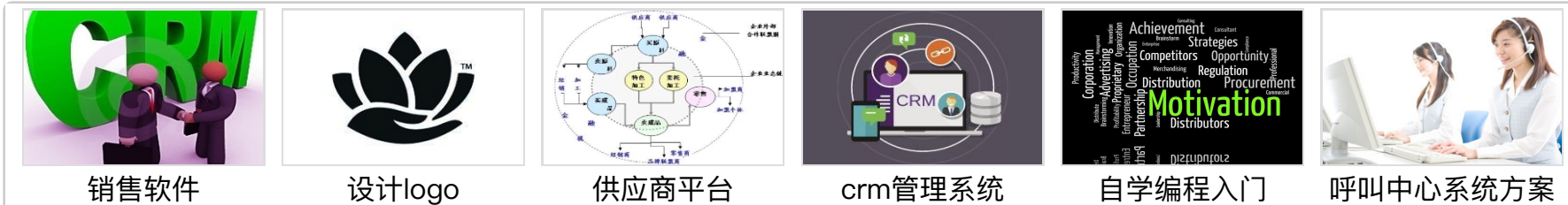


3赞

赞赏

更多好文



Swift：UILabel超出宽度文字的截取

IlronMan

关注

0.319

2021.05.25 11:29:54

字数 853

阅读 722

JKSwiftExtension，测试用例在 JKContentSizeViewController.swift 里面
需求：单行文字可缩放到一定比例，按照Char截取，超出截掉，不允许出现Char被截取一半的情况



效果图

一、常规的剪切方式 (lineBreakMode 设置文字过长时的显示格式)

- 1.1、NSLineBreakMode枚举如下

```
1 public enum NSLineBreakMode : Int {
2     // 以单词为显示单位显示, 后面部分省略不显示
3     case byWordWrapping = 0
4     // 以字符为显示单位显示, 后面部分省略不显示。
5     case byCharWrapping = 1
6     // 剪切与文本宽度相同的内容长度, 后半部分被删除。
7     case byClipping = 2
8     // 前面部分文字以.....方式省略, 显示尾部文字内容
9     case byTruncatingHead = 3
10    // 结尾部分的内容以.....方式省略, 显示头的文字内容
11    case byTruncatingTail = 4
12    // 中间的内容以.....方式省略, 显示头尾的文字内容
13    case byTruncatingMiddle = 5
14 }
```

使用如下：以 byWordWrapping 的形式剪切

```
1 let label = UILabel(frame: CGRect(x: 20, y: 20, width: 100, height: 19))
2 label.backgroundColor = .yellow
3 label.font = UIFont.systemFont(ofSize: 10)
4 label.textAlignment = .left
5 label.textColor = UIColor.brown
6 label.lineBreakMode = .byWordWrapping
7 return label
```

二、本次的需求实现

- 2.1、需求：单行文字可缩放到一定比例，按照Char截取，超出截掉，不允许出现Char被截取一半的情况

- 需求分析：
 - 可缩放一定比例
 - 按照Char截取
 - 不允许出现Char被截取一半的情况
- 实现方式：先获取缩放后的字体大小，后获取指定宽度的第一行文字的内容

- 2.2、具体的核心代码如下

```
1 public struct JKContentSize {}
2
3 public extension JKContentSize {
4     // MARK: 2.1、行数和每行的内容
5     /// 获取已知 width 的 label 的文本行数 & 每一行内容
6     /// - Parameters:
7     ///   - content: 文本的内容
8     ///   - font: 字体
9     ///   - width: 宽度
10    ///   - height: 高度
11    ///   - minimumScaleFactor: 缩放率
12    /// - Returns: 文本行数 & 每一行内容
13    static func linesCountAndLinesContent(content: String, font: UIFont, width: CGFloat) -> (Int, [String]) {
14        let lodFontName = font.fontName
15        let fontSize = getFontSizeForLabel(content: content, minimumScaleFactor: 0.5, fontName: lodFontName)
16        let newFont = UIFont(name: lodFontName, size: fontSize) ?? UIFont.systemFont(ofSize: 10)
17        let c_fn = newFont.fontName as CFString
18        let fp = newFont.pointSize
19        let c_f = CTFontCreateWithName(c_fn, fp, nil)
20
21        let style = NSMutableParagraphStyle()
22        style.lineBreakMode = .byCharWrapping
23        let contentDict = [NSAttributedString.Key.paragraphStyle : style] as [NSAttributedString.Key : Any]
24        let attr = NSAttributedString(string: content, attributes: contentDict)
25        let range = NSRange(location: 0, length: attr.length)
26        attr.addAttributes(contentDict, range: range)
27        attr.addAttribute(NSAttributedString.Key.font, value: c_f, range: range)
28        let frameSetter = CTFramesetterCreateWithAttributedString(attr as CFAttributedString)
29
30        let path = CGMutablePath()
31        /// 2.5 是经验误差值
32        path.addRect(CGRect(x: 0, y: 0, width: width, height: height > (fp * 1.5) ? height : (fp * 1.5)))
33        let framef = CTFramesetterCreateFrame(frameSetter, CRangeMake(0, 0), path, 0)
34        let lines = CTFrameGetLines(framef) as NSArray
35        var lineArr = [String]()
36        for line in lines {
37            let lineRange = CTLineGetStringRange(line as! CTLine)
38            let lineString = content.jk.sub(start: lineRange.location, length: lineRange.length)
39            lineArr.append(lineString as String)
40        }
41        return (lineArr.count, lineArr)
42    }
43
44    // MARK: 获取字体大小
45    /// 获取字体的大小
46    /// - Returns: 字体的大小
47    private static func getFontSizeForLabel(content: String, minimumScaleFactor: CGFloat, fontName: String) -> CGFloat {
48        let text: NSMutableAttributedString = NSMutableAttributedString(attributedString: NSAttributedString(string: content))
49        text.setAttributes([NSAttributedString.Key.font: fontName as Any], range: NSRange(location: 0, length: text.length))
50        let context: NSStringDrawingContext = NSStringDrawingContext()
51        context.minimumScaleFactor = minimumScaleFactor
52        text.boundingBox(with: CGSize(width: width, height: height), options: NSStringDrawingOptions.usesLineFragmentOrigin)
53        let adjustedFontSize: CGFloat = font.pointSize * context.actualScaleFactor
54        return adjustedFontSize
55    }
56 }
```

3人点赞 >

Swift5 知识点

更多精彩内容，就在简书APP



"喜欢，您就打赏！"

赞赏支持

还没有人赞赏，支持一下

IlronMan

一名普通的iOS开发者

总资产768

共写了61.2W字

获得3,023个赞

共1,974个粉丝

关注

音乐平台



写下你的评论...

评论2 赞3 ...