超級簡單ML-App

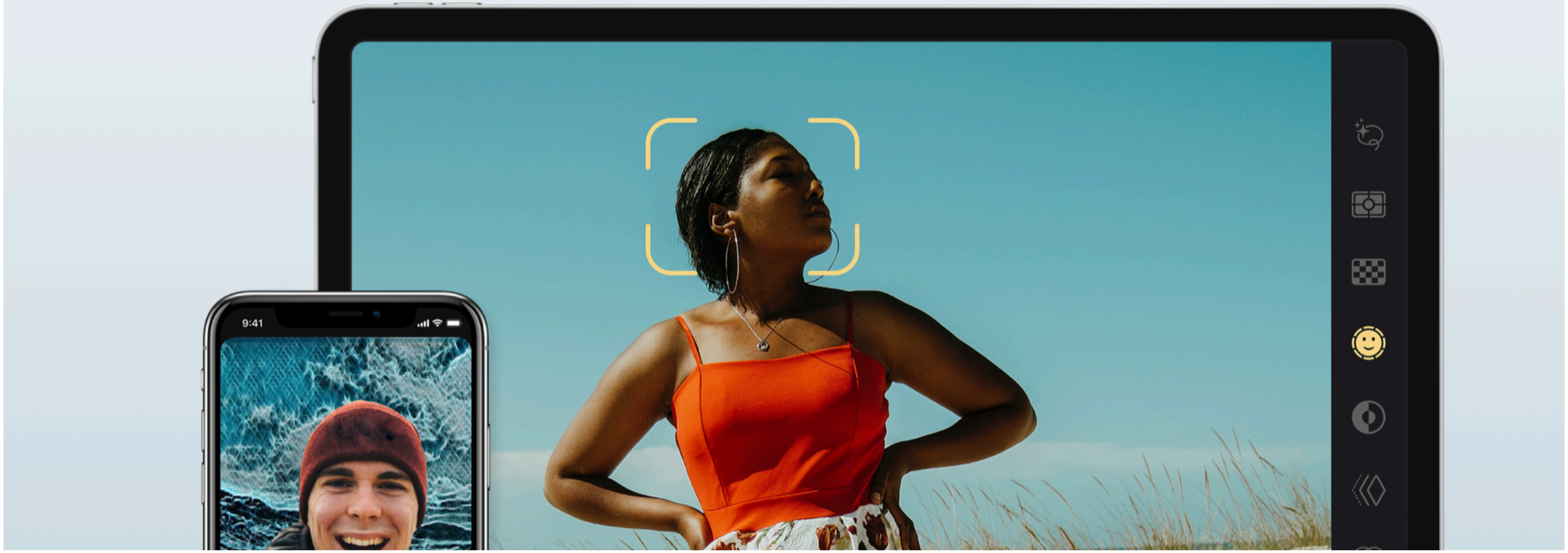
十進位轉2^k進位制 App

可選擇進位制


SUPER
SIMPLE
ML
11:05

Machine Learning

Create intelligent features and enable new experiences for your apps by leveraging powerful on-device machine learning. Learn how to build, train, and deploy machine learning models into your iPhone, iPad, Apple Watch, and Mac apps.



Core ML Models

Build intelligence into your apps using machine learning models from the research community designed for Core ML.

Images Text

Models are in Core ML format and can be integrated into Xcode projects. You can select different versions of models to optimize for sizes and architectures.

The MobileNetv2 architecture trained to classify the dominant object in a camera frame or image.

Hide details ^

Model Info

Links

- Source code in GitHub
- MobileNetV2

Variants

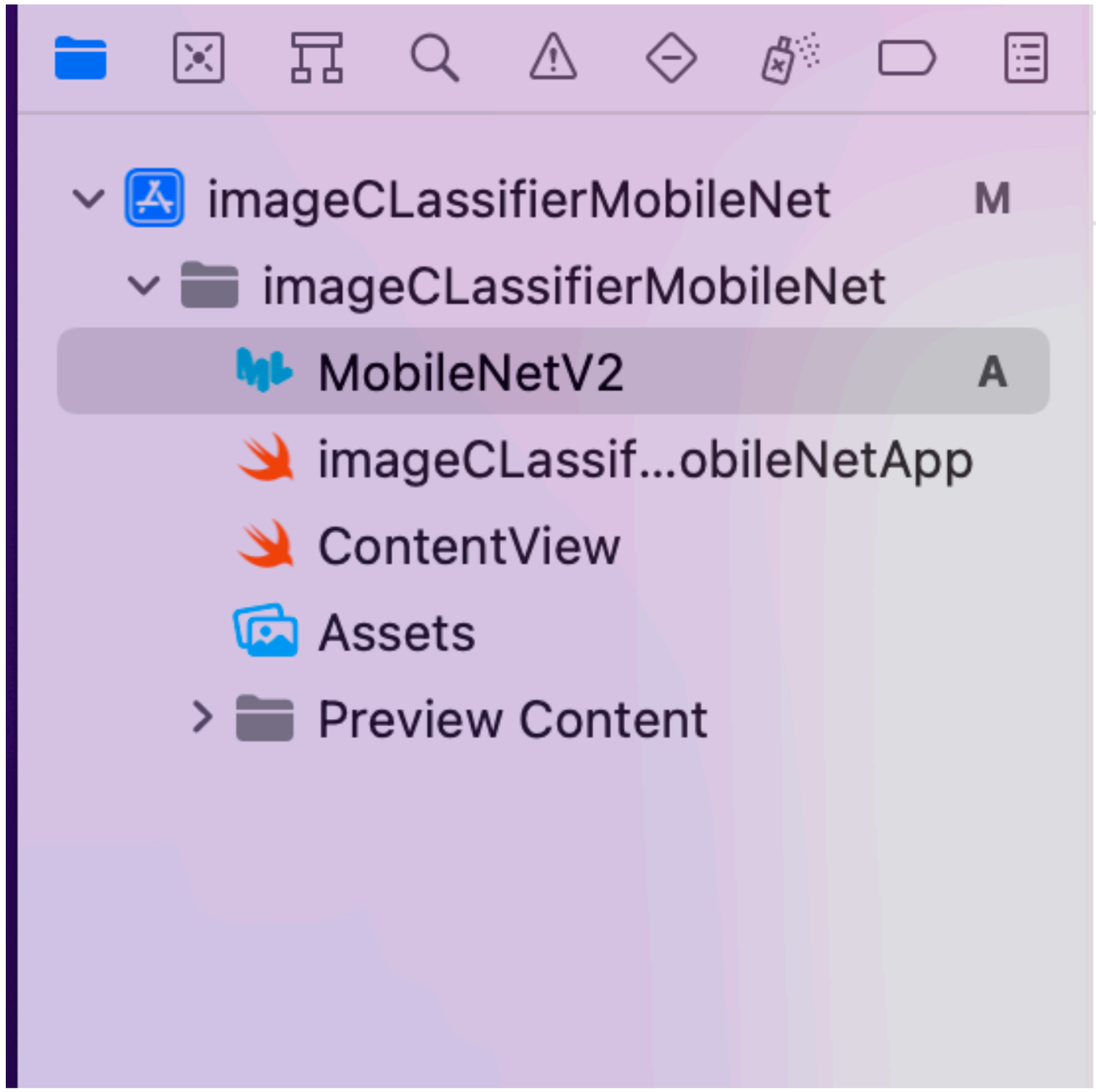
Model Name	Size	Action
MobileNetV2.mlmodel	24.7MB	Download
MobileNetV2FP16.mlmodel	12.4MB	Download
MobileNetV2Int8LUT.mlmodel	6.3MB	Download

MobileNetV2

MobileNetV2.mlmodel

Storing model weights using full precision (32 bit) floating point numbers.
24.7MB

Download


imageCLassifierMobileNet
M
imageCLassifierMobileNet
MobileNetV2
A
imageCLassif...obileNetApp
ContentView
Assets
Preview Content

MobileNetV2

Edit

Model Type	Neural Network Classifier
Size	24.7 MB
Document Type	Core ML Model
Availability	iOS 11.0+ | macOS 10.13+ | tvOS 11.0+ | watchOS 4.0+
Model Class	MobileNetV2

Automatically generated Swift model class

General | Preview | Predictions | Utilities

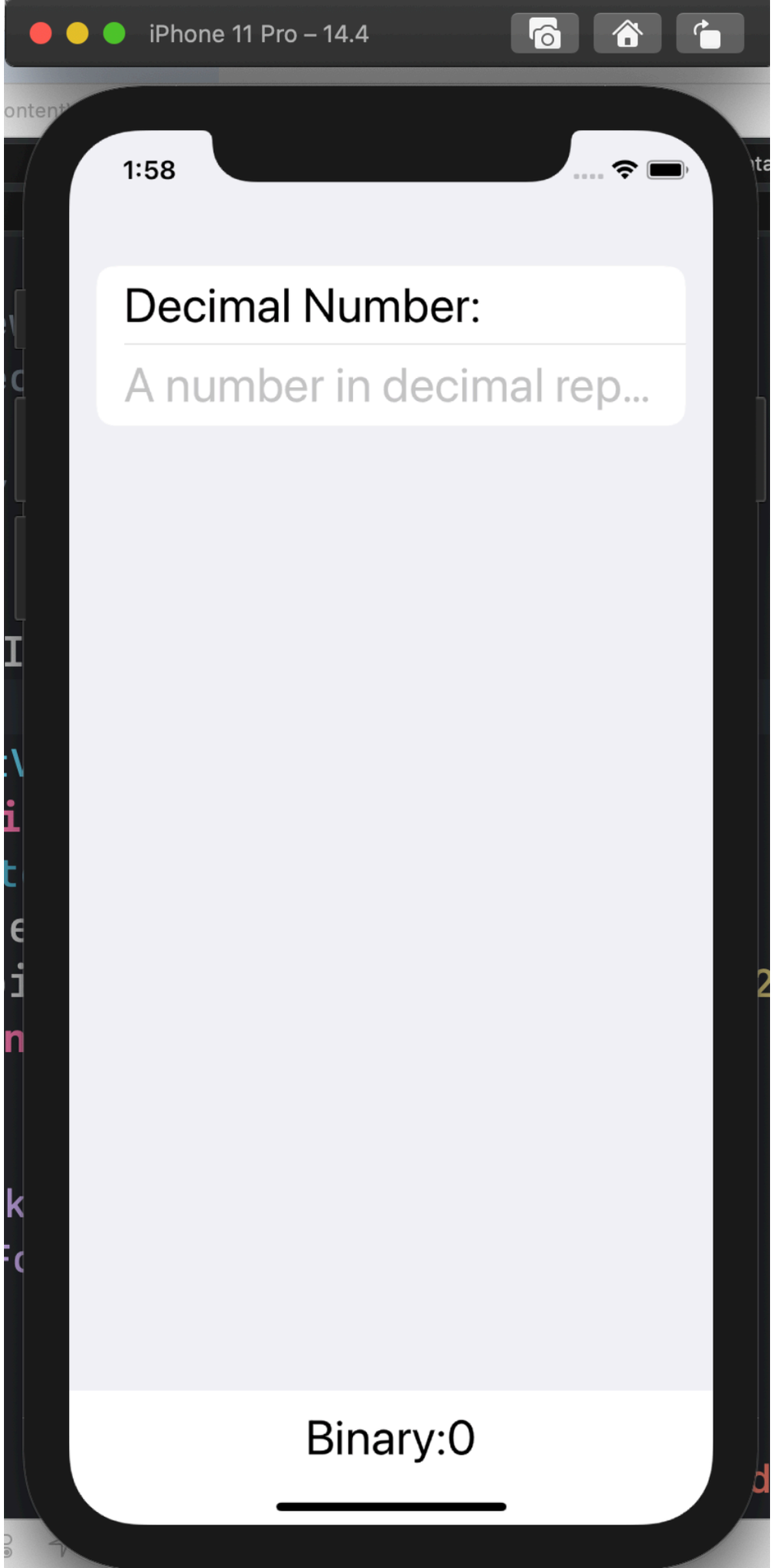
截圖 2022-03...9.48.44.png

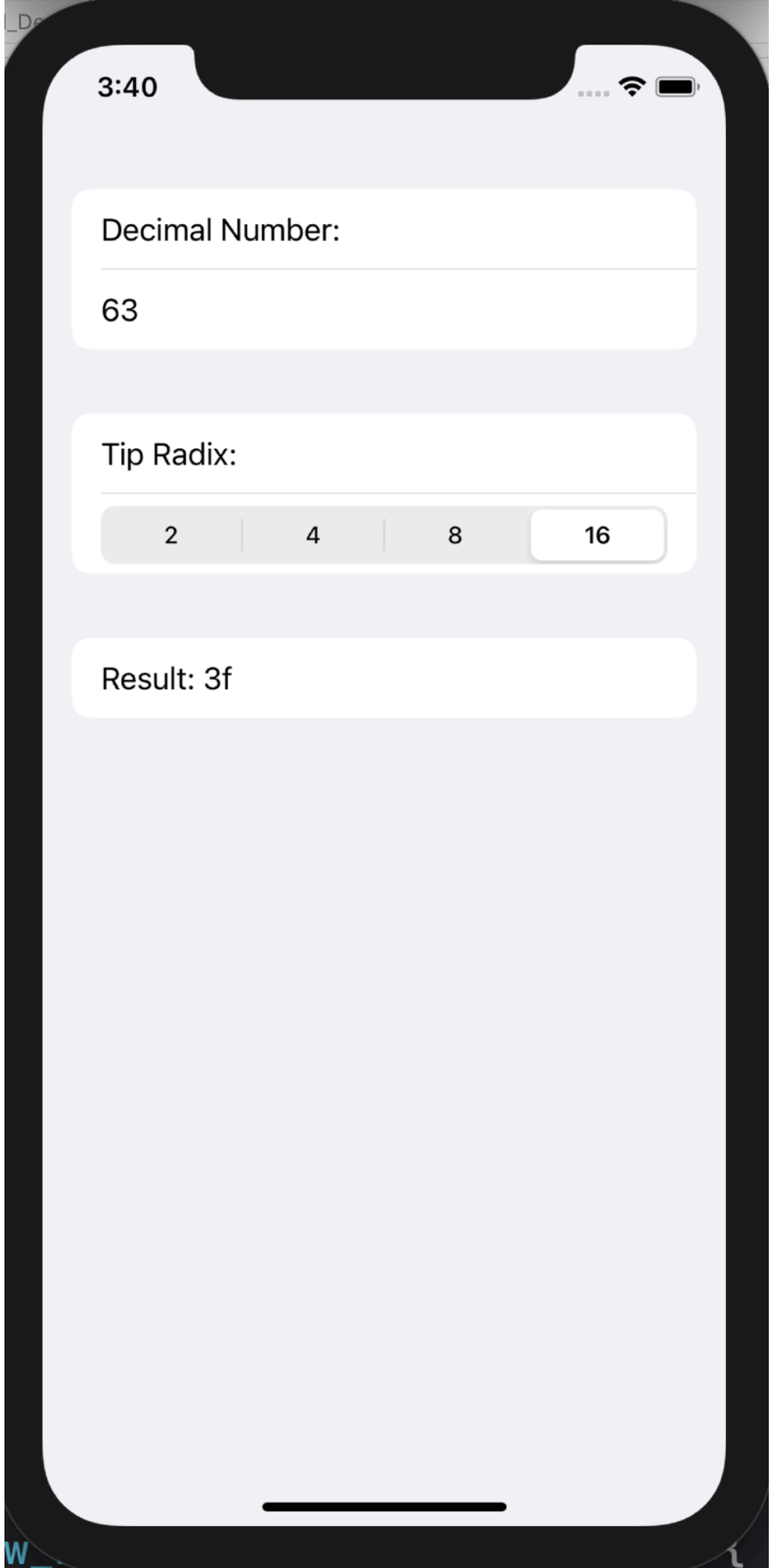
banana

69% confidence

spaghetti squash

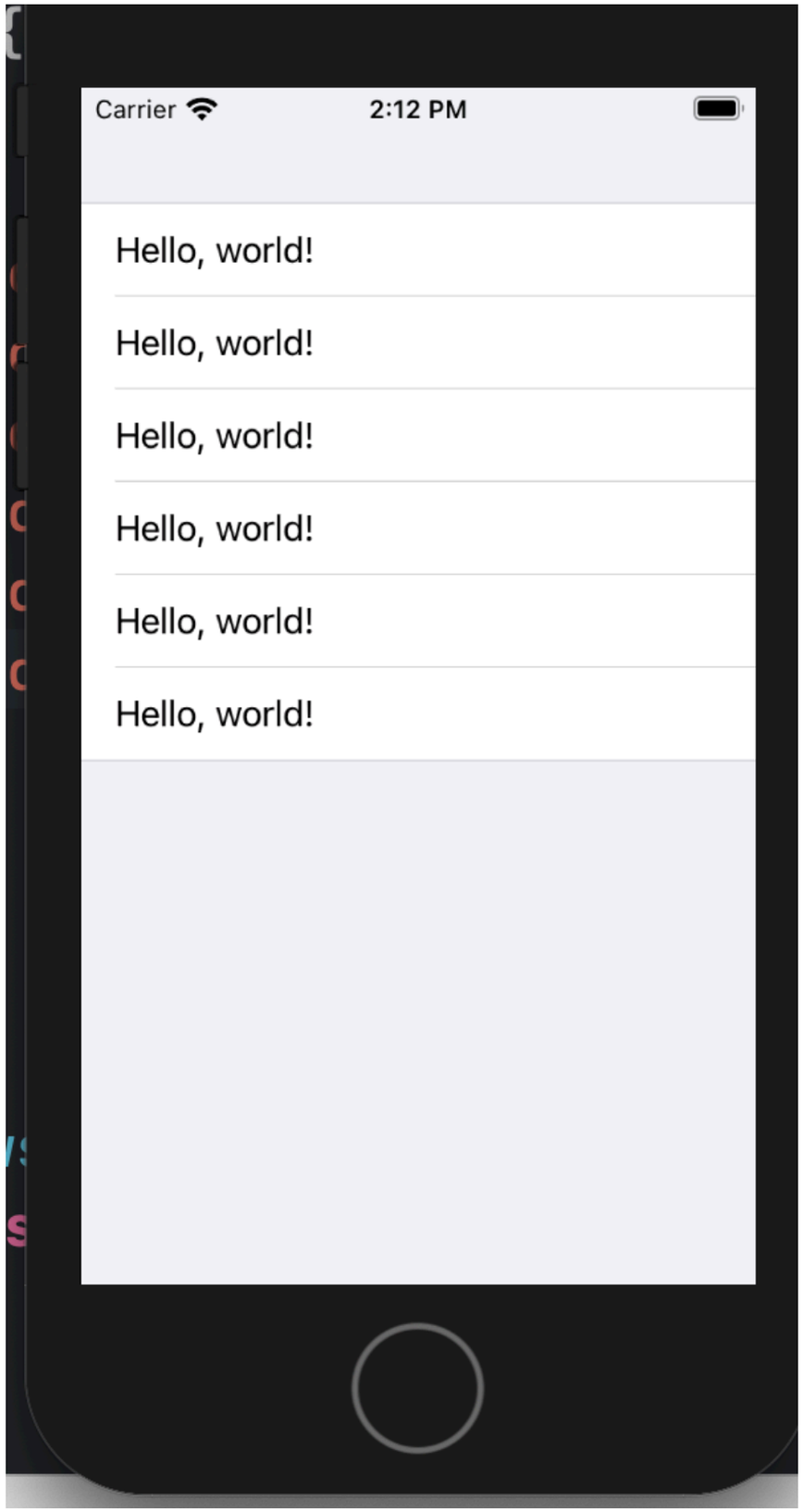
\+ Clear All


iPhone 11 Pro – 14.4
1:58
Decimal Number:
A number in decimal rep...
Binary:0


3:40
Decimal Number:
63
Tip Radix:
2
4
8
16
Result: 3f

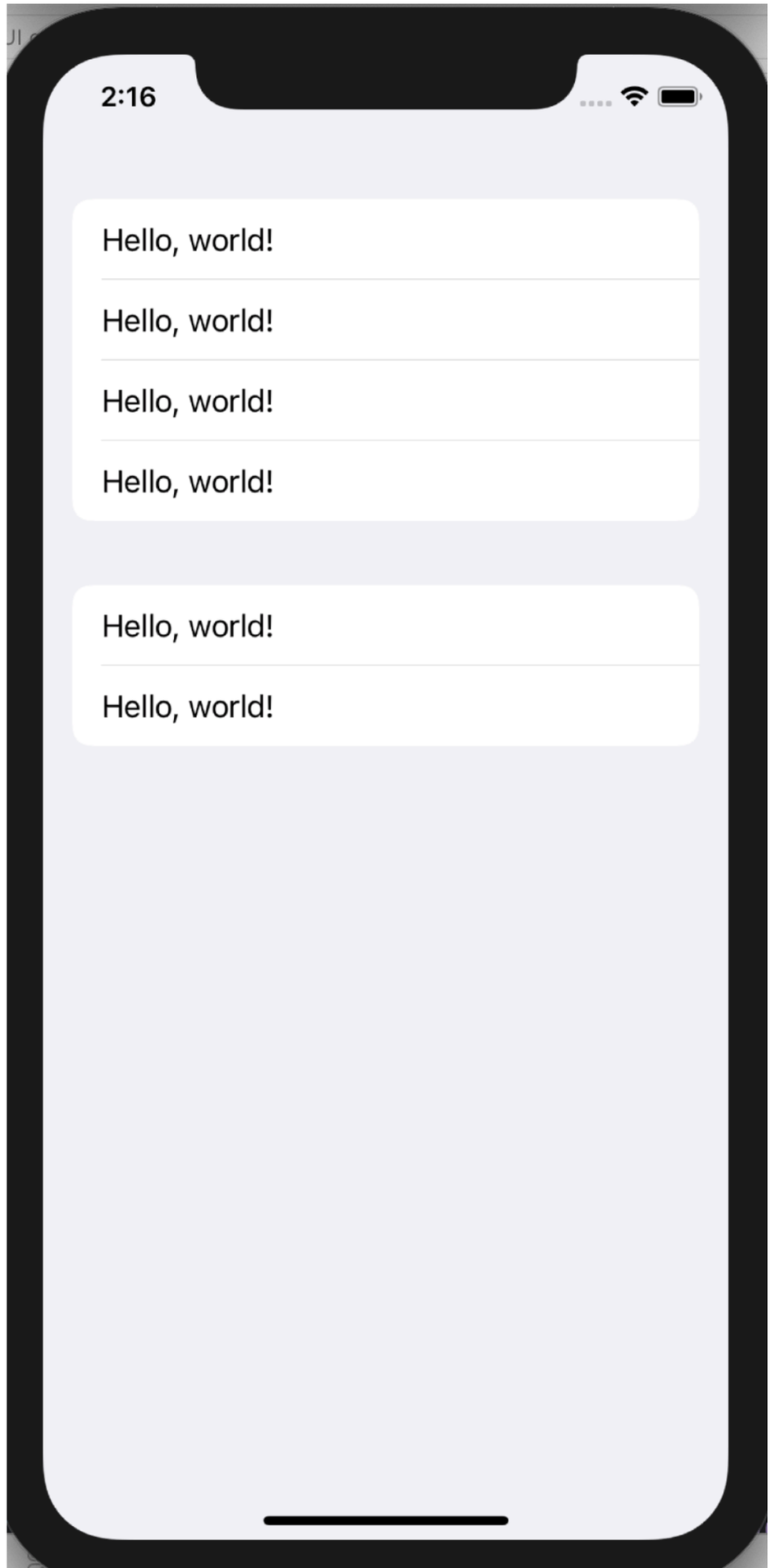

2

```swift
struct ContentView: View {
    var body: some View {
        Form {
            Text("Hello, world!")
            Text("Hello, world!")
            Text("Hello, world!")
            Text("Hello, world!")
            Text("Hello, world!")
            Text("Hello, world!")
        }
    }
}
```



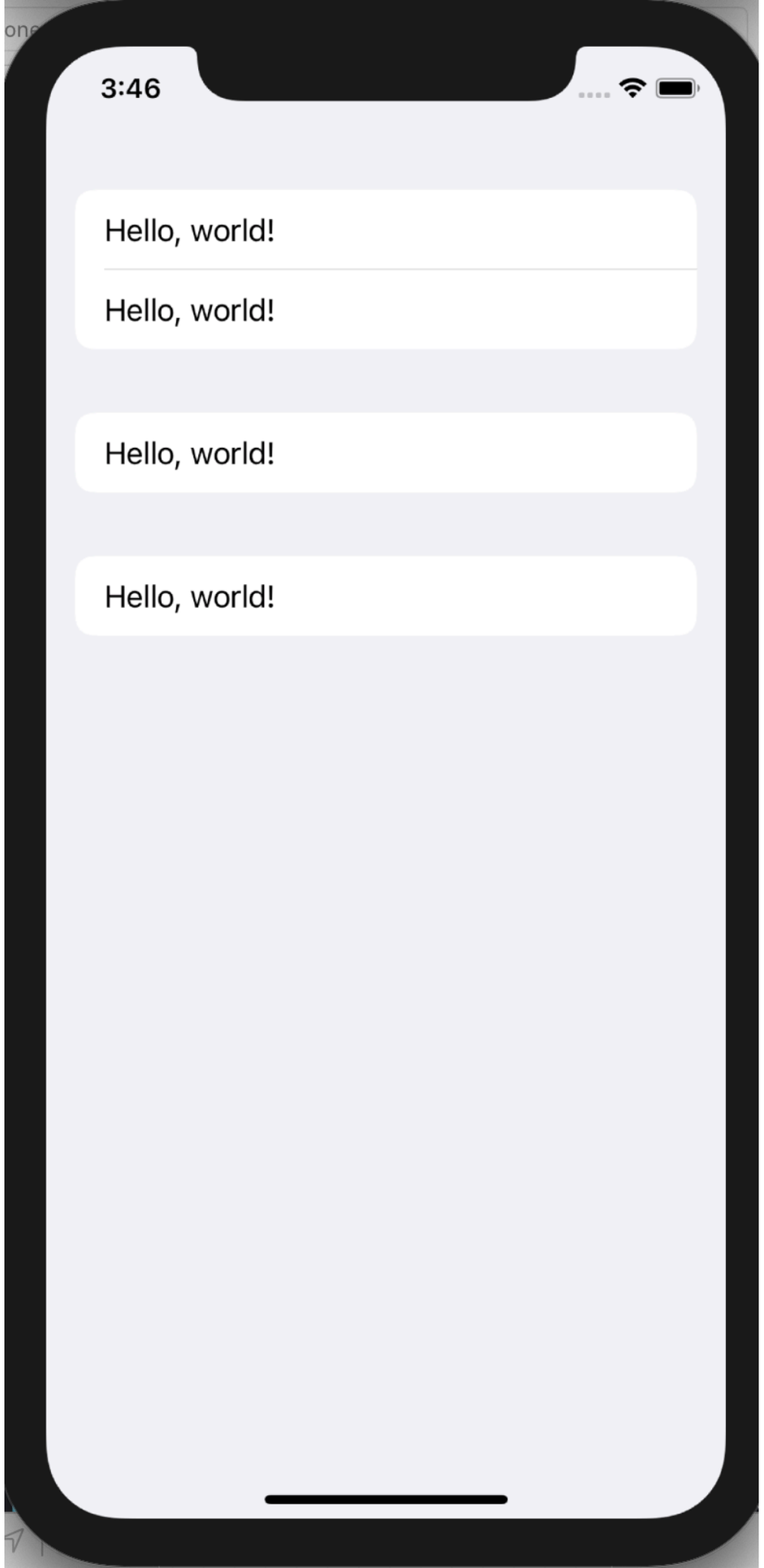
```swift
struct ContentView: View {
    var body: some View {
        Form {
            Section{
                Text("Hello, world!")
                Text("Hello, world!")
                Text("Hello, world!")
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
                Text("Hello, world!")
            }

        }
    }
}
```



Three Sections

```
struct ContentView: View {
    var body: some View {
        Form {
            Section{
                Text("Hello, world!")
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
        }
    }
}
```


3:46
Hello, world!
Hello, world!
Hello, world!
Hello, world!

設定狀態變數

狀態變數
decStr代表輸入的十進位字串

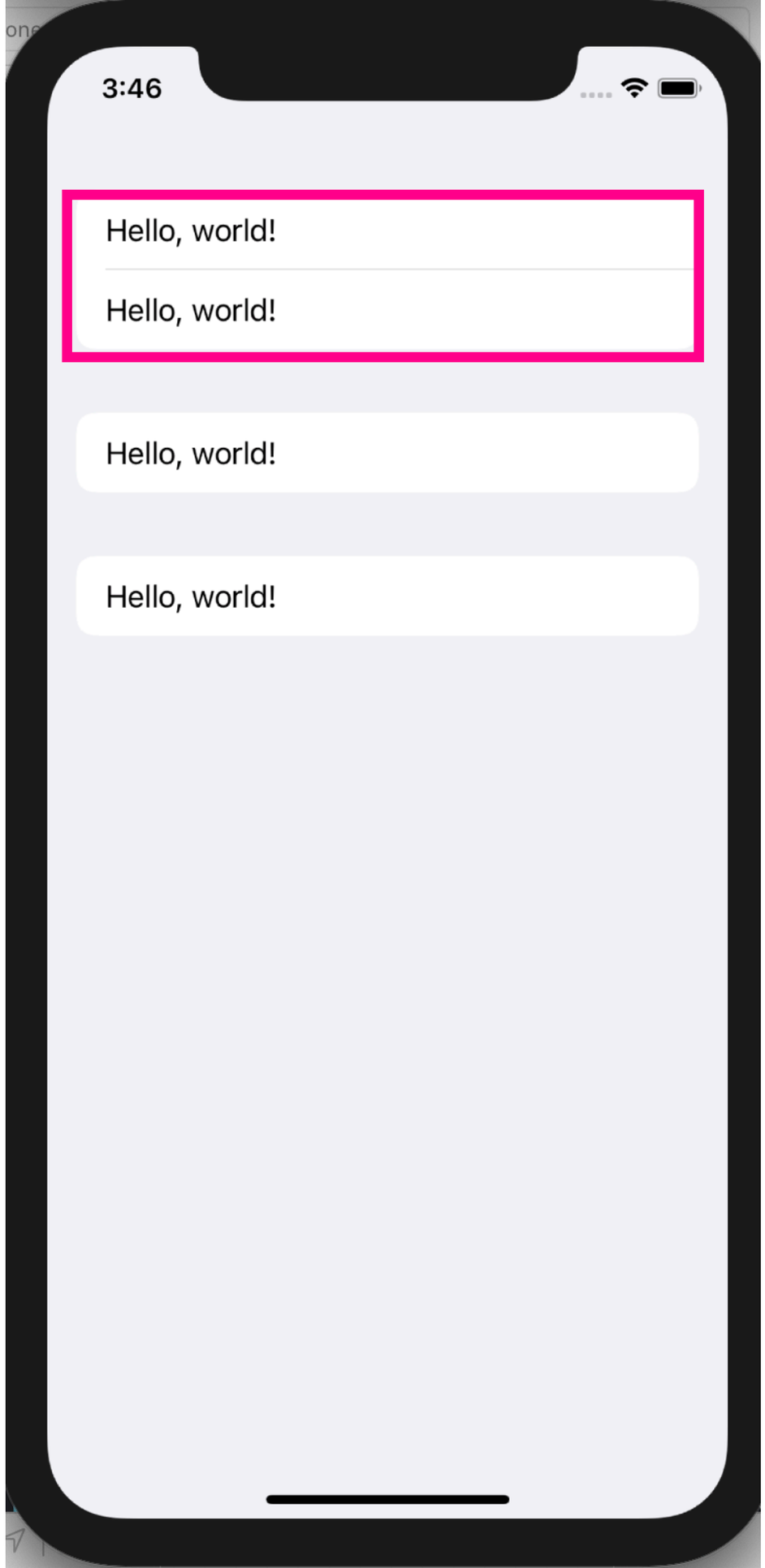
```
struct ContentView: View {
    @State private var decStr = ""
    @State private var tipRadix = 0
```

狀態變數
tipRadix代表進位制選擇，0代表二進位

```swift
struct ContentView: View {
    var body: some View {
        Form {
            Section{
                Text("Hello, world!")
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
        }
    }
}
```

宣告private
狀態變數
decStr
tipRadix

修改第一個Section Text and Textfield

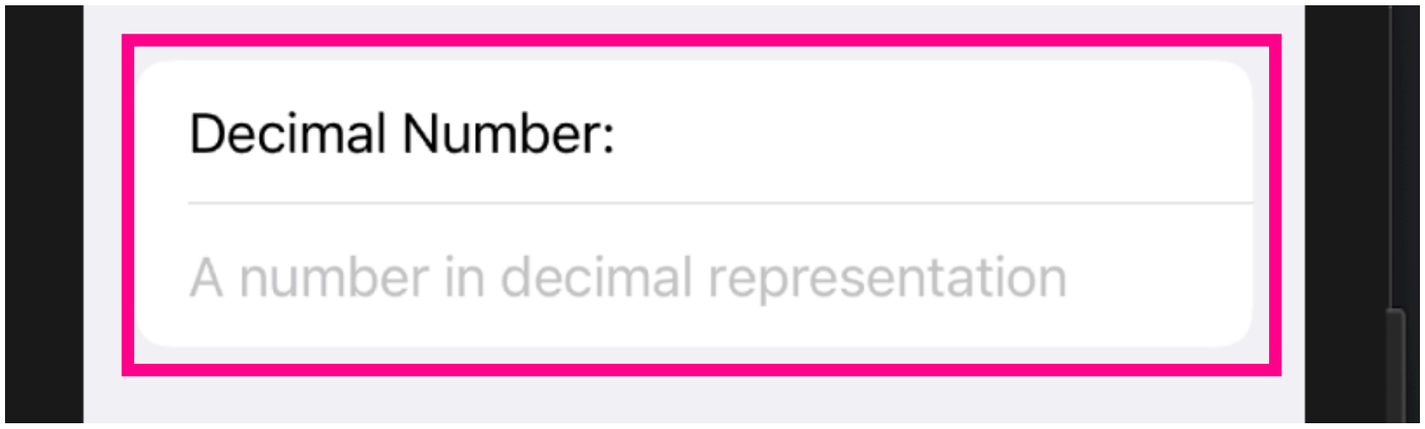


修改第一個Section的顯示內容：改變第一個文字欄位的顯示內容外，並將第二個文字欄位，改為TextField，讓使用者輸入字串

```
Section{
    Text("Decimal Number:")
    TextField("A number in decimal
        representation", text: $decStr)

}
```

連結text到狀態變數，decStr，使用者輸入的字串會自動儲存到decStr中


Decimal Number:
A number in decimal representation

修改第三個Section Text

顯示變數binAfterTip的內容

```
        Section{
            Text("Result: " + String(binAfterTip))

        }
```

宣告字串變數
binAfterTip
將十進位轉為二進位

```swift
let decNum = 22
let bStr = String(decNum, radix: 2)
print(bStr) // prints "10110"
```

```swift
let decNum = 22
let bStr = String(decNum, radix: 2)
print(bStr) // prints "10110"
```

```
22
"10110"
"10110\n"
```

只需改變radix，就可以改變，進位制轉換

```
let decNum = 22
let bStr = String(decNum, radix: 16)
print(bStr) // prints "10110"
```

```
let decNum = 31                                  31
let bStr = String(decNum, radix: 16)             "1f"
print(bStr) // prints "10110"                    "1f\n"
```

```swift
struct ContentView: View {
    @State private var decStr = ""
    @State private var tipRadix = 0
    var binAfterTip: String{
        let decNum = Int(decStr) ?? 0
        let radixNum = 2
        let bStr = String(decNum, radix: radixNum)
        return bStr
    }
```

closure具備回傳功能，當decStr內容改變時回傳值bStr也會改變

設定為二進位轉換

Decimal Number:

34

Result: 100010

修改第二個Section 設計Picker，讓使用者選擇進位制

選擇器的顯示文字

設定為"Tip Select"

將代表不同選擇的tag值，與狀態變數tipRadix連結

```
Section{
    Text("Tip Radix:")
    Picker("Tip Select", selection:
        $tipRadix){
    Text("2").tag(0)
    Text("4").tag(1)
    Text("8").tag(2)
    Text("16").tag(3)

}
.pickerStyle(SegmentedPickerStyle())
}
```

```
Section{
    Text("Tip Radix:")
    Picker("Tip Select", selection:
        $tipRadix){
    Text("2").tag(0)
    Text("4").tag(1)
    Text("8").tag(2)
    Text("16").tag(3)

}
.pickerStyle(SegmentedPickerStyle())
}
```

不同的tag值的顯示文字

選擇二進位 tag值為0
選擇四進位 tag值為1
選擇八進位 tag值為2
選擇十六進位 tag值為3

Tip Radix:

2 | 4 | 8 | 16

Tip Radix:

2 | 4 | 8 | 16

Tip Radix:

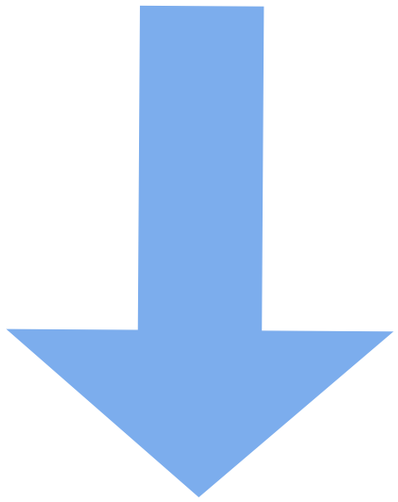
2 | 4 | 8 | 16

Tip Radix:

2 | 4 | 8 | 16

設計變數binAfterTip的Return值

```
let radixNum = 2
```



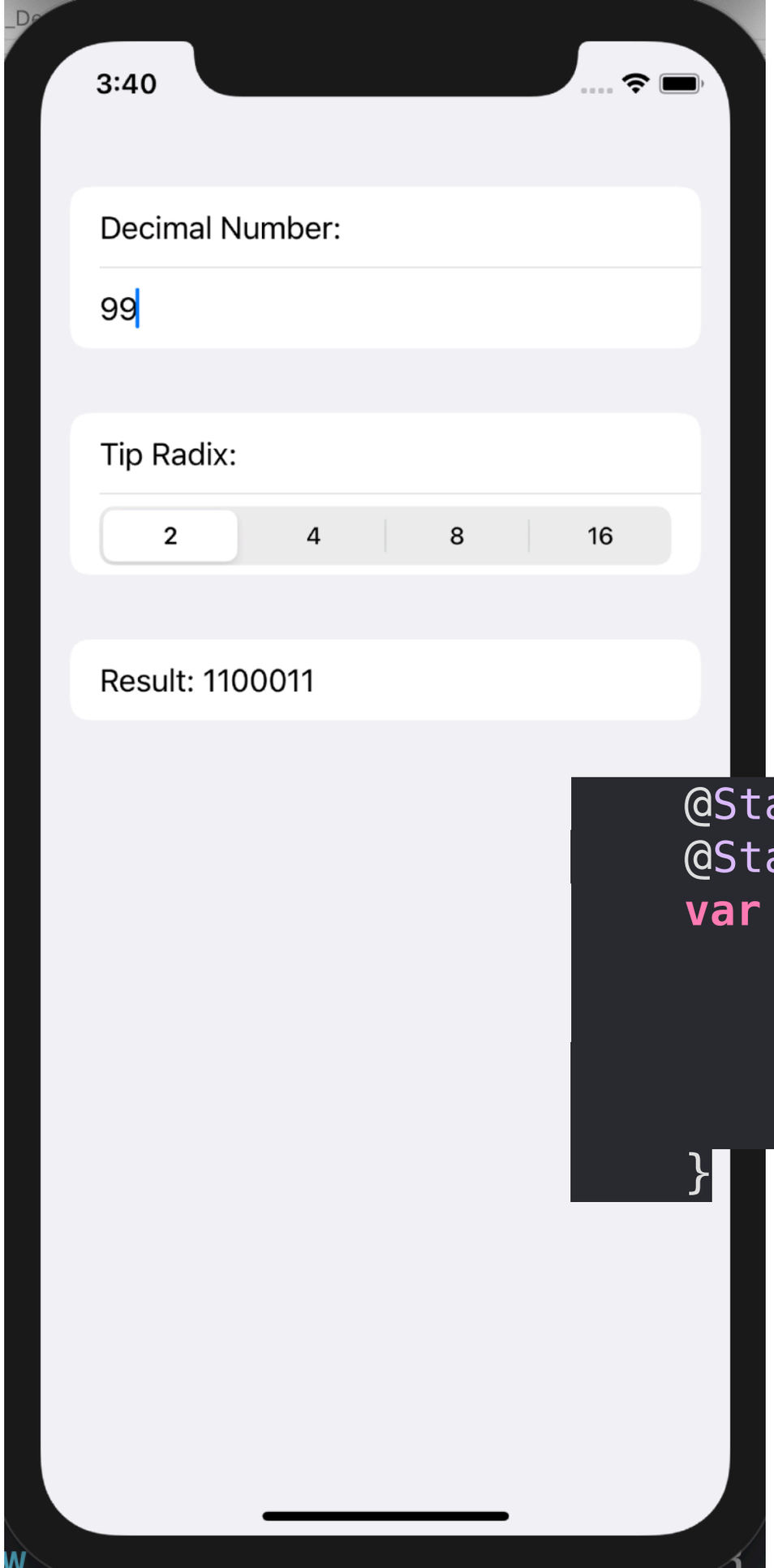
先將tipRadix轉為Double，再將內容加1

```
let radixNum = Int(pow(2,Double(tipRadix)+1))
```

取2的次方數

轉換為整數

```swift
    @State private var decStr = ""
    @State private var tipRadix = 2
    var binAfterTip: String{
        let decNum = Int(decStr) ?? 0
        let radixNum = Int(pow(2,Double(tipRadix)+1))
        let bStr = String(decNum, radix: radixNum)
        return bStr
    }
```



三個變數分別對應到
十進位字串輸入
進位制選擇器變動
轉換結果輸出

```
    @State private var decStr = ""
    @State private var tipRadix = 2
    var binAfterTip: String{
        let decNum = Int(decStr) ?? 0
        let radixNum = Int(pow(2,Double(tipRadix)+1))
        let bStr = String(decNum, radix: radixNum)
        return bStr
    }
```

改變
十進位字串輸入
就會改變
轉換結果

改變
進位制選擇器
也會改變
轉換結果


3:40
Decimal Number:
63
Tip Radix:
2
4
8
16
Result: 3f
3.10