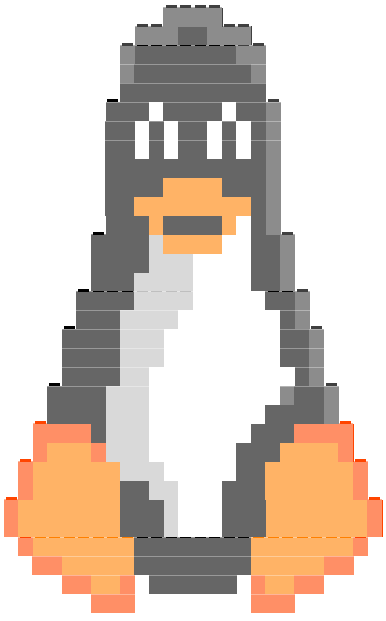
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΦΥΣ 140 Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών (15821)
Χειμερινό Εξάμηνο 2023

Φώτης Πτωχός και Αλέξανδρος Αττίκης

Φροντιστήριο 2

19 Σεπτεμβρίου 2023
15:00 - 17:00



Φροντιστήριο 2

Παράδειγμα 1 Ας ξαναδούμε το δεύτερο πρόγραμμα από τη Διάλεξη 02, για τον υπολογισμό της απόστασης στο χώρο (distance.py) :

tutorial2/ex1.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex1.py
   python3 ex1.py
   script -q ex1.log python3 -i ex1.py

DESCRIPTION:
'''
import numpy as np

# Variable declaration
x1, y1, z1 = 1, 1, 1
x2, y2, z2 = 0, 0, 0

# User input
x2 = float(input("Please provide integer value for x2: "))
if (x2.is_integer() == True):
    print("x2 = ", x2)
else:
    print("x2 is not an integer!")
    quit()

y2 = float(input("Please provide value for y2: "))
if (y2.is_integer() == True):
    print("y2 = ", y2)
else:
    print("y2 is not an integer!")
    quit()

z2 = float(input("Please provide value for z2: "))
if (z2.is_integer() == True):
    print("z2 = ", z2)
else:
    print("z2 is not an integer!")
    quit()


# Calculations
dRsq = (x1 - x2)**2 + (y1 - y2)**2 + (z1 - z2)**2
dR = np.sqrt(dRsq)
print("dR = sqrt(%s) = %s" % (dRsq, dR) )
print("dR = sqrt(%.3f) = %.3f" % (dRsq, dR) )
print("dR = sqrt(%d) = %d" % (dRsq, dR) )

msg = "Thank you for using python3. GOOD BYE!"
print(msg)
```

```
48 print(msg.swapcase())
49
50 quit()
```

Αποτέλεσμα:

```
Please provide integer value for x2: 1
x2 =  1.0
Please provide value for y2: 2
y2 =  2.0
Please provide value for z2: 3
z2 =  3.0
dR = sqrt(5.0) = 2.23606797749979
dR = sqrt(5.000) = 2.236
dR = sqrt(5) = 2
Thank you for using python3. GOOD BYE!
tHANK YOU FOR USING PYTHON3. good bye!
```

Τι έγινε με την τελευταία εντολή print του προγράμματος μας;

Παράδειγμα 2 Το παρακάτω πρόγραμμα δείχνει τρόπους χρήσης για αντικείμενα τύπου λίστας:

tutorial2/ex2.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex2.py
   python3 ex2.py
   script -q ex2.log python3 -i ex2.py

DESCRIPTION:
'''
import numpy as np

# Declare a list with integers
intList = [9, 2, 3, 4, 0, 4, 3, 1, 7]

print("intList = ", intList)
print("len(intList) = ", len(intList))
print("sorted(intList) = ", sorted(intList)) # does not modify the object
print("intList = ", intList)

# Permanently modify the intList (sort() instead of sorted())
intList.sort()
print("\n1) intList = ", intList)

intList.reverse()
print("\n2) intList = ", intList)

rm = intList.pop(0)
print("\n3a) intList = ", intList)
print("3b) len(intList) = ", len(intList))

intList.insert(0, rm)
print("\n4a) intList = ", intList)
print("4b) len(intList) = ", len(intList))

intList.insert( len(intList), -1)
print("\n5a) intList = ", intList)
print("5b) len(intList) = ", len(intList))

list1 = [1, 2, 3]
list2 = ["a", "b", "c"]
newList = list1 + list2
print("="*70)
print("6a) newList = ", newList)
print("6b) len(newList) = ", len(newList))
print("6c) newList[0] = ", newList[0])
print("6d) newList[-1] = ", newList[-1])
print("6e) newList[:1] = ", newList[:1])
print("6f) newList[:2] = ", newList[:2])
print("6g) newList[1:] = ", newList[1:])
print("6h) newList[2:] = ", newList[2:])
```

Παράδειγμα 2 συνεχίζεται...

```
51 print("6i) newList[-1:] = ", newList[-1:])
52 print("6i) newList[-2:] = ", newList[-2:])
53 print("="*70)
54
55 print("*"*70)
56 newList.append("A")
57 newList.extend(["B", "C"])
58 print("7) newList = ", newList)
59 print("*"*70)
60
61 msg = "quit()"
62 print(msg)
63 quit()
```

Αποτέλεσμα:

```
intList =  [9, 2, 3, 4, 0, 4, 3, 1, 7]
len(intList) =  9
sorted(intList) =  [0, 1, 2, 3, 3, 4, 4, 7, 9]
intList =  [9, 2, 3, 4, 0, 4, 3, 1, 7]

1) intList =  [0, 1, 2, 3, 3, 4, 4, 7, 9]

2) intList =  [9, 7, 4, 4, 3, 3, 2, 1, 0]

3a) intList =  [7, 4, 4, 3, 3, 2, 1, 0]
3b) len(intList) =  8

4a) intList =  [9, 7, 4, 4, 3, 3, 2, 1, 0]
4b) len(intList) =  9

5a) intList =  [9, 7, 4, 4, 3, 3, 2, 1, 0, -1]
5b) len(intList) =  10
======================================================================
6a) newList =  [1, 2, 3, 'a', 'b', 'c']
6b) len(newList) =  6
6c) newList[0] =  1
6d) newList[-1] =  c
6e) newList[:1] =  [1]
6f) newList[:2] =  [1, 2]
6g) newList[1:] =  [2, 3, 'a', 'b', 'c']
6h) newList[2:] =  [3, 'a', 'b', 'c']
6i) newList[-1:] =  ['c']
6i) newList[-2:] =  ['b', 'c']
======================================================================
**********************************************************************
7) newList =  [1, 2, 3, 'a', 'b', 'c', 'A', 'B', 'C']
**********************************************************************
quit()
```

Παράδειγμα 3 Απλό πρόγραμμα για τη χρήση εντολών συνθήκης if / else :

tutorial2/ex3.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex3.py
   python3 ex3.py
   script -q ex3.log python3 -i ex3.py


DESCRIPTION:
Introduction to if/else concept
'''
printIf = True
if (printIf):
    msg = "1) Hello world!"
else:
    msg = "1) Goodbye world!"
print(msg)


printIf = False
if (printIf):
    msg = "2) Hello world!"
else:
    msg = "2) Goodbye world!"
print(msg)


print("Quit!")
quit()
```

Αποτέλεσμα:

```
1) Hello world!
2) Goodbye world!
Quit!
```

Παράδειγμα 4 Απλό παράδειγμα για τη χρήση τελεστών σύγκρισης ==, !=, <>, >, <, <=, >= :

tutorial2/ex4.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex4.py
   python3 ex4.py
   script -q ex4.log python3 -i ex4.py


DESCRIPTION:
Use a  "try/except" block to check if the length of the input, after converting it to a string, is equal to 1.
This ensures that the user entered a single character. If the input is a single character, it's considered valid.
Otherwise, we raise a ValueError and print an error message.  This code will handle cases where the user enters
a valid single character or provide an error message for invalid inputs.
'''
import sys

# Get user input and attempt to convert it to an integer. Use try/except to make sure we gen an integer
try:
    key = input(">>> Please enter a number in the range [0,4]: ")
    num = -1

    # Check if the input is a single character
    if len(key) == 1:
        num = int(key)
    else:
        print("Invalid input", key, "! Please enter a single character.")
        #raise Exception("Invalid input! Please enter a single character")
except:
    print("Something went wrong!")
    quit()


msg = "\tThe input number is"
print("If-block 1:")
if (num == 0):
    print(msg + " zero")
elif (num == 1):
    print(msg + " one")
elif (num == 2):
    print(msg + " two")
elif (num == 3):
    print(msg + " two")
elif (num == 4):
    print(msg + " two")
else:
    print("\tInput number is outside range of [0,4]")
```

```
47      sys.exit()
48
49  print("If-block 2:")
50  if (num !=2):
51      print(msg, "not two")
52  if (num < 3):
53      print(msg, "less than two")
54  if (num > 2):
55      print(msg, "greater than two")
56  if (num >= 2):
57      print(msg, "greater or equal to two")
58  if (num <= 2):
59      print(msg, "less or equal to two")
60
61  quit()
```

Αποτέλεσμα:

```
>>> Please enter a number in the range [0,4]: 2
If-block 1:
  The input number is two
If-block 2:
  The input number is less than two
  The input number is greater or equal to two
  The input number is less or equal to two
```

Παράδειγμα 5 Παράδειγμα χρήσης τελεστών σύγκρισης ==, !=, <>, >, <, <=, >= σε συνδιασμό με εντολές συνθήκης if / else :

tutorial2/ex5.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex5.py
   python3 ex5.py
   script -q ex5.log python3 -i ex5.py


DESCRIPTION:
Introduction to comparison operators
'''
myNumber = int(input("Please select a number in the range [0,9]: "))
msg = ""

# If can be used by itself (without else)
if (myNumber == 0):
    msg = "=== A "
    print(msg)

if (myNumber != 1):
    msg = "=== B "
    print(msg)

if (myNumber < 2):
    msg = "=== C "
    print(msg)

if (myNumber > 4):
    msg = "=== D "
    print(msg)

if (myNumber <= 8):
    msg = "=== E "
    print(msg)


myNumber = int(input("Please select another number in the range [0,9]: "))
if (myNumber == 1):
    msg = "=== A "
    print(msg)
elif (myNumber == 2):
    msg = "=== B "
    print(msg)
elif (myNumber <= 6):
    msg = "=== C "
    print(msg)
elif (myNumber < 10):
    msg = "=== D "
    print(msg)
```

Παράδειγμα 5 συνεχίζεται...

```
50 else:
51     msg = "==== Number '%d' is out of range!" % (myNumber)
52     print(msg)
53
54 quit()
```

Αποτέλεσμα:

```
Please select a number in the range [0,9]: 5
=== B
=== D
=== E
Please select another number in the range [0,9]: 9
=== D
```

Παράδειγμα 6 Απλό παράδειγμα στη χρήση λογικών τελεστών `and`, `or`, `not` :

tutorial2/ex6.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex6.py
   python3 ex6.py
   script -q ex6.log python3 -i ex6.py


DESCRIPTION:
More on logical/comparison operators and their combination.


LINKS:
https://www.pythontutorial.net/python-basics/python-logical-operators/
'''
yes = True
no  = False

# If can be used by itself (without else)
if (yes):
    msg = "=== Hello world!"
    print(msg)

if (no):
    msg = "=== This will never be printed"
    print(msg)

if (yes or no):
    msg = "=== This will always be true"
    print(msg)

if (not (yes or no)):
    msg = "=== This will also never be printed"
    print(msg)


print("\n=== Direct use of logical operators booleans")
value = 9.99
print("\tIs the value less than 10?", (value < 10) )
print("\tIs the value not greater than 10?", (not value > 10) )
print("\tIs the value in the range (5, 10]?", (value > 5 and value < 10) )

print("\n=== Direct Precedence of not/and/or operators (descending order):\n\t1) not\n\t2) and\n\t3) or")


a = True
b = True
c = False
d = True
```

Παράδειγμα 6 συνεχίζεται...

```
50 msg = "\n=== Examples for precedence of Logical Operators"
51 msg += " (a = %s, b = %s, c = %s, d = %s)" % (a, b, c, d)
52 print(msg)
53
54 print("\ta or b and c = ", (a or (b and c) ) )
55 print("\ta and b or c and d = ", ( (a and b) or (c and d) ) )
56 print("\ta and b and c or d = ", ( ((a and b) and c) or d) )
57 print("\tnot a and b or c = ", ( ((not a) and b) or c) )
58
59
60 msg = "\n=== Take-away message: Use brackets to make sure!"
61 print(msg)
62
63 quit()
```

Αποτέλεσμα:

```
=== Hello world!
=== This will always be true

=== Direct use of logical operators booleans
  Is the value less than 10? True
  Is the value not greater than 10? True
  Is the value in the range (5, 10]? True

=== Direct Precedence of not/and/or operators (descending order):
  1) not
  2) and
  3) or

=== Examples for precedence of Logical Operators (a = True, b = True, c =
    False, d = True)
  a or b and c =  True
  a and b or c and d =  True
  a and b and c or d =  True
  not a and b or c =  False

=== Take-away message: Use brackets to make sure!
```

Παράδειγμα 7 Απλό πρόγραμμα για τη χρήση της δομής try / except / finally:

tutorial2/ex7.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x e7.py
   python3 ex7.py
   script -q ex7.log python3 -i ex7.py


DESCRIPTION:
Introduction to try/except


LINKS:
https://data-flair.training/blogs/python-exception-handling/
'''
x=1
try:
  print("x = ", x)
except: # NameError:
  print("=== Exception: The variable x is not defined")
finally:
    print("=== This will print no matter what (exception or no exception)")


try:
    newPrice   = float(input("\n=== Type value for new price: "))
    oldPrice   = float(input("=== Type value for old price: "))
    percChange = ((newPrice - oldPrice)*100)/oldPrice

    if percChange < 0:
        result = "\tPrice drop of %s %%" % (abs(percChange))
    else:
        result = "\tPrice increase of %s %%" % (abs(percChange))

    print(result)

except:
    msg = "Error! Please enter a number (not character) for the prices."
    print(msg)


# You can define as many exception blocks as you want, e.g. if you want to
   execute a special block of code for a special kind of error:
# a, b = 1, 0
# try:
#     print("a/b = ", (a/b) )
#     print("This will never be printed. Any statement after an exception in
   the 'try' block are skipped.")
# except TypeError:
#     print("You added values of incompatible types")
```

Παράδειγμα 7 συνεχίζεται...

```
49 # except ZeroDivisionError:
50 #     print("You divided by 0")
51
52 quit()
```

Αποτέλεσμα:

```
x =  1
=== This will print no matter what (exception or no exception)
=== Type value for new price: 80.00
=== Type value for old price: 100.00
  Price drop of 20.0 %
```

Παράδειγμα 8 Απλό πρόγραμμα για τη χρήση της δομής επαναληπτικής διαδικασίας for/while:

tutorial2/ex8.py

```
#!/usr/bin/python3
'''
USAGE:
   chmod +x ex8.py
   python3 ex8.py
   script -q ex8.log python3 -i ex8.py


DESCRIPTION:
Simple loops with for/while
'''
import sys

print("Using a for loop with step +1:")
for i in range(1, 11, +1):
    print("\ti =", i)

print("Using a for loop with step -1:")
for j in range(11, 1, -1):
    print("\tj =", j)

print("Using a for loop with step 20:")
for k in range(0, 100+20, 20):
    print("\tk =", k)


print("Using a while loop:")
count = 1
while count <= 5:
    print("\t", count)
    count += 1

quit()
```

Αποτέλεσμα:

```
Using a for loop with step +1:
  i = 1
  i = 2
  i = 3
  i = 4
  i = 5
  i = 6
  i = 7
  i = 8
  i = 9
```

```
  i = 10
Using a for loop with step -1:
  j = 11
  j = 10
  j = 9
  j = 8
  j = 7
  j = 6
  j = 5
  j = 4
  j = 3
  j = 2
Using a for loop with step 20:
  k = 0
  k = 20
  k = 40
  k = 60
  k = 80
  k = 100
Using a while loop:
   1
   2
   3
   4
   5
```