Python les-materialen

# Tuples

In Python lijken tuples erg op lijsten, maar in tegenstelling tot lijsten zijn ze *immutable (onveranderlijk)*, wat betekent dat ze niet kunnen worden gewijzigd. Je zou tuples gebruiken om dingen te presenteren die niet mogen veranderd worden, zoals **dagen van de week** of **datums op een kalender**.

In deze sectie krijgen we een kort overzicht van het volgende:

1.) Tuples construeren (bouwen)  
 2.) Basis Tuple-methoden  
 3.) Onveranderlijkheid (Immutability)  
 4.) Wanneer worden tuples gebruikt?

Je hebt een intuïtie voor het gebruik van tuples op basis van wat je hebt geleerd over lijsten. We kunnen ze op dezelfde manier behandelen, met als belangrijkste onderscheid (distinction) dat tuples onveranderlijk zijn.

## Tuples construeren

De constructie van een tuple gebruikt () met elementen gescheiden/gesplit door komma’s. Bijvoorbeeld:

# Maak een tuple aan  
t = (1,2,3)

# Controleer het lengthe (len) net als een lijst  
len(t)

3

# Kan ook objecttypen combineren  
t = ('one',2)  
  
# Tonen  
t

('one', 2)

# Gebruik indexering net zoals we hadden met de lijsten gecodeerd  
t[0]

'one'

# Snijden/Sliceren net zoals een lijst  
t[-1]

2

## Basis Tuple-methoden

Tuples hebben ingebouwde methoden, maar niet zoveel als lijsten. Laten we er twee bekijken:

# Gebruik .index om een waarde in te voeren en de index terug te geven  
t.index('one')

0

# Gebruik .count om het aantal keren te tellen dat een waarde verschijnt  
t.count('one')

1

## Onveranderlijkheid/Immutability

Het kan niet genoeg benadrukt worden dat tuples immutable (onveranderlijk) zijn. Laten we eens testen hoe belangrijk deze inhoud is::

t[0]= 'change'

---------------------------------------------------------------------------  
  
TypeError Traceback (most recent call last)  
  
<ipython-input-8-1257c0aa9edd> in <module>()  
----> 1 t[0]= 'change'  
  
  
TypeError: 'tuple' object does not support item assignment

Vanwege deze onveranderlijkheid kunnen tuples niet groeien. Als een tuple eenmaal is gemaakt, kunnen we er niets meer aan toevoegen.

t.append('nope')

---------------------------------------------------------------------------  
  
AttributeError Traceback (most recent call last)  
  
<ipython-input-9-b75f5b09ac19> in <module>()  
----> 1 t.append('nope')  
  
  
AttributeError: 'tuple' object has no attribute 'append'

## Wanneer tuples gebruiken?

Je vraagt ​​je misschien af: “Waarom zou je tuples gebruiken als ze minder beschikbare methoden hebben?” Om eerlijk te zijn, tuples worden niet zo vaak gebruikt als lijsten bij het programmeren, maar worden gebruikt wanneer onveranderlijkheid noodzakelijk is. Als u in uw programma een object doorgeeft en ervoor moet zorgen dat het niet wordt gewijzigd, wordt een tuple uw oplossing. Het biedt een handige bron van gegevensintegriteit (data integrity).

Je zou nu in staat moeten zijn om tuples in je programmering te maken en te gebruiken, evenals hun immutability (onveranderlijkheid) goed genoeg te begrijpen.

Veel success met programmeren!