

Files, Lists en Tuples



BIN-OWE1

Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Linux	
2	Git	Input/output
3	Pseudocode	If/elif/else Booleans
4		For loop
5		Lists and tuples Files (CSV bestanden)
6	Flowchart	Functies
7		Strings CSV bestanden

Studiemateriaal

- Boek: “Starting Out with Python, third edition”
- Reader: “Linux voor Bio-informatici”
- Onderwijs Online
- Kom je er niet uit? → Google



Doelstellingen

- Aan het eind van deze week kan je
 - Bestanden openen in Python en deze parsen
 - Lijsten uitlezen en vullen
 - Tuples aanmaken
- Aan het eind van deze week begrijp je
 - Het verschil tussen lists en tuples
 - Hoe je een bestand kunt inlezen met behulp van Python

Inhoud

- Indices en slicing
- Lists
- Meer over strings
- Tuples

Indices

- Een index is een kaart van de posities
- Denk aan de index van een boek;
 - welk begrip kan ik op welke pagina vinden?
- Voor de computer:
 - Makkelijk: opvragen welk item er op positie [3] staat
 - Moeilijk: opvragen op welke positie “a” staat
- Voor ons:
 - Beiden even makkelijk
 - `s[0]` en `s.index("e")`

Index syntax

`<object> [<index>]`

`<object> [start:stop]`

- Enkel element:

`string[1]`

- Slice:

`string[2:4]`

Positieve en Negatieve indices

- Positieve indices (tellen van voor naar achter, begin bij 0)

[2] [:2] [2:] [2:4]

- Negatieve indices (tellen van achter naar voor, begin bij -1)

[-1][:-1] [-1:] [-4:-1]

Index	0	1	2	3	4	5	6	7	8	9	10	11
Karakter	H	e	l	l	o		w	o	r	l	d	!

Index	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
Karakter	H	e	l	l	o		w	o	r	l	d	!

Samenvatting

- Indices beginnen met tellen vanaf het begin bij 0
- Indices beginnen met tellen vanaf het eind bij -1
- Dit is een zeer handig middel om stukken te filteren

Inhoud

- Indices en slicing
- **Lists**
- Meer over strings
- Tuples

Lists

- Lijsten zijn reeksen met waardes
- Met lijsten is het mogelijk om een aantal bij elkaar horende items op te slaan
- Voorbeelden:
 - `lijst = [1,2,4,7]`
 - `namen = ["Piet", "Jan", "Kees"]`
 - `dieren = ["aap", "konijn", "vos"]`
 - `dingen = [87, "iPhone 5", "konijn", 99, 2]`
- Deze items noemen we ook wel elementen

Elementen in een list

- Lijsten hebben indices

```
lijst = ["aap", "konijn", "vos"]
```

Index	0	1	2
Karakter	aap	konijn	vos

```
print(dieren[1])  
konijn  
print(dieren[3])
```

```
Traceback (most recent call last):  
  File "<pyshell#4>", line 1, in <module>  
    print(dieren[4])  
IndexError: list index out of range
```

Updaten van een list

- Een lijst mag ik ook updaten

```
lijst = ["aap", "konijn", "vos"]
```

Index	0	1	2	3
Karakter	aap	konijn	hond	mus

```
dieren[2] = "hond"
dieren += ["mus"]
print(dieren[3])
mus
```

Met rechte haken geef je aan dat het om een element gaat.
Feitelijk voeg je een lijst toe aan de lijst.

Slicen van een lijst

- Omdat een lijst indices heeft kan ik ook een slice opvragen

```
lijst = ["aap", "konijn", "hond", "mus"]
```

Index	0	1	2	3
Karakter	aap	konijn	hond	mus

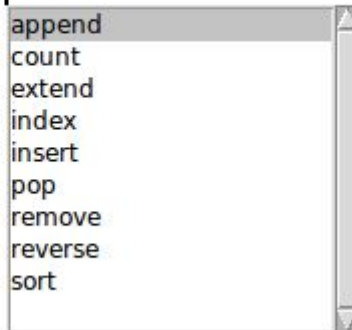
```
print(dieren[2:4])  
['hond', 'mus']
```

Het gekke is hier, dat ook al bestaat index 4 niet, ik toch mag vragen om te printen tot index 4

Methodes van lists

- Ik mag vanalles aan een lijst vragen

```
>>> lijst = ["apen", "noten", "miezen"]  
>>> lijst.|
```



Appenden

```
lijst = ["aap", "konijn", "hond"]
```

Index	0	1	2	3
Karakter	aap	konijn	hond	mus

```
lijst.append("mus")
```


Komt element voor in lijst

- Ik kan controleren of iets voorkomt in een lijst, of de elementen van een lijst bekijken. Dit kan ik ook voor strings doen.

```
for element in lijst:  
    print(element)
```

```
dier = "konijn"  
if dier in lijst:  
    print("Komt voor")
```

Twée dimensionale lists

- Net als dat we for loops kunnen nesten, kunnen we lijsten ook nesten
- Dit noemen we een k dimensionale lijst (waarbij k dus het aantal nests is)
- Voorbeeld van een 2 dimensionale lijst:

```
l = [[1, 2], [3, 4]]
```

```
print (l[0])
```

```
[1,2]
```

```
print (l[1][0])
```

```
3
```

Loopen over 2D list

```
l = [["1", "2"], ["3", "4"], ["5", "6"]]  
for element in l:  
    print(element[0])
```

1

3

5

Samenvatting

- Lists zijn grote verzamelingen aan bij elkaar horende elementen
- Lists maken gebruik van indices
- Lists kan je nesten om een meer dimensionale list te maken
- Zo'n meer dimensionale structuur kan je inzetten als een soort grote tabel of excell sheet

Inhoud

- Indices en slicing
- Lists
- Meer over strings
- Tuples

Wat is een string?

- Een string is een variabele met tekst
- Maar een string is stiekem een lijst met karakters (letters)

Index	0	1	2	3	4	5	6	7	8	9	10	11
Karakter	H	e	l	l	o		w	o	r	l	d	!

Slicing

- Een string kan je ook vragen stellen, zoals welk karakter op een bepaalde positie staat (index)

Index	0	1	2	3	4	5	6	7	8	9	10	11
Karakter	H	e	l	l	o		w	o	r	l	d	!

```
s = "Hello world!"
print ("Resultaat: ",s[1])
```

```
>>> Resultaat: e
```

Slicing

- Met behulp van slicing kan ik dus ook een stuk van een string opvragen

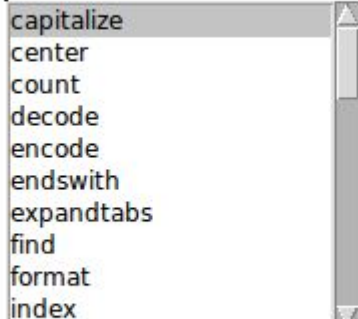
Index	0	1	2	3	4	5	6	7	8	9	10	11
Karakter	H	e	l	l	o		w	o	r	l	d	!

```
s = "Hello world!"
print (s[1:4])
>>> ell
print (s[1:])
>>> ello world!
```


Bewerkingen met Strings

- Iedere string heeft een aantal standaard methodes

```
>>> s = "Hello World!"  
>>> s.
```



- capitalize
- center
- count
- decode
- encode
- endswith
- expandtabs
- find
- format
- index

```
>>> s = "Hello World!"
>>> dir(s)
['__add__', '__class__', '__contains__', '__delattr__', '__doc__'
, '__eq__', '__format__', '__ge__', '__getattribute__', '__ge
titem__', '__getnewargs__', '__getslice__', '__gt__', '__hash__
', '__init__', '__le__', '__len__', '__lt__', '__mod__', '__mul
__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__rep
r__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__s
tr__', '__subclasshook__', '_formatter_field_name_split', '_for
matter_parser', 'capitalize', 'center', 'count', 'decode', 'enc
ode', 'endswith', 'expandtabs', 'find', 'format', 'index', 'isa
lnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', '
isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'rep
lace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rst
rip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase',
'title', 'translate', 'upper', 'zfill']
>>> type(s)
<type 'str'>
>>> |
```

Samenvatting

- Strings zijn een soort lijsten met karakters en gedragen zich daardoor deels als een list

Inhoud

- Indices en slicing
- Lists
- Meer over strings
- **Tuples**

Tuples

- Een tuple is een soort koppel
- Het verschil met een lijst is dat een tuple niet meer te veranderen is nadat deze aangemaakt is (immutable)
- Tuples kennen ook indices
- Tuples zijn heel handig om te gebruiken voor de uitkomst van functies

Lijst

- Mutable
- Creatie:
 - `pietje = [1,4,7]`
- Methodes

```
lijst = ["apen", "noten", "miezen"]  
lijst.
```



append
count
extend
index
insert
pop
remove
reverse
sort

Tuple

- Immutable
- Creatie
 - `pietje = (1,4,7)`
- Methodes

```
tup = (1,2)  
tup.
```



count
index

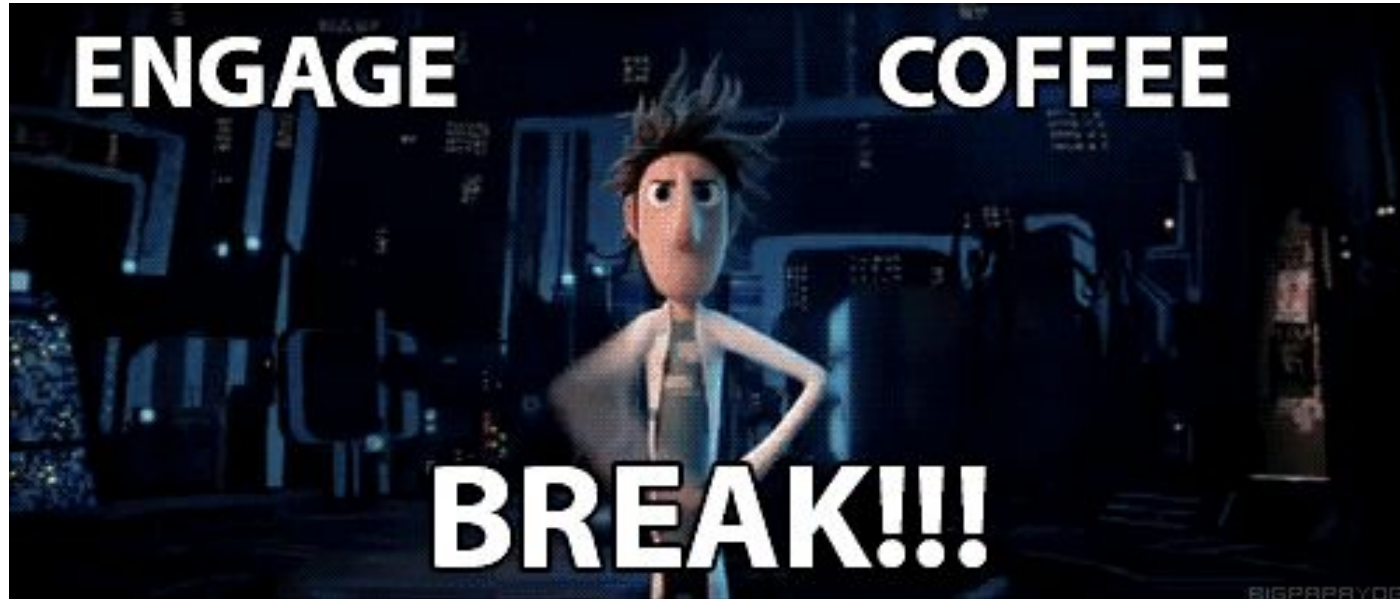
Samenvatting

- Lists, tuples en strings kennen allemaal indices
- Met behulp van indices kan ik slices teruggeven
- Lists en tuples verschillen van elkaar doordat de een veranderbaar is en de ander niet
- Beiden zijn hebben ze hun eigen nut

Opdracht

- Ga naar OnderwijsOnline
- Maak Afvinkopdracht 5
- Bij problemen:
 - Google
 - Klasgenoten
 - Docent
- Lever deze in bij de praktijkdocent





Verantwoording

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/Esther Kok/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites en/of fair use
- Eventuele images zijn opgenomen met vermelding van bron