

Uitwerkingen

Python Assessment



Nijmegen - Utrecht
www.ATComputing.nl

Uitwerking 1: Elementaire datatypes

1.a

Een string bevat tekst in de vorm van Unicode characters, terwijl een bytes object willekeurige bytes bevat.

1.b

- Python 2 heeft een *print-statement*, terwijl dat in Python 3 een *functie* is.
- In Python 3 bevatten strings Unicode characters; in Python 2 zijn dat bytes.
- Python 3 worden vaker *iterators* gebruikt.
- Python 3 kent *views*, Python 2 niet.
- In Python 3 is er een *bytes* datatype bijgekomen, om reeksen ruwe bytes te kunnen representeren.
- In Python 3 kun je text bytes uit een text file naar string decoderen met de `decode()`-method van bytes, wanneer je de text encoding kent. Omgekeerd kun je uit een Python 3 string een reeks text bytes volgens een bepaalde encoding krijgen door de `string-method encode()` te gebruiken.

1.c

In Python 3: geef `print()` een extra *named argument* mee, als volgt:
`print('...', end='')`

In Python 2: voeg een extra `,` toe.

1.d

Ongelimiteerd; slechts beperkt door beschikbaar geheugen.

1.e

Geen verschil.

1.f

```
lijst = []
```

1.g

**

1.h

Ja. `and` wordt gebruikt binnen booleaanse expressies, om een EN-relatie aan te geven; `&` is een operator voor de bit-wise AND-operatie.

1.i

Bijvoorbeeld: `if 'el' in 'hello': ...`

1.j

Mutable basistypen zijn bijvoorbeeld: *list*, *dictionary* en *set*.

Uitwerking 2: Basis statements

2.a

Geen limiet.

2.b

Geen limiet. Er is wel een guideline in PEP8 die zegt dat regels beter niet langer dan 79 tekens zouden moeten zijn.

2.c

Er is geen case-statement. Je gebruikt `if-elif-else` hiervoor.

2.d

Loopconstructies zijn `for-in-else` en `while-else`.

2.e

Het `else`-deel wordt alleen uitgevoerd wanneer de loop *niet* met `break` wordt afgebroken.

2.f

Ja, dat kan met het `raise`-statement.

Uitwerking 3: Strings en lists

3.a

Ja. Dat levert een nieuwe string die bestaat uit de string links van de plus, direct gevolgd door de string rechts van de plus (concatenatie).

3.b

Nee. Dat levert een exception.

3.c

Met een slice-notatie. Bijvoorbeeld: na `sub = "Hello"[1:3]` geldt `sub == 'el'`.

3.d

Strings hebben geen `sort()`-method.

3.e

Gebruik de `split()`-method.

3.f

Die levert een nieuwe string waarbij aan beide uiteinden alle whitespace is weggehaald. Je kunt ook een string-argument mee geven, waardoor `strip()` alle tekens uit dat string-argument aan beide zijden verwijdert.

3.g

Daarvoor wordt een `%` gebruikt.

3.h

Strings hebben een `format()`-method daarvoor.

3.i

Met het `del`-statement. Bijvoorbeeld: `del lijst[4]`
Ook mogelijk met de `pop()`-method.

3.j

Dat is een *iterator*, die telkens de index en waarde van een element uit een reeks teruggeeft.

3.k

```
lijst = [i for i in range(8)]
```

Uitwerking 4: Werken met bestanden

4.a

Nee.

4.b

```
b = f.read(4)
```

4.c

Dat kan niet.

4.d

Dat ligt er aan hoe het tekstbestand geopend is. Als het in text mode is geopend krijg je altijd een string terug; als het in binary mode is geopend krijg je altijd een bytes object terug.

4.e

```
import sys
print('...', file=sys.stderr)
```

4.f

```
f.seek(offset [, how])
```


Uitwerking 5: Unicode, ASCII, UTF-8 en andere encodings

5.a

In Python 2 bestaan strings uit een reeks bytes, in Python 3 zijn dat Unicode characters.

5.b

Bijvoorbeeld: `f = open('filename', 'r', encoding='Latin9')`

5.c

Dan krijg je een `UnicodeEncodeError`-exception, tenzij je als named argument `errors='ignore'` of `errors='replace'` meegeeft.

5.d

Nee, maar een `bytearray` object is wel mutable.

Uitwerking 6: Dictionaries

6.a

```
d = {}
```

6.b

Bijvoorbeeld: `value = d["sleutel"]`

6.c

Dan krijg je een `KeyError`-exception.

6.d

Alleen immutable objects kunnen een sleutel zijn.

6.e

Ja, met de `get()`-method.

Uitwerking 7: Tuples

7.a

Ja, tuples zijn *immutable*, lists zijn *mutable*.

7.b

```
t = ()
```

7.c

Plaats een komma achter het element! Bijvoorbeeld: `t = (42,)`

7.d

Dat kan niet, want een tuple is *immutable*.

7.e

Met de `list()` conversie-functie: `lijst = list(t)`

Uitwerking 8: Sets

8.a

```
s = set()
```

8.b

Dat kan niet, want elementen in een set hebben geen volgorde.

8.c

Een set is *mutable*; een frozenset is *immutable*.

8.d

Met de &-operator: doorsnede = set1 & set2

Uitwerking 9: Functies

9.a

Nee.

9.b

Dat kun je niet aangeven. Wel dat een functie twee argumenten verwacht, maar niet het type ervan.

9.c

Een docstring is documentatie, die met `help()` in de Python shell getoond kan worden.

9.d

Zet in de functiedefinitie achter een argument-naam een `=`, gevolgd door de gewenste default waarde. Bijvoorbeeld: `def myfun(arg=0):`

9.e

Met *starred* en *double starred* argumenten. Zie ook het antwoord op de volgende vraag.

9.f

Een enkele `*` voor een argument betekent dat dat argument een tuple is waarin een *teveel* aan doorgegeven argumenten bij de aanroep van de functie wordt verzameld. Een dubbele `**` voor een argument betekent dat overige argumenten die als *named argument* worden doorgegeven als key-value-pairs in een dictionary worden verzameld.

9.g

Als je een functie aanroept en daarbij een *iterable* doorgeeft, voorafgegaan door een enkele `*`, dan worden de elementen van die iterable als individuele argumenten doorgegeven. De iterable wordt als het ware 'uitgepakt'.

Als je een functie aanroept met een dictionary als argument, voorafgegaan door een dubbele ster (`**`), zullen de elementen van de dictionary als individuele named arguments worden doorgegeven.

9.h

Gebruik de *naam* van de functie als argument.

9.i

De `map()`-functie verwacht een functie en een iterable als argumenten, en zal een nieuwe reeks waarden teruggeven. Deze reeks bestaat uit de returnwaarden van de functie, verkregen door die functie los te laten op elk element van de iterable.

9.j

Een generator-functie gedraagt zich als een iterator: hij zal telkens een volgende waarde genereren. Binnen de generator-functie pas je daarvoor het `yield`-statement toe. De huidige toestand van de generator wordt onthouden, zodat de juiste volgende waarde kan worden gegenereerd bij een volgende iteratieslag.

Uitwerking 10: Modules

10.a

Ja.

10.b

Eerst de directory van waaruit het main script is gestart, daarna de directories uit de environment variabele PYTHONPATH (indien gespecificeerd), en dan de default directories.

Uitwerking 11: Object oriëntatie

11.a

De constructor is de `__init__()`-method.

11.b

Een class variabele ligt binnen de class: er is één exemplaar, die benaderbaar is via de naam van de class.

Een instantievariabele is een variabele die in elke instantie (object) voorkomt.

11.c

Dat is niet mogelijk.

11.d

Gebruik daarvoor `super()`.

11.e

Ja, voor `+` kunnen `__add__()`, `__radd__()`, `__iadd__()` geïmplementeerd worden. Voor `-` zijn dat `__sub__()`, `__rsub__()` en `__isub__()`. En voor `*` zijn dat `__mul__()`, `__rmul__()` en `__imul__()`.

Uitwerking 12: Excepties

12.a

Ja, met

```
except (type1, type2, ...):
```

waarbij *type1* en *type2* Exception classes zijn.

12.b

Ja, wanneer de gebruikte exception class constructor andere soorten argumenten accepteert.

12.c

Door as toe te voegen, bijvoorbeeld:

```
except Exception as ex:
```

Daarna zou je die variabele zichtbaar kunnen maken als volgt:

```
print(ex.args, file=sys.stderr)
```

Of, nog iets eleganter, door de args-tuple uit te laten pakken:

```
print(*ex.args, file=sys.stderr)
```

12.d

Ja. Leid hem af van `Exception` of een geschikte subclass daarvan.

Uitwerking 13: Standard library

13.a

Gebruik `sys.argv`. Dat is een list van strings.

13.b

Gebruik de `sys.exit()`-functie.

- `sys.exit()` zonder argumenten (of de integer waarde 0) geeft succes aan.
- `sys.exit(n)` geeft de integer n terug als exit code.
- Tenslotte, wanneer je `sys.exit('message')` gebruikt, waarbij je dus een foutmelding als string argument gebruikt, zal de foutmelding naar `stderr` worden geschreven, en zal de exit-waarde 1 worden teruggegeven.

13.c

Dat kan met `glob`, of met `fnmatch`.

Uitwerking 14: Reguliere expressies

14.a

Dat is de module `re`.

14.b

Dit is een patroon dat één teken matcht dat alles mag zijn, behalve a, b of c.

14.c

De functie `compile()` zal een reguliere expressie *compileren*, en het resultaat teruggeven als een regex-object. Bij herhaald matchen met deze regex zal dat snellere executie opleveren.

14.d

`[1-9]\d{3} ?[A-Z]{2}`, wat staat voor: een cijfer van 1 tot en met 9, gevolgd door 3 cijfers, dan eventueel een spatie, gevolgd door 2 hoofdletters.

