

Opgaven

Python Assessment



Nijmegen - Utrecht
www.ATComputing.nl

Inleiding

Met dit assessment kun je controleren of je voldoende parate kennis over Python hebt om te beginnen aan de Advanced Python cursus van AT Computing.

De hoofdstukken *Object Orientatie* en *Reguliere Expressies* zijn geen vereiste voorkennis voor de Advanced Python cursus. Ze zijn hier opgenomen zodat dit assessment ook kan dienen als test voor de beheersing van de stof van onze “gewone” Python cursus.

De antwoorden op de vragen zijn beschikbaar als een apart document.

Tenzij anders aangegeven geldt een vraag binnen Python 3 context.

Opgave 1: Elementaire datatypes

1.a

Wat is het verschil tussen een string en een bytes object?

1.b

Noem 3 verschillen tussen Python 2 en Python 3.

1.c

Hoe onderdruk je een regelovergang bij het afdrukken van een string?

1.d

Hoe groot kan een integer maximaal worden?

1.e

Wat is het verschil tussen een string die wordt gemaakt met dubbele quotes (") en met enkele quotes (')?

1.f

Hoe maak je een lege list aan?

1.g

Met welke operator kun je machtsverheffen?

1.h

Is er verschil tussen and en &?

1.i

Hoe bepaal je op eenvoudige wijze of een string voorkomt in een grotere string?

1.j

Noem twee mutable basistypen.

Opgave 2: Basis statements

2.a

Hoe lang mag een Python statement maximaal zijn?

2.b

Hoe lang mag een regel Python code maximaal zijn?

2.c

Hoe ziet het `case`-statement in Python eruit?

2.d

Welke loop-constructies kent Python allemaal?

2.e

Wat betekent het `else`-deel in een Python loop?

2.f

Is het mogelijk om in je eigen code een exception te genereren? Met welk statement?

Opgave 3: Strings en lists

3.a

Kun je strings bij elkaar optellen?

3.b

Kun je een string bij een getal optellen? En andersom?

3.c

Hoe geef je een sub-string aan?

3.d

Wat geeft de string-method `sort()`-method terug? En als de string al gesorteerd is?

3.e

Waarmee krijg je een list van alle woorden in een string?

3.f

Wat doet de `strip()`-method?

3.g

Waarmee wordt de positie gemarkeerd voor substitutie in klassieke formattering?

3.h

Welke method gebruikt Python's moderne formattering?

3.i

Hoe verwijder je een element uit een list?

3.j

Wat doet de `enumerate()`-functie?

3.k

Geef een voorbeeld van list-comprehension. Hoe maak je gebruikmakend van list-comprehension een list met elementwaarden [0, 1, 2, 3, 4, 5, 6, 7]?

Opgave 4: Werken met bestanden

4.a

Is er verschil als je een binair bestand opent in Linux en Windows?

4.b

Hoe lees je 4 bytes uit een binair bestand?

4.c

Hoe lees je 4 bytes uit een tekstbestand?

4.d

Geeft in Python 3 een lees-opdracht op een tekst-bestand een string of een bytes object terug?

4.e

Hoe kun je in Python 3 schrijven naar stderr?

4.f

Met welke method kun je de file offset veranderen?

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

5.a

Wat zijn de elementen van een string in Python 2 en in Python 3?

5.b

Hoe open je een bestand met een encoding die afwijkt van de standaard?

5.c

Wat gebeurt er als je een string met een Euro-teken schrijft naar een bestand dat geopend is met een ASCII-encoding?

5.d

Is een bytes object mutable?

Opgave 6: Dictionaries

6.a

Hoe maak je een lege dictionary?

6.b

Hoe krijg je de waarde die hoort bij een bepaalde sleutel?

6.c

Wat gebeurt er als je de waarde vraagt van een niet-bestaande sleutel?

6.d

Kan elk object een sleutel zijn in een dictionary?

6.e

Kan een dictionary een default waarde teruggeven voor niet-bestaande sleutels?

Opgave 7: Tuples

7.a

Is er een verschil tussen tuples en lists?

7.b

Hoe maak je een lege tuple?

7.c

Hoe maak je een tuple met één element?

7.d

Hoe voeg je een element toe aan een tuple?

7.e

Hoe maak je van list van een tuple?

Opgave 8: Sets

8.a

Hoe maak je een lege set aan?

8.b

Hoe kun je het laatste element van een set verwijderen?

8.c

Wat is het verschil tussen een set en een frozenset?

8.d

Hoe verkrijg je de doorsnede van twee sets?

Opgave 9: Functies

9.a

Moet een functie altijd een `return` statement bevatten?

9.b

Hoe geef je in een functie-definitie aan dat de functie twee integer argumenten verwacht?

9.c

Waarvoor is een docstring?

9.d

Hoe geef je default waarden aan voor argumenten?

9.e

Waarmee kun je een variabel aantal argumenten mogelijk maken?

9.f

Wat betekenen enkele `*` en dubbele `**` asterisks in een functie definitie?

9.g

En wat betekenen ze in een functie-aanroep?

9.h

Hoe geef je een functie door als argument van een andere functie?

9.i

Wat doet de `map()`-functie?

9.j

Wat doet een generator-functie?

Opgave 10: Modules

10.a

Wordt immediate code in een module ook uitgevoerd bij het `import` statement?

10.b

Welke directories worden doorzocht bij het importeren?

Opgave 11: Object oriëntatie

11.a

Welke method wordt aangeroepen bij het maken van een object?

11.b

Wat is het verschil tussen een class variabele en een instantievariabele?

11.c

Hoe geef je aan dat een method of variabele *private* is, met andere woorden of die method of variabele alleen zichtbaar is binnen de class?

11.d

Met welke functie kun je opvragen wat de super-class van een object is?

11.e

Kun je operatoren zoals +, - en * definiëren voor je eigen classes?

Opgave 12: Excepties

12.a

Kun je met één `except`-clause meerdere excepties opvangen?

12.b

Is het mogelijk om bij het `raise` statement de exceptie een argument anders dan een string mee te geven?

12.c

Hoe kun je bij een `try`-statement het exception-object opvangen in een variabele?

12.d

Kun je een eigen exceptie class maken?

Opgave 13: Standard library

13.a

Hoe vraag je op welke argumenten op de commandoregel aan het huidige script zijn meegegeven?

13.b

Hoe kun je voortijdig een Python script (netjes) beëindigen?

13.c

Met welke module kun je voor bestandsnamen wildcard uitdrukkingen gebruiken (bijv: *.png)?

Opgave 14: Reguliere expressies

14.a

Wat is het module om te werken met reguliere expressies?

14.b

Wat betekent `[^abc]`?

14.c

Wat doet de functie `compile()`?

14.d

Kun je een reguliere expressie opstellen die matcht op Nederlandse postcodes?