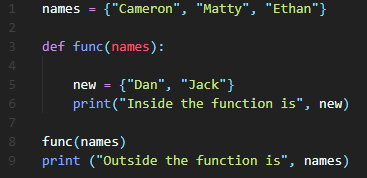
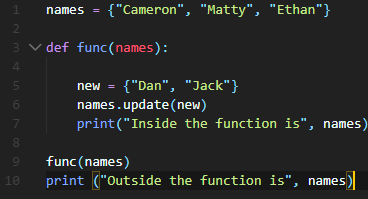
Pass by Value

In pass-by-value, the function receives a copy of the argument objects passed to it by the caller, stored in a new location in memory. The function then effectively supplies its own box to put the value in, and there is no longer any relationship between either the variables or the objects referred to by the function and the caller. The objects happen to have the same value, but they are totally separate, and nothing that happens to one will affect the other.

In pass by value, when a value is altered within a subroutine, it is only changed within the subroutine. This is because it is a local variable inside the subroutine, meaning it is not changed outside.

Pass by Reference

In pass-by-reference, the box (the variable) is passed directly into the function, and its contents (the object represented by the variable) implicitly come with it. Inside the function context, the argument is essentially a complete alias for the variable passed in by the caller. They are both the exact same box, and therefore also refer to the exact same object in memory.

In pass by reference, when a value is altered within a subroutine, it is also altered outside the subroutine. This is because it is essentially treated as a global variable, so if it is changed anywhere it changes everywhere.