2023

An introduction to Python programming Language for beginners

BABAK ZOLGHADR-ASLI

SESSION THREE | EXERCISES & CHEATSHEETS

UNIVERSITY OF QUEENSLAND & UNIVERSITY OF EXETER

Contents

I. Examples

```
>>> a = 10
>>> b = 10
>>> a == b
True
>>> id(a), id(b)
(94820449549088, 94820449549088)
>>> a != b
False
>>> a = 2
>>> b = 3
>>> c = 5
>>> a > b
False
>>> a>b
False
>>> a >= a
True
>>> c < b
False
>>> c <= c
True
```

```
>>> (a > b) and (a < 10)
False
>>> (True) and (True)
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
>>> (a > b) or (a == 1)
False
>>> (a<=3) or (b>=100)
True
>>> (True) or (True)
True
>>> (False) or (True)
True
>>> (True) or (False)
True
>>> False or False
False
>>> not(True)
False
>>> not False
True
>>> a = 5
>>> b = 6
>>> if b > a:
      print('b>a')

b>a

>>> a, b = 5, 6

>>> if b > a:

  print(1)

1

>>> a, b = 5, 6

>>> if b < a:

  print(1)

>>> if True:

  print(1)

1

>>> if False:

  print(1)

>>> a = 5

>>> b = 6

>>> if (b > a) and (b > 3):

  print('b>a and b>3')

b>a and b>3

>>> a = 5

>>> b = 4

>>> if a > b:

  print('a>b')

else:

  print('a<=b')

a>b

>>> a, b = 3, 4

>>> if a > b:

  print('a>b')
```

```
else:
  print('a<=b')
>>> a = 5
>>> b = 0
>>> if (a > b) and (b > 1):
  print('a>b')
else:
  print('a<=b or b<=1')
a<=b or b<=1
>>> a = 5
>>> if a == 3:
  print('a is 3')
elif a == 2:
  print('a is 2')
elif a == 1:
  print('a is 1')
else:
  print('a is not 3, 2, nor 1')
a is not 3, 2, nor 1
>>> b = 9
>>> if b<8:
  print('b<8')
elif b<6:
  print('b<6')
else:
  print('b>=8')
b>=8
>>> b = 5
>>> if b<8:
  print('b<8')
elif b<6:
  print('b<6')
else:
  print('b>=8')
b<8
>>> b = 5
>>> if b<6:
  print('b<6')
elif b<8:
  print('b<8')
else:
  print('b>=8')
b<6
>>> b = 9
>>> if b<6:
  print('b<6')
elif b<8:
  print('b<8')
else:
  print('b>=8')
b>=8
>>> 5 in [1, 3, 2]
False
>>> 1 in [1, 2, 3]
True
>>> 5 in (5, 3, 4)
True
>>> # What do you think would be printed here (2 min)
>>> (1, 2) in {(1, 2), (2, 3), 'f'}
True
>>> 'apple' in {'apple':1, 'banana':2, 'orange':3}
True
>>> dict_obj = {'apple':1, 'banana':2, 'orange':3}
>>> values = dict_obj.values()
>>> items = dict_obj.items()
>>> keys = dict_obj.keys()
>>> 1 in values
True
>>> 'apple' in keys
True
>>> ('apple', 1) in items
True
>>> for item in [1, 2, 3]:
  print(item)
1
2
3
>>> for i in (1, 2, 3):
  print(i)
1
2
3
>>> for character in 'Hi':
  print(character)
H
i
>>> for element in {'apple':1, 'orange':2}:
  print(element)

apple

orange

>>> x = [1, 2, 3, 4]

>>> list_new = list()

>>> for i in x:

  y = 2*i**2

  list_new.append(y)

>>> list_new

[2, 8, 18, 32]

>>> count = [5, 5, 2, 4]

>>> sum(count)

16

>>> result = 0

>>> for j in count:

  result += j # result = result + j

>>> result

16

>>> shopping_list = {'apple':2, 'orange':3, 'banana':4, 'kiwi':3 }

>>> 2 + 3 + 4 + 3

12

>>> sum_val = 0

>>> for i in shopping_list:

  sum_val += shopping_list[i]

>>> sum_val

12

>>> sum(shopping_list.values())

12

>>> sum_val = 0
>>> for i in shopping_list.values():
  sum_val += i
>>> sum_val
12
>>> range(5)
range(0, 5)
>>> type(range(5))
range
>>> range(1,5)
range(1, 5)
>>> range(1,5,2)
range(1, 5, 2)
>>> for i in range(3):
  print(i)
0
1
2
>>> for i in range(1, 4):
  print(i)
1
2
3
>>> for i in range(2, 6, 2):
  print(i)
2
4
>>> for i in range(5, 2):
  print(i)
>>> type(range(5,2))
```

```
range
>>> for i in range(5,2,-1):
  print(i)
5
4
3
>>> range(5)[0]
0
>>> range(5,2,-1)[-1]
3
>>> n = 5
>>> fib_seq = [0, 1, 1, 2, 3]
>>> for i in range(n):
  index = len(fib_seq)-1
  x =  fib_seq[index-1]+fib_seq[index]
  fib_seq.append(x)
>>> fib_seq
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
>>> fib_seq = [0, 1, 1, 2, 3]
>>> fib_seq[4-1]
2
>>> list_a = ['apple', 'orange', 'banana']
>>> list_b = [1, 2, 5]
>>> for i in range(len(list_a)):
  print(list_a[i], ':', list_b[i])
apple : 1
orange : 2
banana : 5
>>> list_a = ['apple', 'orange', 'banana']
>>> list_b = [1, 2, 5]
>>> for i, x in enumerate(list_a):
  print(x, ':', list_b[i])
apple : 1
orange : 2
banana : 5
>>> list_a = ['apple', 'orange', 'banana']
>>> for index, element in enumerate(list_a):
  print(index, ':', element)
0 : apple
1 : orange
2 : banana
>>> list_a = ['apple', 'orange', 'banana']
>>> list_b = [1, 2, 5]
>>> for i in zip(list_a, list_b):
  print(i)
('apple', 1)
('orange', 2)
('banana', 5)
>>> list_a = ['apple', 'orange', 'banana']
>>> list_b = [1, 2, 5]
>>> for i, j in zip(list_a, list_b):
  print(i, j)
apple 1
orange 2
banana 5
>>> i = 3
>>> while i >= 0:
  print(i)
  i = i - 1 # i -= 1
3
2
1
0
n = 5
>>> for i in range(5):
  print(i)
0
1
2
3
4
>>> n = 5
>>> i = 0
>>> while i<5:
  print(i)
  i += 1
0
1
2
3
4
>>> for i in range(5):
  if i%2 == 0:
    print(i)
  else:
    print()
0
```

```
2


4
>>> a = {5, 6, 10}
>>> if type(a) == set:
  for i in a:
    print(i)
else:
  print('Error')
10
5
6
a = 5
>>> if a == 2:
  print('a is 2')
  if a == 3:
    print('a is 3')
  else:
    print('inner statement ')
>>> if a == 2:
  print('a is 2')
  if a == 3:
    print('a is 3')
else:
  print('outer statement ')
outer statement
>>> for i in range(5):
  if i == 3:
    break
  print(i)
0
1
2
>>> for i in range(5):
  if i == 3:
    continue
  print(i)
0
1
2
4
>>> for i in range(5):
  pass
>>> a = [i for i in range(5)]
>>> a
[0, 1, 2, 3, 4]
>>> b = {i**2 for i in a}
>>> b
{0, 1, 4, 9, 16}
>>> c = tuple(x for x in b if x%2 == 0)
>>> c
(0, 4, 16)
>>> d = frozenset('even' if i%2 == 0 else i**2 for i in range(5))
>>> d
frozenset({1, 9, 'even'})
>>> keys = ['a', 'b', 'c']
>>> values = (1, 2, 3)
>>> {i:j for i, j in zip(keys, values)}

{'a': 1, 'b': 2, 'c': 3}

>>> [i + 2*j for (i, j) in zip(range(3), range(1,4))]

[2, 5, 8]
```

II. Exercises

1. Evaluate the flowing comparisons. Guess what would be printed in the terminal. Execute the codes to see how you did.

A.

```
>>> 3.0 == 3
```

B.

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

C.

```
>>> set('BABAK') == set('KABAB')
```

D.

```
>>> a = "let's test the transition of property"

>>> b = a

>>> b += 'OK'

>>> b == a
```

E.

```
>>> a = [1, 2, 3]

>>> b = a

>>> b.append(4)

>>> a == b
```

2. Evaluate the flowing comparisons. Guess what would be printed in the terminal. Execute the codes to see how you did.

A.

```
>>> True and False
```

B.

```
>>> True or False
```

C.

```
>>> (5 < 3) or (3 >= 3) and False
```

D.

```
>>> False or True and True
```

E.

```
>>> False and True or True
```

3. Here we have two `list` objects each containing a set of elements. Write a program to do a pair-wised addition on the elements in these lists.

```
>>> a = [1, 5, 10]

>>> b = [19, 15, 10]
```

4. Students grades are stored in a `list` object as follows. The passing level for this specific course is 60. Do go over the list and print the grade and test whether the student has pass the course or not. Do create a list.

```
>>> grades = [75, 60, 50, 100, 80, 72]
```

5. The grades of students in Hydroinformatics are as follows; in order to take this course you need to have “B-” or higher in “advanced water resources management”. Evaluate their scores to see how many are eligible to take this course.

```
>>> grades = ['B+', 'B', 'D', 'F', 'D-', 'C+', 'C-', 'B+', 'A+', 'F']
```

6. The following program is there to represent the following mathematic function. What seems to be the problem with this code? Test the program with `x = 6`. Try to debug the program and run it again.

$$f(x) = \begin{cases} 6 & 6 \leq x \\ 5 & 5 \leq x < 6 \\ 4 & x < 5 \end{cases}$$

III. Recap.

A *cheatsheet* for comparison operators in Python.

Comparison operator	Description
`==`	Equal
`!=`	Not equal
`>`	Greater than
`>=`	Greater than or equal to
`<`	Less than
`<=`	Less than or equal to

A *cheatsheet* for logical operators in Python.

Logical operator	Description
`and`	Returns `True` if both conditions can be hold, otherwise `False` would be returned
`or`	Returns `True` if at least one conditions holds, otherwise `False` would be returned
`not`	Reverses the result of the comparison

A *cheatsheet* for some of the Python's built-in functions.

Built-in functions	Description
`range()`	Returns an iterable object that produces a sequence of integers.
`enumerate()`	Returns a stream of tuples contain a paired index and items of the original iterable object
`zip()`	Aggregates the iterables passed to the function and returns a series `tuple` objects where the elements are pairwise matched
`isinstance()`	Tests multiple options as type of an instance simultaneously.

CONTACT

@babak_zolghadr

babakzolghadrasli.wordpress.co

@babakzolghadrasli

EMAILS

b.zolghadrasli@uq.net.au
bz267@exeter.ac.uk

BABAK ZOLGHADR-ASLI

QUEX-JOINT PH.D. CANDIDATE

RESEARCH AREA

- Water resources planning and management
- Climate change
- Sustainable development
- Decision-Making paradigms
- Deep Uncertainty
- Optimization
- Machine Learning
- Data Mining

AWARDS & HONORS

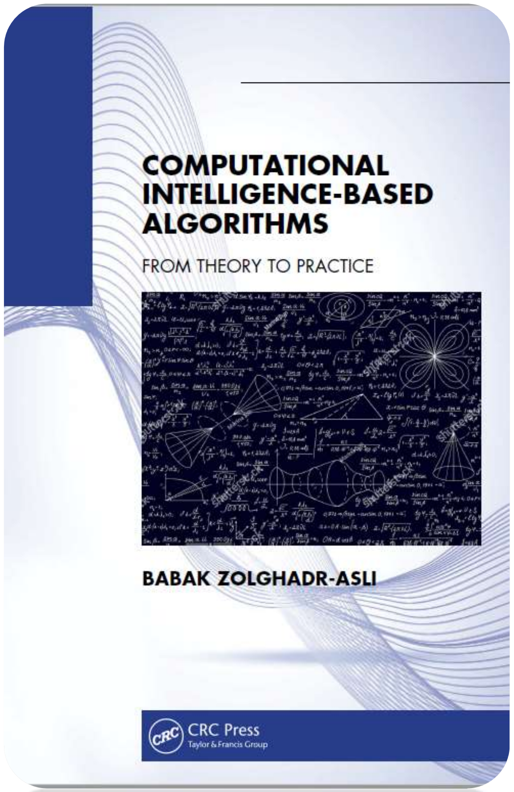
Outstanding researcher award in "the 26th Research Festival", University of Tehran (2017); Outstanding student award in "the 8th International Festival and Exhibition", University of Tehran (2018); Outstanding M.Sc. thesis award in "the 5th National Festival of Environment", Tehran Iran (2018); Winner of the "Prof. Alireaz Sepaskhah" 1st Scientific Award in water engineering [Shiraz University] (2019); Excellent Reviewer, Journal of Hydro Science & Marine Engineering (2020).

SELECTED PUBLICATION

1. Zolghadr-Asli, B., Naghdyzadegan Jahromi, M., Wan, X., Enayati, M., Naghdizadegan Jahromi, M., Tahmasebi Nasab, M., Pourghasemi, H.R., & Tiefenbacher, J.P. (2023). "Uncovering the Depletion Patterns of Inland Water Bodies via Remote Sensing, Data Mining, and Statistical Analysis." Water, 15(8), 1508.
2. Zolghadr-Asli, B. (2023). "No-free-lunch-theorem: A page taken from the computational intelligence for water resources planning and management." Environmental Science and Pollution Research, DOI: 10.1007/s11356-023-26300-1.
3. Zolghadr-Asli, B. (2023). "Computational intelligence-based optimization algorithms: From theory to practice," CRC Press, (Typesetting and finalizing the publisher requirements).

FOR A FULL LIST VISIT: HERE

SCAN ME



COMPUTATIONAL INTELLIGENCE-BASED ALGORITHMS

FROM THEORY TO PRACTICE

BABAK ZOLGHADR-ASLI

CRC Press
Taylor & Francis Group

Coming out soon .... HOPEFULLY!!!!

Chapter 9: Harmony Search Algorithm

SCAN ME

QUEX INSTITUTE

INTERNATIONAL SYMPOSIUM

The University of Queensland
Australia

University of Exeter

Stay in touch

@babak_zolghadr

babakzolghadrasli.wordpress.com

@babakzolghadrasli

b.zolghadrasli@uq.net.au
bz267@exeter.ac.uk

SCAN ME

QUEX INSTITUTE
INTERNATIONAL SYMPOSIUM

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

University of Exeter