

# Akademia Pythona

## II Typy i operacje

KN Pythona wita na kursie Pythona.

Plan:

- Wprowadzenie do typów obiektów Pythona
- Typy liczbowe
- Wprowadzenie do typów dynamicznych
- Łańcuchy znaków
- Listy
- Słowniki
- Krotki, pliki oraz pozostałe

Hierarchia programow w Pythonie:

- Programy skladaja sie z modulow
- Moduly zawieraja instrukcje
- Instrukcje zawieraja wyrazenia
- Wyrazenia tworza i przetwarzaja obiekty

Po co korzysta sie z typow wbudowanych:

- Obiekty wbudowane sprawiaja, ze programy pisze sie latwo
- Obiekty wbudowane sa komponentami rozszerzen
- Obiekty wbudowane sa bardziej wydajne od wlasnych struktur danych
- Obiekty wbudowane sa standardowa czescia jezyka

# Najwazniejsze typy danych w Pythonie

Typ obiektu	Przykładowy literal
Liczby	1234, 3.14159, Decimal, Fraction
Lancuchy znakow	'KNJPython', 'Abc'
Listy	[1, [2, 'trzy'], ], [1, 2, 3]
Słowniki	{'kolo': 'pythona', 'wyklad': 2}
Krotki	(1, 2, 'trzy', 3.14158)
Pliki	plik = open('test.txt', 'r')
Zbiory	set('cba'), {'a', 'b', 'c'}
Inne typy podstawowe	Boolean, typy, None
Typy jednostek programu	Funkcje, moduly, klasy
Typy powiazane z implementacja	Kod skompilowany, slady stosu

```
:> 123 + 222
```

```
345
```

```
:> 1.5 * 4
```

```
6.0
```

```
:> 2 ** 2
```

```
4
```

```
:> 2 ** 134
```

```
?
```

# Liczby - podstawy

```
:> len(str(2 ** 1000000000))
```

```
:> import math
```

```
:> math.pi
```

```
3.14159...
```

```
:> math.sqrt(85)
```

```
9.21...
```

```
:> import random
```

```
:> random.random()
```

```
?
```

```
:> random.choice([1, 2, 3, 4])
```

```
?
```

# Lancuchy znakow

Lancuch znakow - sekwencja

```
:> S = 'Python'
```

```
:> S
```

Python

```
:> len(S)
```

6

```
:> S[0]
```

P

```
:> S[1]
```

y



# Operacje na sekwencjach

:> S[-1]

n

:> S[-2]

o

Formalnie:  $S[-1] ==> S[\text{len}(S) - 1]$

:> S[1:3]

yt

Typy niezmiennosc:

- Lancuchy znakow
- Liczby
- Krotki

Typy zmienne:

- Listy
- Slowniki

## Listy:

- Listy są uporządkowanymi kolekcjami dowolnych obiektów
- Dostęp do elementów list można uzyskać za pomocą pozycji przesunięcia
- Listy mają zmienną długość, są niejednorodne i można je dowolnie zagnieżdżać
- Listy należą do zmiennych sekwencji
- Listy są tablicami referencji do obiektów

```
:> l = []
```

```
:> l = list()
```

```
:> l = [1, 2, 3]
```

```
:> l += [4]
```

```
:> l
```

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

```
:> l.append(5)
```

```
:> l.pop(0)
```

```
1
```

```
:> l
```

```
[2, 3, 4, 5]
```

Listy Pythona wspierają dowolne zagnieżdżanie obiektów:

```
:> m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
:> m[0][2]
```

3

```
:> col2 = [row[1] for row in m]
```

```
:> col2
```

```
[2, 5, 8]
```

## Slowniki:

- Dostep do slownikow odbywa sie po kluczu, a nie wartosci przesuniecie
- Slowniki sa nieuporzadkowanymi kolekcjami dowolnych obiektow
- Slowniki maja zmienna dlugosc, sa heterogeniczne i moga dowolnie zagniezdzone
- Slowniki naleza do kategorii zmiennych odwzorowan
- Slowniki sa tabelami referencji do obiektow (tablice asocjacyjne)



```
:> d = {'product': 'ball', 'price': 12.25, 'quant': 23}
```

```
:> d = {}
```

```
:> d['price'] = 32
```

```
:> sklep = {'products': ['product1', 'product2'], 'localization':  
'12.3243|23.4322'}
```

```
:> sklep['products'][0]
```

```
product1
```

## Krotki:

- Krotki sa uporządkowanymi kolekcjami dowolnych obiektow
- Dostep do krotek odbywa sie po wartosci przesuniecie
- Krotki naleza do kategorii niezmiennych sekwencji
- Krotki maja stala dlugosc, sa heterogeniczne i mozna je dowolnie zagniezdzać
- Krotki sa tablicami referencji do obiektow

```
:> t = (1, 2, 3, 4)
```

```
:> len(t)
```

```
4
```

```
:> t + (5, 6)
```

```
(1, 2, 3, 4, 5, 6)
```

```
:> t[0]
```

```
1
```

```
:> t[0] = 12
```

```
?
```

# Krotki - zagniezdzenie

```
:> t = ('abc', [1, 2, (3, 4)])
```

```
:> t[1][2][2]
```

```
4
```

```
:> f = open('out.txt', 'w')  
:> f.write('python file contents')  
:> f.close()
```

```
:> f = open('out.txt')
```

```
:> text = f.read()
```

```
:> text
```

```
python file contents
```

Zbiory:

```
:> x = set('a')
```

```
:> y = {'a'}
```

```
:> x & y # czesc wspolna
```

```
:> x | y # suma zbiorow
```

```
:> x - y # roznica
```



Liczby o stałej precyzji:

```
:> import decimal
```

```
:> d = decimal.Decimal('3.141')
```

# Inne typy podstawowe

Liczby wymierne:

```
:> from fractions import Fraction
```

```
:> f = Fraction(3, 17)
```

```
:> f + 1
```

# Inne typy podstawowe

Zmienne logiczne:

```
:> bool('abc')
```

True

# Znaczenie True i False w Pythonie

Obiekt	Wartosc
'python'	True
''	False
[]	False
[1]	True
{}	False
1	True
0.0	False
None	False