General

Custom comparison operators

To explain how Python must compare two objects of a self-defined data type (*class*), a specific implementation for the comparison operators must be provided. This can be done by overloading the following magical methods:

method	operator
___lt___	$<$
___le___	$\leq$
___gt___	$>$
___ge___	$\geq$
___eq___	$=$
___ne___	$\neq$

Please note that in most cases you'll have the opportunity to define most of these comparison operators based on the other comparison operators. For example, two object are different if they are not equal.

Gebruik van `self`

If you work with classes, you need to make a distinction between two kinds of variables. There are object properties that can be referenced in all class methods and there are local variables of methods that are only acccessible in the method where they are defined. Only the names of the object properties need to be prefixed with `self`. Variables that are local to a method (the local variables) do not need to be prefixed with `self`, and its considered very bad programming style if you do so.

Initialize object properties in intialisation method

Before you start with the implementation of a class, you must first determine which properties the objects of the class will have. Each of these properties will correspond to a variable that is prefixed with `self.`. These variables describe the internal state of the individual objects and can be addressed in all class methods. It's always a good idea to define the object properties in the `__init__` method, where you assign them an initial value.