A value type is a data type that holds data within its own memory allocation. All structures are considered value types, even if they contain reference type members. Other examples of value types are booleans, characters, arrays, strings, dictionaries and all numeric data types to name a few. Value types are stored on the stack and enhance the performance speed of a program due to the quick recovery of value type data.

A reference type in swift shares a single copy of its data, causing every new instance to point to the same copy, leading to the same address in the memory heap. Reference types are known for conserving memory since it does not allocate new memory for every instance of a variable. Some examples of reference types are classes, closures, functions and objects.

Value types are used for their speed and reference types are known for their optimal use of memory. Programs that are intended for performance speed use predominantly value types and try to stay away from reference types when possible because reference types focus on memory optimization but cause the performance speed to suffer. However, when writing a program that would need to optimize memory usage, reference types are the ones that are used. Value types make program execution fast, but uses up a lot of memory as a consequence for the speed.

Value types are also utilized in iOS, Swift and ObjC for the ability to reason with your code easier and not worry about other parts of code changing your variables. Since reference types are pointers to one value, if the value is changed anywhere in your code without your knowledge or awareness, it could be very difficult to spot.