

## I Podstawowe różnice między Python 2.x a 3.x

1. Uruchom zintegrowane środowisko programistyczne (IDLE) dla Python 3.x
2. Linia poleceń interpretera (Python Shell) jako kalkulator – sprawdź i zanotuj uzyskane wyniki:
  - dodawanie (np.  $7+9$ )
  - odejmowanie (np.  $8-2$ )
  - mnożenie (np.  $3*5$ )
  - dzielenie (np.  $5/2$ )
  - potęgowanie (np.  $4**2$ ; `pow(4, 2)`)
  - część całkowita (np.  $5//3$  oraz  $5.0//3$ )
  - reszta z dzielenia [modulo] (np.  $5\%3$  oraz  $5.0\%3$ )
  - dzielenie z resztą (np. `divmod(5, 3)`)
3. Powtórz zadanie 2 (dla tych samych operandów) w IDLE dla Python 2.x. Czy są jakieś różnice?
4. Porównaj w obu wersjach Pythona (2.x i 3.x) wyniki następujących działań:

```
>>> w = 'żółć'
>>> print(w)      # lub 'print w' w Python2.x
...
>>> w
...
```

Czy są jakieś różnice w wynikach w zależności od wersji Python'a? Jeśli tak, to dlaczego?

## II Zmienne i dynamiczny system typów

Zmienne w Pythonie, to de facto odniesienia do obiektów. Obiekty (np. wartości) są przechowywane w pamięci, a operator (=) łączy w pamięci odniesienie do obiektu („zmienną”) wraz z tym obiektem (wartością). Nie ma zatem potrzeby podawania deklaracji „zmiennej” ani określania typu wartości przez nią „przechowywanej”.

1. Napisz program (IDLE → File → New Window, następnie zapisz w pliku np. `typy.py` i uruchom: F5), który będzie przypisywał zmiennej `x` różne typy danych (`str`, `int`, `float`, `complex`, `boolean`), wyświetlał jej zawartość oraz typ danych przez nią przechowywany. (podpowiedź: funkcja `type(x)` zwraca typ zmiennej `x`).
2. Wykonaj następujące działania (w Python Shell):

```
>>> z='ABC'
>>> w='ABC'
>>> z is w
...
>>> z=1
>>> w=1
>>> z is w
...
>>> z=1.0
>>> w=1.0
>>> z is w
...
>>> z=1.0
>>> w=z
>>> z is w
...
```

Zanotuj wyniki. Skąd się biorą różnice? Co robi operator **is**?

3. Sprawdź, jakie wartości będą „przechowywały” zmienne **x** i **y** po wykonaniu polecenia:

```
>>> x,y = 1, 'dwa'
```

i co się stanie, gdy wykonamy:

```
>>> x,y = y,x
```

### III Pomoc podręczna oraz dokumentacja dostępna on-line

W tej części laboratorium zapoznasz się z możliwościami, jakie oferuje polecenie **help**. W linii poleceń możemy wpisać **help()** i przejść do trybu interaktywnego bądź uzyskać od razu pomoc dotyczącą interesującego nas obiektu np. **help(int)** udostępni nam informację o klasie **int**. Jeśli pomoc nie mieści się na jednym ekranie (jednej stronie) wówczas możemy użyć strzałek do przesuwania tekstu. Aby zamknąć pomoc, należy wpisać **q** lub **quite()**.

W trybie interaktywnym polecenie **keywords** wyświetli listę dostępnych słów kluczowych języka Python. Pomoc na temat poszczególnych słów kluczowych jest dostępny po jego wpisaniu. Polecenie **modules** wyświetli listę dostępnych modułów, natomiast poleceniem **topics** możemy wyświetlić listę tematów dotyczących w języka Python.

Pomoc on-line jest dostępna na stronie <http://docs.python.org>. Możemy wybrać, jakiej wersji Python'a dotyczy pomoc, a następnie wyszukać interesującego nas zagadnienia w wyszukiwarce

(Quick search) bądź przeglądając rozdziały dokumentacji języka Python.

1. Wykorzystując pomoc podręczną bądź on-line sprawdź, jakie metody dla klasy **str** są dostępne.
2. Wynotuj metody, które umożliwiają:
  - zamianę liter z małych na WIELKIE (kilka możliwości, znajdź jak najwięcej),
  - zliczanie wystąpień ciągów znaków (np. 'cy') w tekście (np. 'na cacy tacy cykuta z cytata z Tacyta'),
  - zamianę fragmentu tekstu na inny,
  - sprawdzenie, czy ciąg znaków zawiera same cyfry (kilka możliwości, znajdź możliwie najwięcej),
  - podział tekstu na pojedyncze słowa i zapis ich np. w formie listy,
  - łączenie znaków (np. będących elementami listy [list] lub krotki [tuple]) w dłuższe ciągi tekstowe.

#### IV Ciągi tekstowe

1. Napisz skrypt typu „*hello world*”, który będzie pytał o imię użytkownika, a następnie wyświetlał napis z jego imieniem np. **Witaj, Tomasz!**
2. Ze strony [http://pl.wikipedia.org/wiki/Wiersz\\_stychiczny](http://pl.wikipedia.org/wiki/Wiersz_stychiczny) przekopij cały zamieszczony tam fragment inwokacji Pana Tadeusza A.Mickiewicza i przypisz go do zmiennej **tekst** (jako ciąg tekstowy wielowierszowy!).  
Następnie uzupełnij skrypt o fragment kodu, który obliczy ilość znaków w tekście, zliczy samogłoski, spółgłoski, spacje i inne znaki (tj. przecinki, kropki, wykrzykniki itd.) oraz wyświetli następujący tekst:  
**Tekst składa się z: nn znaków, w tym: xx samogłosek, yy spółgłosek, zz spacji oraz mm innych znaków.** (gdzie nn, xx, yy, zz i mm to odpowiednio wyliczone wartości).
3. Wyświetl co drugi, co trzeci i co siódmy znak tekstu ze zmiennej **tekst** z zadania 2.
4. Dopisz do skryptu kod, który kolejno:
  - wyświetli tylko pierwszy wers inwokacji,
  - zamieniał litery tego wersu na wielkie i wyświetli wynik,
  - zamieni słowo Litwo! na Polsko! (ot, poprawimy wieszczą ;) i wyświetli wynik.
5. Napisz skrypt, który będzie pobierał od użytkownika trzy cyfry (trzy razy po jednej cyfrze) liczby trzycyfrowej, a następnie wyświetli tę liczbę oraz kwadrat tej liczby.
6. Napisz skrypt, który będzie *zawijał białe znaki*, tzn. będzie usuwał tabulacje, znaki nowej linii oraz wielokrotne spacje, zastępując je pojedynczymi spacjami.  
Przykład: zmiana zawierająca tekst wielowierszowy:

```
w=''To   jest       jakiś  
rozstrzelony tekst!'''
```

zostanie przedstawiona w jednej linijce (słowa rozdzielone pojedynczą spacją):

```
To jest jakiś rozstrzelony tekst!
```

Podpowiedź: skup się na metodzie **str.join** i **str.split**.