Internal and external parameter names in Swift are used to differentiate between the names of parameters as they are used inside a function and the names of parameters as they are used outside of a function.

Internal parameter names are the names of parameters as they are used inside a function. They are used to access the values of the parameters within the function body.

External parameter names are the names of parameters as they are used outside of a function. They are the names that are used to pass values to the function when it is called.

By default, the internal and external parameter names are the same. However, you can override the external parameter names by specifying them in the function definition.

For example, the following function definition has two parameters with different internal and external names:

Swift

```swift
func greet(person name: String) {
  print("Hello, \(name)!")
}
```

The internal parameter name for the first parameter is `name`, and the external parameter name is `person`.

To call this function, you would use the external parameter name, `person`:

Swift

```swift
greet(person: "Alice")
```

This would print the following output to the console:

```
Hello, Alice!
```

Using internal and external parameter names can make your code more readable and maintainable. For example, it can be helpful to use different parameter names if the internal names are not descriptive or if the external names are more meaningful to the caller of the function.

Here are some additional tips for using internal and external parameter names:

- Use internal parameter names to make your code more readable and maintainable.
- Use external parameter names to make your code more readable and understandable to the caller of the function.
- Use consistent formatting when using internal and external parameter names.
- Use descriptive names for both internal and external parameter names.

I hope this helps!