#### CALabel

###### 类说明

标签文字类，用于显示文本。

###### 基类

[CAView](../CAView.docx)

###### 属性

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 属性名 | 说明 |
| protected | [Text](#Text) | label的文本内容 |
| protected | [fontName](#fontName) | label的字体 |
| protected | [fontSize](#fontSize) | label的字体大小 |
| protected | [Color](#TextColor) | label的字体颜色 |
| protected | [VerticalTextAlignmet](#VerticalTextAlignmet) | 文本的竖直对齐方式 |
| protected | [TextAlignment](#TextAlignment) | 文本的水平对齐方式 |
| protected | [NumberOfLine](#NumberOfLine) | label的行数 |

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| public | [createWithFrame](#createWithFrame) | 创建一个label |
| public | [createWithCenter](#createWithCenter) | 创建一个label |

###### 属性介绍

**Text**

类型：string

解释：label的文本内容

**fontName**

类型：string

解释：label使用的字体

**fontSize**

类型：usigned int

解释：字体的大小

**Color**

类型：ccColor4B

解释：文本的颜色

**VerticalTextAlignmet**

类型：CAVerticalTextAlignment

解释：文本的竖直对齐方式，枚举类型。

typedef enum

{

CAVerticalTextAlignmentTop,

CAVerticalTextAlignmentCenter,

CAVerticalTextAlignmentBottom,

} CAVerticalTextAlignment;

第一种是顶部对齐方式，第二种是中心对齐方式，第三种是底部对齐方式。

**TextAlignment**

类型：CATextAlignment

解释：文本的水平对齐方式，枚举类型。

typedef enum

{

CATextAlignmentLeft,

CATextAlignmentCenter,

CATextAlignmentRight,

} CATextAlignment;

第一种是左对齐方式，第二种是中心对齐方式，第三种是右对齐方式。

示例：

CALabel\* c\_Label =

CALabel::createWithCenter(CCRect(size.width\*0.5,size.height\*0.5,220,500));

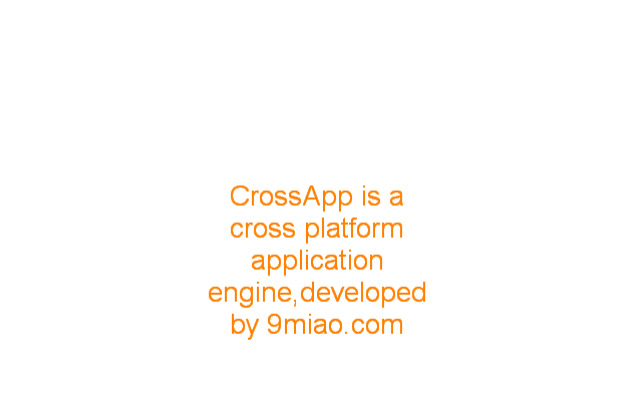
c\_Label->setText("CrossApp is a cross platform application engine,developed by 9miao.com");

c\_Label->setTextAlignment(CATextAlignmentCenter);

c\_Label->setColor(CAColor\_orange);

c\_Label->setFontSize(28);

this->getView()->addSubview(c\_Label);



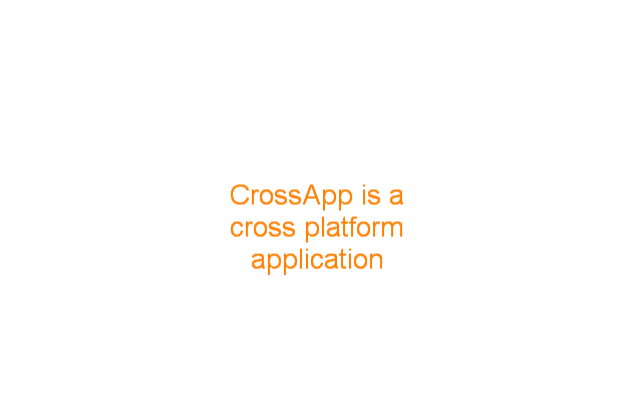
**NumberOfLine**

类型：unsigned int

解释：文本的行数

示例：

c\_Label->setNumberOfLine(3);



###### 方法介绍

static CALabel\* createWithFrame(const CCRect& rect)

返回值：CALabel\*

解释：创建一个label，并指定frame。

static CALabel\* createWithCenter(const CCRect& rect)

返回值：CALabel\*

解释：创建一个label，并指定center。