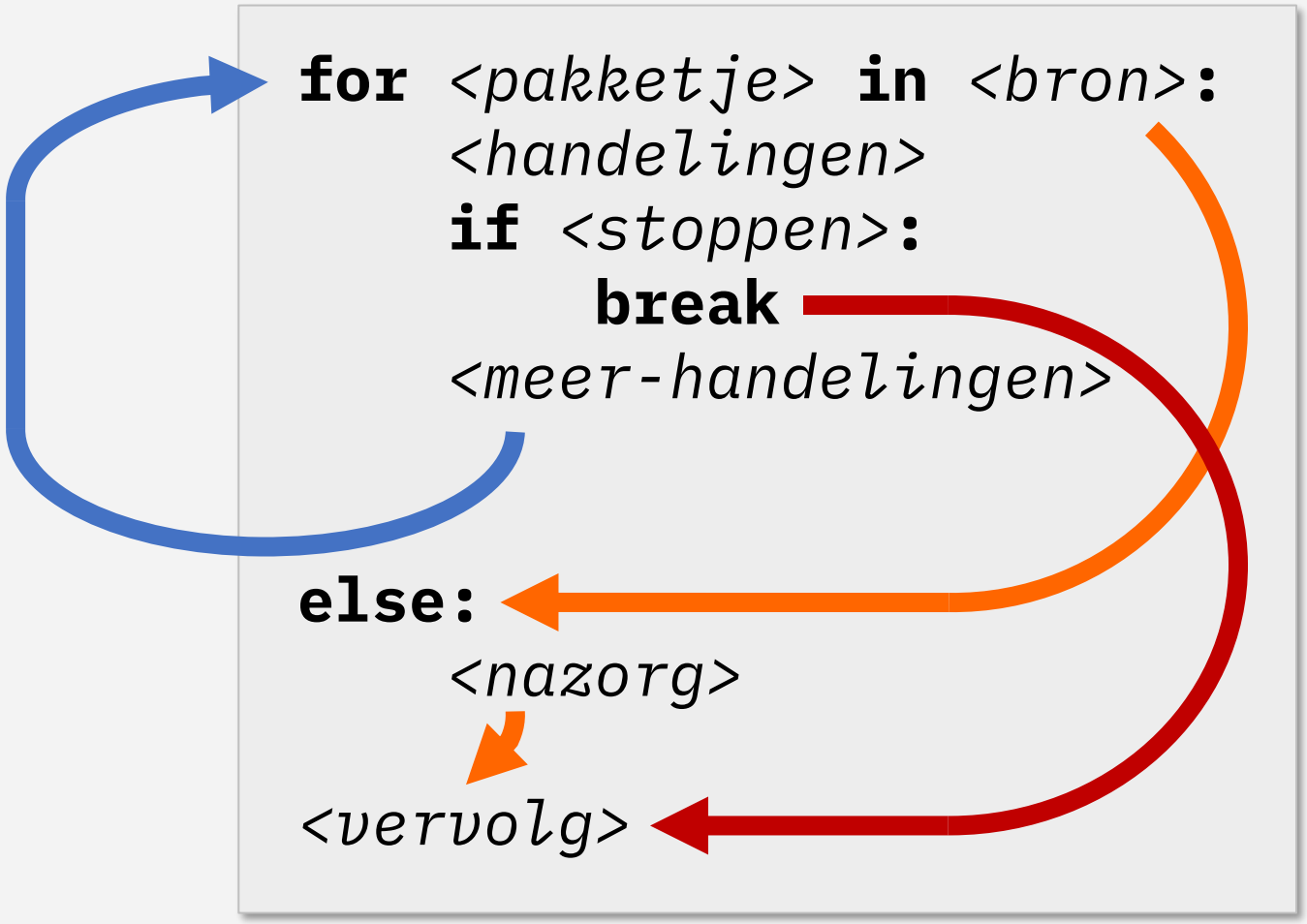


EEN BOOM VAN ITERIOLEN IN PYTHON

Meindert Meindertsma — 11 november 2026

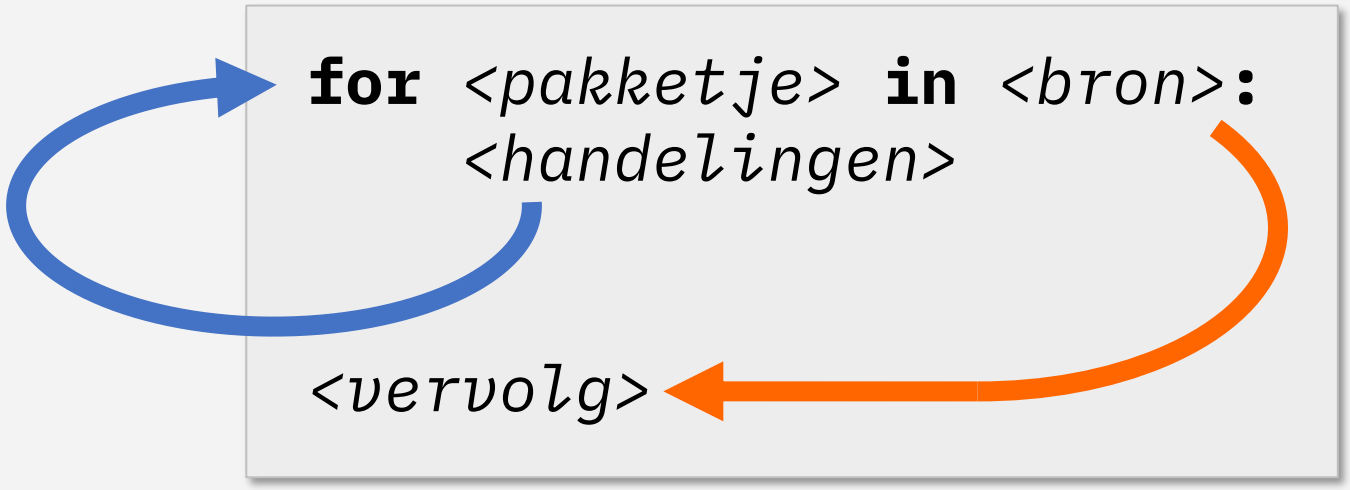
Hoofdstuk 7: **for … break … else …**



wanneer er niets meer leverbaar is

wanneer <stoppen> True is

Waar het in hoofdstuk 23 om draait



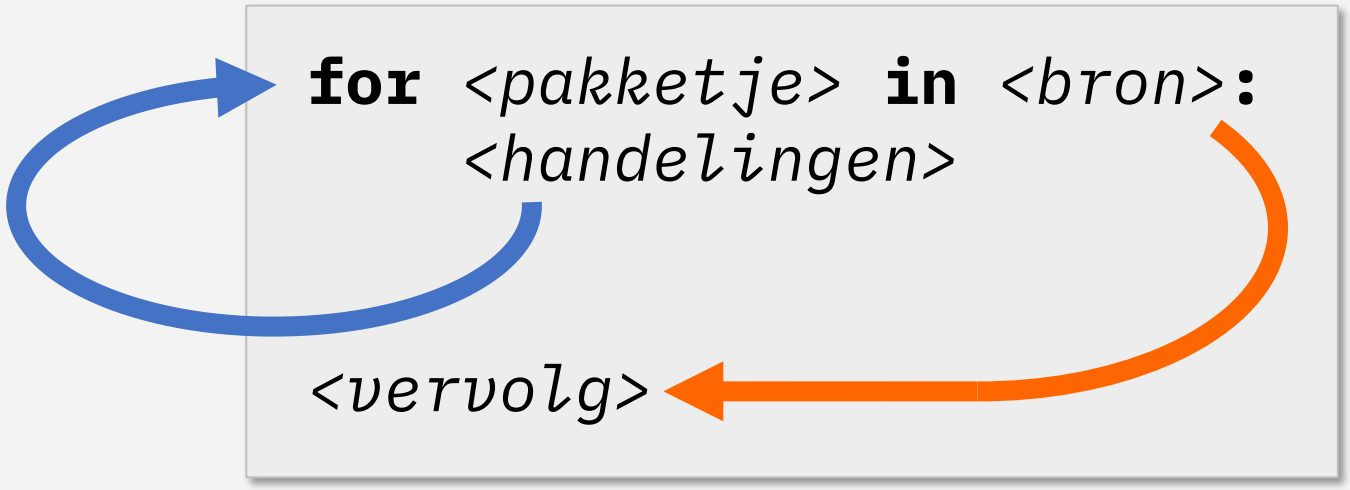
Het object *<bron>* representeert:

- een aanwezige voorraad van ‘pakketjes’, of
- een maker of verwerker van ‘pakketjes’ die op afroep zullen worden geproduceerd dan wel op andere wijze beschikbaar gesteld.

Technisch bezien:

- bestaat een *<pakketje>* uit een of meer iteratievariabelen;
- is *<bron>* een **‘iterabel’** object (zie slide 5 en verder).

Hoe krijgt het for-statement zijn pakketjes?



for vraagt *<bron>* via haar methode **__iter__** om haar **'iterator'** beschikbaar te stellen. Deze methode geeft een *<iterator>* als returnwaarde terug.

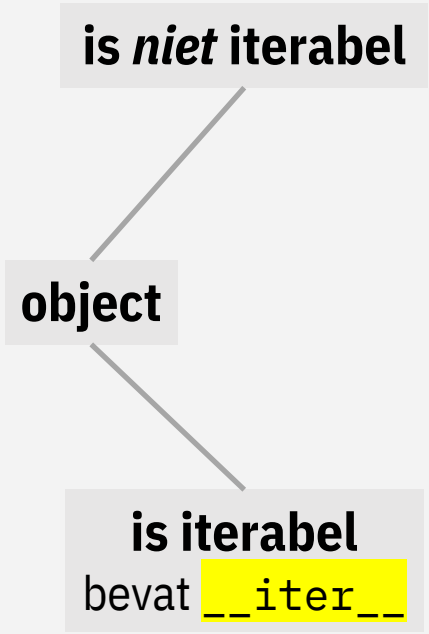
Vervolgens bestelt **for** bij die *<iterator>* via zijn methode **__next__** een 'pakketje'. Levering door *<iterator>*`.__next__()` volgt, tenzij er niets meer leverbaar is. In dat geval genereert de methode de exception **StopIteration**.

for blijft via *<iterator>*`.__next__` pakketjes afroepen totdat de exception **StopIteration** optreedt. Dat is het sein om het itereren te stoppen en de verdere verwerking over te laten aan *<vervolg>* of de eventuele **else**-tak.

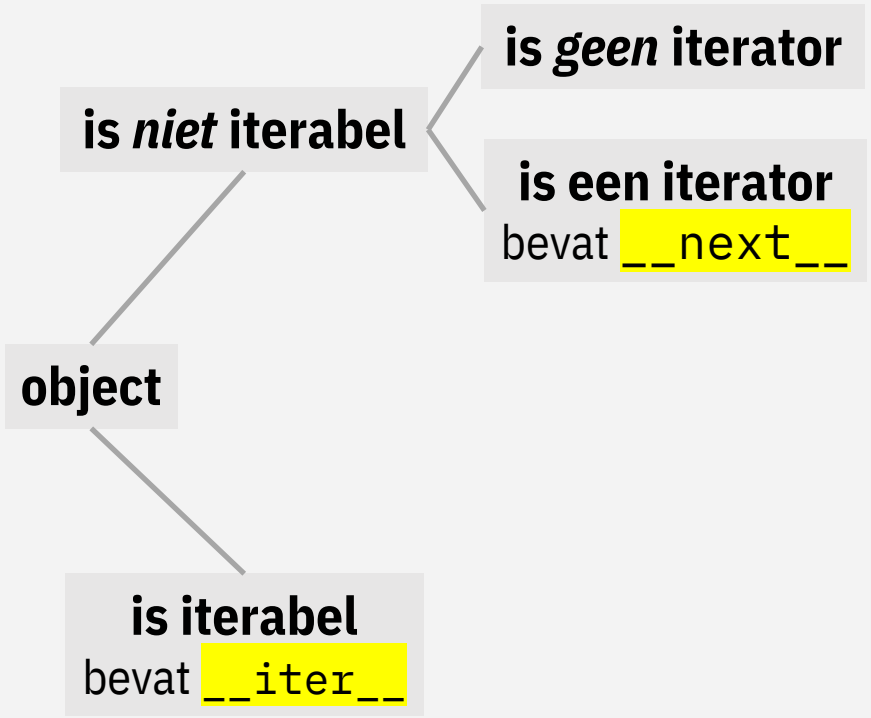
Objecten gecategoriseerd naar *protocollen*

object

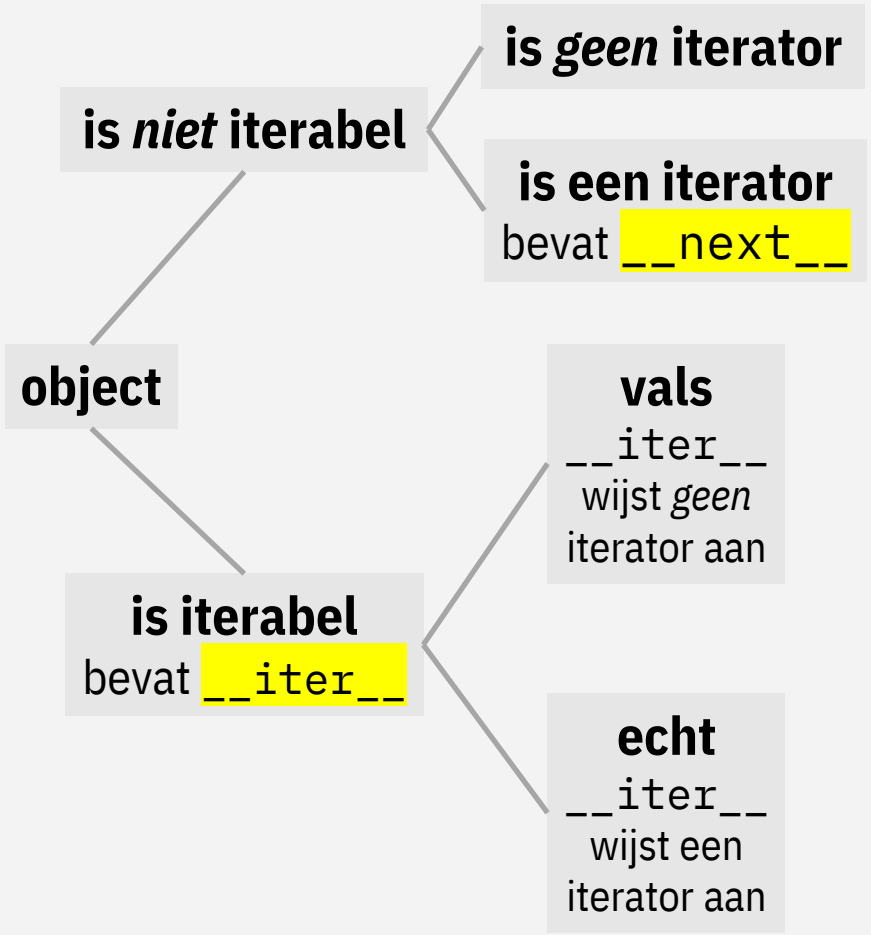
2 categorieën



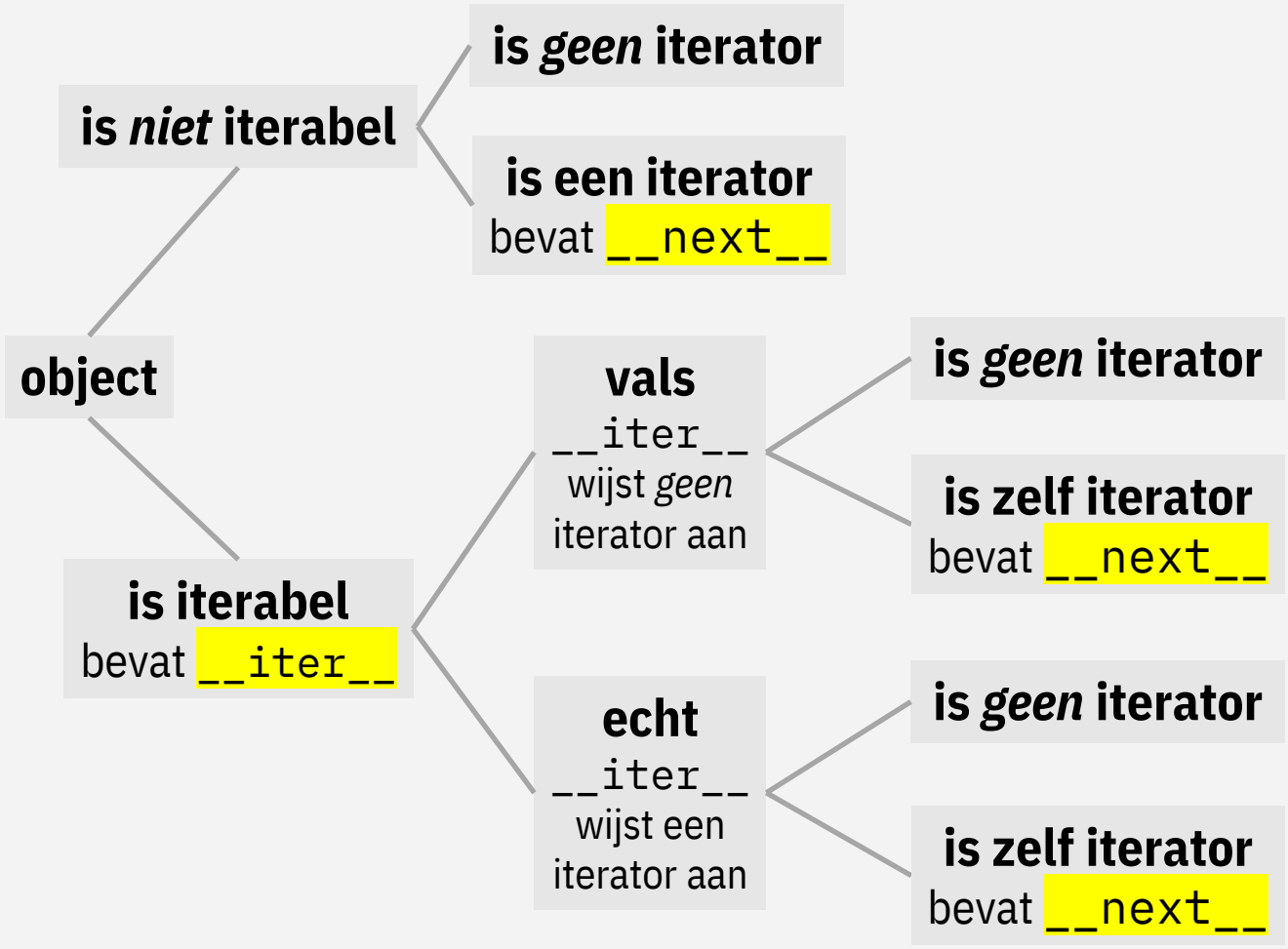
3 categorieën



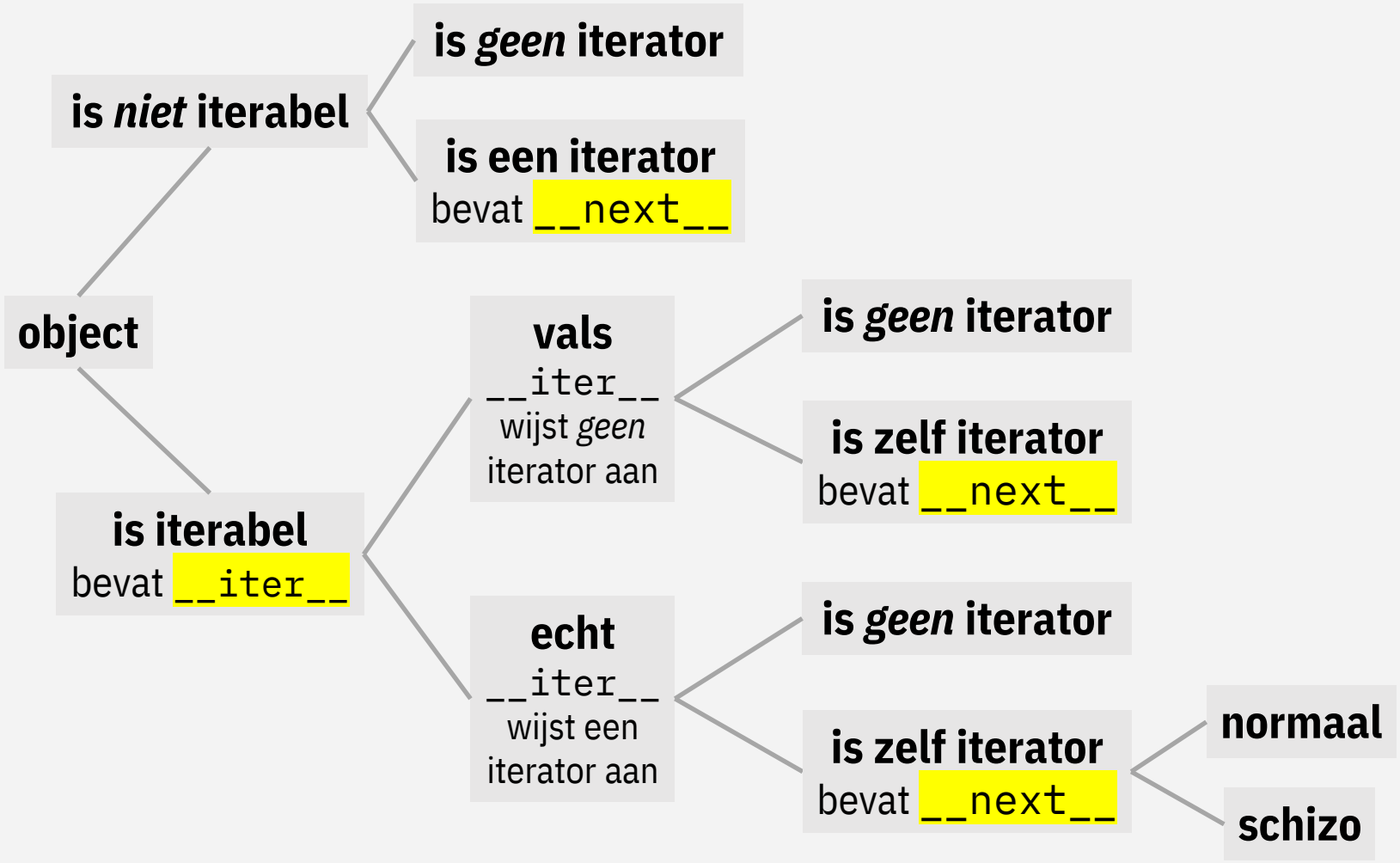
4 categorieën



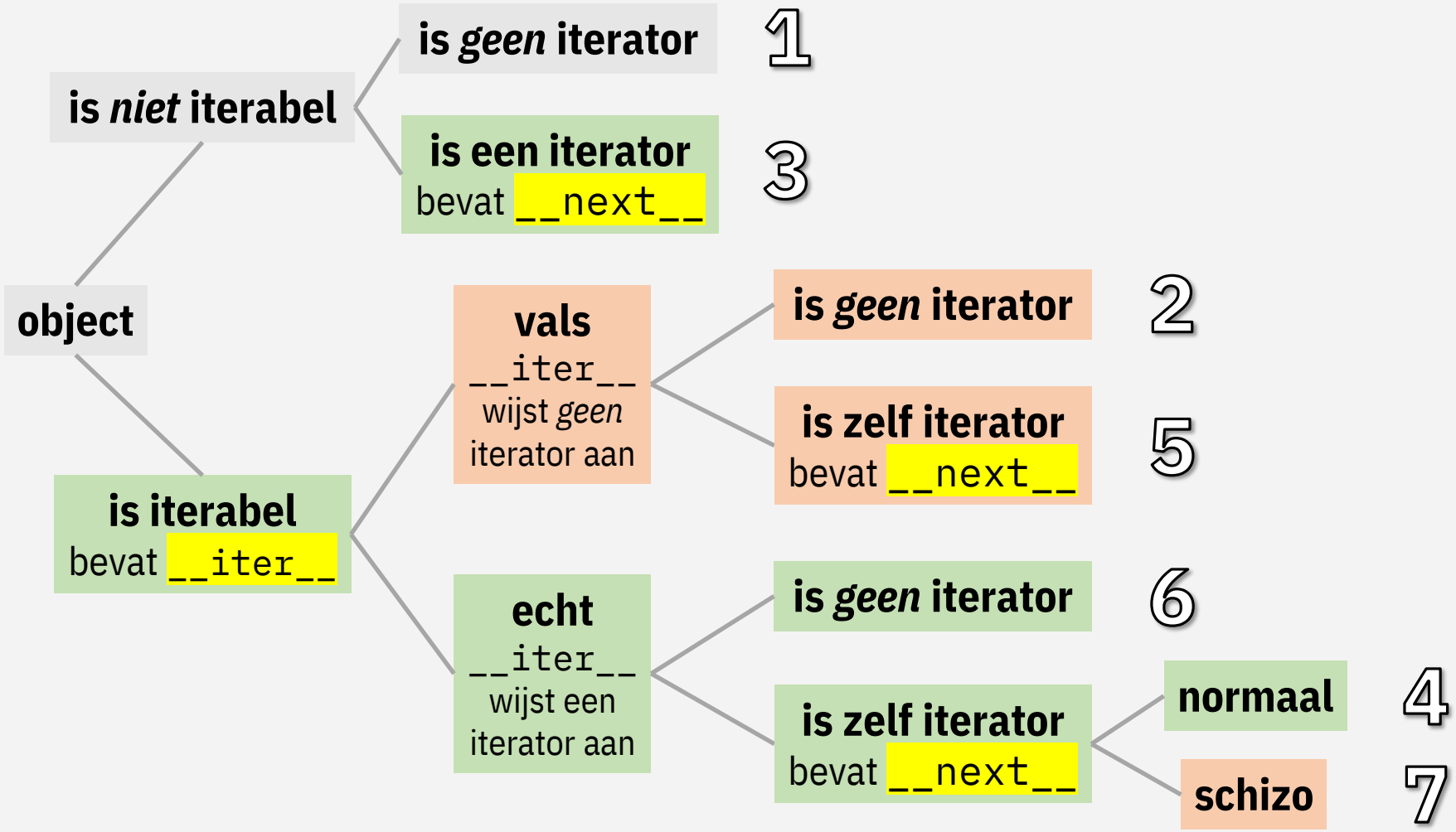
6 categorieën



7 categorieën



7 categorieën



Wat de functie **iter(**<*bron*>**)** doet

1. Kijken of <*bron*> een methode **__iter__** bevat. Indien niet en ook geen methode **__getitem__***), dan error:
   "TypeError: …… object is not iterable"
2. De methode <*bron*>**.__iter__** aanroepen. Deze retourneert een object <*iterator*> (dit kan wel of niet hetzelfde object als <*bron*> zijn).
3. Kijken of <*iterator*> een methode **__next__** bevat. Zo niet, dan error:
   "TypeError: iter() returned non-iterator of type ……"
4. De **iterator** <*iterator*> retourneren.

Vervolgens kan de functie **next** met <*iterator*> aan de slag of kan de methode <*iterator*>**.__next__** rechtstreeks worden aangeroepen.

*) De iteratievoorzieningen rond de methode **__getitem__** zijn een relict van Python 2. De methode wordt pas aangesproken als **__iter__** niet bestaat. Spronck behandelt de mogelijke rol bij iteraties via **for** niet, en ik doe dat hier evenmin.

Wat de functie **iter(***<callable>*, *<sentinel>***)** doet

1. Kijken of *<callable>* werkelijk aanroepbaar is. Indien niet, dan error: “TypeError: iter(v, w): v must be callable”
2. De meegegeven *<callable>* omzetten naar een **callable_iterator** object dat de methoden **__iter__** en **__next__** bevat.
3. De methode **__iter__()** retourneert zichzelf.
4. De methode **__next__()** roept de *<callable>* aan.
5. **__next__()** retourneert de door *<callable>* geretourneerde waarde,
6. tenzij deze waarde gelijk is aan *<sentinel>*. In dat geval genereert **__next__()** de exception **StopIteration**.

Wat de functie **next(***<iterator>[, <default>]***)** doet

1. Nagaan of *<iterator>* een methode **__next__** bevat. Zo niet, dan error: "TypeError: …… object is not an iterator"
2. De methode *<iterator>*.**__next__** aanroepen om een pakket van waarden uit dat object op te halen.
3. Deze waarden als returnwaarden doorgeven.
4. Kan **__next__()** niets meer leveren en genereert het daarom de exception **StopIteration**, dan hangt het ervan af of **next** een tweede argument *<default>* meegekregen heeft :
   a. Zo ja, dan de exception afvangen en de waarde *<default>* retourneren.
   b. Zo nee, dan de uitzonderingstoestand **StopIteration** aanhouden en zo eindigen.

Literatuur

- Pieter Spronck: *De programmeursleerling – Leren coderen met Python 3*
  versie 1.1.1, 2024
  (Nederlandse vertaling van *The Coder's Apprentice*).

- Meindert Meindertsma: *Itereren met while & for*
  18 en 22 januari 2024 (op 11 november 2026 bijgewerkt)
  (PDF “while-and-for-2026.pdf” van een PowerPoint-presentatie).
  In dit verband zijn de slides 7–10 en 13–16 het meest interessant.

- ____: *7 iteriolen in Python*
  22 januari en 28 september 2024 (op 11 november 2026 bijgewerkt)
  (PDF “iteriolen-2026.pdf” van een PowerPoint-presentatie).

- Kritische reacties op mijn producten zijn welkom (bijvoorbeeld via mijn e-mailadres **ledepot2023@ymail.com**).