

使用方式
外层方法
内部结构

概念区分
equalTo() 与 mas_equalTo() 的区别
offset()与inset()的区别

官方Github链接 -> [Masonry](#)

如果不知道怎么用, 请直接参考里脊串的 [Masonry介绍与使用实践\(快速上手Autolayout\)](#)

UIScrollView中添加subView如果需要使用Masonry. 参考 里脊串的 [UIScrollView与Autolayout的那点事](#)

使用方式

外层方法

```
- (NSArray *)mas_makeConstraints:(void(^)(MASConstraintMaker *make))block;
- (NSArray *)mas_updateConstraints:(void(^)(MASConstraintMaker *make))block;
- (NSArray *)mas_remakeConstraints:(void(^)(MASConstraintMaker *make))block;

/*
    mas_makeConstraints 只负责新增约束 Autolayout不能同时存在两条针对于同一对象的约束 否则会报错
    mas_updateConstraints 针对上面的情况 会更新在block中出现的约束 不会导致出现两个相同约束的情况
    mas_remakeConstraints 则会清除之前的所有约束 仅保留最新的约束
*/
```

内部结构

1. 四个边距同时设定

```
make.edges.equalTo(otherView).with.insets(UIEdgeInsetsMake(5,5,5,5));
//make与otherView的上下左右间距差均为5;
```

2. 相对某view的某边的距离问题(equalTo())

```
make.top.equalTo(otherView).with.offset(10);  
//make与otherView的上间距为10;
```

3. 相对某view的某边的距离问题(mas_equalTo())

```
make.top.mas_equalTo(otherView.mas_top).offset(10);  
//make与otherView的上间距为10;
```

4. 多控件处于同一水平线居中

```
make.centerY.equalTo(@[sub2,sub3]);
```

5. 多控件处于同一垂直线居中

```
make.centerX.equalTo(@[sub2,sub3]);
```

6. 控件在superView中居中

```
make.center.equalTo(superView);
```

7. 控件在superView中水平居中

```
make.centerY.equalTo(superView);
```

或者是

```
make.centerY.equalTo(superView.centerY);
```

8. 控件在superView中垂直居中

```
make.centerX.equalTo(superView);
```

或者是

```
make.centerX.equalTo(superView.centerX);
```

9. Aview的左边距离Bview的右边10px

```
make.left.mas_equalTo(Bview.mas_right).offset(10);
```

10. 权值的设置

```
make.width.equalTo(@0).priority(UILayoutPriorityRequired);
//UILayoutPriorityRequired 1000
//UILayoutPriorityDefaultHigh 750
//UILayoutPriorityDefaultLow 250
```

11. 两个控件右对齐

```
make.left.equalTo(Bview.mas_left);
```

12. 单一约束生效/单一约束无效

初始化约束:

```
property (nonatomic, strong) MASConstraint *constr;
方法体中创建约束:
[view mas_makeConstraints:^(MASConstraintMaker *make) {
    constr = make.height.equalTo(@0).priority(UILayoutPriorityRequired);
}];
```

约束无效:

```
[constr deactivate];
```

约束生效:

```
[constr activate];
```

概念区分

equalTo() 与 mas_equalTo() 的区别

```
前者的参数为object,后者为前者的封装, 参数仅支持 *NSNumber* *CGPoint* *CGSiz
ze* *UIEdgeInsets*.
```

offset()与inset()的区别

offset就是作用在单个属性的时候用 比如make.top.equalTo(self.view.mas_top).xxxx 这时候 equalTo的对象 是要指定具体属性的 比如mas_top,mas_right

insets就是同时作用多个属性的时候用 比如

make.left.right.bottom.equalTo(self.view).xxxx 这时候 equalTo的对象 是不指定具体的属性的

insets只是对offset封装了一下而已 避免offset写一大堆 用inset一句话就搞定了