TEXTURE

moka

in let us: Go 2019 spring

GOAL

오늘의 목표

GOAL

오늘의 목표

Texture 쓰세요

TEXTURE ??

AsyncDisplayKit

Texture is an iOS framework built on top of UIKit that keeps even the most complex user interfaces smooth and responsive.

TEXTURE ??

AsyncDisplayKit

Texture is an iOS framework built on top of UIKit that keeps even the most complex user interfaces smooth and responsive.

Texture는 복잡한 사용자 인터페이스를 부드럽고 반응성 좋은 UIKit 위에 구축된 iOS 프레임 워크입니다.

😵 😵 😵 😵 😵

쉽게 말하면,

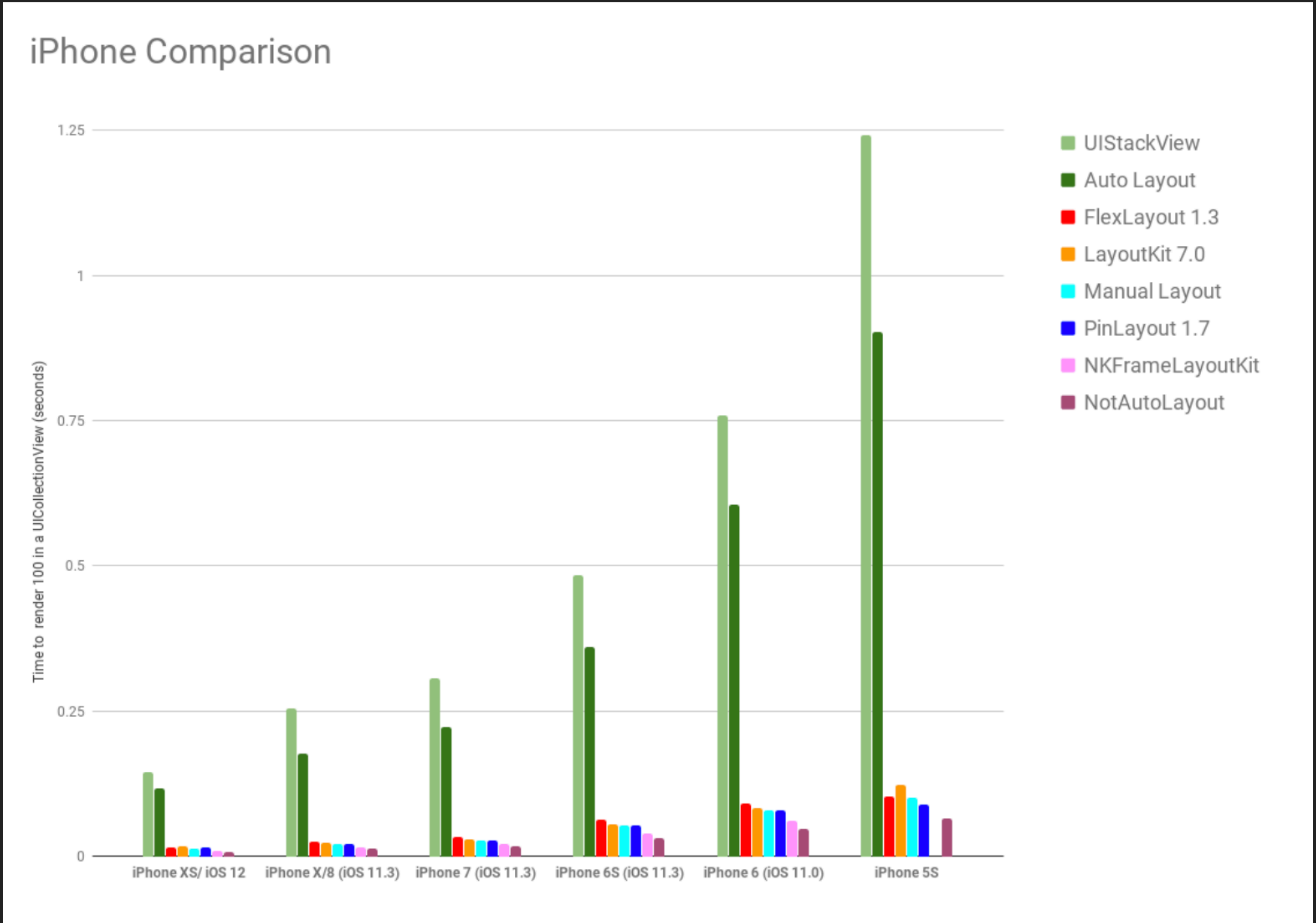
“story board 또는 xib(nib) 등 없이 코드로 UI 를 그리는 프레임워크” 입니다.

쉽게 말하면,

“빠르고 좋은”

Why Texture？

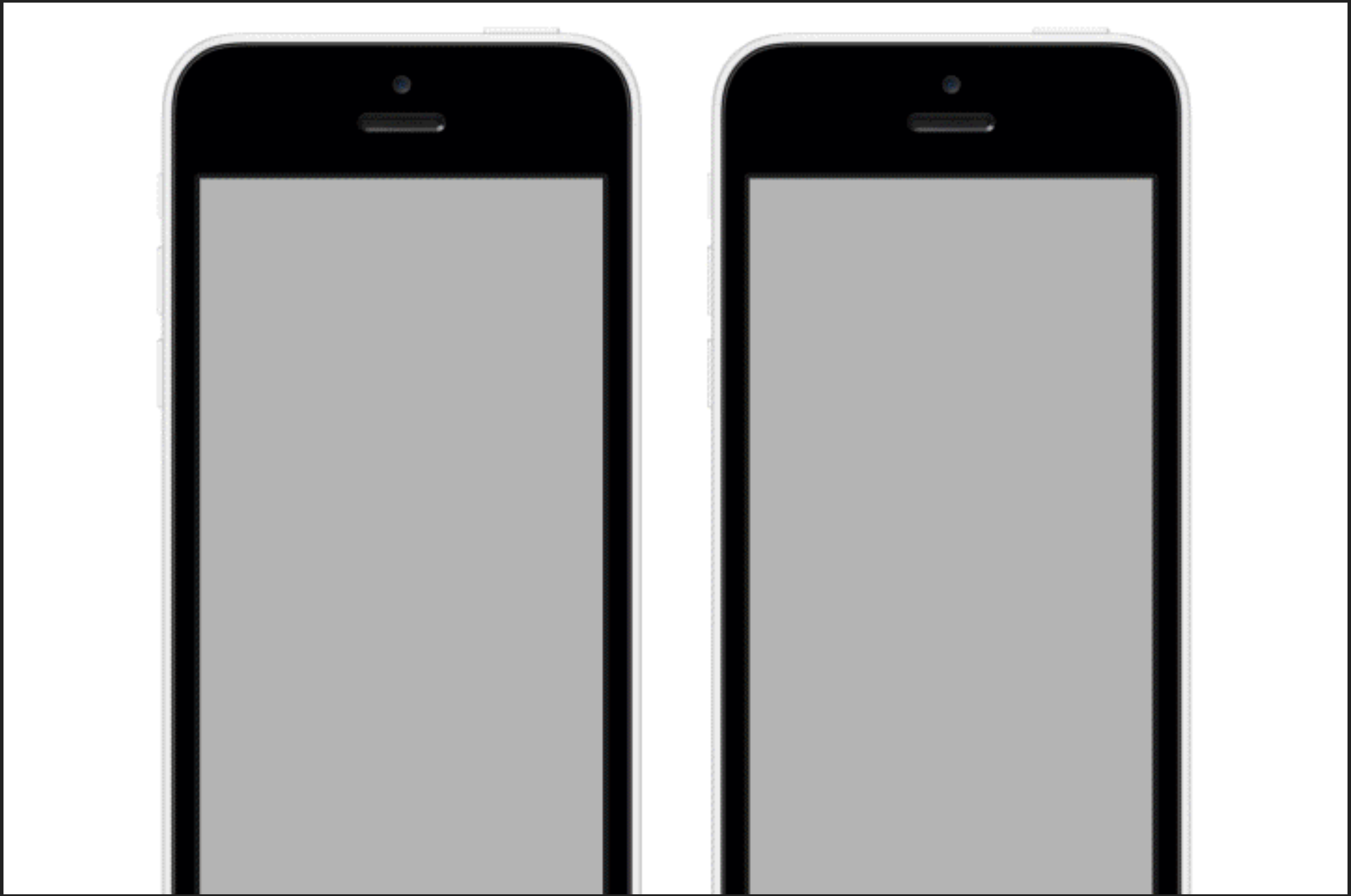
AutoLayout ⋯



https://github.com/layoutBox/LayoutFrameworkBenchmark

Pinterest Slack(?)

빙글



https://medium.com/@Pinterest_Engineering/re-architecting-pinterest-039-s-ios-app-e0a2d34a6ac2

장점 Texture ❤️

코드이다

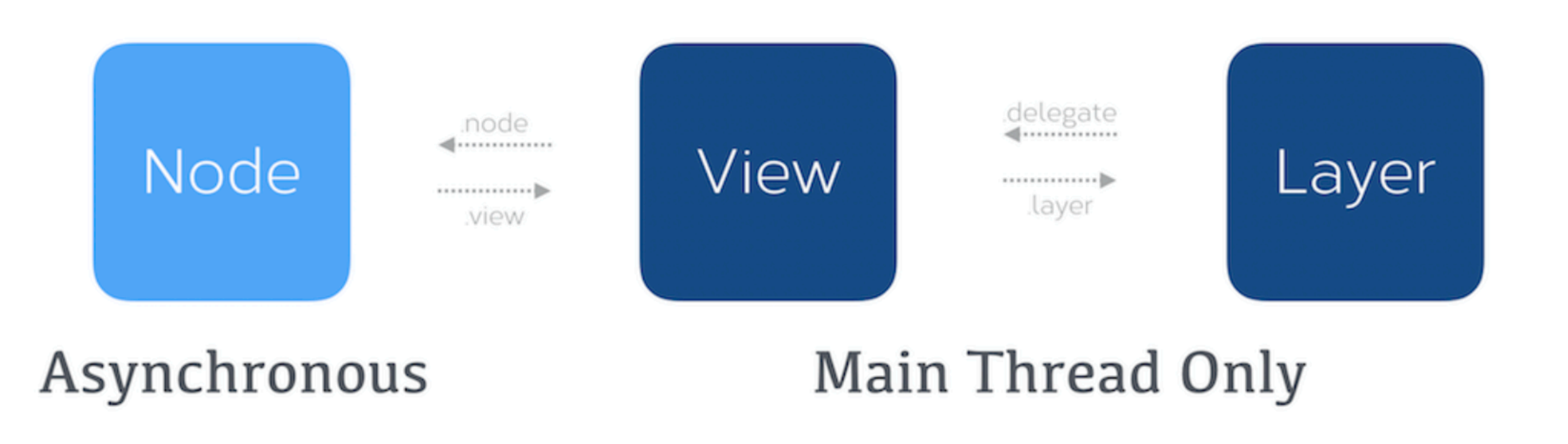
재사용이 용이하다 (모듈화)

반응, 랜더링이 빠르다 (내부적으로 설계가 잘되어있다 / Thread 안전성)

협업에 용이하다

가독성이 좋다 (FlexBox) / 다른 코드로 그리는 도구들에 비해

AppCode & Xcode 동시에 사용 가능하다


Node
.node
.view
View
.delegate
.layer
Layer
Asynchronous
Main Thread Only

단점 Texture 👀

러닝커브 .. ?

자료가 적다

문서나 예제에 objective C 가 많다

HOW TEXTURE ??

Show sample

Node

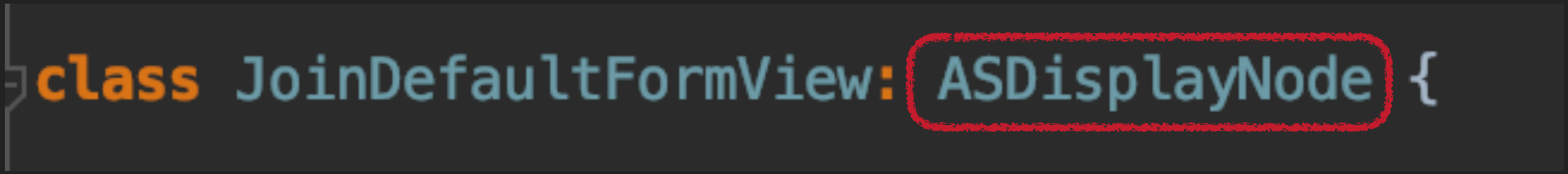
- ASDisplayNode
    - init (background thread)
    - didLoad (main thread)
    - layoutSpecThatFits (background thread)
    - layout

* background thread 에서 view & layer 접근에 주의

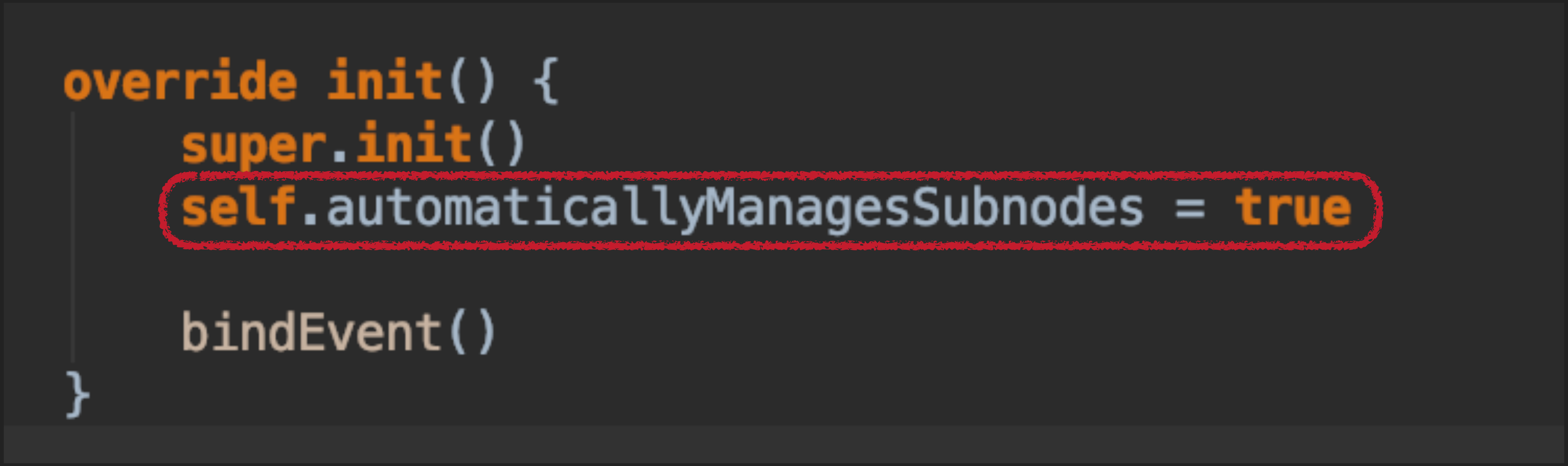
LayoutSpec

pod ‘Texture’ (in Podfile)

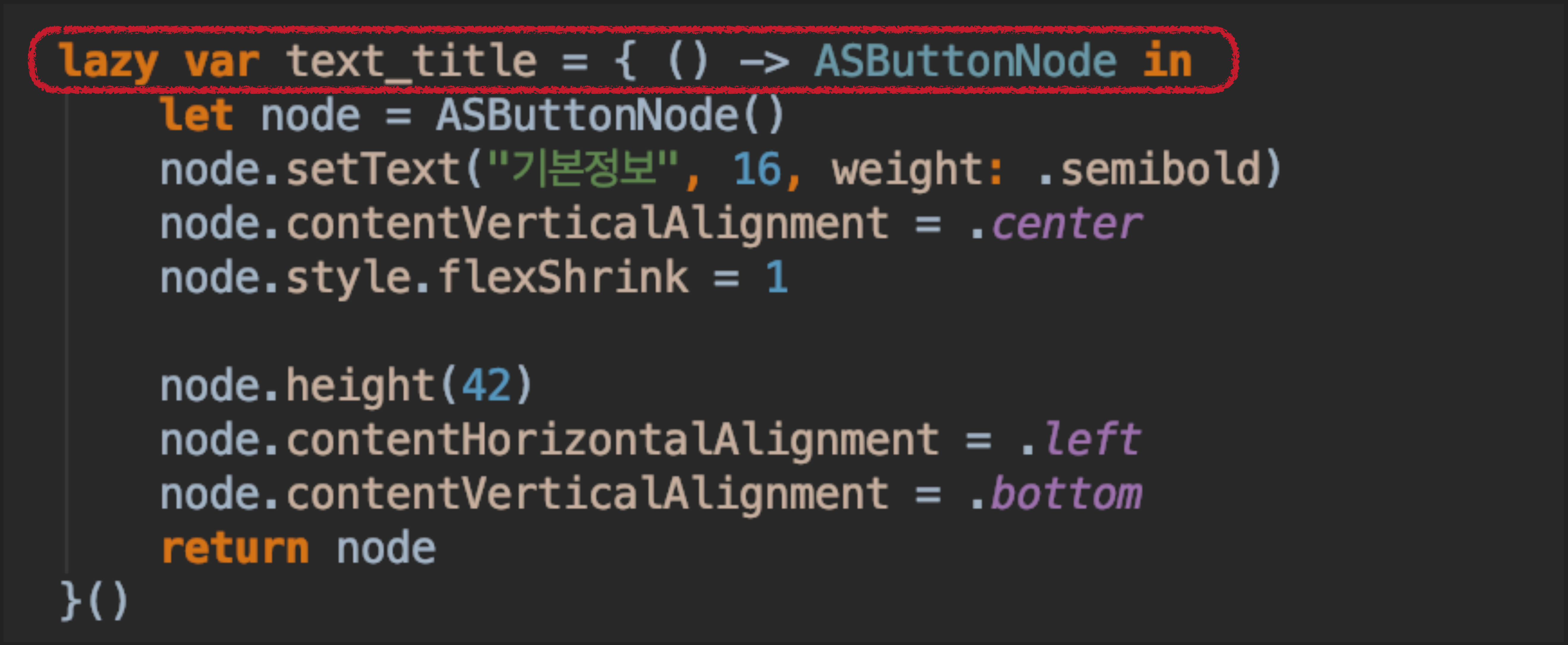
pod install



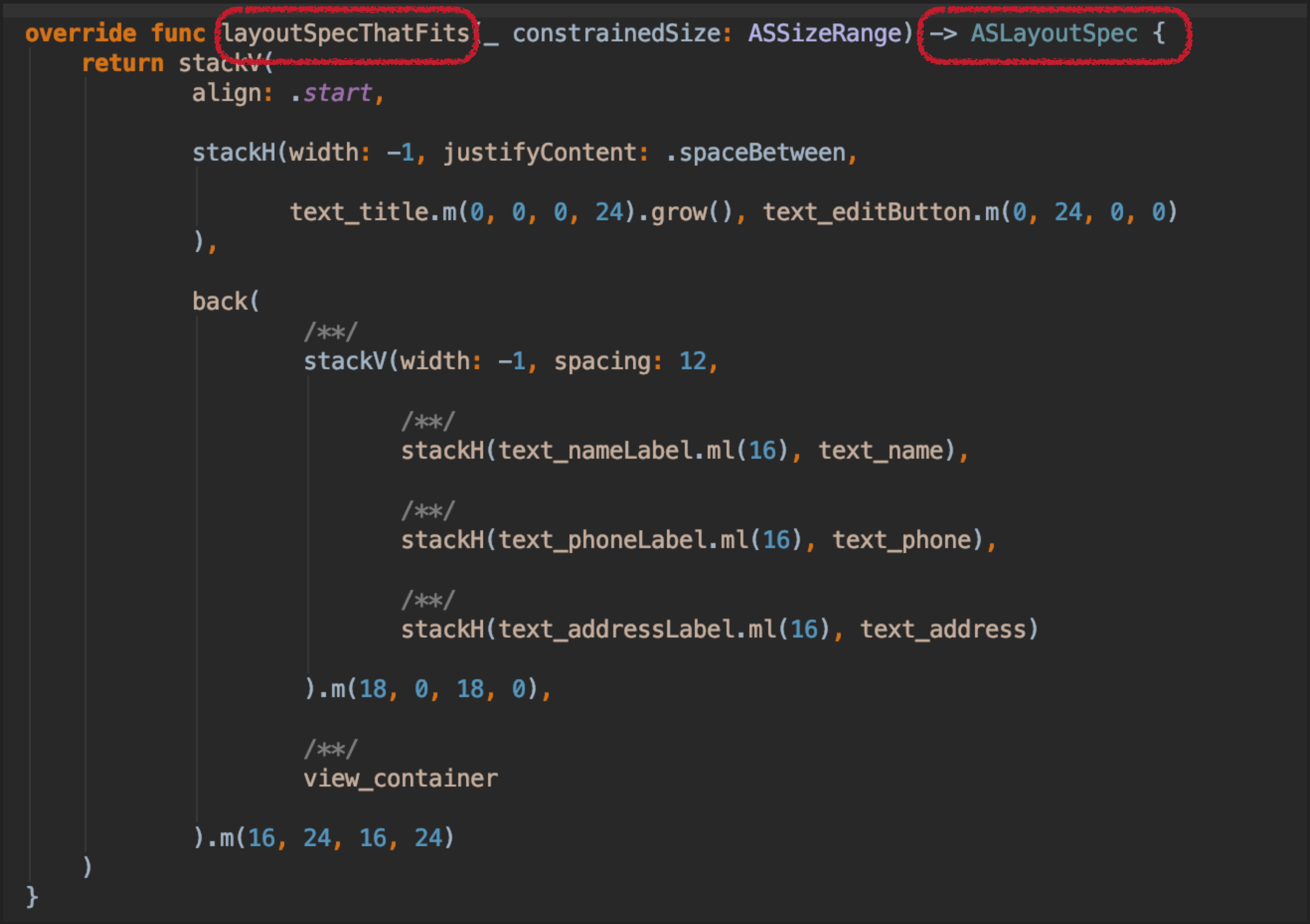
```
class JoinDefaultFormView: ASDisplayNode {
```



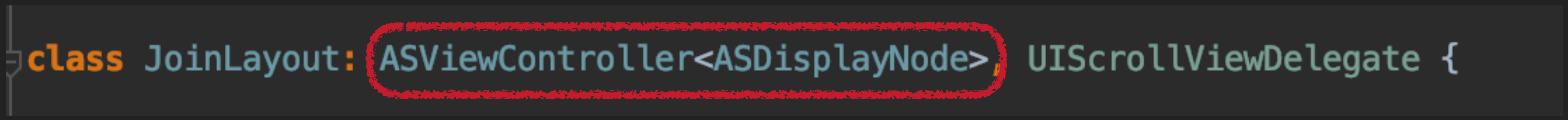
```
override init() {
    super.init()
    self.automaticallyManagesSubnodes = true

    bindEvent()
}
```



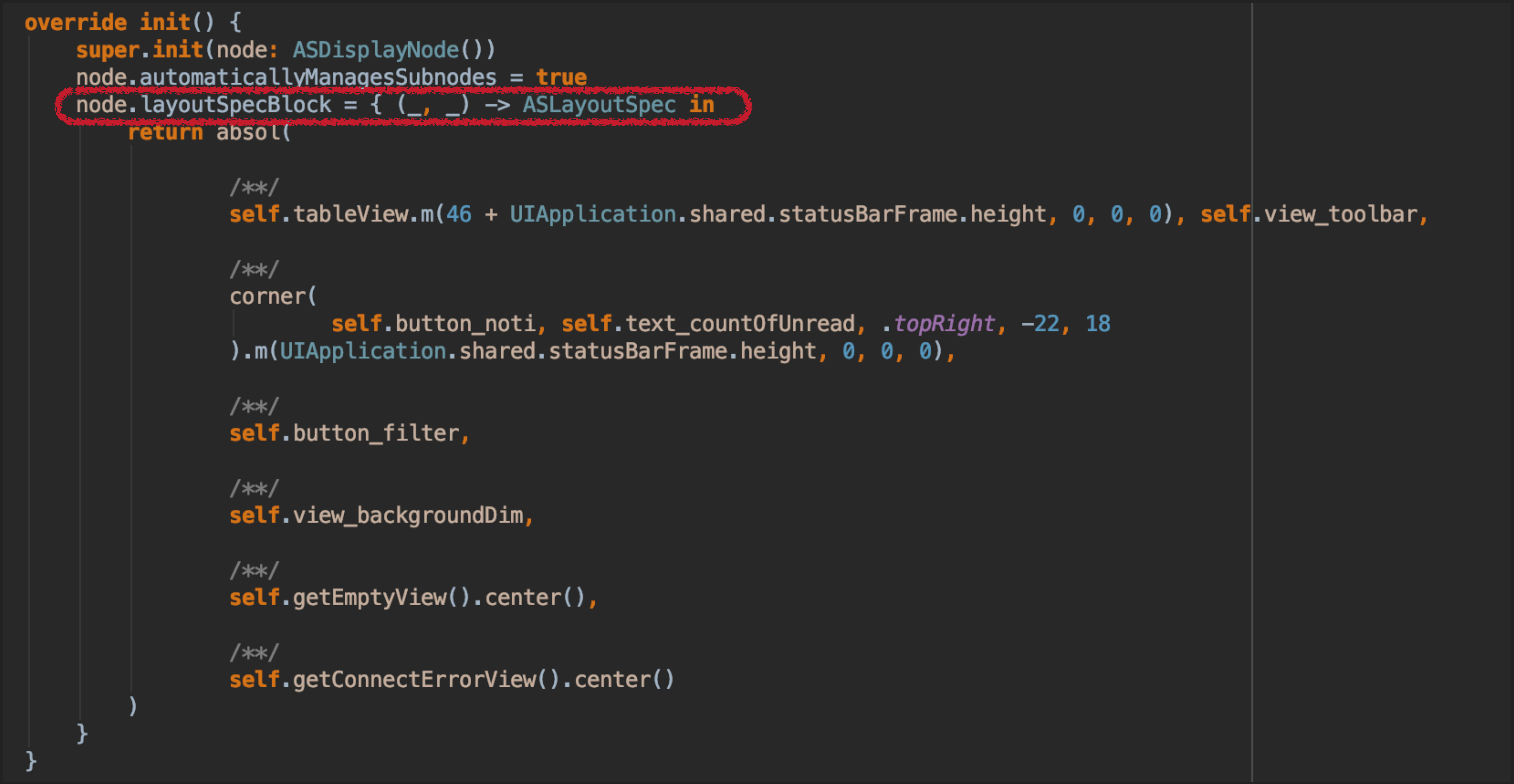
```swift
lazy var text_title = { () -> ASButtonNode in
    let node = ASButtonNode()
    node.setText("기본정보", 16, weight: .semibold)
    node.contentVerticalAlignment = .center
    node.style.flexShrink = 1

    node.height(42)
    node.contentHorizontalAlignment = .left
    node.contentVerticalAlignment = .bottom
    return node
}()
```



```
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start,

            stackH(width: -1, justifyContent: .spaceBetween,

                   text_title.m(0, 0, 0, 24).grow(), text_editButton.m(0, 24, 0, 0)
            ),

            back(
                    /**/
                    stackV(width: -1, spacing: 12,

                           /**/
                           stackH(text_nameLabel.ml(16), text_name),

                           /**/
                           stackH(text_phoneLabel.ml(16), text_phone),

                           /**/
                           stackH(text_addressLabel.ml(16), text_address)

                    ).m(18, 0, 18, 0),

                    /**/
                    view_container

            ).m(16, 24, 16, 24)
    )
}
```

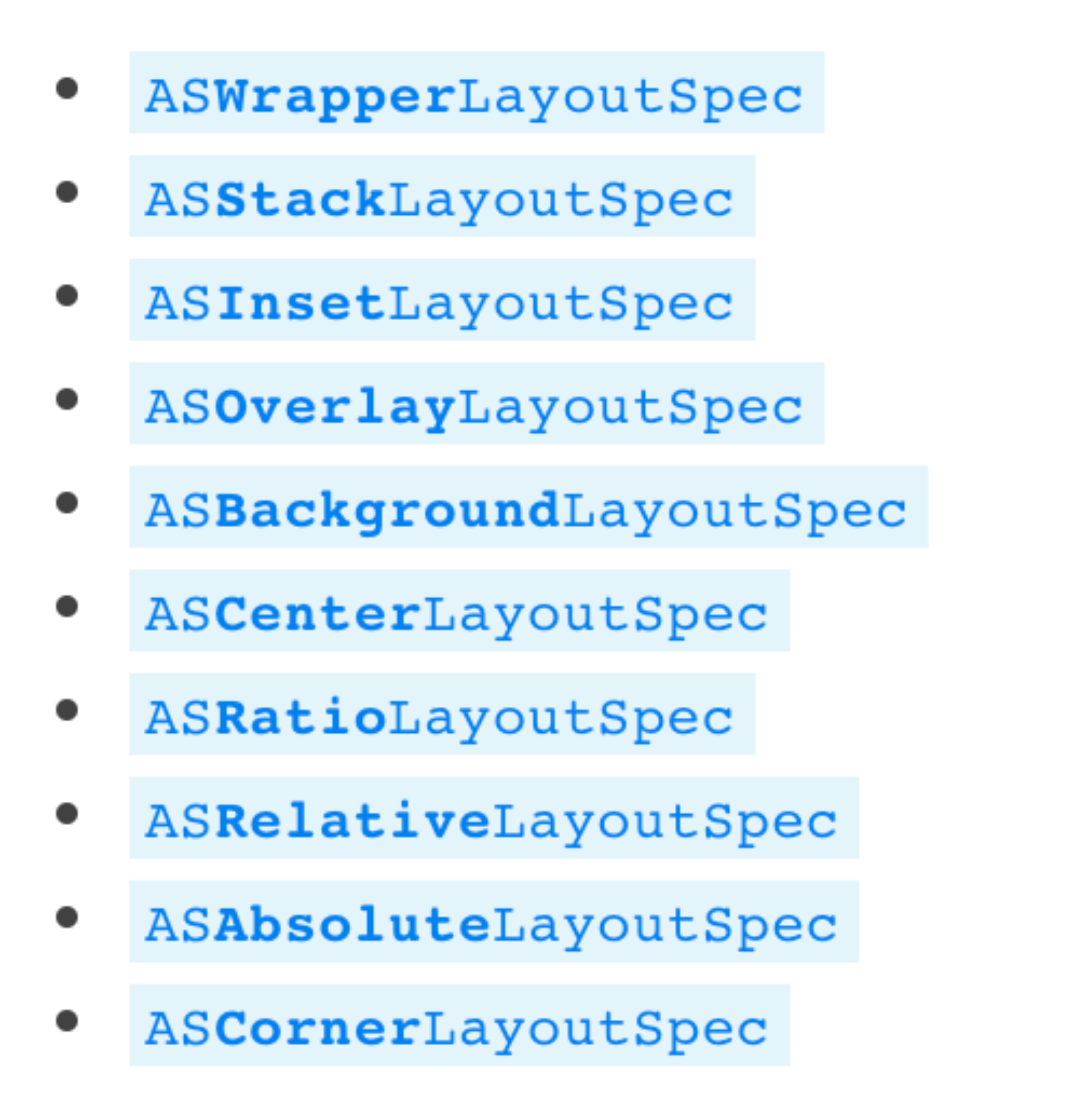


```
class JoinLayout: ASViewController<ASDisplayNode>, UIScrollViewDelegate {
```



```
override init() {
    super.init(node: ASDisplayNode())
    node.automaticallyManagesSubnodes = true
    node.layoutSpecBlock = { (_, _) -> ASLayoutSpec in
        return absol(

                /**/
                self.tableView.m(46 + UIApplication.shared.statusBarFrame.height, 0, 0, 0), self.view_toolbar,

                /**/
                corner(
                        self.button_noti, self.text_countOfUnread, .topRight, -22, 18
                ).m(UIApplication.shared.statusBarFrame.height, 0, 0, 0),

                /**/
                self.button_filter,

                /**/
                self.view_backgroundDim,

                /**/
                self.getEmptyView().center(),

                /**/
                self.getConnectErrorView().center()
        )
    }
}
```

LayoutSpec

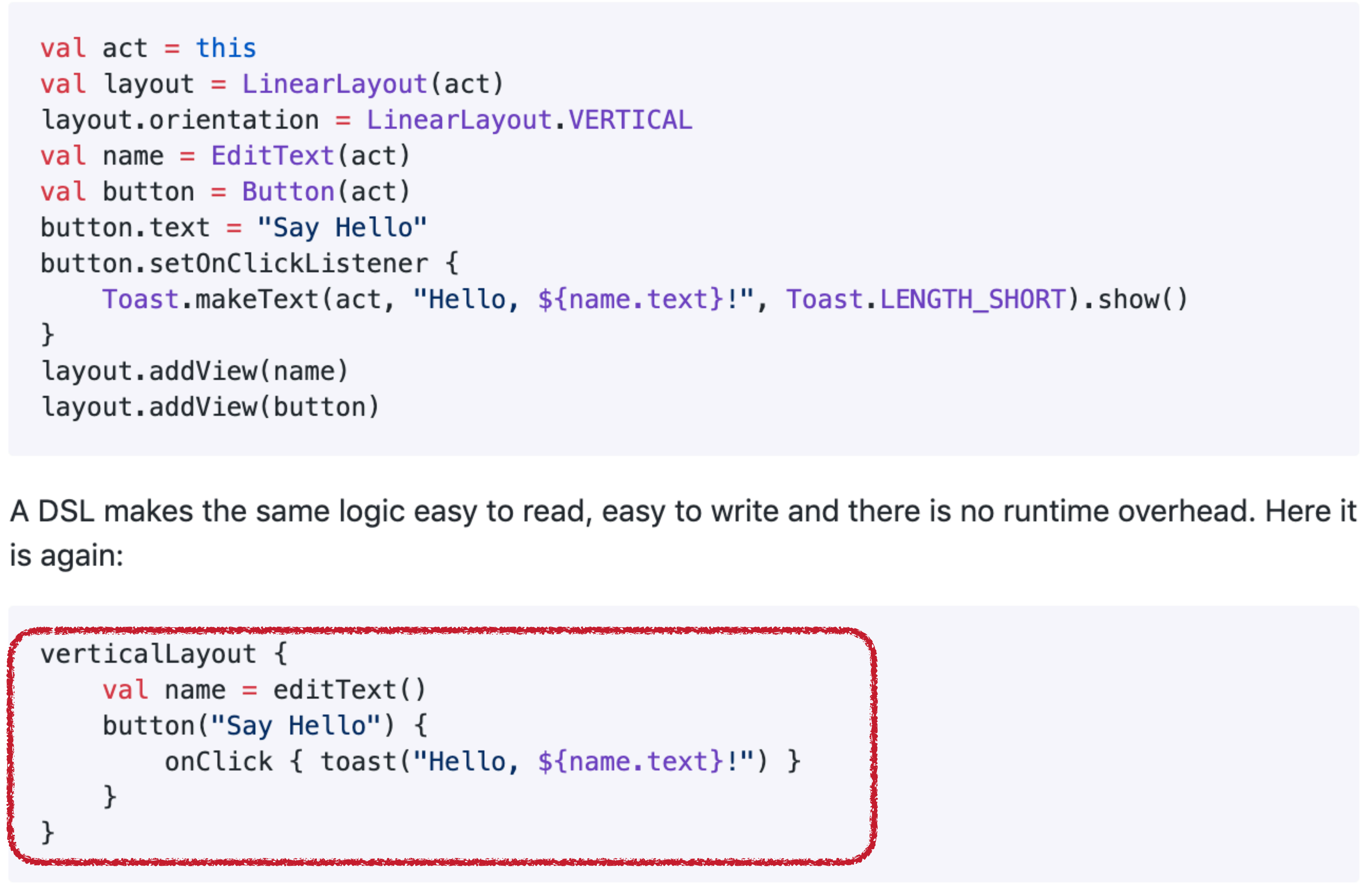


- `ASWrapperLayoutSpec`
- `ASStackLayoutSpec`
- `ASInsetLayoutSpec`
- `ASOverlayLayoutSpec`
- `ASBackgroundLayoutSpec`
- `ASCenterLayoutSpec`
- `ASRatioLayoutSpec`
- `ASRelativeLayoutSpec`
- `ASAbsoluteLayoutSpec`
- `ASCornerLayoutSpec`

가독성

코드는 읽기 쉬워야 한다

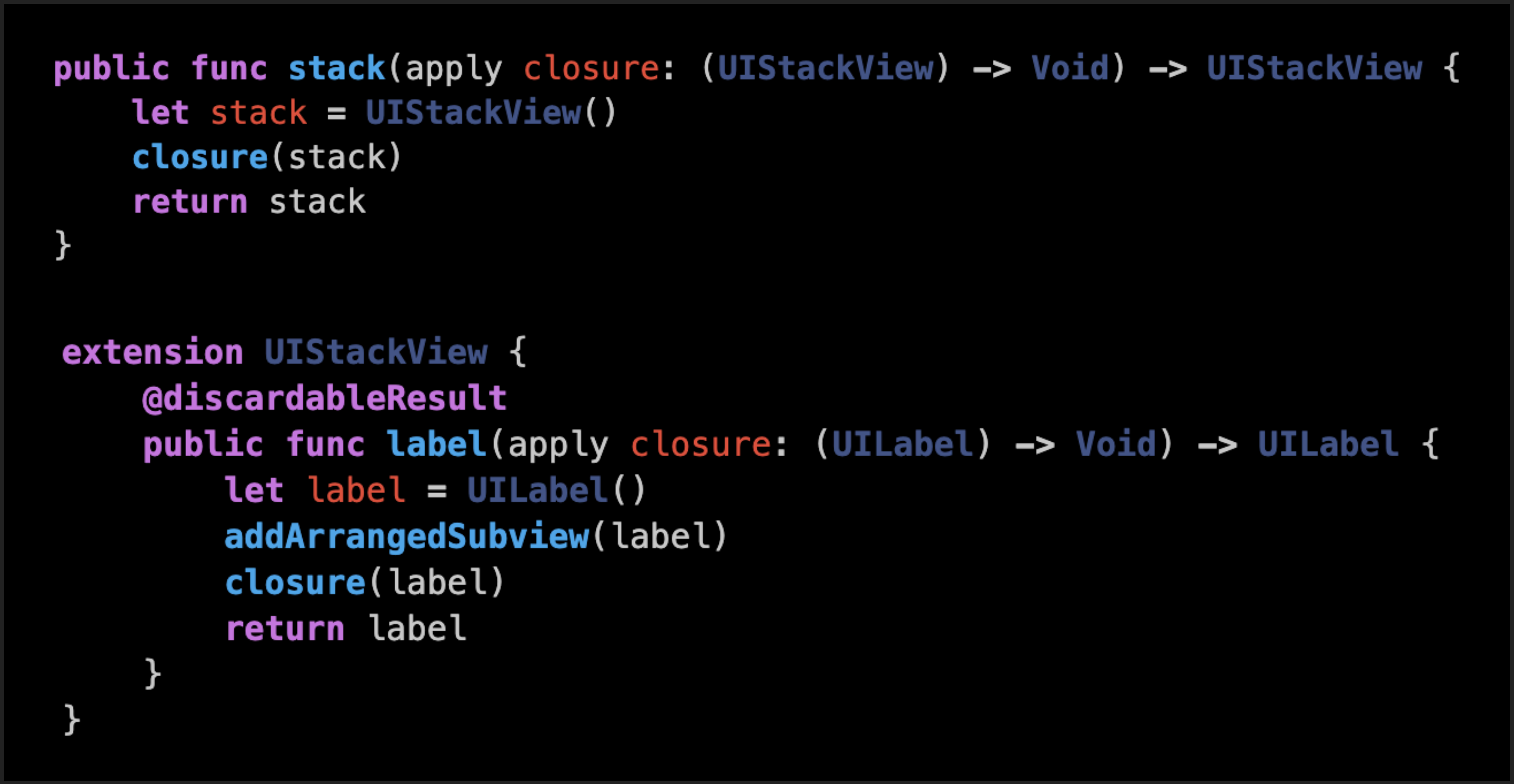
Anko / DSL



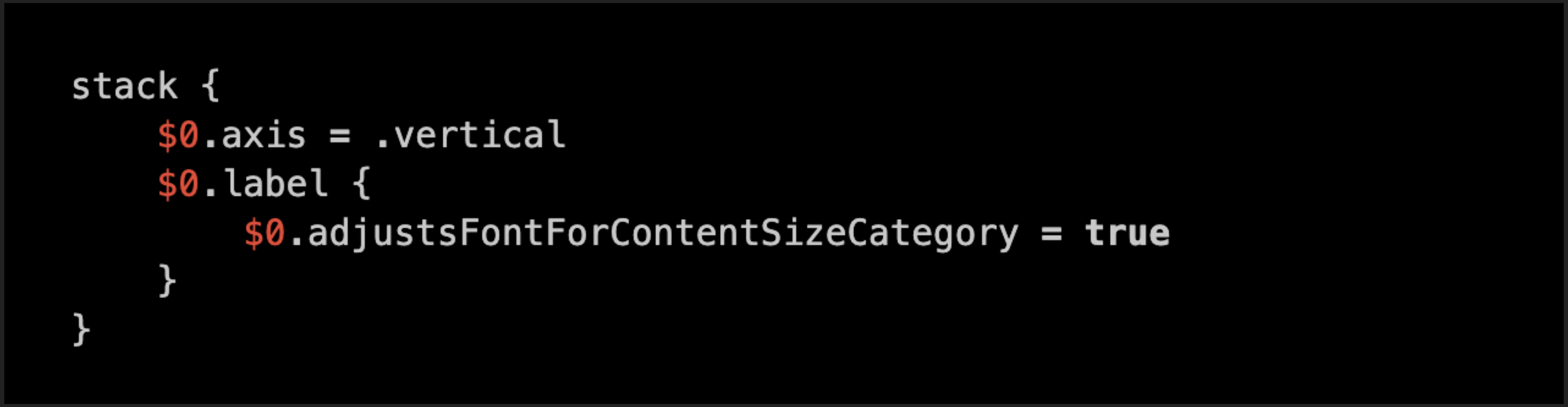
```
val act = this
val layout = LinearLayout(act)
layout.orientation = LinearLayout.VERTICAL
val name = EditText(act)
val button = Button(act)
button.text = "Say Hello"
button.setOnClickListener {
    Toast.makeText(act, "Hello, ${name.text}!", Toast.LENGTH_SHORT).show()
}
layout.addView(name)
layout.addView(button)
```

A DSL makes the same logic easy to read, easy to write and there is no runtime overhead. Here it is again:

```
verticalLayout {
    val name = editText()
    button("Say Hello") {
        onClick { toast("Hello, ${name.text}!") }
    }
}
```



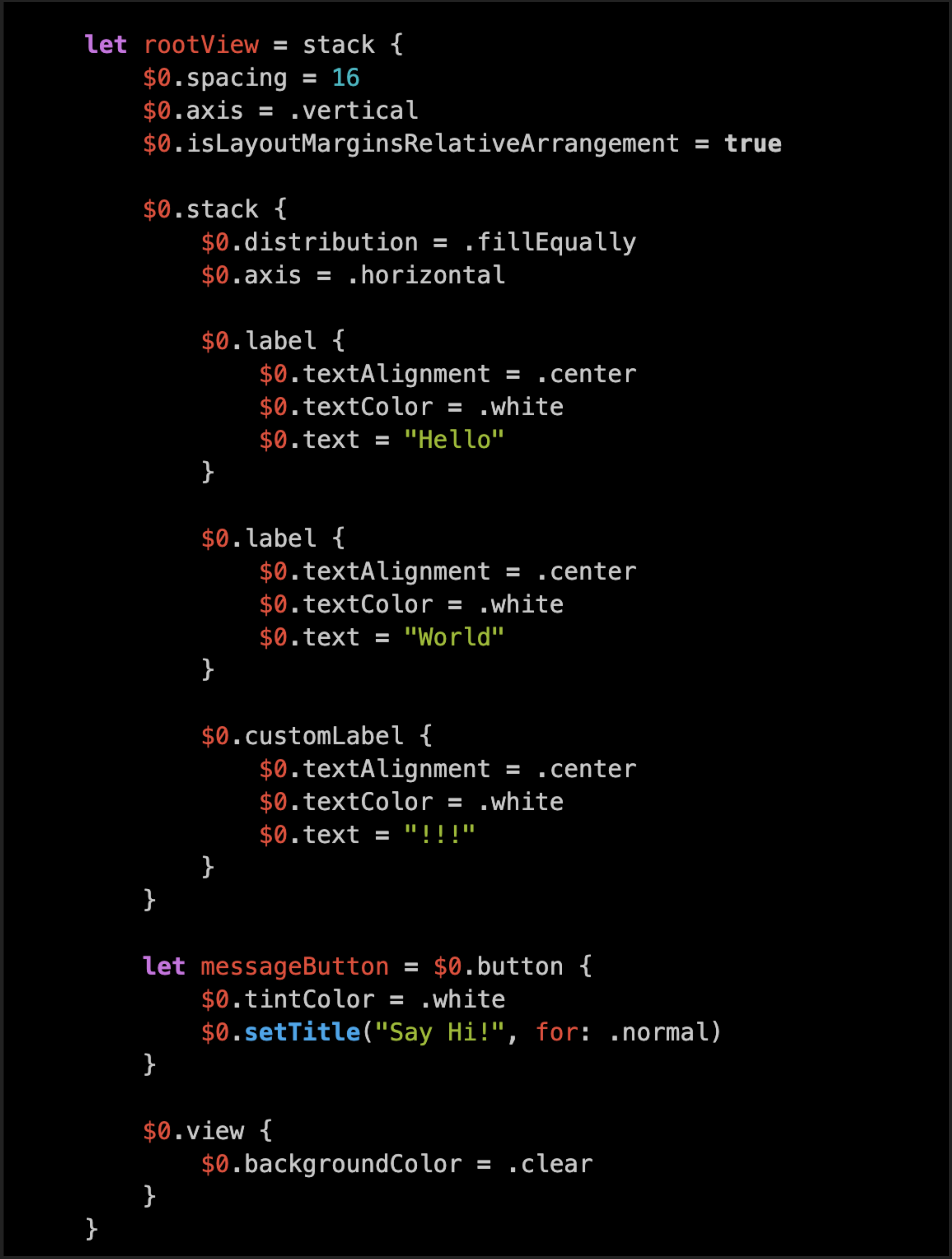
```swift
public func stack(apply closure: (UIStackView) -> Void) -> UIStackView {
    let stack = UIStackView()
    closure(stack)
    return stack
}

extension UIStackView {
    @discardableResult
    public func label(apply closure: (UILabel) -> Void) -> UILabel {
        let label = UILabel()
        addArrangedSubview(label)
        closure(label)
        return label
    }
}
```



```swift
stack {
    $0.axis = .vertical
    $0.label {
        $0.adjustsFontForContentSizeCategory = true
    }
}
```

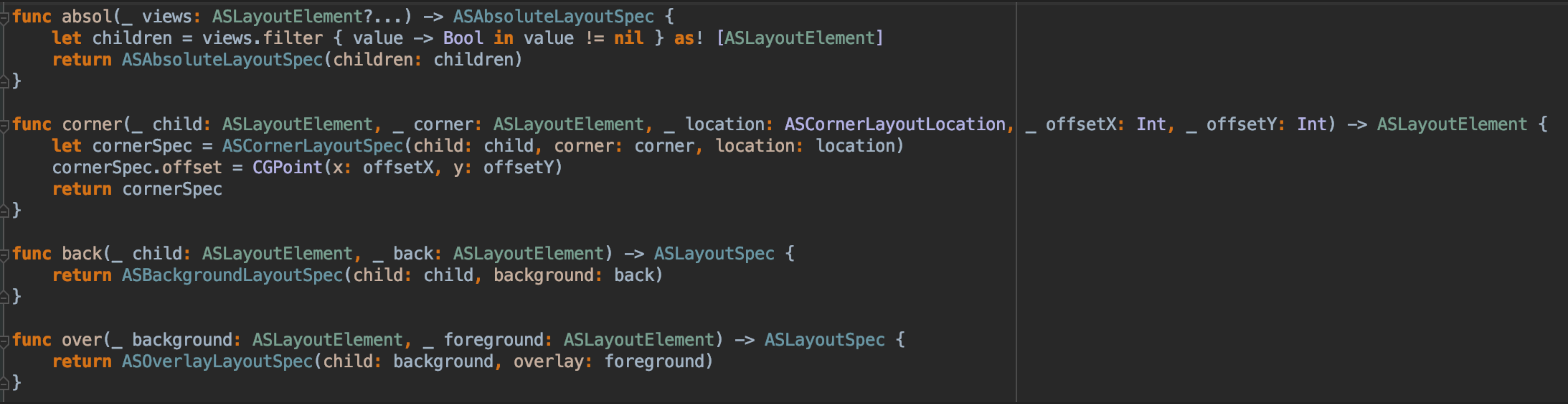
Swift / DSL

UIKit

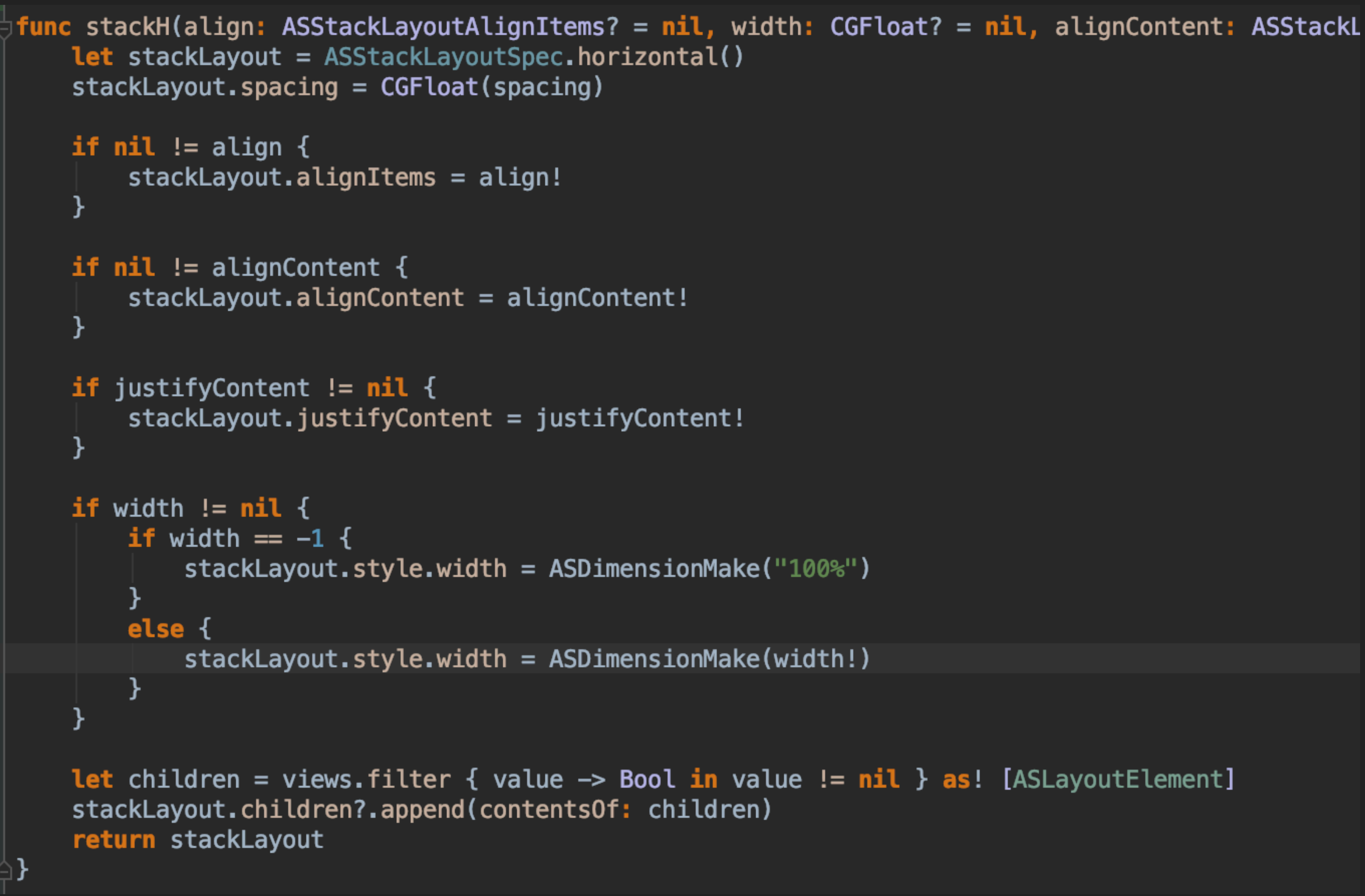


```swift
let rootView = stack {
    $0.spacing = 16
    $0.axis = .vertical
    $0.isLayoutMarginsRelativeArrangement = true

    $0.stack {
        $0.distribution = .fillEqually
        $0.axis = .horizontal

        $0.label {
            $0.textAlignment = .center
            $0.textColor = .white
            $0.text = "Hello"
        }

        $0.label {
            $0.textAlignment = .center
            $0.textColor = .white
            $0.text = "World"
        }

        $0.customLabel {
            $0.textAlignment = .center
            $0.textColor = .white
            $0.text = "!!!"
        }
    }

    let messageButton = $0.button {
        $0.tintColor = .white
        $0.setTitle("Say Hi!", for: .normal)
    }

    $0.view {
        $0.backgroundColor = .clear
    }
}
```

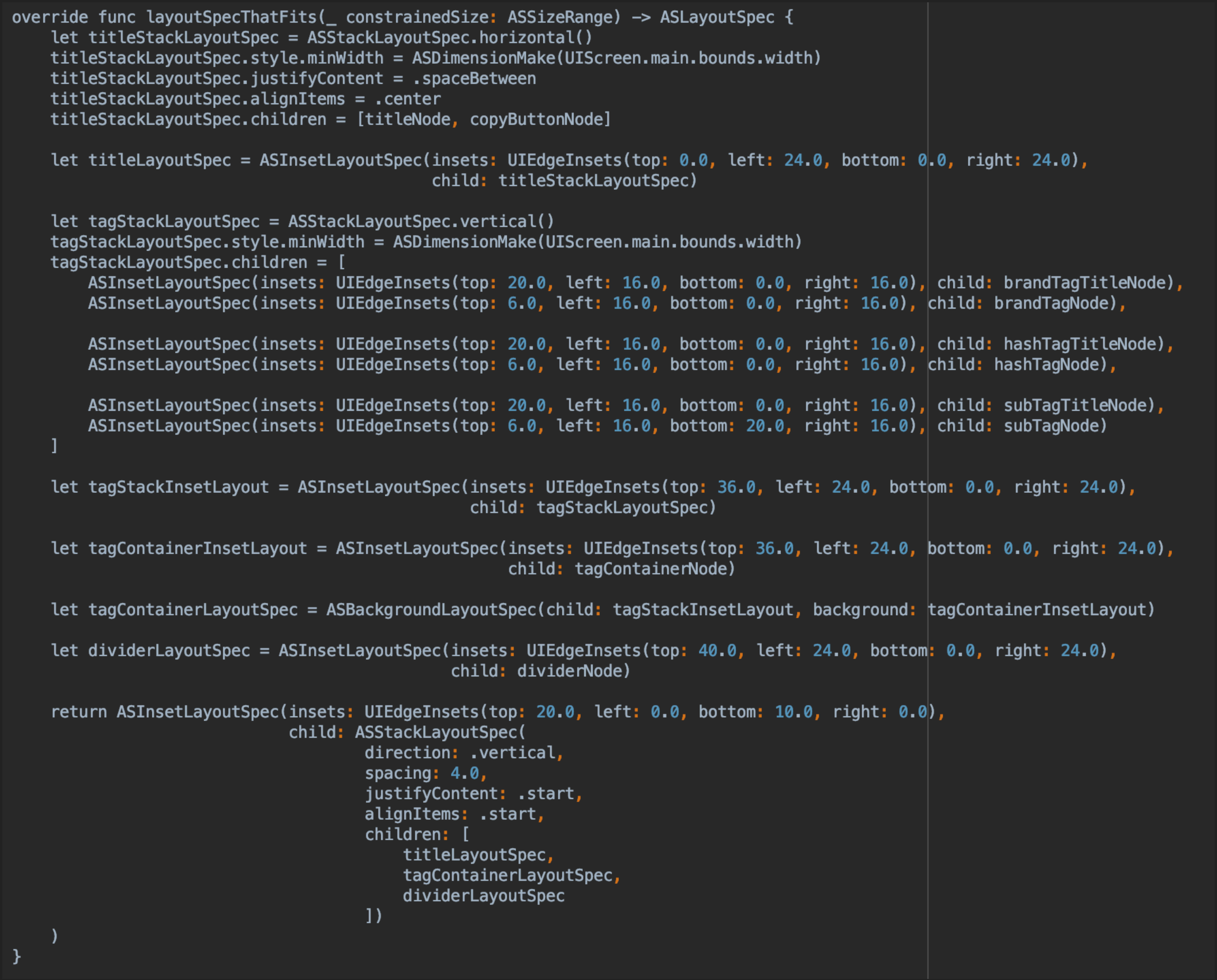
Swift



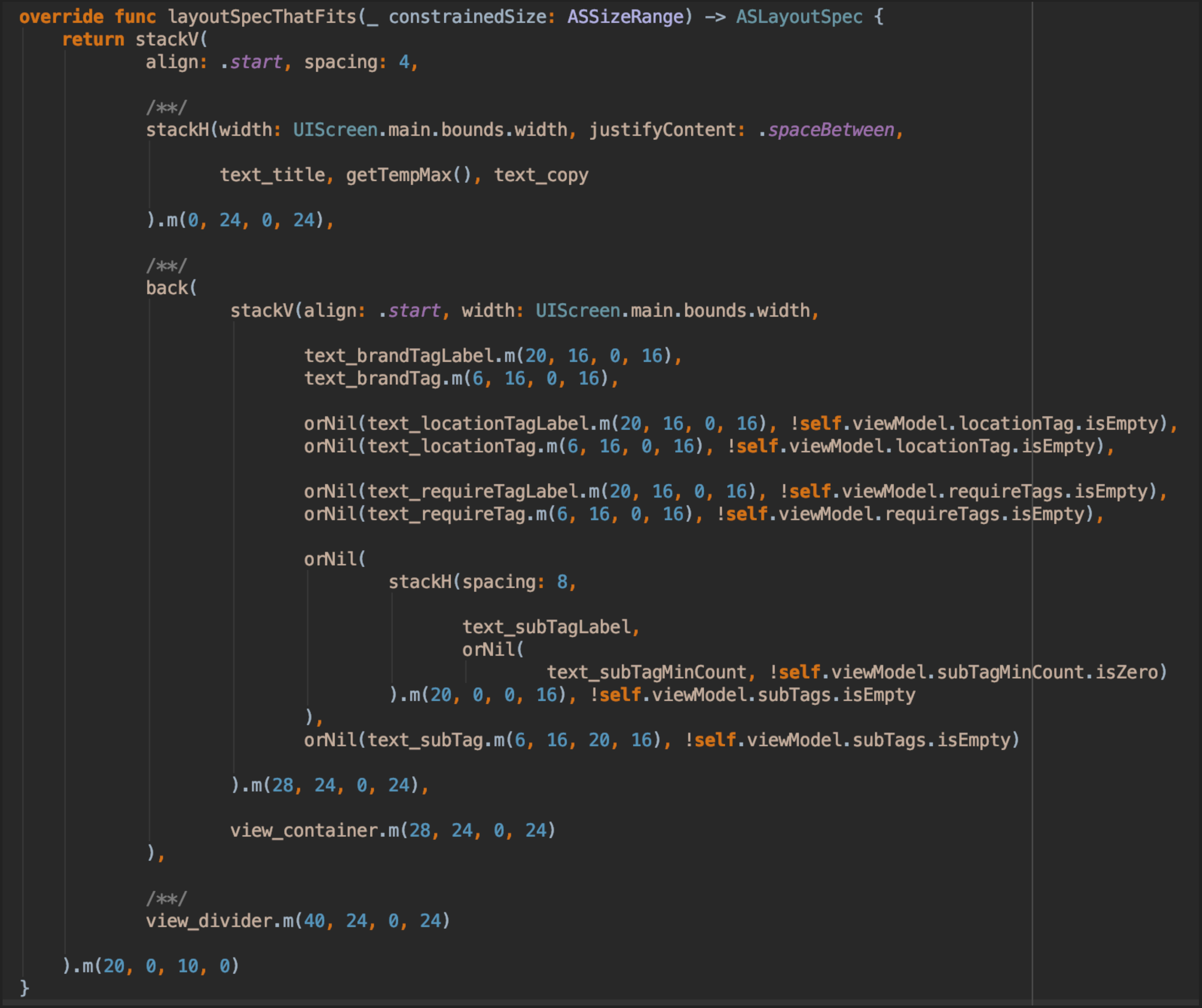
```swift
func absol(_ views: ASLayoutElement?...) -> ASAbsoluteLayoutSpec {
    let children = views.filter { value -> Bool in value != nil } as! [ASLayoutElement]
    return ASAbsoluteLayoutSpec(children: children)
}

func corner(_ child: ASLayoutElement, _ corner: ASLayoutElement, _ location: ASCornerLayoutLocation, _ offsetX: Int, _ offsetY: Int) -> ASLayoutElement {
    let cornerSpec = ASCornerLayoutSpec(child: child, corner: corner, location: location)
    cornerSpec.offset = CGPoint(x: offsetX, y: offsetY)
    return cornerSpec
}

func back(_ child: ASLayoutElement, _ back: ASLayoutElement) -> ASLayoutSpec {
    return ASBackgroundLayoutSpec(child: child, background: back)
}

func over(_ background: ASLayoutElement, _ foreground: ASLayoutElement) -> ASLayoutSpec {
    return ASOverlayLayoutSpec(child: background, overlay: foreground)
}
```



```swift
func stackH(align: ASStackLayoutAlignItems? = nil, width: CGFloat? = nil, alignContent: ASStackL
    let stackLayout = ASStackLayoutSpec.horizontal()
    stackLayout.spacing = CGFloat(spacing)

    if nil != align {
        stackLayout.alignItems = align!
    }

    if nil != alignContent {
        stackLayout.alignContent = alignContent!
    }

    if justifyContent != nil {
        stackLayout.justifyContent = justifyContent!
    }

    if width != nil {
        if width == -1 {
            stackLayout.style.width = ASDimensionMake("100%")
        }
        else {
            stackLayout.style.width = ASDimensionMake(width!)
        }
    }

    let children = views.filter { value -> Bool in value != nil } as! [ASLayoutElement]
    stackLayout.children?.append(contentsOf: children)
    return stackLayout
}
```

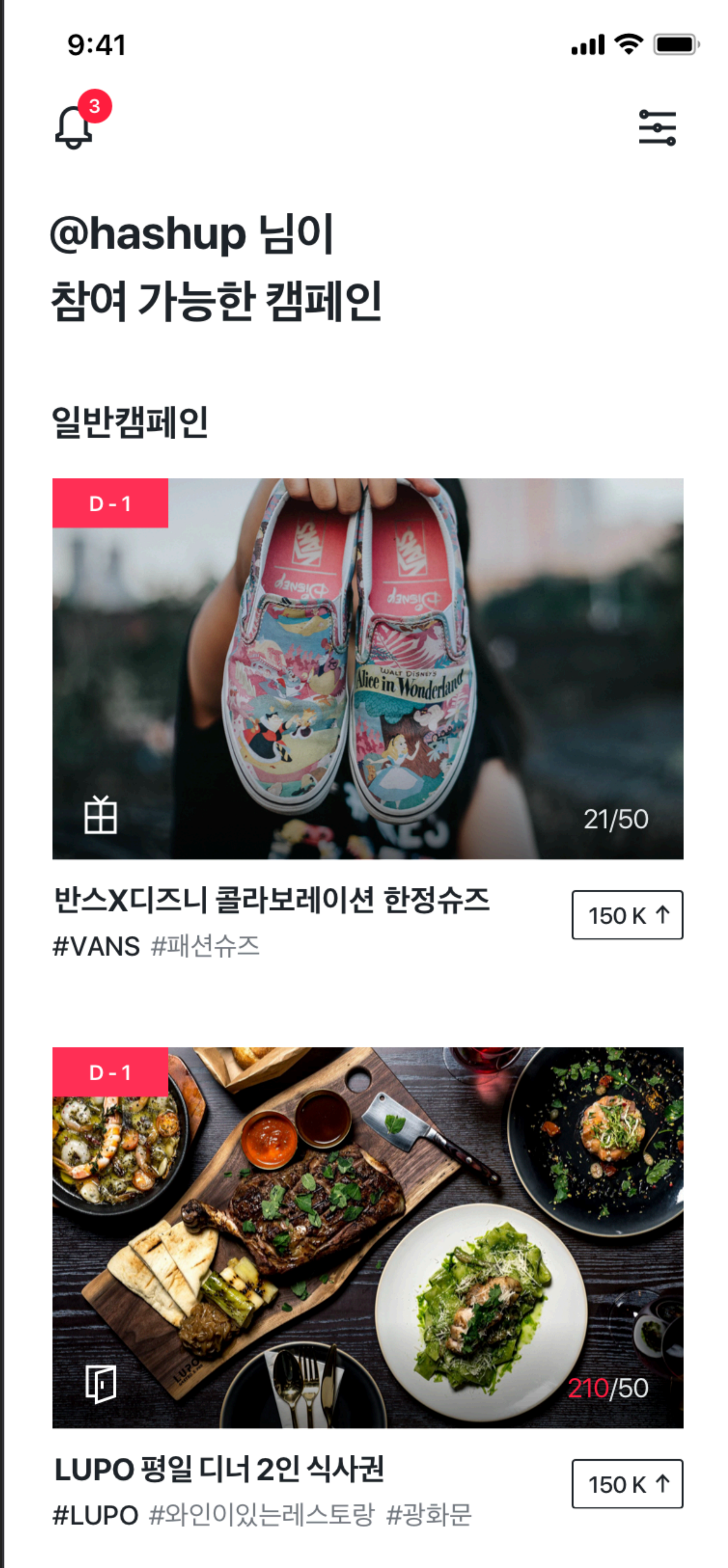


```
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    let titleStackLayoutSpec = ASStackLayoutSpec.horizontal()
    titleStackLayoutSpec.style.minWidth = ASDimensionMake(UIScreen.main.bounds.width)
    titleStackLayoutSpec.justifyContent = .spaceBetween
    titleStackLayoutSpec.alignItems = .center
    titleStackLayoutSpec.children = [titleNode, copyButtonNode]

    let titleLayoutSpec = ASInsetLayoutSpec(insets: UIEdgeInsets(top: 0.0, left: 24.0, bottom: 0.0, right: 24.0),
                                            child: titleStackLayoutSpec)

    let tagStackLayoutSpec = ASStackLayoutSpec.vertical()
    tagStackLayoutSpec.style.minWidth = ASDimensionMake(UIScreen.main.bounds.width)
    tagStackLayoutSpec.children = [
        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 20.0, left: 16.0, bottom: 0.0, right: 16.0), child: brandTagTitleNode),
        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 6.0, left: 16.0, bottom: 0.0, right: 16.0), child: brandTagNode),

        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 20.0, left: 16.0, bottom: 0.0, right: 16.0), child: hashTagTitleNode),
        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 6.0, left: 16.0, bottom: 0.0, right: 16.0), child: hashTagNode),

        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 20.0, left: 16.0, bottom: 0.0, right: 16.0), child: subTagTitleNode),
        ASInsetLayoutSpec(insets: UIEdgeInsets(top: 6.0, left: 16.0, bottom: 20.0, right: 16.0), child: subTagNode)
    ]

    let tagStackInsetLayout = ASInsetLayoutSpec(insets: UIEdgeInsets(top: 36.0, left: 24.0, bottom: 0.0, right: 24.0),
                                                child: tagStackLayoutSpec)

    let tagContainerInsetLayout = ASInsetLayoutSpec(insets: UIEdgeInsets(top: 36.0, left: 24.0, bottom: 0.0, right: 24.0),
                                                    child: tagContainerNode)

    let tagContainerLayoutSpec = ASBackgroundLayoutSpec(child: tagStackInsetLayout, background: tagContainerInsetLayout)

    let dividerLayoutSpec = ASInsetLayoutSpec(insets: UIEdgeInsets(top: 40.0, left: 24.0, bottom: 0.0, right: 24.0),
                                              child: dividerNode)

    return ASInsetLayoutSpec(insets: UIEdgeInsets(top: 20.0, left: 0.0, bottom: 10.0, right: 0.0),
                             child: ASStackLayoutSpec(
                                     direction: .vertical,
                                     spacing: 4.0,
                                     justifyContent: .start,
                                     alignItems: .start,
                                     children: [
                                         titleLayoutSpec,
                                         tagContainerLayoutSpec,
                                         dividerLayoutSpec
                                     ])
    )
}
```



```
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start, spacing: 4,

            /**/
            stackH(width: UIScreen.main.bounds.width, justifyContent: .spaceBetween,

                   text_title, getTempMax(), text_copy

            ).m(0, 24, 0, 24),

            /**/
            back(
                    stackV(align: .start, width: UIScreen.main.bounds.width,

                           text_brandTagLabel.m(20, 16, 0, 16),
                           text_brandTag.m(6, 16, 0, 16),

                           orNil(text_locationTagLabel.m(20, 16, 0, 16), !self.viewModel.locationTag.isEmpty),
                           orNil(text_locationTag.m(6, 16, 0, 16), !self.viewModel.locationTag.isEmpty),

                           orNil(text_requireTagLabel.m(20, 16, 0, 16), !self.viewModel.requireTags.isEmpty),
                           orNil(text_requireTag.m(6, 16, 0, 16), !self.viewModel.requireTags.isEmpty),

                           orNil(
                                   stackH(spacing: 8,

                                          text_subTagLabel,
                                          orNil(
                                                  text_subTagMinCount, !self.viewModel.subTagMinCount.isZero)
                                   ).m(20, 0, 0, 16), !self.viewModel.subTags.isEmpty
                           ),
                           orNil(text_subTag.m(6, 16, 20, 16), !self.viewModel.subTags.isEmpty)

                    ).m(28, 24, 0, 24),

                    view_container.m(28, 24, 0, 24)
            ),

            /**/
            view_divider.m(40, 24, 0, 24)

    ).m(20, 0, 10, 0)
}
```

Example



@hashup 님이 참여 가능한 캠페인

일반캠페인

반스X디즈니 콜라보레이션 한정슈즈

150 K ↑

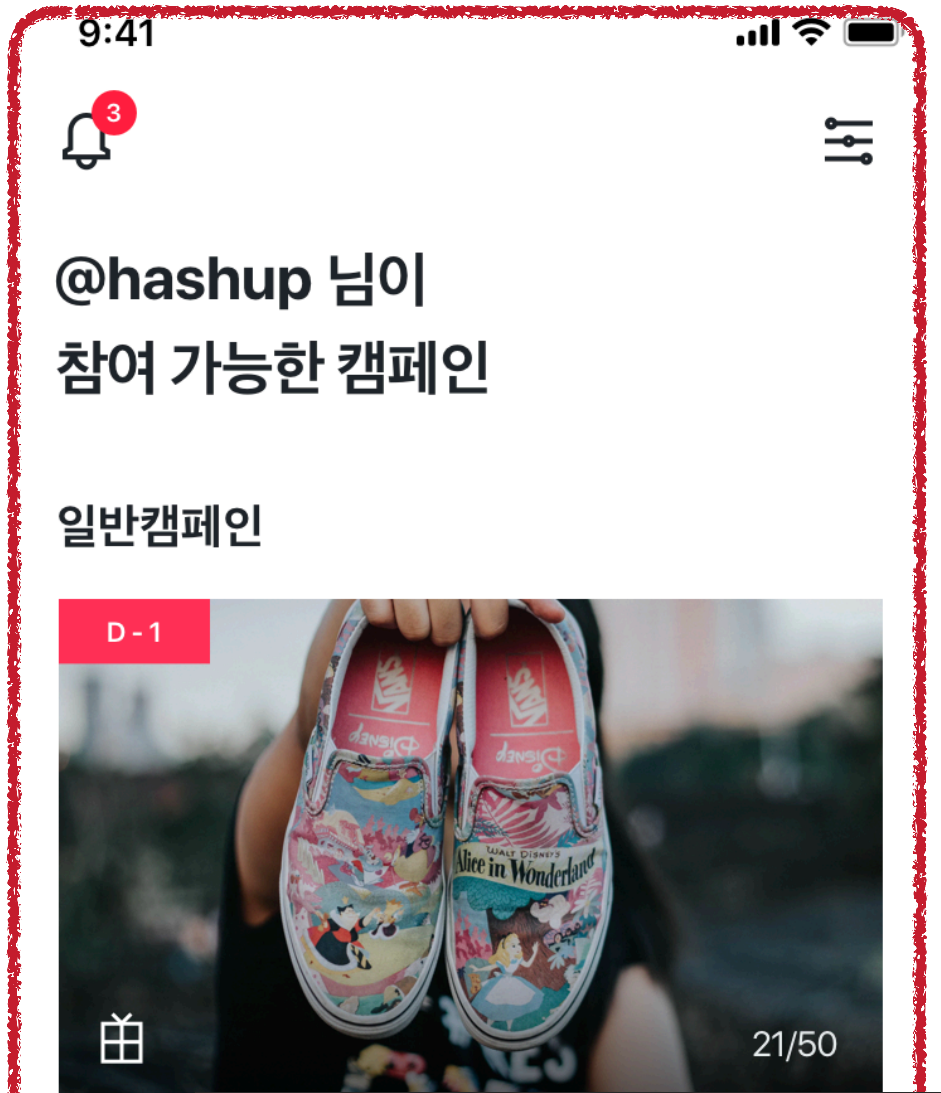
#VANS #패션슈즈

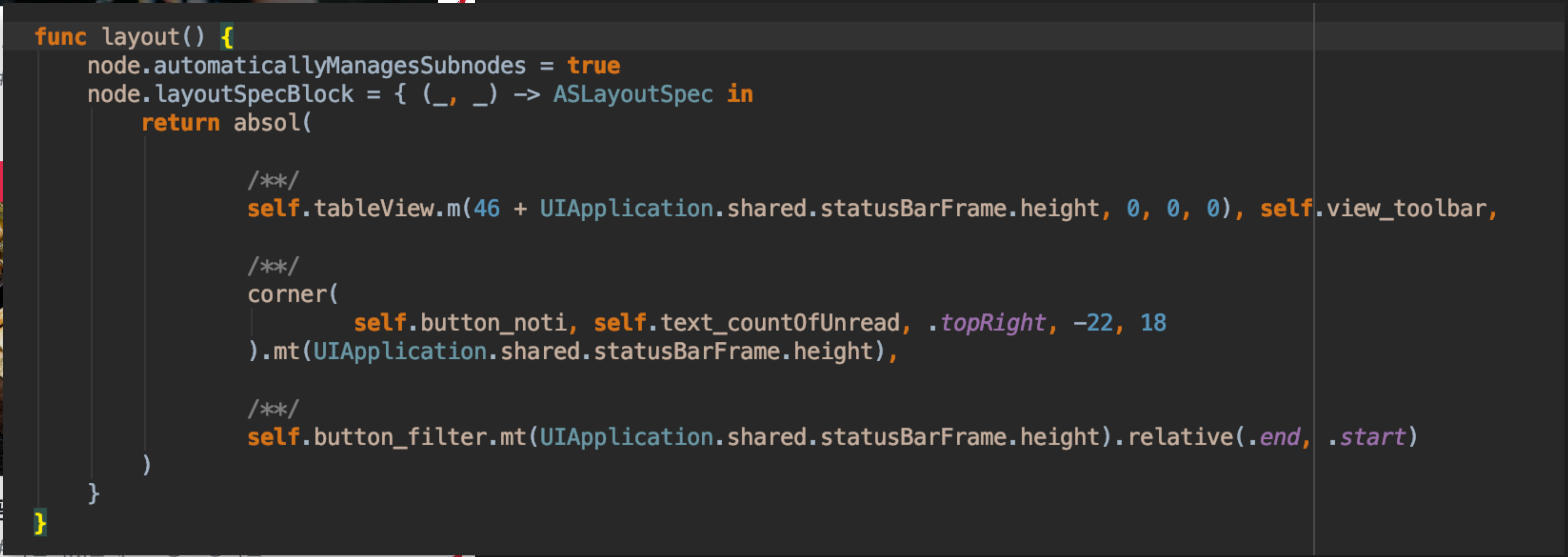
LUPO 평일 디너 2인 식사권

150 K ↑

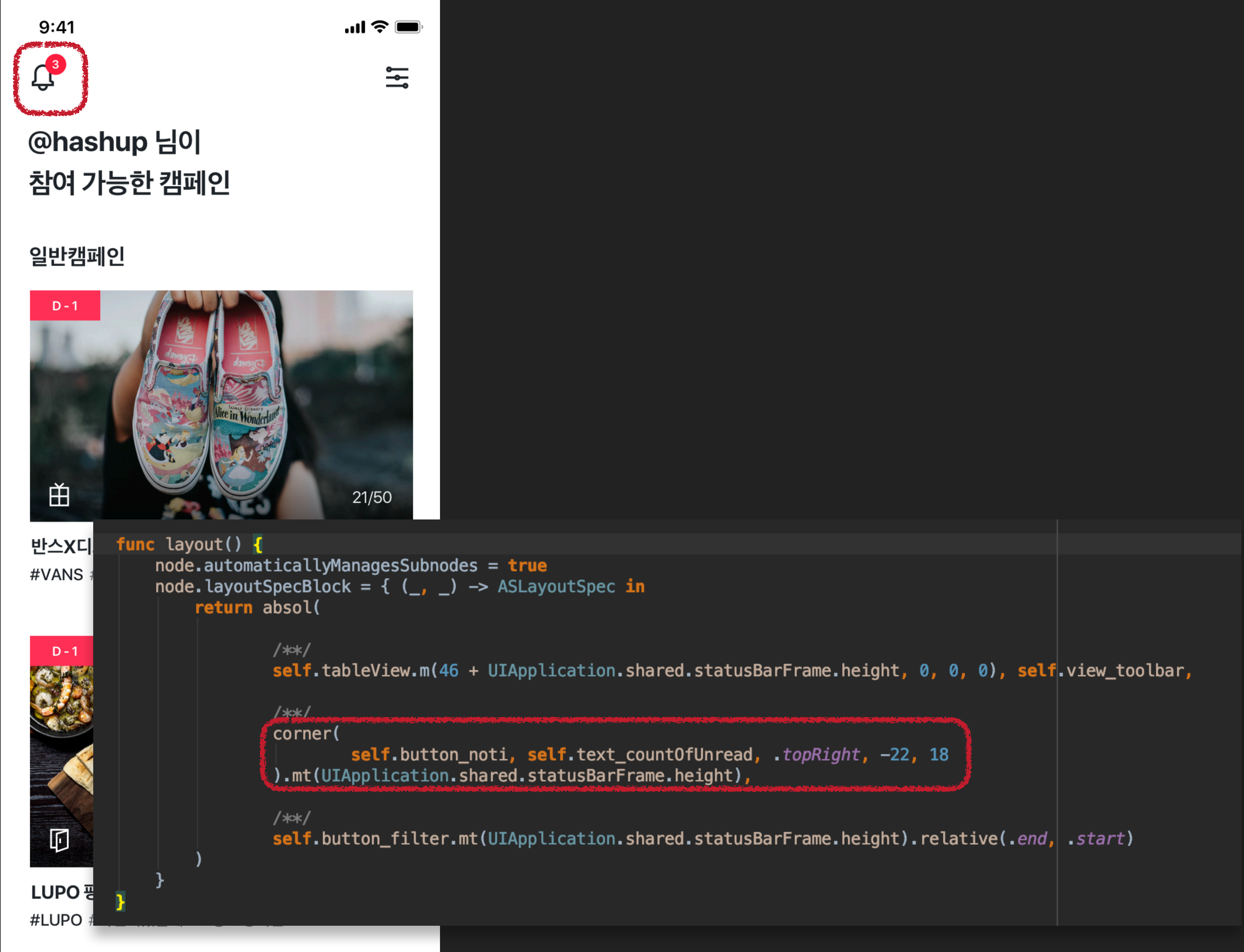
#LUPO #와인이있는레스토랑 #광화문

ASViewController

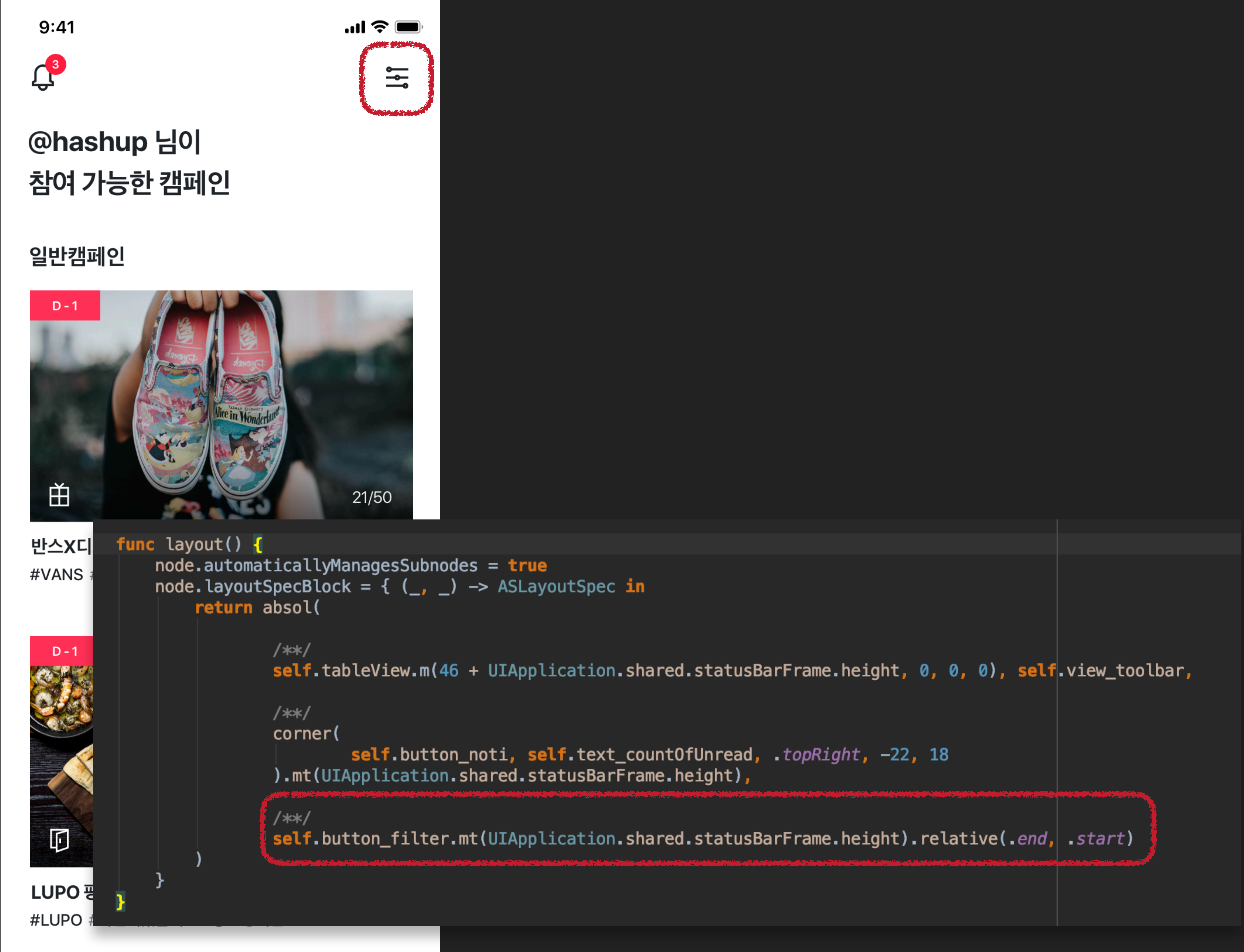




```
func layout() {
    node.automaticallyManagesSubnodes = true
    node.layoutSpecBlock = { (_, _) -> ASLayoutSpec in
        return absol(

                /**/
                self.tableView.m(46 + UIApplication.shared.statusBarFrame.height, 0, 0, 0), self.view_toolbar,

                /**/
                corner(
                        self.button_noti, self.text_countOfUnread, .topRight, -22, 18
                ).mt(UIApplication.shared.statusBarFrame.height),

                /**/
                self.button_filter.mt(UIApplication.shared.statusBarFrame.height).relative(.end, .start)
        )
    }
}
```



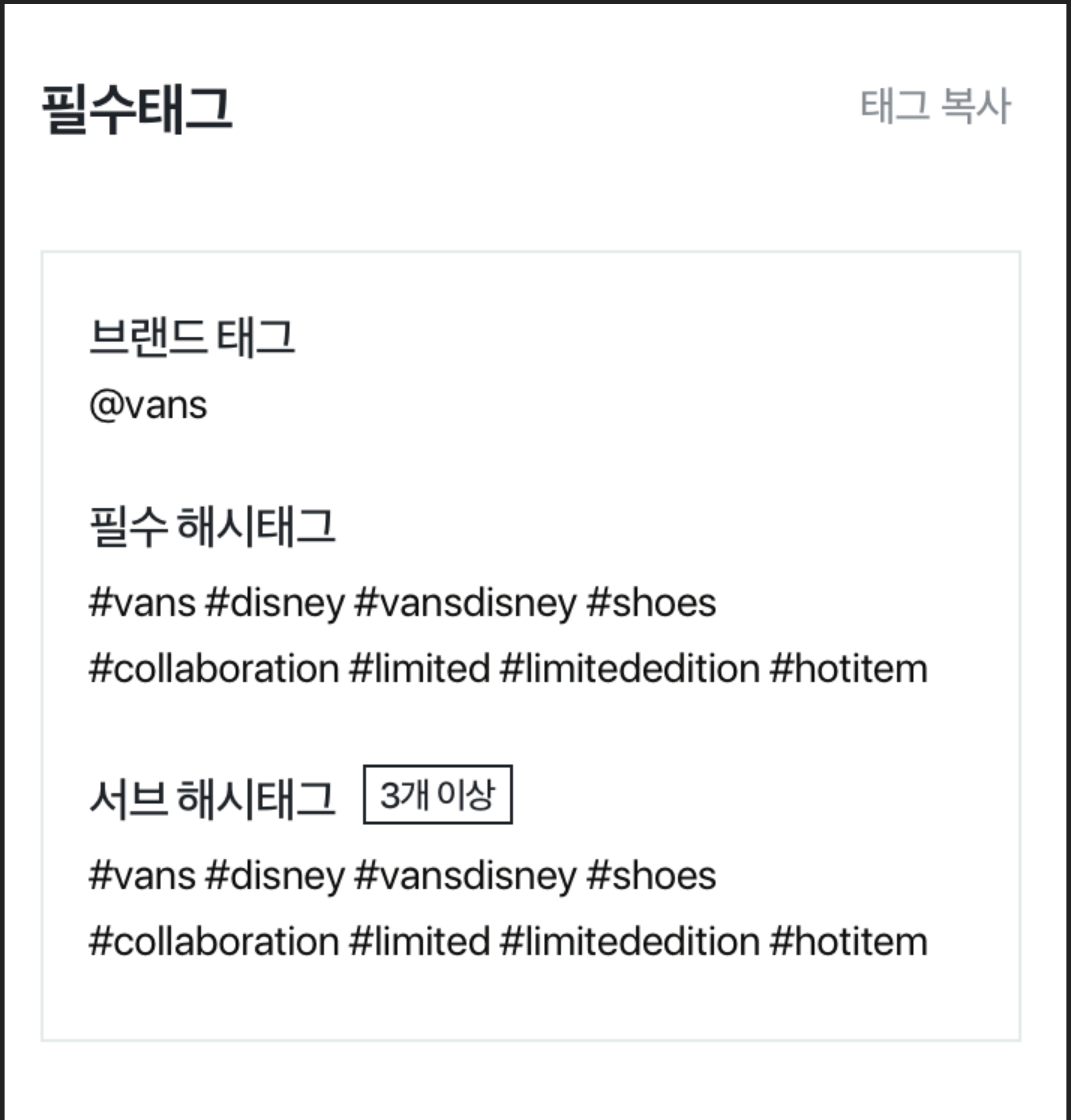
```swift
func layout() {
    node.automaticallyManagesSubnodes = true
    node.layoutSpecBlock = { (_, _) -> ASLayoutSpec in
        return absol(

                /**/
                self.tableView.m(46 + UIApplication.shared.statusBarFrame.height, 0, 0, 0), self.view_toolbar,

                /**/
                corner(
                        self.button_noti, self.text_countOfUnread, .topRight, -22, 18
                ).mt(UIApplication.shared.statusBarFrame.height),

                /**/
                self.button_filter.mt(UIApplication.shared.statusBarFrame.height).relative(.end, .start)
        )
    }
}
```



@hashup 님이
참여 가능한 캠페인

일반캠페인

```
func layout() {
    node.automaticallyManagesSubnodes = true
    node.layoutSpecBlock = { (_, _) -> ASLayoutSpec in
        return absol(

                /**/
                self.tableView.m(46 + UIApplication.shared.statusBarFrame.height, 0, 0, 0), self.view_toolbar,

                /**/
                corner(
                        self.button_noti, self.text_countOfUnread, .topRight, -22, 18
                ).mt(UIApplication.shared.statusBarFrame.height),

                /**/
                self.button_filter.mt(UIApplication.shared.statusBarFrame.height).relative(.end, .start)
        )
    }
}
```



필수태그

브랜드 태그

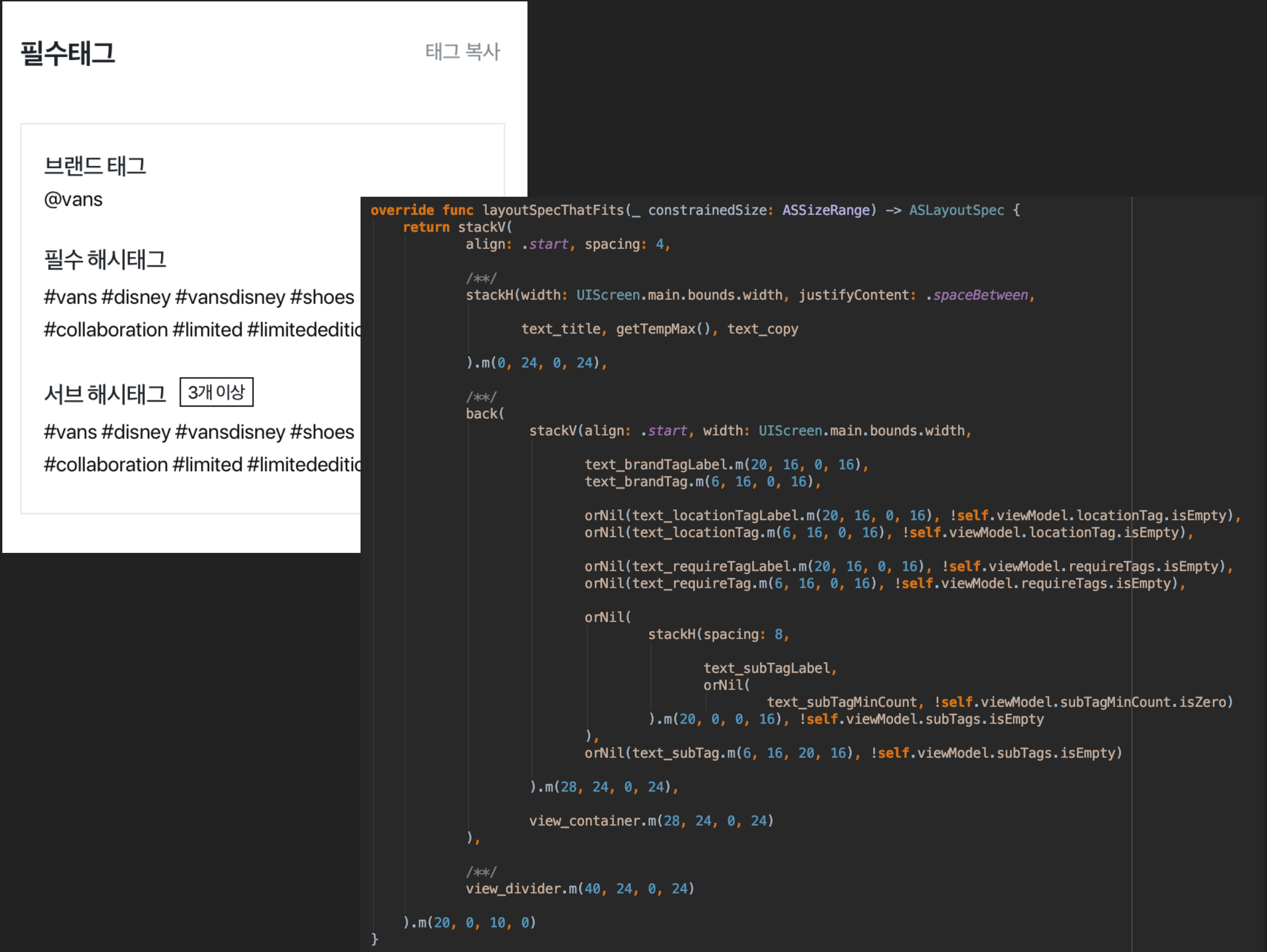
@vans

필수 해시태그

#vans #disney #vansdisney #shoes
#collaboration #limited #limitededition #hotitem

서브 해시태그 3개 이상

#vans #disney #vansdisney #shoes
#collaboration #limited #limitededition #hotitem



필수태그

태그 복사

브랜드 태그

@vans

필수 해시태그

#vans #disney #vansdisney #shoes

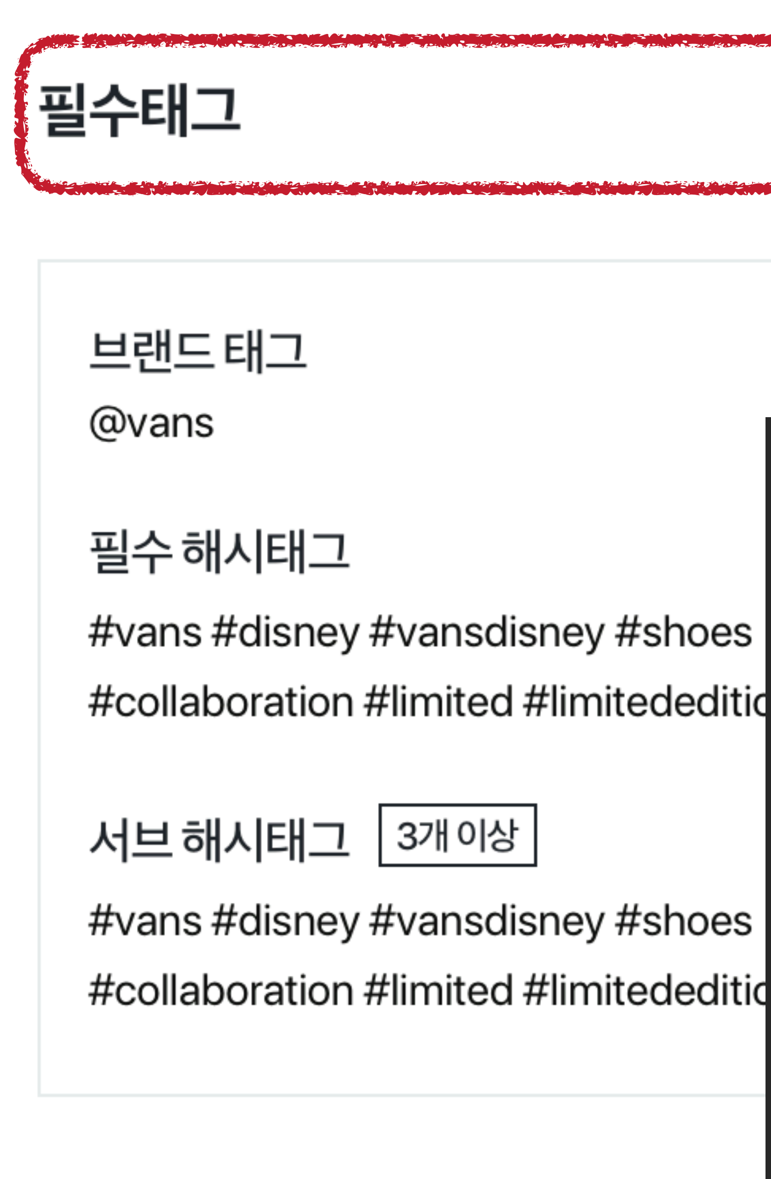
#collaboration #limited #limitededitic

서브 해시태그 3개 이상

#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitic

```swift
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start, spacing: 4,

            /**/
            stackH(width: UIScreen.main.bounds.width, justifyContent: .spaceBetween,

                   text_title, getTempMax(), text_copy

            ).m(0, 24, 0, 24),

            /**/
            back(
                    stackV(align: .start, width: UIScreen.main.bounds.width,

                           text_brandTagLabel.m(20, 16, 0, 16),
                           text_brandTag.m(6, 16, 0, 16),

                           orNil(text_locationTagLabel.m(20, 16, 0, 16), !self.viewModel.locationTag.isEmpty),
                           orNil(text_locationTag.m(6, 16, 0, 16), !self.viewModel.locationTag.isEmpty),

                           orNil(text_requireTagLabel.m(20, 16, 0, 16), !self.viewModel.requireTags.isEmpty),
                           orNil(text_requireTag.m(6, 16, 0, 16), !self.viewModel.requireTags.isEmpty),

                           orNil(
                                   stackH(spacing: 8,

                                          text_subTagLabel,
                                          orNil(
                                                  text_subTagMinCount, !self.viewModel.subTagMinCount.isZero)
                                   ).m(20, 0, 0, 16), !self.viewModel.subTags.isEmpty
                           ),
                           orNil(text_subTag.m(6, 16, 20, 16), !self.viewModel.subTags.isEmpty)

                    ).m(28, 24, 0, 24),

                    view_container.m(28, 24, 0, 24)
            ),

            /**/
            view_divider.m(40, 24, 0, 24)

    ).m(20, 0, 10, 0)
}
```



필수태그 태그 복사

브랜드 태그

@vans

필수 해시태그

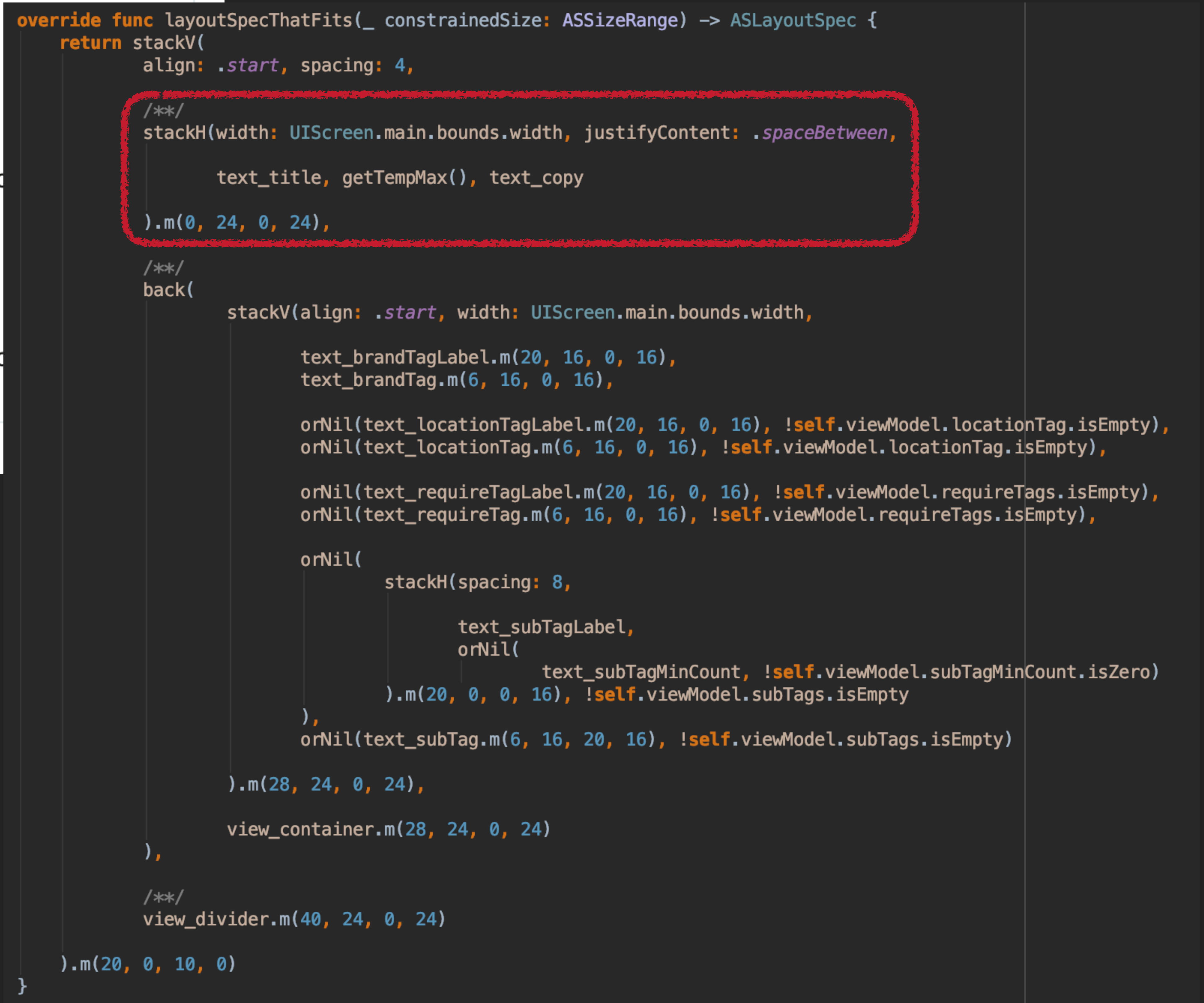
#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio

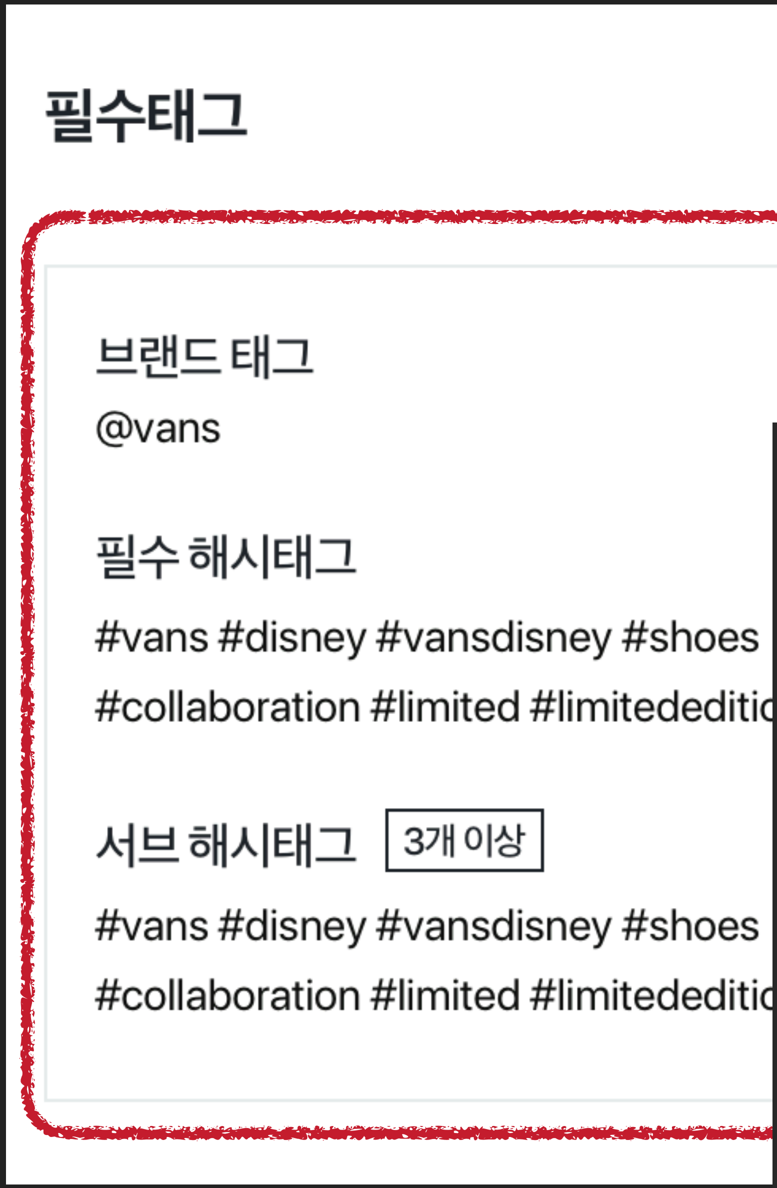
서브 해시태그 3개 이상

#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio



```
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start, spacing: 4,

            /**/
            stackH(width: UIScreen.main.bounds.width, justifyContent: .spaceBetween,

                   text_title, getTempMax(), text_copy

            ).m(0, 24, 0, 24),

            /**/
            back(
                    stackV(align: .start, width: UIScreen.main.bounds.width,

                           text_brandTagLabel.m(20, 16, 0, 16),
                           text_brandTag.m(6, 16, 0, 16),

                           orNil(text_locationTagLabel.m(20, 16, 0, 16), !self.viewModel.locationTag.isEmpty),
                           orNil(text_locationTag.m(6, 16, 0, 16), !self.viewModel.locationTag.isEmpty),

                           orNil(text_requireTagLabel.m(20, 16, 0, 16), !self.viewModel.requireTags.isEmpty),
                           orNil(text_requireTag.m(6, 16, 0, 16), !self.viewModel.requireTags.isEmpty),

                           orNil(
                                   stackH(spacing: 8,

                                          text_subTagLabel,
                                          orNil(
                                                  text_subTagMinCount, !self.viewModel.subTagMinCount.isZero)
                                   ).m(20, 0, 0, 16), !self.viewModel.subTags.isEmpty
                           ),
                           orNil(text_subTag.m(6, 16, 20, 16), !self.viewModel.subTags.isEmpty)

                    ).m(28, 24, 0, 24),

                    view_container.m(28, 24, 0, 24)
            ),

            /**/
            view_divider.m(40, 24, 0, 24)

    ).m(20, 0, 10, 0)
}
```



필수태그 태그 복사

브랜드 태그

@vans

필수 해시태그

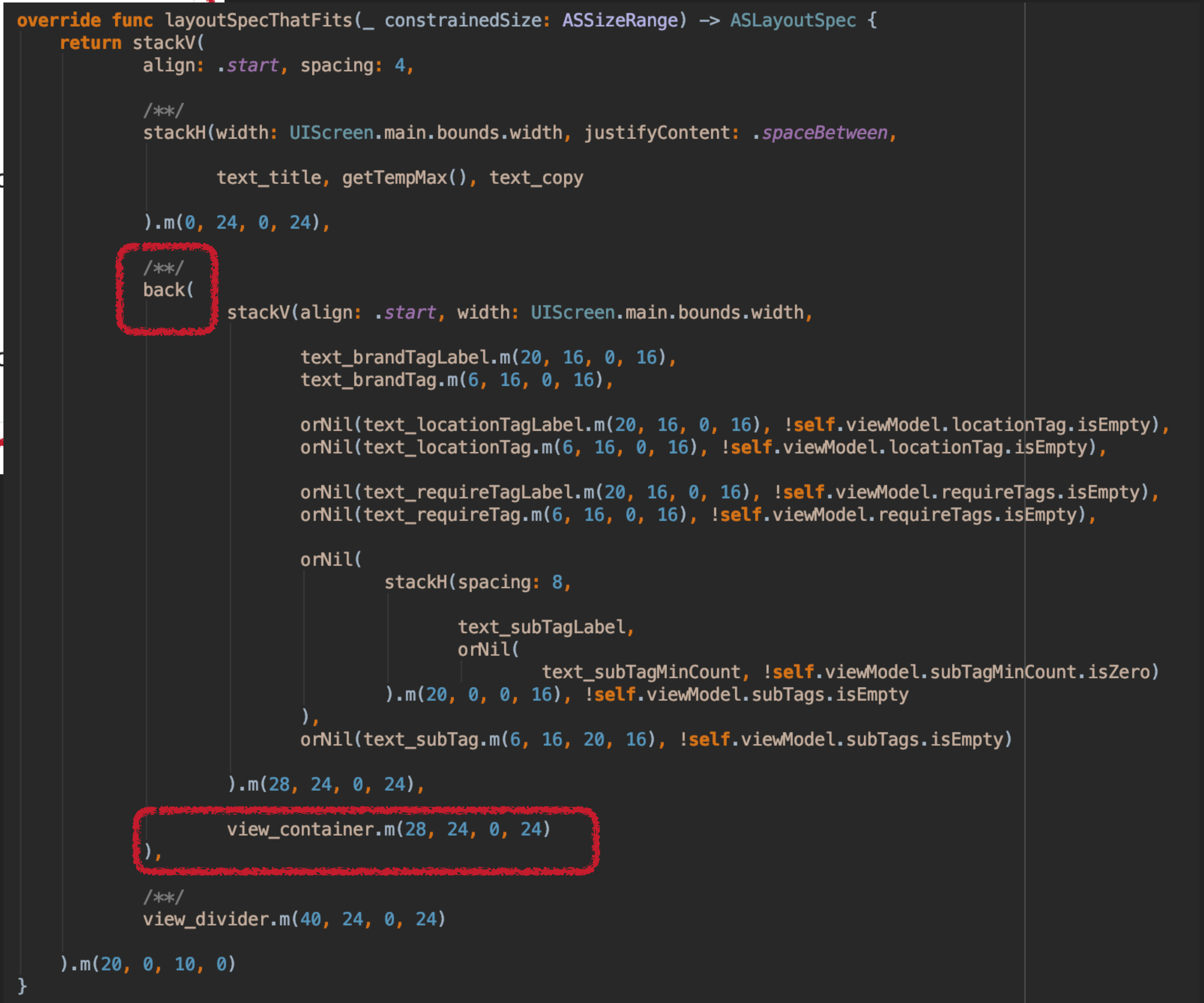
#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio

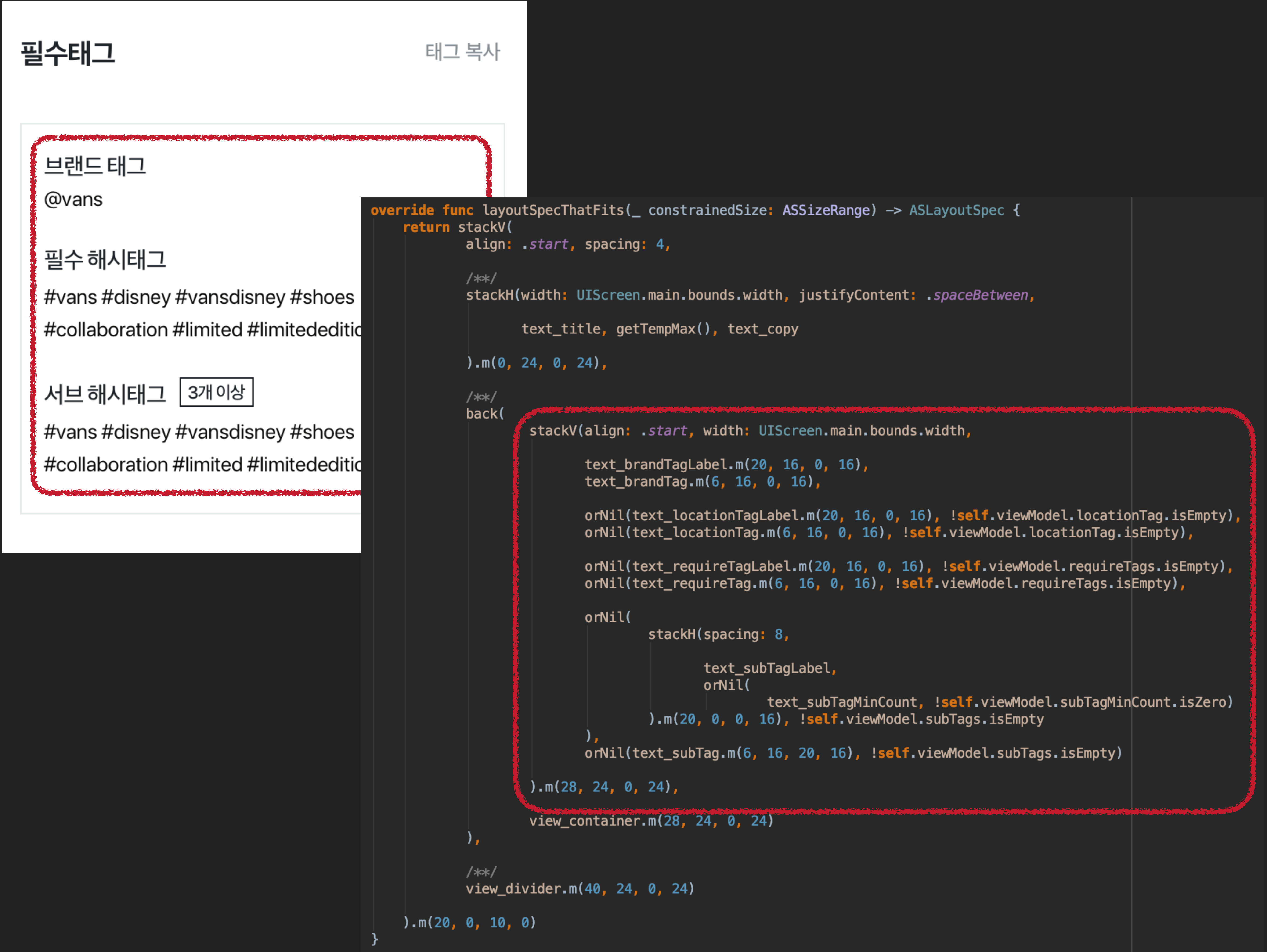
서브 해시태그 3개 이상

#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio



```swift
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start, spacing: 4,

            /**/
            stackH(width: UIScreen.main.bounds.width, justifyContent: .spaceBetween,

                    text_title, getTempMax(), text_copy

            ).m(0, 24, 0, 24),

            /**/
            back(
                stackV(align: .start, width: UIScreen.main.bounds.width,

                        text_brandTagLabel.m(20, 16, 0, 16),
                        text_brandTag.m(6, 16, 0, 16),

                        orNil(text_locationTagLabel.m(20, 16, 0, 16), !self.viewModel.locationTag.isEmpty),
                        orNil(text_locationTag.m(6, 16, 0, 16), !self.viewModel.locationTag.isEmpty),

                        orNil(text_requireTagLabel.m(20, 16, 0, 16), !self.viewModel.requireTags.isEmpty),
                        orNil(text_requireTag.m(6, 16, 0, 16), !self.viewModel.requireTags.isEmpty),

                        orNil(
                                stackH(spacing: 8,

                                        text_subTagLabel,
                                        orNil(
                                                text_subTagMinCount, !self.viewModel.subTagMinCount.isZero)
                                ).m(20, 0, 0, 16), !self.viewModel.subTags.isEmpty
                        ),
                        orNil(text_subTag.m(6, 16, 20, 16), !self.viewModel.subTags.isEmpty)

                ).m(28, 24, 0, 24),

                view_container.m(28, 24, 0, 24)
            ),

            /**/
            view_divider.m(40, 24, 0, 24)

    ).m(20, 0, 10, 0)
}
```



필수태그

태그 복사

브랜드 태그

@vans

필수 해시태그

#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio

서브 해시태그 3개 이상

#vans #disney #vansdisney #shoes

#collaboration #limited #limitededitio

```swift
override func layoutSpecThatFits(_ constrainedSize: ASSizeRange) -> ASLayoutSpec {
    return stackV(
            align: .start, spacing: 4,

            /**/
            stackH(width: UIScreen.main.bounds.width, justifyContent: .spaceBetween,

                   text_title, getTempMax(), text_copy

            ).m(0, 24, 0, 24),

            /**/
            back(
                    stackV(align: .start, width: UIScreen.main.bounds.width,

                           text_brandTagLabel.m(20, 16, 0, 16),
                           text_brandTag.m(6, 16, 0, 16),

                           orNil(text_locationTagLabel.m(20, 16, 0, 16), !self.viewModel.locationTag.isEmpty),
                           orNil(text_locationTag.m(6, 16, 0, 16), !self.viewModel.locationTag.isEmpty),

                           orNil(text_requireTagLabel.m(20, 16, 0, 16), !self.viewModel.requireTags.isEmpty),
                           orNil(text_requireTag.m(6, 16, 0, 16), !self.viewModel.requireTags.isEmpty),

                           orNil(
                                   stackH(spacing: 8,

                                          text_subTagLabel,
                                          orNil(
                                                  text_subTagMinCount, !self.viewModel.subTagMinCount.isZero)
                                   ).m(20, 0, 0, 16), !self.viewModel.subTags.isEmpty
                           ),
                           orNil(text_subTag.m(6, 16, 20, 16), !self.viewModel.subTags.isEmpty)

                    ).m(28, 24, 0, 24),

                    view_container.m(28, 24, 0, 24)
            ),

            /**/
            view_divider.m(40, 24, 0, 24)

    ).m(20, 0, 10, 0)
}
```

Thank you