

关于ios：setNeedsLayout与setNeedsUpdateConstraints和layoutIfNeeded vs updateConstraintsIfNeeded

📅 2019-06-02

🏷️ [autolayout](#) [ios](#) [nslayoutconstraint](#) [objective-c](#)

setNeedsLayout vs. setNeedsUpdateConstraints and layoutIfNeeded vs updateConstraintsIfNeeded

我知道自动布局链基本上由3个不同的过程组成。

- 更新约束
- 布局视图(这里是计算框架的地方)
- 显示

我不太清楚的是，`-setNeedsLayout` 和 `-setNeedsUpdateConstraints` 之间的内在区别。来自Apple Docs：

设置需求布局

Call this method on your application's main thread when you want to adjust the layout of a view's subviews. This method makes a note of the request and returns immediately. Because this method does not force an immediate update, but instead waits for the next update cycle, you can use it to invalidate the layout of multiple views before any of those views are updated. This behavior allows you to consolidate all of your layout updates to one update cycle, which is usually better for performance.

设置需要更新约束

When a property of your custom view changes in a way that would impact constraints, you can call this method to indicate that the constraints need to be updated at some point in the future. The system will then call updateConstraints as part of its normal layout pass. Updating constraints all at once just before they are needed ensures that you don't needlessly recalculate constraints when multiple changes are made to your view in between layout passes.

当我想在修改约束后为视图设置动画，并为我通常调用的更改设置动画时，例如：

```
1 [UIView animateWithDuration:1.0f delay:0.0f usingSpringWithDamping:0.5f initialSprin
2     [self.modifyConstrView setNeedsUpdateConstraints];
3     [self.modifyConstrView layoutIfNeeded];
4     } completion:NULL];
```

我发现，如果我使用 `-setNeedsLayout` 而不是 `-setNeedsUpdateConstraints`，一切都能按预期工作，但是如果我用 `-updateConstraintsIfNeeded` 更改 `-layoutIfNeeded`，动画就不会发生。我试图得出自己的结论：

- `-updateConstraintsIfNeeded` 只更新约束，但不强制布局进入进程，因此原始帧仍然保留
- `-setNeedsLayout` 也调用 `-updateContraints` 方法

那么什么时候可以用一个代替另一个呢？关于布局方法，我需要在约束发生变化的视图上调用它们还是在父视图上调用它们？

► 相关讨论

广告 X

同时运行Windows和Mac应用程序

Mac与Window融合。在Mac上运行您喜爱的Windows软件。Mac用户的最优之选。充分利用Mac。立即试用!

Parallels.cn

了解详情



你的结论是对的。基本方案是：

- `setNeedsUpdateConstraints` 确保将来调用 `updateConstraintsIfNeeded` 调用 `updateConstraints`。
- `setNeedsLayout` 确保将来调用 `layoutIfNeeded` 时调用 `layoutSubviews`。

当调用 `layoutSubviews` 时，它也调用 `updateConstraintsIfNeeded`，所以在我的经验中很少需要手动调用它。实际上，除了在调试布局时，我从未调用过它。

使用 `setNeedsUpdateConstraints` 更新约束也是非常罕见的，objc.io——一个必须了解自动付款的人——说：

If something changes later on that invalidates one of your constraints, you should remove the constraint immediately and call setNeedsUpdateConstraints. In fact, that's the only case where you should have to trigger a constraint update pass.

另外，根据我的经验，我从来没有必要使约束失效，也没有在代码的下一行中设置 `setNeedsLayout`，因为新的约束要求一个新的布局。

经验法则则是：

- 如果直接操纵约束，请调用 `setNeedsLayout`。
- 如果您更改了某些条件(如偏移量或smth)，这些条件将更改重写的 `updateConstraints` 方法(一种建议的更改约束的方法，btw)中的约束，则调用 `setNeedsUpdateConstraints`，大多数情况下，在这之后调用 `setNeedsLayout`。
- 如果您需要上述任何操作立即生效，例如，当您需要在布局通过后学习新的帧高度时，请将其附加到 `layoutIfNeeded` 中。

另外，在您的动画代码中，我相信 `setNeedsUpdateConstraints` 是不需要的，因为约束在动画之前是手动更新的，动画只根据新旧的不同重新布局视图。

► 相关讨论



回过头来的回答是非常正确的。不过，我想添加一些其他细节。

下面是一个典型的uiview循环图，它解释了其他行为：



- 我发现，如果我使用 `-setNeedsLayout` 而不是 `-setNeedsUpdateConstraints`，一切都能按预期工作，但如果我用 `-updateConstraintsIfNeeded` 更改 `-layoutIfNeeded`，动画就不会发生。

`updateConstraints` 通常不做什么事情。它只是解决约束，直到调用 `layoutSubviews` 时才应用约束。所以动画需要调用 `layoutSubviews`。

- `setNeedsLayout`调用also-updateContaints方法

不，这是不必要的。如果您的约束没有被修改，uiview将跳过对 `updateConstraints` 的调用。您需要显式调用 `setNeedsUpdateConstraint` 来修改流程中的约束。

要呼叫 `updateConstraints`，您需要执行以下操作：

```
1 [view setNeedsUpdateConstraints];
2 [view setNeedsLayout];
3 [view layoutIfNeeded];
```

► 相关讨论



广告 X

同时运行Windows和Mac应用程序

Mac与Window融合。在Mac上运行您喜爱的Windows软件。Mac用户的最优之选。充分利用Mac。立即试用!

Parallels.cn

了解详情