('account2 =' + CurrToStr (account2));
  ShowMessage ('Account3 =' + CurrToStr (account3));
efeito;
```

PROFESSOR MARCOS NIELSEN TIPOS DE DADOS ACEITOS PELO DELPHI

Double

Tipo

Um tipo de ponto flutuante de apoio a cerca de 15 dígitos de precisão

Sistema
de unidade

tipo de casal ;

Descrição

O tipo **Double** é o objetivo geral tipo de ponto flutuante em Delphi. É o equilíbrio entre a capacidade / precisão versus armazenamento / desempenho.

Ele suporta cerca de 15 dígitos de precisão em um intervalo de $2,23 \times 10^{-308}$ a $1,79 \times 10^{308}$.

Exemplo de código: Mostrando a precisão ea capacidade de Double valores

```
var
  account1, account2, account3, account4 : Double;
begin
  account1 := 0.1234567890123456789;      // 20 decimal places
  account2 := 1.234567890123456789E308;   // Highest exponent value
  account3 := account1 + account2;
  account4 := 9.9E308;                    // Treated as infinite

  ShowMessage ('Account1 = '+FloatToStr(account1));
  ShowMessage ('Account2 = '+FloatToStr(account2));
  ShowMessage ('Account3 = '+FloatToStr(account3));
  ShowMessage ('Account4 = '+FloatToStr(account4));
end;
```

PROFESSOR MARCOS NIELSEN TIPOS DE DADOS ACEITOS PELO DELPHI

String

Um tipo de dados que armazena uma cadeia de caracteres

Sistema de unidade

Tipo

- 1 tipo **String** ;
- 2 tipo **String** [FixedSize constante];

Descrição

A **String** tipo de dados é usado para armazenar seqüências de caracteres, como sentenças.

String é, por padrão, na verdade, tratados como uma **AnsiString** . Ele pode ser tratado como um **ShortString** se o **\$ LongStrings** diretiva de compilador está definido **Off** (o padrão é **No**).

Um **AnsiChar** pode conter qualquer número de caracteres, restritos apenas pela memória.

Versão 2 entanto, as forças da cadeia para ser um **ShortString** definindo uma **FixedSize** (até 255 caracteres) da cadeia. Isto é particularmente importante quando a criação de muitas cordas, e especialmente quando armazenar strings em registros (como no exemplo 2).

Strings podem ser atribuídos a partir de outros textos, a partir de funções que retornam uma seqüência, e com concatenações como no código de exemplo.

Exemplo de código: Atribuir a uma corda e em seguida, adicionando um pouco mais

```
var
  myString: String ;
começar
  // Atribuir uma frase famosa para essa seqüência
  myString: = 'Olá Mundo';

  // Adicionar a esta cadeia
  myString: = myString + ', como estão todos? ";

  // Mostra o valor final myString
  ShowMessage ('myString =' myString +);
feito;
```

[Mostrar código de unidade integral](#)

```
Olá Mundo, como estão todos?
```

PROFESSOR MARCOS NIELSEN
TIPOS DE DADOS ACEITOS PELO DELPHI

Real Um tipo de ponto flutuante de apoio a cerca de 15 dígitos de
precisão

Obsoletos

Sistema
de unidade

tipo **Real** ;

Descrição

O **Real** é o tipo obsoleto.

Você deve usar o equivalente atual: **Double**

PROFESSOR MARCOS NIELSEN TIPOS DE DADOS ACEITOS PELO DELPHI

Pointer

Tipo

Define um uso geral Ponteiro para a memória de dados com base

Sistema de unidade

Tipo **Pointer** ;

Descrição

O **ponteiro do** tipo fornece um ponteiro de uso geral para qualquer variável baseada na memória. Ou seja, uma que é acessada por referência.

Objetos, ansistrings e matrizes são exemplos de variáveis de referência com base.

Mas seja **advertido** : os ponteiros não tipados são perigosas - normalmente é sempre melhor usar uma referência de ponteiro específico para o tipo de dados você está usando. Só então você pode agir de acordo com o ponteiro, como no exemplo.

Exemplo de código: Referindo-se à forma atual usando um ponteiro variável

```
var
  generalPtr : Pointer; // A pointer to anything
  formPtr    : ^TForm;  // A pointer to a form object

begin
  // The current unit's form is addressable via the self keyword
  generalPtr := Addr(self);

  // We can assign this pointer to the form pointer
  formPtr := generalPtr;

  // And set the form caption to show this
  formPtr.Caption := 'Test program';
end;
```