**Reference vs Value**

In the Swift programming language, there are two categories of type: Reference type and Value type. Reference type is where a single copy of data is shared, and its type is usually a class. Value type holds a unique copy of the data, such as enum or struct. Reference type has a shared instance where it can be passed around and referenced by various variables. Value type, however, is completely different from reference types.

The difference between a Reference type and a Value type is that a Reference type has a pointer with another memory that keeps a hold of the real data, and this is stored in the heap. Whereas a Value type keeps a hold of its data in its own memory allocation, and it’s stored in the stack.

Swift portrays reference types as a class, which is similar to Objective-C. So, everything that is inherited from NSObject, it is stored as a reference type. There are various kinds of value types, for example, tuples, struct and enum. Object-C utilises value types in number literals like NSInteger or C structures like CGPoint. Swift standard library uses Value types exclusively.

Diagram

Description automatically generated