00ex_introduction

March 14, 2020

1. The MickeyMouse problem

a) Write a program that prints the numbers from 1 to 100. But for multiples of three print Mickey instead of the number and for the multiples of five print Mouse. For numbers which are multiples of both three and five print MickeyMouse

b) Put the result in a tuple and substitute Mickey with Donald and Mouse with Duck

In [1]:
```
"""
for i in range (1,100):
    if i%3==0 and i%5!=0:
        print("Myckey")
    if i%5==0 and i%3!=0:
        print("Muose")
    if i%3==0 and i%5==0:
        print("MickeyMuose")
    if i%3!=0 and i%5!=0:
        print(i)
"""

a=[]

for i in range(1,100):
    if i%3==0 or i%5==0:
        if i%5==0:
            if i%3==0:
                a.append("MickeyMouse")
            else:
                a.append("Mouse")
        else:
            a.append("Mickey")
    else:
        a.append(i)

for i in a: print (i)

b=a
```

```
        for i in range (0,99):
            if b[i]=="MickeyMouse":
                b[i]="DonaldDuck"
            if b[i]=="Mouse":
                b[i]="Duck"
            if b[i]=="Mickey":
                b[i]="Donald"
        at=tuple(x for x in b)
        print(at)
```

```
1
2
Mickey
4
Mouse
Mickey
7
8
Mickey
Mouse
11
Mickey
13
14
MickeyMouse
16
17
Mickey
19
Mouse
Mickey
22
23
Mickey
Mouse
26
Mickey
28
29
MickeyMouse
31
32
Mickey
34
Mouse
Mickey
37
```

```
38
Mickey
Mouse
41
Mickey
43
44
MickeyMouse
46
47
Mickey
49
Mouse
Mickey
52
53
Mickey
Mouse
56
Mickey
58
59
MickeyMouse
61
62
Mickey
64
Mouse
Mickey
67
68
Mickey
Mouse
71
Mickey
73
74
MickeyMouse
76
77
Mickey
79
Mouse
Mickey
82
83
Mickey
Mouse
```

```
86
Mickey
88
89
MickeyMouse
91
92
Mickey
94
Mouse
Mickey
97
98
Mickey
(1, 2, 'Donald', 4, 'Duck', 'Donald', 7, 8, 'Donald', 'Duck', 11, 'Donald', 13, 14, 'DonaldDuc
```

2. The swap function
Write a function that swap the values of two input variables x and y (whatever the type). Try to do that also without a temporary variable

```
In [2]: def canonic_swap (x,y):
            t=x
            x=y
            y=t
            return x,y

        def python_swap (x,y):
            return y,x

        x=input("input x")
        y=input("input y")
        print (x,y)
        x, y = canonic_swap(x,y)
        print(x,y)
        x, y = python_swap(x,y)
        print (x,y)
```

```
input x 5
input y 6


5 6
6 5
5 6
```

3. Computing the distance

Write a function that calculates and returns the euclidean distance between two points *u* and *v*, where *u* and *v* are both 2-tuples *(x,y)*. For example, if *u=(3,0)* and *v=(0,4)*, the function should return 5

```
In [3]: import math
        def dist (a, b):
            return math.sqrt(pow(a[0]-b[0],2) + pow(a[1]-b[1],2))
        a= (3,0)
        b=(0,4)
        print (dist(a,b))
```

```
5.0
```

4. Counting letters

Write a program to calculate the number of times each character occurs in a given string *s*. Ignore differneces in capitalization

```
In [4]: s="Write a program that prints the numbers from 1 to 100. \
        But for multiples of three print Mickey instead of the number and for the multiples of
        For numbers which are multiples of both three and five print MickeyMouse"

        a={}
        for letter in s:
            f = False
            for i in a:
                if i==letter:
                    f=True
            if f==False:
                a[letter]=1
            else:
                a[letter]+=1
        print(a)
```

```
{'W': 1, 'r': 17, 'i': 14, 't': 19, 'e': 22, ' ': 41, 'a': 7, 'p': 8, 'o': 13, 'g': 1, 'm': 8,
```

5. Isolating the unique

Write a function that determines and count the unique numbers in the list *l*

```
In [5]: #as unique i intended number that appears only one time

        l = [157, 157, 36, 45, 58, 3, 74, 96, 64, 45, 31, 10, 24, 19, 33, 86, 99, 18, 63, 70,
         85, 63, 47, 56, 42, 70, 84, 88, 55, 20, 54, 8, 56, 51, 79, 81, 57, 37, 91,
         1, 84, 84, 36, 66, 9, 89, 50, 42, 91, 50, 95, 90, 98, 39, 16, 82, 31, 92, 41,
         45, 30, 66, 70, 34, 85, 94, 5, 3, 36, 72, 91, 84, 34, 87, 75, 53, 51, 20, 89, 51, 20]

        u=0
        for x in l:
            f=0
            for y in l:
                if x==y:
                    f+=1
            if f<2:
                #print(x, "is unique")
                u+=1
        print("total uniques: ", u)
```

```
total uniques:  37
```

6. Combination of functions
Write two functions - one that returns the square of a number, and one that returns the cube. Now write a third function that returns the number raised to the 6th power using the two previous functions.

```
In [6]: def square (x):
            return x*x
        def cube   (x):
            return x*x*x
        def sixth  (x):
            return square(x)*cube(x)

        x=5
        print (x)
        x=square(x)
        print (x)

        y=5
        print (y)
        y=cube(y)
        print (y)

        z=5
        print (z)
        z=sixth(z)
        print (z)
```

```
5
25
5
125
5
3125
```

7. Cubes
Create a list of the cubes of x for x in *[0, 10]* using:

a) a for loop

b) a list comprehension

```
In [7]: a=[]
        for i in range (0,10):
            a.append(cube(i))
        print (a)

        b=[cube(x) for x in range (0,10)]
        print (b)
```

```
[0, 1, 8, 27, 64, 125, 216, 343, 512, 729]
[0, 1, 8, 27, 64, 125, 216, 343, 512, 729]
```

8. Nested list comprehension

A Pythagorean triple is an integer solution to the Pythagorean theorem $a^2 + b^2 = c^2$. The first Pythagorean triple is (3,4,5). Find and put in a tuple all unique Pythagorean triples for the positive integers a, b and c less than 100.

```
In [8]: a=tuple([(x,y,z) for x in range (1,100) for y in range (x,100) for z in range (y,100)
        print (a)
```

```
((3, 4, 5), (5, 12, 13), (6, 8, 10), (7, 24, 25), (8, 15, 17), (9, 12, 15), (9, 40, 41), (10,
```

9. Normalization

Write a function that takes a tuple of numbers and returns it with the entries normalized to one

```
In [9]: import math

        #questa funzione è intesa come norma del vettore 1
        def norm(a):
            t=0
            for n in a:
                t+=n*n
            b=[x/math.sqrt(t) for x in a]
            c=tuple(b)
            return c
        #questa è intesa come valore massimo del vettore 1
        def norm2(a):
            t=max(a)
            b=[x/t for x in a]
            c=tuple(b)
            return c
        a=(3,5,8)
        print(a)
        a=norm (a)
        print (a)
        a=norm2 (a)
        print (a)
```

```
(3, 5, 8)
(0.30304576336566325, 0.5050762722761054, 0.8081220356417685)
(0.37500000000000006, 0.6250000000000001, 1.0)
```

```
In [ ]:
```