

# Snapkit

## 结构

- ConstraintView
  - 在 iOS 中就是 UIView,
  - UIView 和 NSView 的别名, 统一不同平台的变量名
- ConstraintViewDSL
  - UIView 通过 extension 增加一个类型为 ConstraintViewDSL 的 snap 属性, 并持有 UIView
- ConstraintAttributesDSL
  - 提供 布局属性: left, right

## 如何设置约束

- ConstraintMaker
  - 在构造函数里, 禁用了 AutoresizeMask
  - ConstraintDescription 数组包含了用户设置的各个属性
  - makeConstraints 函数 来保存用户设置的约束
- ConstraintAttributes
  - 是 OptionSet 结构, 内部采用的是 位域, 可以由多个属性组合表示另一个属性。比如 size 是由 width 和 height 组合而成。
  - 在 swift 中, 枚举没办法将多个枚举选项组成一个值。
- ConstraintDescription
  - 一个完整的约束描述类, 里面包含了 ConstraintAttributes 在内的各种与约束有关的元素, 一个 ConstraintDescription 实例, 就可以提供与一种约束有关的所有内容。

## 设置完后如何处理

makeConstraints 函数如何在外部通过一个闭包来写约束关系的?

- 1, 初始化 ConstraintMaker 对象
- 2, 调用外部闭包函数, 将设置的属性保存到 ConstraintDescription 数组中。
- 3, 从 description 数组中遍历所有的约束, 并激活。