1. It is important for Python to define an order in which the decorators are applied in order to define a specific method in which these decorators are applied, to prevent randomness in how the program is processed. For example, if you created two decorators, one which incremented the first argument by 1, and a second that multiplies two to the first argument, the end result produced by Python would be different based on the order in which the decorators are processed.
2. The designers of Python chose to apply the decorators in the reverse order in which they're specified in order to keep the same process in which someone would pass the function into another function. Because it makes more sense that the last decorator would be the first decorator that the function is passed into, it is the last decorator that is applied first, and then it continues in reverse order.