

Algemeen

Zelfgedefinieerde vergelijksoperatoren

Om twee objecten van een zelfgedefinieerd type te kunnen vergelijken, moet een specifieke implementatie voor de vergelijksoperatoren voorzien worden. Dit kan door volgende magische methodes te overladen:

methode	operator
<code>__lt__</code>	<code><</code>
<code>__le__</code>	<code><=</code>
<code>__gt__</code>	<code>></code>
<code>__ge__</code>	<code>>=</code>
<code>__eq__</code>	<code>=</code>
<code>__ne__</code>	<code>!=</code>

Let er op dat het in de meeste gevallen volstaat om sommige van deze vergelijksoperatoren te implementeren op basis van andere vergelijksoperatoren. Zo zijn twee objecten doorgaans verschillend als ze niet gelijk zijn aan elkaar.

Gebruik van `self`

Als je werkt met klassen, dan is het belangrijk dat je onderscheid maakt tussen twee soorten variabelen. Er zijn objecteigenschappen die kunnen aangesproken worden in alle klassemethoden en er zijn lokale variabelen die enkel toegankelijk zijn binnen de methode waarin ze gedefinieerd worden. Enkele de namen van de objecteigenschappen moeten voorafgegaan worden door `self`. De lokale variabelen van een methode moeten niet voorafgegaan worden door `self`, en het getuigt van een zeer slechte programmeerstijl als je dat toch doet.

Objecteigenschappen initialiseren in de initialisatiemethode

Voor je begint aan de implementatie van een klasse, moet je eerste bepalen welke eigenschappen de objecten van die klasse moeten hebben. Elk van die eigenschappen komt dan overeen met een variabele die voorafgegaan wordt door de prefix `self..` Deze variabelen beschrijven de interne toestand van de individuele objecten en kunnen in alle methoden van de klasse gebruikt worden. Het is altijd een goed idee om deze objecteigenschappen te definiëren in de `__init__` methode en er daar een initiële waarde aan toe te kennen.