

Getting the hang of APIs or exploring module integration using an interactive session with the interpreter. This should come prior to writing tests and code.

Writing tests may be premature in a statically typed languages until you have experimented with the type signatures you will use.

Put both the class and its companion inside some wrapper object.

Certain aspects of coding with packages and visibility modifiers.

AnyRef.##

Pass the possibly `null` reference to the `Option` factory method. It will return `None` or `Some` as appropriate, which can be pattern matched against.

Treating `Option` as a collection, using `for` expressions and methods like `foreach`, `map`, and `filter`.

Another Option of the same type.

Enforcing the existence of the `canEquals` method correct implementations of `equals` in polymorphic classes require.