The `import` keyword in Swift is used to make the symbols defined in another module available to your code. Modules are self-contained units of code that can be reused and shared.

To import a module, you simply use the `import` keyword, followed by the module name. For example, the following code imports the `UIKit` module, which defines the user interface classes for iOS apps:

```
Swift
import UIKit
```

Once you have imported a module, you can use the symbols that it defines in your code. For example, the following code creates a `UILabel` object, which is a class defined in the `UIKit` module:

```
Swift
let label = UILabel()
```

You can also import specific symbols from a module. To do this, you use the `import` keyword, followed by the module name and a period, followed by the symbol name. For example, the following code imports the `sin()` function from the `math` module:

```
Swift
import math.sin
```

Once you have imported a specific symbol, you can use it in your code without having to specify the module name. For example, the following code calculates the sine of 90 degrees:

```
Swift
let sine = sin(90)
```

Importing modules is a powerful way to organize your code and make it more reusable. By using modules, you can break down your code into smaller, more manageable pieces. You can also share modules with other developers, so that they can use your code in their own apps.

Here are some additional tips for using `import` in Swift:

- You can use `import` to import all of the symbols defined in a module, or you can import specific symbols from a module.
- You can use `import` to import modules from the standard library, or you can import modules that you have created yourself.
- You can use `import` to import modules that are available on the device, or you can import modules that are available on a remote server.

Overall, `import` is a valuable tool for any Swift developer. It allows you to organize your code and make it more reusable.