

# Limited import clarification and its effect

**try! Swift Tokyo 2019**  
**Tomoya Hirano @DeNA**

# Import

# Import Declaration

1. attributes `import` module
2. attributes `import` module.`submodule`
3. attributes `import` `importKind` module.`symbolName`

# Import Declaration

1. `attributes import module`
2. `attributes import module.submodule`
3. `attributes import importKind module.symbolName`

# 1. Attributes

- Specify how to import symbols.
- Optionally
- `@testable` / `@_exported`

# @testable

- For test and development.
- Import internal method as public.

# @\_exported

- Import symbols as own code.
- Undocumented attributes

## 2. Submodule import

1. attributes `import` module

2. attributes `import` module.submodule

3. attributes `import importKind` module.symbolName



# Example: SceneKit initializer

```
MyPlayground
1 import SceneKit
2
SCNScene.init

[M] SCNScene init()
[M] SCNScene? init(coder: NSCoder)
[M] SCNScene? init(named: String)
[M] SCNScene init(url: URL, options: [SCNSceneSource.Loading
[M] SCNScene? init(named: String, inDirectory: String?, optio
[M] Void initialize()
[M] Void performSelector(inBackground: Selector, with: A
[M] Int index(ofAccessibilityElement: Any)
```

# Example: SceneKit initializer

MyPlayground

```
1 import SceneKit
2 import SceneKit.ModelIO
3
4 SCNScene.init|
```

M	SCNScene	<u>init</u> ()
M	SCNScene?	<u>init</u> (coder: NSCoder)
M	SCNScene?	<u>init</u> (named: String)
M	SCNScene	<u>init</u> (mdlAsset: MDLAsset)
M	SCNScene	<u>init</u> (url: URL, options: [SCNSceneSource.LoadingOptions])
M	SCNScene?	<u>init</u> (named: String, inDirectory: String?, options: [SCNSceneSource.LoadingOptions])
M	Void	<u>initialize</u> ()
M	Void	<u>performSelector</u> ( <u>in</u> Background: Selector, <u>with</u> :

# 3. Kind import

1. attributes `import` module

2. attributes `import` module.submodule

3. attributes `import importKind` module.symbolName

# 3. Kind import

struct	class	enum	protocol
typealias	func	let	var

# 3. Kind import

- Individual import is only top-level.
- Can not import separately of overloaded functions.



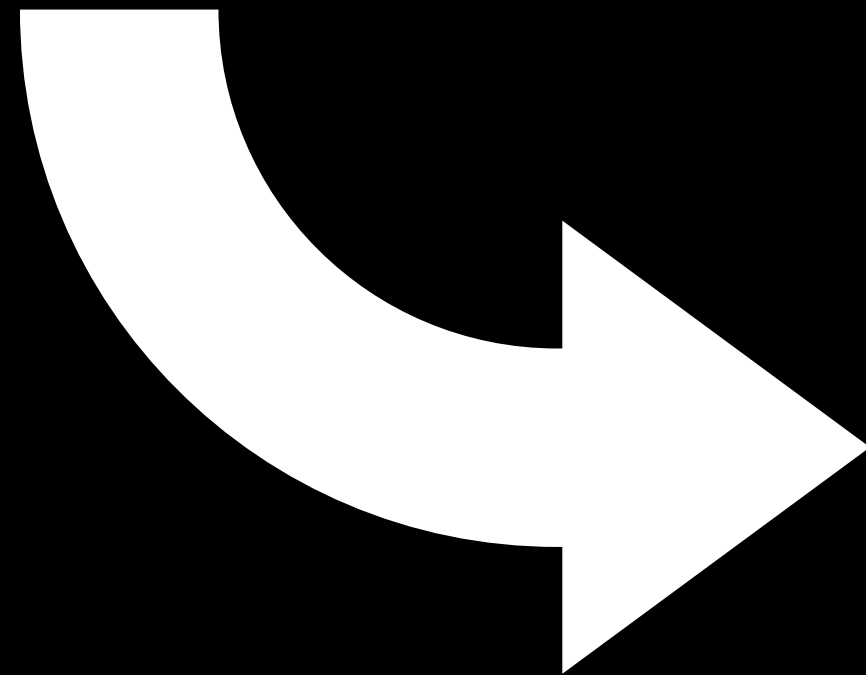
# Example: Conflict method name

```
NamespaceConflict > Sources > Fox > Fox.swift > No Selection
1 import Foundation
2
3 public func printEmoji() {
4     print("🐱")
5 }
6
```

```
NamespaceConflict > Sources > Cat > Cat.swift > printEmoji()
1 import Foundation
2
3 public func printEmoji() {
4     print("🐱")
5 }
6
```

```
1 import Cat
2 import Fox
3
4 printEmoji()
5
```

! Ambiguous use of 'printEmoji()'



```
1 import Cat
2 import func Fox.printEmoji
3
4 printEmoji()
5 |
```



Program ended with exit code: 0

# Import Options

1. Attributes
2. Submodule import
3. Kind import



**Can build time be shortened?**

# 4. Measuring the build times

- made 100,000 methods.
- `import Core` vs `import func Core.function0`
- Try 10 times

# Build time

	Basic Import	Kind Import
Average	277.645s	285.105s
Max	309.46s	312.99s
Min	273.48s	265.25s

**Can binary size be small?**

# Binary size

	Basic Import	Kind Import
Filesize	7336KB	7336KB
Md5	0b10a59ce0a75decbb 2f120520c4fbc1	0b10a59ce0a75decbb 2f120520c4fbc1

# Pros & Cons

	Build time	Binary Size	Namespace
Simple Import	Same	Same	Feel free
Kind specify	Same	Same	Necessary