ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
**ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ
ΜΗΧΑΝΙΚΩΝ**
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

Ακαδ. Έτος: 2016-2017

Δευτέρα, 3/10/2016

Μάθημα: Εφαρμογές Η/Υ

Διδάσκοντες:
Ν.Δ. Λαγαρός (Επικ. Καθηγητής), Αθ. Στάμος (ΕΔΙΠ), Χ. Φραγκουδάκης (ΕΔΙΠ)

Παραδείγματα για την 1η παράδοση – Εισαγωγικά περί Python

1. Μικρά εισαγωγικα προγράμματα σε Python ελαστικότητας

```
#Add integers
a = 2
b = 3
c = a + b



#Text delimeters
a = 2
b = 3
c = a + b
print(c)
print("a+b=", c)
print('a+b=', c)
print("'a'+'b'=", c)
print("""'a'+"b"=""", c)



#Operations on integers
a = 2
b = 3
print("a+b=", a+b)
print("a-b=", a-b)
print("a*b=", a*b)
print("a/b=", a/b)
print("a//b=", a//b)
print("a**b=", a**b)



#Operations on reals (float)
a = 3.1415
b = 2.718
print("a+b=", a+b)
print("a-b=", a-b)
print("a*b=", a*b)
print("a/b=", a/b)
print("a//b=", a//b)
print("a**b=", a**b)



#Operations on complex
```

```
a = 3.1415+1.1j
b = 2.718-7.7j
print("a+b=", a+b)
print("a-b=", a-b)
print("a*b=", a*b)
print("a/b=", a/b)

c = complex(7.1, 7.2)
print("c=", c)
print("a**c=", a**c)



#Operations on fractions
from fractions import Fraction
a = Fraction(1, 2)
b = Fraction(3, 5)
print("a=", a)
print("b=", b)
print("a+b=", a+b)
print("a-b=", a-b)
print("a*b=", a*b)
print("a/b=", a/b)
print("a//b=", a//b)
print("a**b=", a**b)
print("a**3=", a**3)



#Automatic conversion to long integers
j = 3
k = j**15
print("k=", k)

j = 1234567890
k = j**15
print("k=", k)



#Operations on decimals/arbitrary precision
from decimal import Decimal, getcontext
print("float")
a = 1
b = 7
print("a/b=", a/b)


print("Decimal")
a = Decimal(1)
b = Decimal(7)
print("a=", a)
print("b=", b)
print("a/b=", a/b)

print("Decimal 100")
c = getcontext()      #Get control object
c.prec = 100          #Set precision to 100 digits
print("a=", a)
print("b=", b)
print("a/b=", a/b)
```

```
#Tuples
c = 5, 6, 7      #This is a comment: parentheses are not necessary
d = 11,          #A tuple with one element
e = 100, 200
print("c=", c)
print("d=", d)
print("e=", e)
f = c + d + e
print("combined=", f)
print("sum=", sum(f))



#Tuple indexing
a = 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
b = tuple(range(0, 110, 10))
print("a=", a)
print("b=", b)
print("a[3]=", a[3])
print("a[-1]=", a[-1])
print("a[-3]=", a[-3])
c = a[2:7]
print("a[2:5]=", c)
c = a[2:9:3]
print("a[2:9:3]=", c)
print("a[2:]=", a[2:])
print("a[2:-1]=", a[2:-1])
print("a[:4]=", a[:4])



#List indexing
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
b = list(range(0, 110, 10))
print("a=", a)
print("b=", b)
print("a[3]=", a[3])
print("a[-1]=", a[-1])
print("a[-3]=", a[-3])
c = a[2:7]
print("a[2:5]=", c)
c = a[2:9:3]
print("a[2:9:3]=", c)
print("a[2:]=", a[2:])
print("a[2:-1]=", a[2:-1])
print("a[:4]=", a[:4])



#list mutations
a = [20, 40, 60, 100]
print("a=", a)
a.append(120)
print("a=", a)
a.insert(2, 500)
print("a=", a)
del a[2]
print("a=", a)
a[1:3] = [-1, -2, -3, -4, -5]
print("a=", a)



#Immutable tuples and mutable lists
```

```
a = (1, 5, 10)
b = a
print("**Immutable***")
print("a=", a)
print("b=", b)

a = (2, -2)
print("after")
print("a=", a)
print("b=", b)


a = [1, 5, 10]
b = a
print("**Mutable***")
print("a=", a)
print("b=", b)

a.append(3145)
print("after")
print("a=", a)
print("b=", b)



#Copy list explicitly
a = [1, 5, 10]
b = a.copy()
print("**Mutable copy***")
print("a=", a)
print("b=", b)

a.append(3145)
print("after")
print("a=", a)
print("b=", b)



#Nested tuples
c = 5, 6, (100, 200), 7      #Nested
print("c=", c)
print("c[1]=", c[1])
print("c[2]=", c[2])
print(sum(c))      #Does not work



#Nested lists
c = [5, 6, (100, 200), 7]      #Nested
print("c=", c)
print("c[1]=", c[1])
print("c[2]=", c[2])

c[2] = list(c[2])
print("c=", c)
print("c[2]=", c[2])

x = c[2]
x.insert(1, [300,400,500])
print("c=", c)
print("c[2]=", c[2])
```

```
# -*- coding: iso-8859-7 -*-
#Text indexing
t = "Ntua-Civil"
print("t=", t)
print("t[3:6]=", t[3:6])
u = t[3:6] + t[7:-1]
print("u=", u)

t = "ΕΜΠ-Πολιτικοί"
print("t=", t)
print("t[::-1]=", t[::-1])
```

```
# -*- coding: iso-8859-7 -*-
#Text concatenation
a = "Ntua"
b = "Civil"
t = a + "-" + b

n = 231
u = "students " + str(231)
print("u=", u)
```

```
# -*- coding: iso-8859-7 -*-
#Text is immutable
t = "ΕΜΠ-Πολιτικοί"
print("t=", t)
t[2]="XX"    #Error: strings are immutable
```

```
# -*- coding: iso-8859-7 -*-
#list with numbers and text
t = "ΕΜΠ-Πολιτικοί"
a = [1, 2, 3, t, 4, 5]
print("t=", t)
print("a=", a)
print("sum=", sum(a))   #Error
```

```
#Hypotenuse of right triangle
from math import sqrt, hypot
a =3
b = 4
c = sqrt(a**2 + b**2)
d = hypot(a, b)
print("c=", c)
print("d=", d)
```

```
#Hypotenuse of right triangle with matrices
from math import sqrt
from numpy import array
a = array([3, 4])
d = a**2
e = sum(d)
c = sqrt(e)
print("c=", c)
```

```
#Hypotenuse of right triangle with matrices
from math import sqrt
from numpy import array
a = array([3, 4])
c = sqrt(sum(a**2));
print("c=", c)
```

```
#Hypotenuse of right triangle: read sides from keyboard
from math import sqrt, hypot
a = float(input("a: "))
b = float(input("b: "))
c = hypot(a, b)
print("c=", c)
```

```
#Hypotenuse of right triangle: show results to window
from math import sqrt, hypot
from tkinter import Tk, Label
a = 6.0
b = 8.0
c = hypot(a, b)

win = Tk()
q = Label(win, text="a="+str(a), width=20)
q.grid()
q = Label(win, text="b="+str(b))
q.grid()
q = Label(win, text="c="+str(c))
q.grid()
win.mainloop()
```

```
#Hypotenuse of right triangle: GUI
from math import sqrt, hypot
from tkinter import Tk, Label, Entry, Button, END
a = 6.0
b = 8.0
c = hypot(a, b)

win = Tk()
q = Label(win, text="a=")
q.grid(row=0, column=0)
wa = Entry(win, width=20)
wa.grid(row=0, column=1)

q = Label(win, text="b=")
q.grid(row=1, column=0)
wb = Entry(win, width=20)
wb.grid(row=1, column=1)

q = Label(win, text="c=")
q.grid(row=2, column=0)
wc = Entry(win, width=20)
wc.grid(row=2, column=1)

q = Button(win, text="Compute", command=lambda: wc.delete(0, END) or
wc.insert(1, hypot(float(wa.get()), float(wb.get())) ))
```

```
q.grid(row=3, column=1)

win.mainloop()



#Hypotenuse of right triangle: GUI advanced
from math import sqrt, hypot
from tkinter import Tk, Label, Entry, Button, END
a = 6.0
b = 8.0
c = hypot(a, b)

win = Tk()
q = Label(win, text="a=")
q.grid(row=0, column=0)
wa = Entry(win, width=20)
wa.grid(row=0, column=1, sticky="we")

q = Label(win, text="b=")
q.grid(row=1, column=0)
wb = Entry(win, width=20)
wb.grid(row=1, column=1, sticky="we")

q = Label(win, text="c=")
q.grid(row=2, column=0)
wc = Entry(win, width=20)
wc.grid(row=2, column=1, sticky="we")

def compute():
    "Compute the sum."
    wc.delete(0, END)
    a = float(wa.get())
    b = float(wb.get())
    c = hypot(a, b)
    wc.insert(1, c)

q = Button(win, text="Compute", command=compute)
q.grid(row=3, column=1)
win.columnconfigure(1, weight=1)

win.mainloop()



#Hypotenuse of right triangle: GUI advanced with error chechking
from math import sqrt, hypot
from tkinter import Tk, Label, Entry, Button, END
from tkinter.messagebox import showinfo, ERROR

a = 6.0
b = 8.0
c = hypot(a, b)

win = Tk()
q = Label(win, text="a=")
q.grid(row=0, column=0)
wa = Entry(win, width=20)
wa.grid(row=0, column=1, sticky="we")

q = Label(win, text="b=")
q.grid(row=1, column=0)
wb = Entry(win, width=20)
wb.grid(row=1, column=1, sticky="we")
```

```
q = Label(win, text="c=")
q.grid(row=2, column=0)
wc = Entry(win, width=20)
wc.grid(row=2, column=1, sticky="we")

def compute():
    "Compute the sum."
    try:
        a = float(wa.get())
    except ValueError as e:
        showinfo("Error in a", str(e), parent=win, icon=ERROR)
        return
    try:
        b = float(wb.get())
    except ValueError as e:
        showinfo("Error in b", str(e), parent=win, icon=ERROR)
        return
    c = hypot(a, b)
    wc.delete(0, END)
    wc.insert(1, c)

q = Button(win, text="Compute", command=compute)
q.grid(row=3, column=1)
win.columnconfigure(1, weight=1)

win.mainloop()
```