

# Wnioskowanie statystyczne

28 lutego 2014

# Python

Python 2.7.3 (default, Jan 13 2013, 11:20:46)

[GCC 4.6.3] on linux2

Type "help", "copyright", "credits" or "license" for more information.

```
>>> 2*3
```

```
6
```

```
>>> c=2*3
```

```
>>> c
```

```
6
```

```
>>> a=2
```

```
>>> b=3
```

```
>>> a*b
```

```
6
```

# Python: listy

```
>>> lista = [1, 3, 5, 7, 9]
```

```
>>> lista
```

```
[1, 3, 5, 7, 9]
```

```
>>> lista[0]
```

```
1
```

```
>>> lista[4]
```

```
9
```

```
>>> lista[5]
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
IndexError: list index out of range
```

```
>>> lista[2:]
```

```
[5, 7, 9]
```

```
>>> lista[:2]
```

```
[1, 3]
```

```
>>> lista[1::]
```

```
[3, 5, 7, 9]
```

```
>>> lista[1:]
```

```
[3, 5, 7, 9]
```

```
>>> lista[1::2]
```

```
[3, 7]
```

# Python: konstrukcja if

```
if warunek1 :  
    polecenie1  
elif warunek2 :  
    polecenie2  
else :  
    polecenie3
```

```
a = 2  
if a == 2 :  
    print "wszystko OK"  
else :  
    print "błąd!"
```

# Python: funkcje

```
def nazwa_funkcji :  
    treść_funkcji
```

```
def zagadka(n) :  
    if n > 1 :  
        return zagadka(n-1) * n  
    else :  
        return 1
```

# Python: funkcje

```
def nazwa_funkcji :  
    treść_funkcji
```

```
def zagadka(n) :  
    if n > 1 :  
        return zagadka(n-1) * n  
    else :  
        return 1
```

**silnia.py**

# Python: import

`import moduł`

`wynik = moduł.metoda(parametry)`

`import moduł as nazwa`

`wynik = nazwa.metoda(parametry)`

`from moduł import metoda`

lub

`from moduł import *`

`wynik = metoda(parametry)`

# Python: for

*for zmienna in zakres :*  
*polecenia*

*zakresem może być np. :*

- gotowa lista
  - for i in [1, 3, 5, 7, 9]
  - for i in lista
  - for i in range(1, 10, 2)
- “specjalny” obiekt
  - for i in xrange(1, 10, 2)



# Python: for

*for zmienna in zakres :*  
*polecenia*

```
for i in [1,2,3] :  
    print i
```

```
i = 1  
print i  
i = 2  
print i  
i = 3  
print i
```

# Python: while

*while warunek :*  
*polecenia*

```
n = 123
while n > 1 :
    if n % 2 == 0 :
        n = n / 2
    else :
        n = 3*n+1
print n
```

# Python: list comprehension

wynik = [*wyrażenie* for *zmienna* in *lista*]

```
>>> a = range(5,11)
```

```
>>> a
```

```
[5, 6, 7, 8, 9, 10]
```

```
>>> b = [x**2 for x in a]
```

```
>>> b
```

```
[25, 36, 49, 64, 81, 100]
```

```
>>> c = [42 for i in range(0,5)]
```

```
>>> c
```

```
[42, 42, 42, 42, 42]
```

# Rysowanie histogramu

```
>>> import pylab
```

```
>>> z = [ (n**2)%10 for n in range(0,1000) ]
```

```
>>> pylab.hist(z)
```

```
>>> pylab.show()
```

# Zadanie

Napisz funkcję, która dla wszystkich liczb naturalnych od 2 do N (N odpowiednio duże) obliczy ilość kroków w algorytmie “ $3n+1$ ” oraz wyrysuje histogram otrzymanych wartości

*Wskazówka.* Rozpocznij od napisania funkcji, która obliczy ilość kroków w algorytmie “ $3n+1$ ” dla danego startowego n.

```
while n > 1 :  
    if n % 2 == 0 :  
        n = n / 2  
    else :  
        n = 3*n+1
```