Get Started

Outline

1. Overview
2. Basic syntax
3. Compare `let` and `var`

1. Overview

1. What is Swift?
2. Why Swift?

1.1 What is Swift?

- Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns.

- Developed by Apple Inc for iOS, iPadOS, macOS, watchOS, tvOS, Linux and z/OS, Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C written for Apple products.

1.2 Why Swift?

- Swift programming closely resembles natural English because of its good syntactic sugar.

```swift
// Swift
let count = 10
var price = 23.55

let firstMessage = "Swift is awesome. "
let secondMessage = "What do you think?"
var message = firstMessage + secondMessage

print(message)
```

```objc
// Objective-C
const int count = 10;
double price = 23.55;

NSString *firstMessage = @"Swift is awesome. ";
NSString *secondMessage = @"What do you think?";
NSString * message = [NSString stringWithFormat:@"%@%@"
, firstMessage, secondMessage];

NSLog(@"%@", message);
```

1.2 Why Swift?

- Swift supports ARC for all APIs with Objective-C only supports within the Cocoa API.

- Swift switches from static libraries to dynamic libraries.

2. Basic syntax

1. Define variables
2. Define functions

2.1 Define variables

```swift
// Swift
let count = 10
var name = "John"
let explicitDouble: Double = 70
```

2.2 Define functions

```swift
// Swift
func getString(from array: [String], with index: Int) -> String {
    return array[index]
}
```

3. Compare `let` and `var`

- Both **`let`** and **`var`** are used when defining variables.
- **`let`** is used for defining constants which **do not** change their value.
- **`var`** is used for defining mutable variables which **do** change their value.
- Swift strongly encourages you to use constants wherever possible because it's safer: if you say "this value will never change," then Swift will refuse to let you change it even by accident.

3. Compare `let` and `var` (cont)

```
let x = 10
x = 20 // Cannot assign to value: 'x' is a 'let' constant
// Fix-it: Change 'let' to 'var' to make it mutable

var y = 20 // Variable 'y' was never mutated; consider changing to 'let' constant
print(y)
```

Question & Answer?