IOS 文本框及其代理类的使用

IOS 库中提供了为所有 UIKit 组件，都提供了两个定义，一个是类定义、一个是协议定义。如文本框：

UITextField : 提供了界面的各个属性，仅限于界面

UITextFieldDelegate : 提供了能够对文本框所做操作的定义

API 原文

The UITextFieldDelegate protocol defines the messages sent to a text field delegate as part of the sequence of editing its text. All of the methods of this protocol are optional.

翻译

UITextFieldDelegete 协议定义了在一个文本框中，按照顺序，编辑里面文字的时候，传递给它的 delegete 的消息。

协议方法说明

- (BOOL)textFieldShouldBeginEditing:(UITextField \*)textField{

//返回一个BOOL值，指定是否循序文本字段开始编辑

return YES;

}

- (void)textFieldDidBeginEditing:(UITextField \*)textField{

//开始编辑时触发，文本字段将成为first responder，一般默认

[textField becomeFirstResponder];

}

- (BOOL)textFieldShouldEndEditing:(UITextField \*)textField{

//返回BOOL值，指定是否允许文本字段结束编辑，当编辑结束，文本字段会让出first responder

//要想在用户结束编辑时阻止文本字段消失，可以返回NO

//这对一些文本字段必须始终保持活跃状态的程序很有用，比如即时消息

return NO;

}

- (BOOL)textField:(UITextField\*)textField shouldChangeCharactersInRange:(NSRange)range replacementString:(NSString \*)string{

//当用户使用自动更正功能，把输入的文字修改为推荐的文字时，就会调用这个方法。

//这对于想要加入撤销选项的应用程序特别有用

//可以跟踪字段内所做的最后一次修改，也可以对所有编辑做日志记录,用作审计用途。

//要防止文字被改变可以返回NO

//这个方法的参数中有一个NSRange对象，指明了被改变文字的位置，建议修改的文本也在其中

return YES;

}

/////////////////////////////////////

//限制用户输入字符长度，超过长度，自动截取

- (BOOL)textField:(UITextField \*)textField shouldChangeCharactersInRange:(NSRange)range replacementString:(NSString \*)string

{

if ([textField.text length] > MAXLENGTH)

{

textField.text = [textField.text substringToIndex:MAXLENGTH-1];

return NO;

}

return YES;

}

/////////////////////////////////////

- (BOOL)textFieldShouldClear:(UITextField \*)textField{

//返回一个BOOL值指明是否允许根据用户请求清除内容

//可以设置在特定条件下才允许清除内容

return YES;

}

-(BOOL)textFieldShouldReturn:(UITextField \*)textField{

//返回一个BOOL值，指明是否允许在按下回车键时结束编辑

//如果允许要调用resignFirstResponder 方法，这回导致结束编辑，而键盘会被收起

[textField resignFirstResponder];//查一下resign这个单词的意思就明白这个方法了

return YES;

}

--------------------------------

练习一：掌握 UITextFieldDelegate 与 UITextField 配合应用

(1)当text field 获得焦点时，是把text field 的背景色设置为绿色.

- (BOOL)textFieldShouldBeginEditing:(UITextField \*)textField{

NSLog(@"textFieldShouldBeginEditing:");

return YES;

}

- (void)textFieldDidBeginEditing:(UITextField \*)textField {

NSLog(@"textFieldDidBeginEditing:");

textField.backgroundColor = [UIColor greenColor];

}

(2)当text field失去焦点之前会调用textFieldShouldEndEditing。在示例中，我们使用textFieldShouldEndEditing:让背景色返回最初的颜色。

- (BOOL)textFieldShouldEndEditing:(UITextField \*)textField{

NSLog(@"textFieldShouldEndEditing:");

textField.backgroundColor = [UIColor whiteColor];

return YES;

}

- (void)textFieldDidEndEditing:(UITextField \*)textField{

NSLog(@"textFieldDidEndEditing:");

}

(3)当用户输入字符的时候，需要监测输入字符的长度，大于10，不允许输入

- (BOOL)textField:(UITextField \*)textField shouldChangeTextInRange:(NSRange)range replacementText:(NSString \*)text{

NSCharacterSet \*doneButtonCharacterSet = [NSCharacterSet newlineCharacterSet];

NSRange replacementTextRange = [text rangeOfCharacterFromSet:doneButtonCharacterSet];

NSUInteger location = replacementTextRange.location;

if (textView.text.length + text.length > 10){

if (location != NSNotFound){

[textView resignFirstResponder];

}

return NO;

}

else if (location != NSNotFound){

[textView resignFirstResponder];

return NO;

}

return YES;

}

每次用户通过键盘输入字符时，在字符显示在text Field之前，textField:shouldChangeCharactersInRange:replacementString方法会被调用。这个方法中可以方便的定位测试用户输入的字符，并且限制用户输入特定的字符。在上面的代码中，我使用done键来隐藏键盘：通过检测看replacement文本中是否包含newLineCharacterSet任意的字符。

如果有一个字符是来自newLineCharacterSet的，那么说明done按钮被按过了，因此应该将键盘隐藏起来。另外，在用户每次输入内容时，还检测text Field当前文本内容的长度，如果大于140个字符，则返回NO，这样text Field就可以限制输入10个字符了。

练习二：与文本框类似的一个组件，UITextView 允许用户输入多行文字，请自己查询 API ，练习上述同样的例子，仔细理解 UITextView 与 UITextViewDelegate 之间的配合使用。