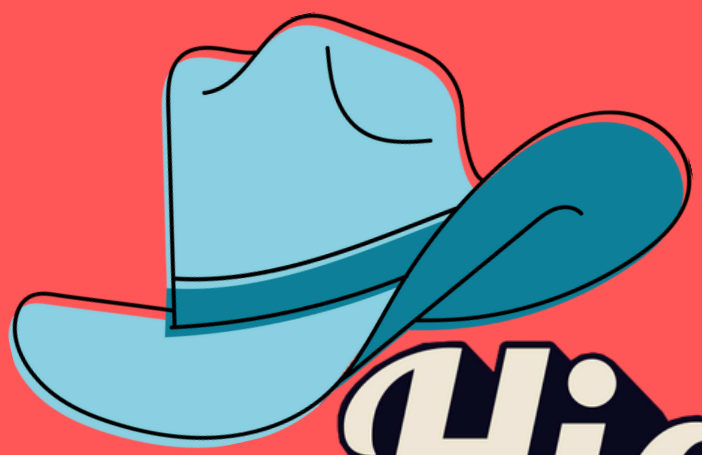

Higher Order Function

IN SWIFT

SWIPE →

by Shubam Gupta

👉 What is a Higher-Order Function?

In Swift, a higher-order function is a function that:

- Takes another function (closure) as a parameter, or
- Returns a function

“Swift uses them heavily to write clean, readable, and expressive code.”

🔹 Why should iOS developers care?

- Less boilerplate code
- More readable logic
- Functional & declarative style
- Widely used in SwiftUI, Combine, and collections

Most Common Higher-Order Functions in Swift.

- **map – Transform elements**

Used when you want to change each element.

```swift
let numbers = [1, 2, 3, 4]

let squaredNumbers = numbers.map { $0 * $0 }

// Result: [1, 4, 9, 16]
```

🧠 Each element is transformed, count remains same.

- **filter – Select elements**

Used when you want to keep only matching elements.

```swift
let numbers = [1, 2, 3, 4, 5, 6]

let evenNumbers = numbers.filter { $0 % 2 == 0 }

// Result: [2, 4, 6]
```

🧠 Elements may reduce based on condition.

- **reduce – Combine into single value**

Used to combine all values into one.

```
let numbers = [1, 2, 3, 4]

let total = numbers.reduce(0) { $0 + $1 }

// Result: 10
```

🧠 Initial value + combining logic.

- **forEach – Perform action**

Used when you don’t need a return value.

```
let names = ["Swift", "iOS", "UIKit"]

names.forEach {
    print($0)
}
```

⚠️ Unlike map, **forEach** doesn’t return anything.

- **compactMap – Transform + remove nil**

Used to unwrap optionals safely.

```
let values = ["1", "2", "abc", "4"]

let numbers = values.compactMap { Int($0) }

// Result: [1, 2, 4]
```

🧠 Best way to avoid force unwraps.

Real-World iOS Example

```
let users = [
    User(name: "John", isActive: true),
    User(name: "Alex", isActive: false),
    User(name: "Sara", isActive: true)
]

let activeUserNames = users
    .filter { $0.isActive }
    .map { $0.name }

// Result: ["John", "Sara"]
```

✅ Clean ✅ Readable ✅ Expressive code

💬 Why use higher-order functions?

They improve readability, reduce loops, and promote functional programming.

💬 Are they faster than loops?

Performance is similar, but clarity is the real win.

💬 Can Higher-Order Functions cause memory issues?

Yes, if closures capture self strongly.

```
items.forEach { [weak self] item in
    self?.process(item)
}
```

🧠 One-Line Summary

Higher-order functions help write cleaner, safer, and more expressive Swift code.

THANK YOU
BY SHUBAM GUPTA