

## 絵に文字を描写する ~TextOut~

ここでは、Imageなどで読み込んだ絵に文字を描写する方法について説明します。

ゲームなんかを作っていると読み込んだ絵や背景などに文字を書き込みたいときが出てくるでしょう。そのような時は、ImageやFormなどのキャンバスに文字を描写する関数「TextOut」を使いましょう。

では、早速始めてみましょう。

### 1. 下準備

まずは、FormにImageを一つおいてください。その後、プログラム上でFormの大きさを500×500にし、Imageの大きさをFormに合わせましょう。

次に、ペイントなどで500×500の絵を適当に描いて、Imageに読みこんでください。

### 2. 文字を書く

では、実際に文字を書いてみましょう。

Imageに文字を書くだけならForm1のfastcallに書いてもらって結構です。

```
fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    Form1->Height = 500;
    Form1->Width = 500;
    Image1->Top = 0;
    Image1->Left = 0;

    //画像の読み込み(詳しくはコピーレクト参照)
    Image1->Picture->LoadFromFile("ライブラリ2.bmp");

    //サイズ、色、文字の指示
    Image1->Height = 500;
    Image1->Width = 500;
    Image1->Canvas->Font->Size = 20;
    Image1->Canvas->Font->Color = clBlack;
    Image1->Canvas->TextOut(20,10,"こんにちわ");
}
```

「Canvas」と言うのは Image の絵を書くための紙、又はそのままキャンバスのことです。今、Canvas には先ほど読み込んだ絵がかかれています。

そのキャンバスに今から書く文字の大きさ、色など指定しています。もちろん書式も変えることができます。

そして、「TextOut」これが、キャンバスに文字を描写する関数です。中身は

TextOut(書き始める x 座標, 書き始める y 座標, ” 書きたい文字” )  
今回は、Image のキャンバスに x 座標 20、y 座標 10 の位置から” こんにちは” と書き込む処理になっています。

これを実行するとこうなるはずですが



書いた文字の範囲が白い四角になっているでしょう。それは、TextOut で書いた文字の背景がデフォルトでは白に設定されているからです。この白の部分を透過して背景にしたいときは次の文を入れてください。

Image1->Canvas->Brush->Style = bsClear ;  
実行すると背景が透けるようになります。



#### 参考

文字を描写する `Textout` と組み合わせて使うと便利なものを上げておきます。

`TextOut` で書かれた文字の範囲を知るには、`TextExtent` や `TextWidth`、`TextHeight` などがあります。

また、`TextRect` は範囲内のみ、文字が現れるようにするメソッドです。範囲からはみ出た部分は切り取られ表示されません。