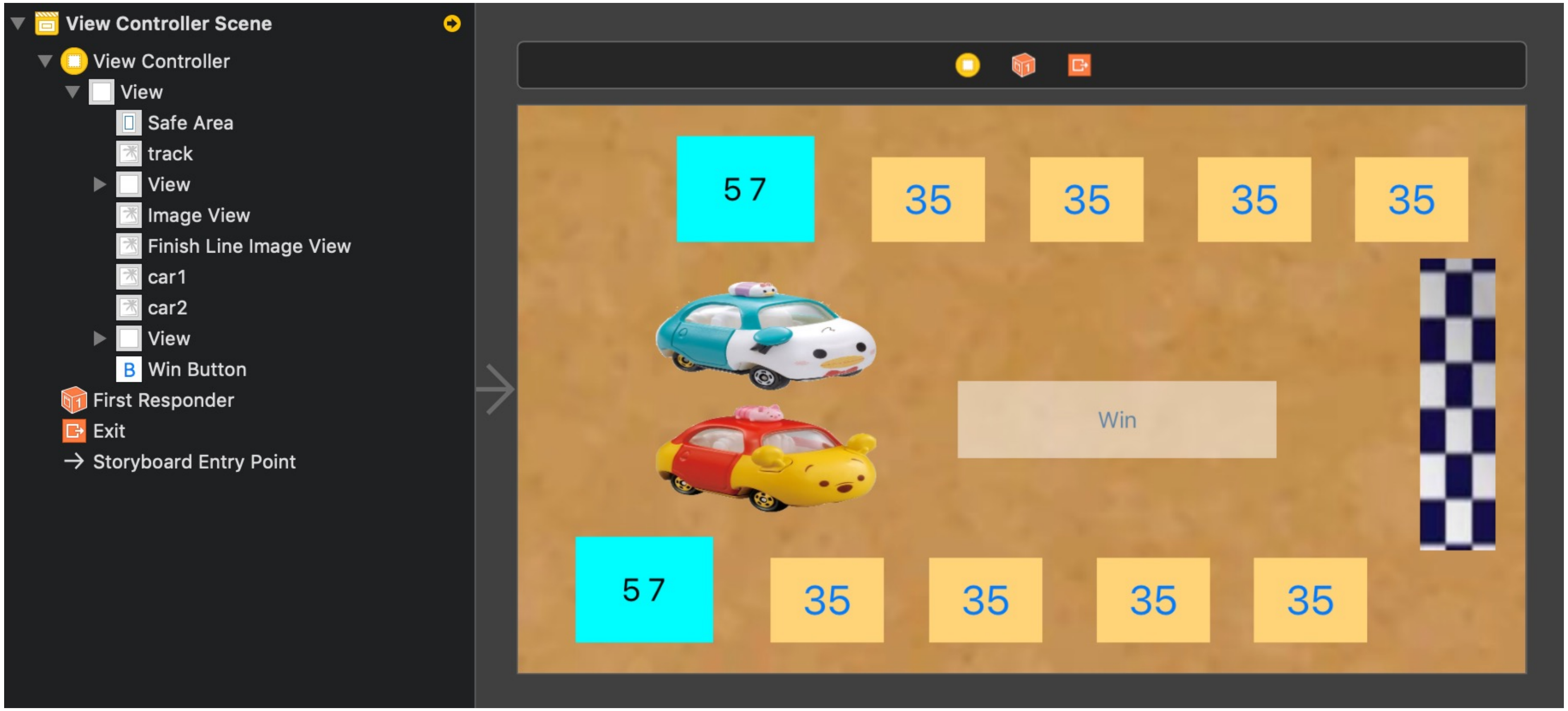
新手的 iOS App 練功坊 1
乘法賽車 PK 大賽

http://makeiosapp.strikingly.com

彼得潘

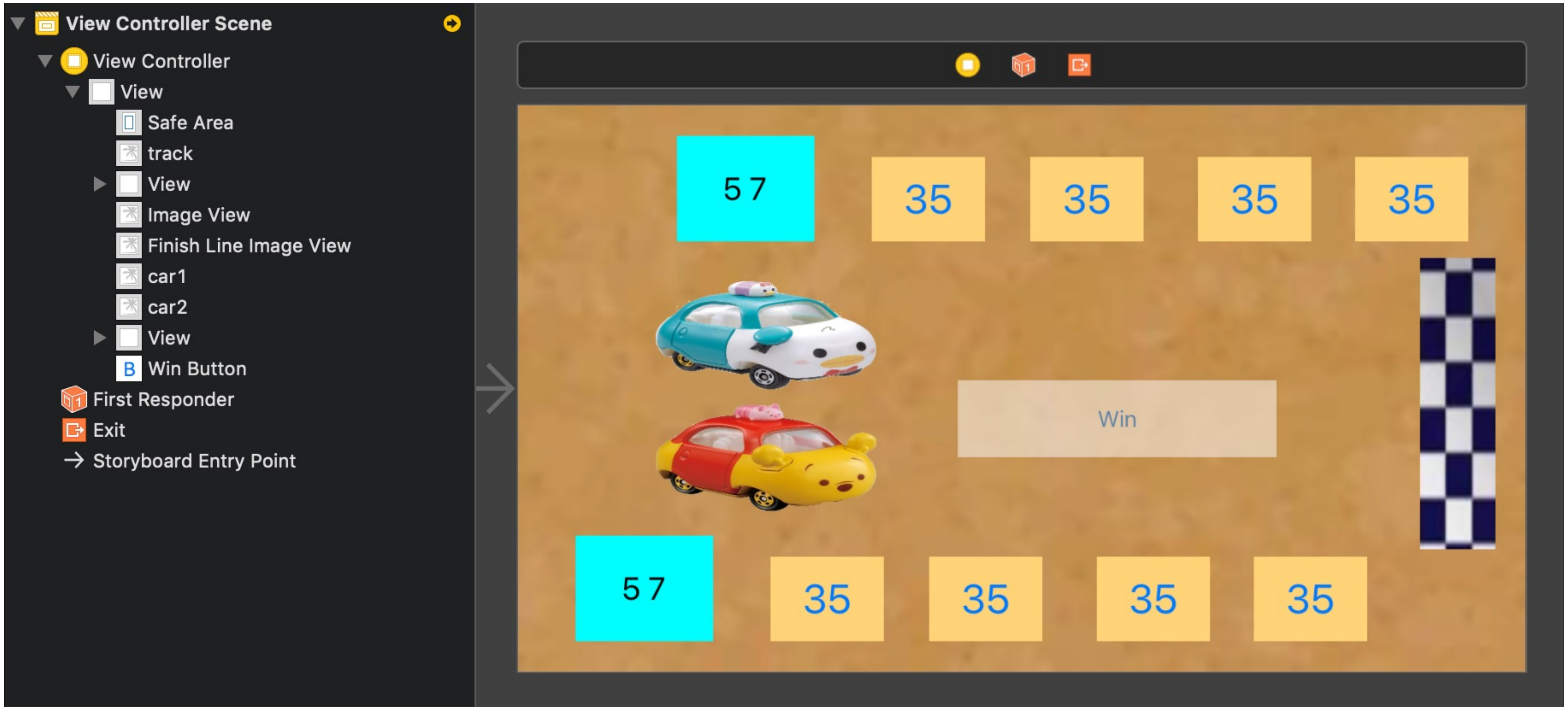
http://apppeterpan.strikingly.com

製作畫面


View Controller Scene
- View Controller
  - View
    - Safe Area
    - track
    - View
    - Image View
    - Finish Line Image View
    - car1
    - car2
    - View
    - Win Button
- First Responder
- Exit
- Storyboard Entry Point

https://bit.ly/2R0XFDL

拉 outlet


View Controller Scene
View Controller
View
Safe Area
track
View
Image View
Finish Line Image View
car1
car2
View
Win Button
First Responder
Exit
Storyboard Entry Point
5 7
35
35
35
35
Win
5 7
35
35
35
35

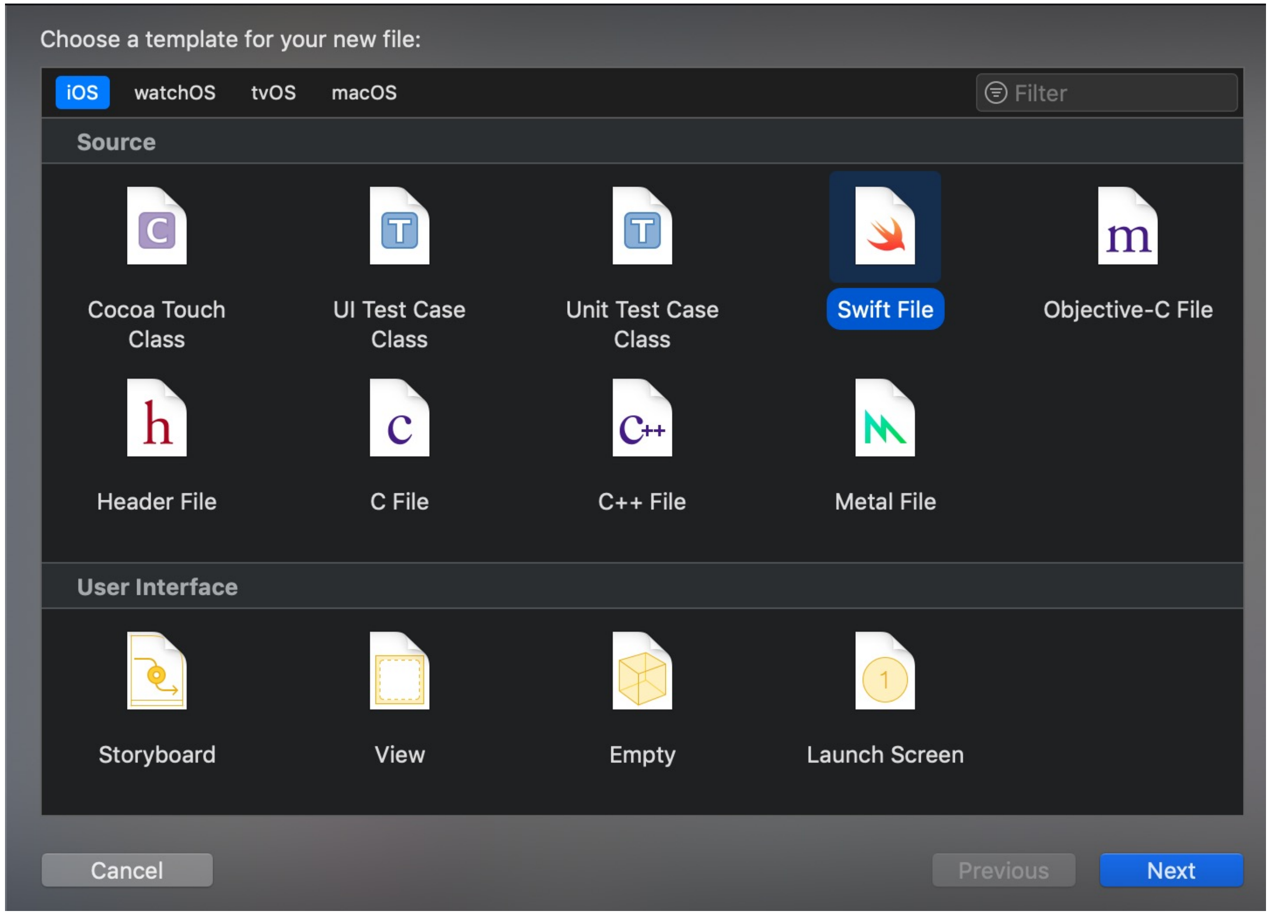
outlet collection

https://bit.ly/2AiR6aq

旋轉 view 的方法

https://bit.ly/2AbEC43

定義問題

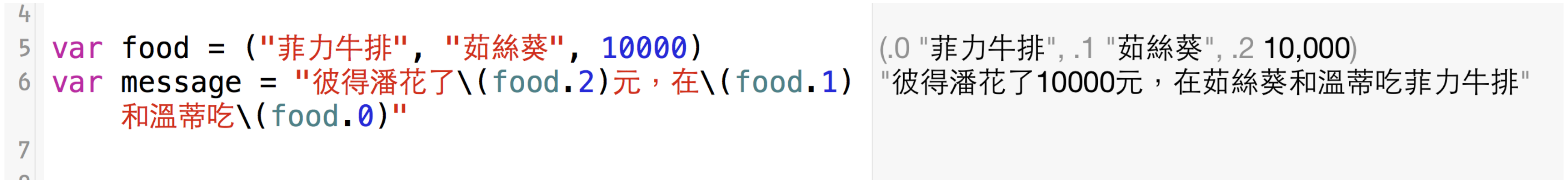

Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next

定義問題

```swift
struct Question {
    let title: (Int, Int)
    let choices: [Int]
}
```

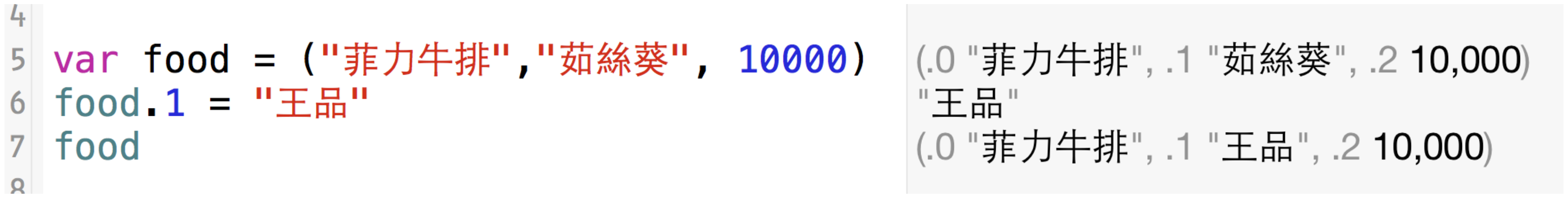
tuple ?

結合資料的輕便tuple



```swift
var food = ("菲力牛排", "茹絲葵", 10000)
var message = "彼得潘花了\(food.2)元，在\(food.1)
    和溫蒂吃\(food.0)"
```

```
(.0 "菲力牛排", .1 "茹絲葵", .2 10,000)
"彼得潘花了10000元，在茹絲葵和溫蒂吃菲力牛排"
```

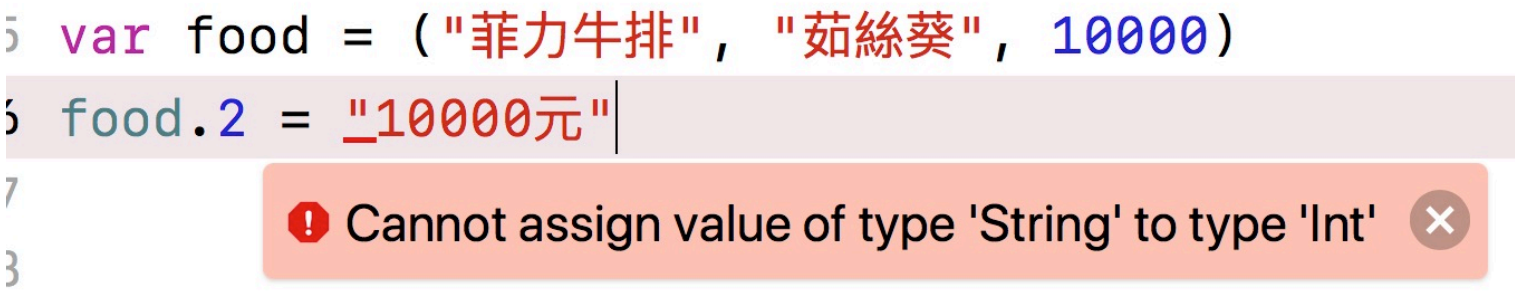


```swift
var food = ("菲力牛排","茹絲葵", 10000)
food.1 = "王品"
food
```

```
(.0 "菲力牛排", .1 "茹絲葵", .2 10,000)
"王品"
(.0 "菲力牛排", .1 "王品", .2 10,000)
```

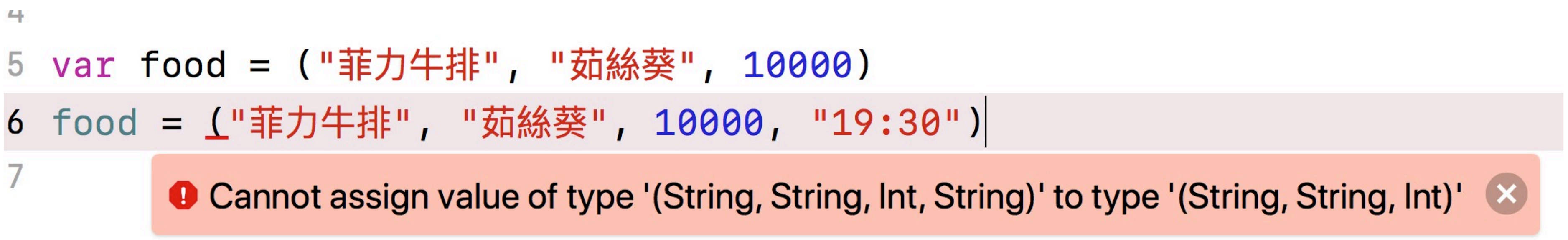
Apple 範例:
let http404Error = (404, "Not Found")

結合資料的輕便tuple



```
var food = ("菲力牛排", "茄絲葵", 10000)
food.2 = "10000元"
```

Cannot assign value of type 'String' to type 'Int'



```
var food = ("菲力牛排", "茄絲葵", 10000)
food = ("菲力牛排", "茄絲葵", 10000, "19:30")
```

Cannot assign value of type '(String, String, Int, String)' to type '(String, String, Int)'

結合資料的輕便tuple

```
var food = (name: "菲力牛排", restaurant: "茹絲葵", price: 10000)

var message = "彼得潘花了\(food.price)元，在\(food.restaurant)和溫蒂吃\(food.name)"
```

```
var food: (name: String, restaurant: String, price: Int) = ("菲力牛排", "茹絲葵",
10000)

var message = "彼得潘花了\(food.price)元，在\(food.restaurant)和溫蒂吃\(food.name)"

var (foodName, foodRestaurant, foodPrice) = food
message = "彼得潘花了\(foodPrice)元，在\(foodRestaurant)和溫蒂吃\(foodName)"
```

以 tuple 存取 dictionary & array

```swift
var foodDictionary = ["name": "菲力牛排", "price": "$1000"]
for (key, value) in foodDictionary {
    print(key, value)
}

var foods = ["菲力牛排", "松阪豬"]
var i = 0
for food in foods {
    print(i, food)
    i = i + 1
}

for (i, food) in foods.enumerated() {
    print(i, food)
}
```

用 tuple 圓 function 回傳多個資料的夢

```swift
func eat(name: String) -> (String, Int) {
    if name == "早餐" {
        return ("菲力牛排", 1000)
    } else {
        return ("神戶牛排", 10000)
    }
}

var food = eat(name: "晚餐")
print(food.0, food.1)
```

問題的 array

```
var questions = [Question]()
```

進行到第幾題

```
var questionIndexes = [Int](repeating: 0, count: 2)
```

array 有 2 個成員，內容都是 0

亂數

https://bit.ly/2EvZrLV

建立問題

```
func createQuestion() -> Question {
        let numberRange = 1...9
        let number1 = Int.random(in: numberRange)
        let number2 = Int.random(in: numberRange)
        let choiceRange = 0...3
        let answerIndex = Int.random(in: choiceRange)
        var choices = [Int]()
        for i in choiceRange {
            if i == answerIndex {
                let number = number1 * number2
                choices.append(number)
            } else {
                let number = Int.random(in: 1...99)
                choices.append(number)
            }

        }
        let question = Question(title: (number1, number2), choices: choices)
        questions.append(question)
        return question
    }
```

建立問題

可能產生重覆的選項

```
func createQuestion() -> Question {
       let numberRange = 1...9
       let number1 = Int.random(in: numberRange)
       let number2 = Int.random(in: numberRange)
       let choiceRange = 0...3
       let answerIndex = Int.random(in: choiceRange)
       var choices = [Int]()
       for i in choiceRange {
           if i == answerIndex {
               let number = number1 * number2
               choices.append(number)
           } else {
               let number = Int.random(in: 1...99)
               choices.append(number)
           }

       }
       let question = Question(title: (number1, number2), choices: choices)
       questions.append(question)
       return question
   }
```

```swift
    let choiceNumbers = [Int](1...99)

    func createQuestion() -> Question {

        let numberRange = 1...9
        let number1 = Int.random(in: numberRange)
        let number2 = Int.random(in: numberRange)
        let choiceRange = 0...3
        let answerIndex = Int.random(in: choiceRange)
        let answer = number1 * number2

        var choices = choiceNumbers.filter { (number) -> Bool in
            return answer != number
        }
        choices.shuffle()
        choices[answerIndex] = answer

        let question = Question(title: (number1, number2), choices: 
Array(choices[choiceRange]))
        questions.append(question)
        return question
    }
```

產生 1 ~ 99 的 array

```swift
let choiceNumbers = [Int](1...99)
```

shuffle & shuffled

ed / ing : 原本資料不變，
產生改變後的資料回傳

```
var name = "peter"
name.append(" pan")
name.appending(" pan")
```

Array & ArraySlice

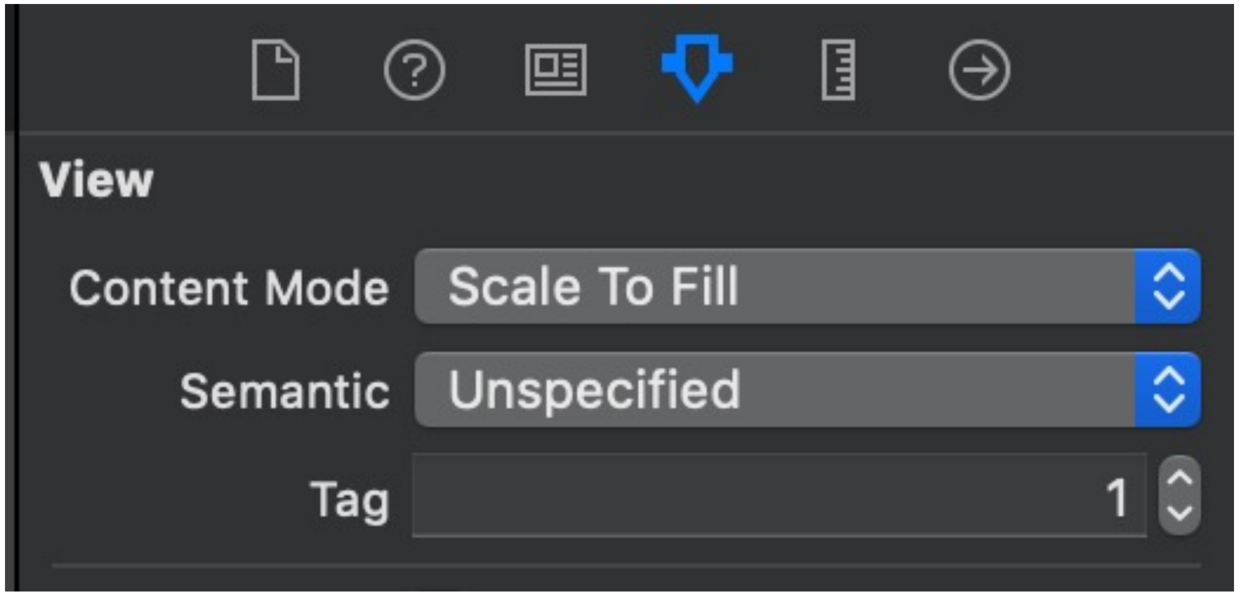
```
 let question = Question(title: (number1, number2), choices:
Array(choices[choiceRange]))
```

拉 action

多個元件可連到同一個 function

```
@IBAction func choiceButtonPressed(_ sender: UIButton)
```

利用 tag 判斷點選的 button



View

Content Mode: Scale To Fill

Semantic: Unspecified

Tag: 1

答對時移動車子

https://bit.ly/2EzbPeh

調整元件位置的五種方法

判斷是否到達終點

```
if carImageViews[questionViewTag].frame.maxX >= finishLineImageView.frame.minX {
            if questionViewTag == 0 {
                winButton.setTitle("Yellow Win", for: .normal)

            } else {
                winButton.setTitle("White Win", for: .normal)

            }
            winButton.isHidden = false
        }
```

重玩

動畫

```
UIView.animate(withDuration: 0.5) {
    self.carImageViews[questionViewTag].center.x += 50

}
```

完整版

https://bit.ly/2pYIGA0

計時

https://swifteducation.github.io/teaching_app_development_with_swift/stopwatch.html