

## Laboratorium 1 — Zapoznanie się ze środowiskiem pracy.

### Część organizacyjna

**Cel przedmiotu** Zapoznanie studentów z zasadami tworzenia programów i składnią języka Python. Ukształtowanie praktycznych umiejętności programowania w języku Python.

### Zakres materiału

- L1 Zapoznanie się ze środowiskiem pracy. Uruchamianie programów.
- L2 Ćwiczenia w programowaniu proceduralnym.
- L3 Ćwiczenia w wykorzystaniu kolekcji w języku Python.
- L4 Ćwiczenia w korzystaniu z pakietów. Tworzenie i użycie własnych modułów i pakietów.
- L5 Ćwiczenia w programowaniu obiektowym. Tworzenie własnych typów danych. Tworzenie własnych kolekcji.
- L6 Ćwiczenia w odczycie i zapisie plików tekstowych, binarnych i XML.
- L7 Opracowanie programów realizujących obliczenia inżynierskie z wykorzystaniem bibliotek Numpy i Scipy.
- L8 Wizualizacja wyników obliczeń - ćwiczenia w użyciu biblioteki Matplotlib.
- L9 Testowanie programów w języku Python.
- L10 Ćwiczenia w programowaniu GUI. Projekt końcowy.
- L11 Zaliczenie laboratorium.

### Wymagane oprogramowanie

- Python 3.x lub jego dystrybucje np.: Anaconda (zalecana).
- Opcjonalnie środowisko programistyczne z możliwością debugowania kodu np.: PyCharm, Visual Studio 2017 lub nowszy (zalecany), QT Creator.
- Pakiety: numpy, scipy, pillow (lub pil), matplotlib, pyqt.

### Tryb pracy

- Skrypty do napisania na zajęciach (łatwe, średnie, trudne).
- Zapowiedziane wejściówki na komputerze (waga 0.3).
- Programy pisane na zajęciach (waga 0.15).
- Projekt końcowy (waga 0.4).

Ocena końcowa:

$$ocena = \frac{0.3 \cdot \text{śr. wejściówki} + 0.15 \cdot \text{śr. programy} + 0.4 \cdot \text{projekt}}{0.85}$$



## Zapoznanie się ze środowiskiem pracy.

Python jako język skryptowy nie wymaga kompilacji kodu. Do uruchomienia napisanego skryptu wymagany jest jedynie zainstalowany interpreter. Interpreter Pythona można uruchomić bezpośrednio z poziomu konsoli (cmd, powershell, bash) (w przypadku Windowsa warto upewnić się, że został on dodany do zmiennej PATH) lub poprzez jedno z wielu środowisk programistycznych takich jak PyCharm, Visual Studio, czy Eclipse.

Celem dzisiejszych zajęć będzie zapoznanie się z przykładowymi środowiskami oraz napisanie i uruchomienie pierwszego programu.

---

**Zadanie 1** Uruchom interpreter Pythona z poziomu konsoli, wykorzystując polecenie „python”, a następnie wyświetl napis „Hello World!” stosując funkcję print.

**Zadanie 2** Za pomocą notatnika napisz swój pierwszy skrypt o poniższej treści, zapisz powstały skrypt jako „witaj.py”, a następnie uruchom z poziomu konsoli.

```
imie = input("Jak masz na imię?") #pobranie danych
print('Witaj', imie)
```

Zauważ, że w przypadku języka Python znakiem oddzielającym kolejne linie kodu jest znak nowej linii „ENTER”.

---

W celu instalacji pakietów, można posłużyć się poleceniem pip. W tym celu należy w konsoli (cmd) wpisać polecenie:

```
pip install nazwa_pakietu
```

Aby odinstalować pakiet, należy skorzystać z polecenia:

```
pip uninstall nazwa_pakietu
```

W celu wyświetlenia listy aktualnie zainstalowanych pakietów, można posłużyć się poleceniem list:

```
pip list
```

lub z modyfikatorem:

```
pip list -o
```

w celu wyświetlenia przestarzałych pakietów lub

```
pip list -u
```

w celu wyświetlenia aktualnych pakietów.

Dla Condy polecenia wyglądają podobnie — należy jedynie użyć polecenia conda, zamiast pip. Dodatkowo przydatnym poleceniem jest update w połączeniu z modyfikatorem –all, które powoduje aktualizację wszystkich zainstalowanych pakietów.

---

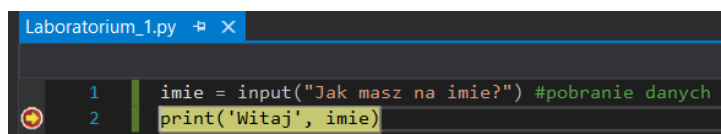
**Zadanie 3** Wraz ze standardową instalacją Pythona dostarczone jest proste graficzne środowisko programistyczne „IDLE”. Pozwala ono na pisanie, debugowanie oraz uruchamianie skryptów. Uruchom „IDLE” a następnie: załaduj wcześniej napisany skrypt, przełącz „IDLE” w tryb debugowania i uruchom załadowany skrypt.

**Zadanie 4** W kodzie z poprzedniego zadania, usuń nawias zamykający od funkcji print. Jak zachowa się interpreter?



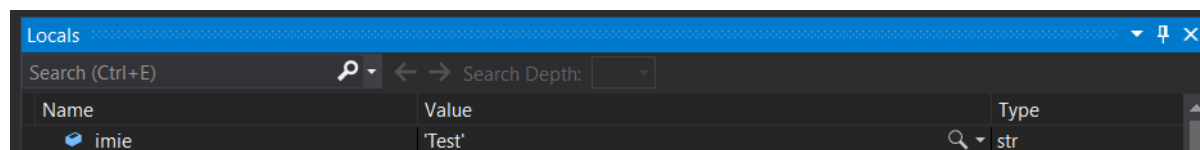
**Zadanie 5** W kodzie z zadania 3 zadania, zmień nazwę funkcji `print` na `prin`? Jak zachowa się interpreter?

**Zadanie 6** Znacznie potężniejszym narzędziem jeśli chodzi o funkcjonalność jest środowisko Visual Studio z zainstalowanym pakietem Python Tools. Poza zaawansowanymi możliwościami debugowania pozwala ono chociażby na zarządzanie projektami składającymi się z wielu plików. Twoim zadaniem będzie utworzenie projektu języka Python „Python Application”, dodanie do niego utworzonego wcześniej skryptu, wstawienie „breakpoint-a” na drugiej linii i uruchomienie skryptu.



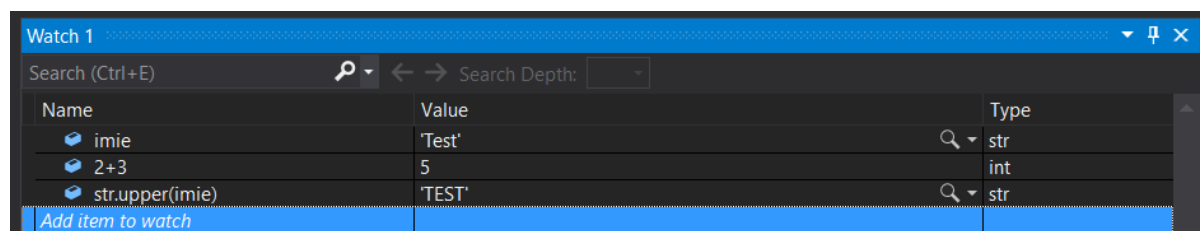
```
Laboratorium_1.py
1  imie = input("Jak masz na imie?") #pobranie danych
2  print('Witaj', imie)
```

W momencie zatrzymania skryptu na „breakpointie” w zakładce **Locals** wyświetlone zostaną wszystkie dostępne w tym momencie zmienne lokalne w postaci trójek (nazwa, wartość, typ) np.:



Name	Value	Type
imie	'Test'	str

Zakładka **Watch** pozwala natomiast na ręczne wprowadzenie interesujących nas zmiennych lub też sprawdzenie całych wyrażeń, możemy tutaj np. umieścić wywołanie funkcji dla danej zmiennej, sprawdzić wynik sumy dwóch zmiennych.



Name	Value	Type
imie	'Test'	str
2+3	5	int
str.upper(imie)	'TEST'	str

W przypadku uruchomienia skryptu z aktywnym „breakpoint-em” w zakładce **Output** możemy spodziewać się dodatkowych informacji o przebiegu skryptu:

```
Loaded '__main__'
Loaded 'runpy'
Witaj test
The thread 'MainThread' (0x1) has exited with code 0 (0x0).
The program 'python.exe' has exited with code 0 (0x0).
```

### Pobieranie informacji od użytkownika.

Do pobierania danych od użytkownika służy funkcja `input`, która przyjmuje jeden parametr typu `string` i zwraca także napis. W celu konwersji zwracanej wartości, należy posłużyć się jedną z dostępnych funkcji do konwersji danych, takich jak np. `int`, `float`.

```
x = int(input("Ile masz lat? "))
print("Masz", x, "lat")
```

**Zadanie 7** Pobierz od użytkownika następujące informacje: kierunek studiów, rok studiów, średnią uzyskaną w poprzednim semestrze studiów. Pamiętaj, aby dokonać odpowiedniej konwersji pobranych danych. Wyświetl pobrane informacje w następującej postaci: "Kierunek: xxx, Rok studiów: yyy, Średnia: zzz".

### Wyświetlanie informacji/danych.

Do wyświetlania tekstu na ekranie komputera służy funkcja `print`. Poza domyślnym wywołaniem z jedną wartością

```
print(x)
```

można użyć jej do wyświetlania wielu zmiennych/tekstu podając kolejne elementy po przecinkach:

```
print("x", a, 1.0)
```

W przypadku, gdy chcemy umieścić zmienną po środku tekstu bez jego rozbijania na części możemy wykorzystać f-stringi:

```
a = 3
b = 5
print(f"x = {a}, y = {b}")
```

Dodatkowo, Python umożliwia skonfigurowanie formatu wyjścia z wykorzystaniem dwóch parametrów:

- `sep` — separator wykorzystywany do rozdzielania kolejnych wyświetlanych elementów; domyślnie jest to znak spacji,
- `end` — znak umieszczany na koniec wyświetlanej frazy (czyli na koniec każdego `print`); domyślnie jest to znak nowej linii.

```
print("a", "b", "c", sep = "-*-", end = " | ")
print("kolejna linia")
```

wynik:

```
a-*-b-*-c | kolejna linia
```

**Zadanie 8** Pobierz od użytkownika: adres zamieszkania, kod pocztowy i miasto. Wyświetl pobrane informacje w jednej linii rozdzielając je przecinkami. W tym celu zastosuj wywołanie funkcji `print` z odpowiednim parametrem.

### Zmienne w Pythonie.

Typ zmiennych w języku Python jest ustalany automatycznie przez interpreter i nie jest on podawany przy tworzeniu zmiennej. Przy czym nie sprawia to, że język Python jest językiem słabo typowanym. Zastosowane podejście jest zbliżone do tworzenia zmiennych z użyciem słowa `auto` w `C++` lub `var` znanego z `C#`. Aby utworzyć zmienną podajemy jej nazwę, a następnie wartość poprzedzoną znakiem „`=`”. W celu sprawdzenia typu danych, można skorzystać z funkcji `type`.

```
a = 5
type(a)
```

wynik:

```
<class 'int'>
```



W Pythonie nie brakuje oczywiście zmiennych typu `bool`, które mogą przyjmować jedną z dwóch wartości `True` lub `False`.

W przypadku zmiennych logicznych (`bool`) istnieje możliwość stosowania operacji logicznych, takich jak `and`, `or`, `not`.

---

**Zadanie 9** Wykorzystując funkcje konwersji danych (`int()`, `float()`) oraz funkcję `input`, utwórz skrypt, który pobierze od użytkownika jego wzrost w centymetrach oraz wagę w gramach, a następnie wyświetli wagę w kilogramach i wzrost w metrach wraz ze stosownym komunikatem oraz obliczy wskaźnik BMI i go wyświetli.

**Zadanie 10** Napisz program, bez wykorzystania instrukcji warunkowych, który pobierze od użytkownika pewną liczbę i sprawdzi, czy jej wartość jest większa od 10.

**Zadanie 11** Napisz program, bez wykorzystania instrukcji warunkowych, który pobierze od użytkownika pewną liczbę i sprawdzi, czy:

- a) podana liczba jest większa od 10 i jednocześnie mniejsza od 20;
- b) podana liczba jest mniejsza od 0 lub większa od 20.

---

### Komentarze w Pythonie.

W Pythonie istnieją dwa typy komentarzy:

- jednoliniowy oznaczany znakiem `#`,
- wieloliniowy rozpoczynający i kończący się `"""`.

---

### Typy danych.

W Pythonie dostępne są następujące wbudowane typy danych:

- liczbowe: `int`, `float`, `complex`;
- tekstowe: `str`;
- logiczne: `bool`;
- sekwencje: `list`, `tuple`;
- słowniki: `dict`;
- zbiory: `set`, `frozenset`;
- binarne: `bytes`, `bytearray`.

Wyróżnić możemy także klasę `slice(start, stop, step)`, wykorzystywaną przy indeksowaniu.