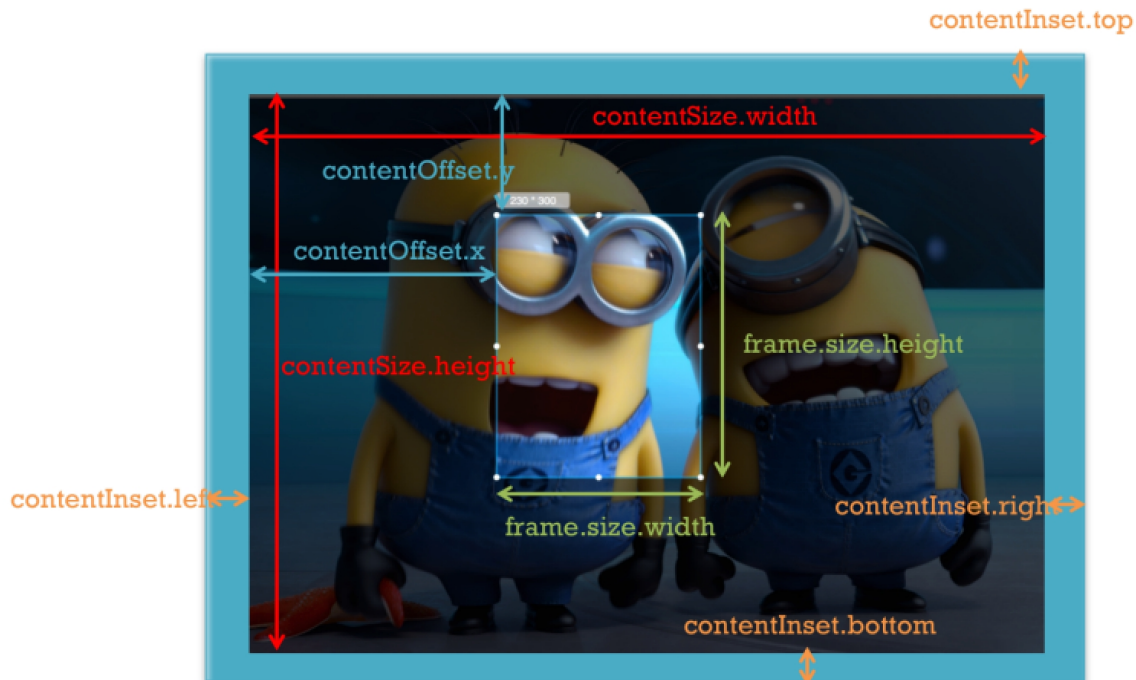


# UIScrollView:

## 1. UIScrollView的常见属性:



## 2. UIScrollView的代理:

如果想在 `UIScrollView` 正在滚动或滚动到某个位置或者停止滚动时做一些特定的操作，就要求我们能够监听到 `UIScrollView` 的整个滚动过程。而为 `UIScrollView` 设置代理（**delegate**）对象，可以完成监听的任务，给它的代理发送相应的消息，通过代理得知它的滚动情况。

# UITableView:

## 1. UITableView展示数据的方式:

`UITableView` 需要一个数据源(**dataSource**)来显示数据。它会向数据源查询一共有多少行数据以及每一行显示什么数据等。所以，需要时实现 `UITableViewDataSource` 协议来完成 `UITableView` 的设置。

## 2. UITableView展示数据的代码实现:

- 设置UITableView的组（**Section**）数量:

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView;
```

- 设置UITableView各组中行（**Row**）的数量:

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:
(NSInteger)section;
```

- 设置UITableView各行所展示的内容：

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
(NSIndexPath *)indexPath;
```

### 3.Cell的重用机制：

- 假设某个 UITableView 有100个数据需要显示，即需要100个 cell，会消耗很大的内存，然而屏幕中最多只能一次性显示10个 cell，那么就只需要创建(10+1)个；
- 当滑动 tableView 的时候，刚离开视图的 cell 会被放到复用池中，等下一个cell需要显示时，会先看复用池中有没有cell如果有的时候，就从复用池中拿出来cell，没有的话就重新创建cell。
- 实现方法：

```
UITableViewCell *cell = [tableView
 dequeueReusableCellWithIdentifier:CellIdentifier];
```

## 其他问题总结：

### 1.self.a和\_a的区别：

- 使用 self.a 会调用getter/setter方法，而 \_a 并不会调用getter/setter方法,它是直接访问实例变量并赋值。
- self 和 \_ 的赋值方式不同，self进行深拷贝，\_ 进行浅拷贝。
- 同时，使用 \_ 是获取不到父类的属性，因为它只是对局部变量的访问。

Tip：应尽量使用 self，但是在进行懒加载的代码中，不要使用 self，否则会引起循环调用。

### 2.alloc/init和new的区别：

- 两种方式创建对象现在基本上一样，区别就是使用 new 只能默认 init 进行初始化，alloc 方式可以使用其它的 init 开头的方法进行初始化。
- 其实默认的 init 方法中，什么都没有做,直接返回了 self，所以,如果没有重写 init 的话，[Class alloc] 和 [[Class alloc]init] 是等价的。

### 3.objectForKey&valueForKey&valueForKeyPath:

- `objectForKey` 仅用于字典，通过其键查找对象。
- `valueForKey` 是通过显示调用getter获取对象。
- `valueForKeyPath` 则是可以能找到所有层级的键值对，相当于多次调用 `valueForKey`。