
Laboratorium 1

Środowisko pracy i pierwsze programy

Wstęp do programowania

Autor listy zadań: Grzegorz Madejski

Cel zajęć

Nauka programowania. Pisanie prostych programów z użyciem typowych konstrukcji (instrukcje warunkowe, pętle, itp.).

Sposób zaliczenia przedmiotu

Na ocenę z przedmiotu składają się:

- a) Krótkie kartkówki
- b) Jeden większy sprawdzian (kolokwium).
- c) Mały projekt programistyczny na koniec semestru.

Sprawdź sylabus! 😊

Język programowania

Wybieramy Pythona. Jest to język popularny, prosty, dobrze udokumentowany i nadaje się jako pierwszy język do nauki.

Jak programować?

- Pythona należy zainstalować korzystając ze strony <https://www.python.org/downloads/>. Na komputerach w pracowni komputerowej już jest zainstalowany i tego nie robimy!

Jeśli mamy Linuxa, można pisać programy wykorzystując dowolny edytor (nawet konsolowy) np. Vi/Vim, Gedit, Geany, Atom, Sublime Text.

Programy uruchamiamy w konsoli poprzedzając nazwę pliku komendą python lub python3 (w zależności od wersji).

- W Pythonie wygodnie programuje się też w specjalnych środowiskach programistycznych np.
 - a) Visual Studio Code <https://code.visualstudio.com/>
 - b) PyCharm <https://www.jetbrains.com/pycharm/>
 - c) Spyder <https://www.spyder-ide.org/>
 - d) Anaconda <https://www.anaconda.com/products/individual>
- Te środowiska są szczególnie przydatne, gdy nasze projekty staną się większe i będą korzystały z wielu paczek.

- W razie problemów z Pythonem, można ostatecznie korzystać ze strony <https://replit.com/languages/python3>
Wówczas programy piszemy na stronie i niczego nie trzeba instalować. Ale nie zalecam tego sposobu.

Na zajęciach zachęcam zmierzyć się z Visual Studio Code lub z PyCharmem. Osoby bardziej zaawansowane mogą wybrać inny sposób.

Przydatne linki

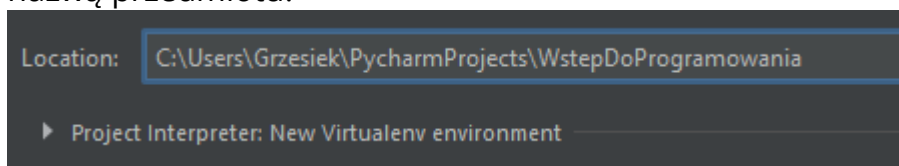
Proszę przejrzeć poniższe strony:

- Fajny tutorial o programowaniu w Pythonie:
<https://www.w3schools.com/python/default.asp>
Oferuje nawet możliwość testowania swoich programów online.
- Oficjalny tutorial Pythona po polsku:
<https://docs.python.org/pl/3/tutorial/index.html>
- Jak ładnie pisać w Pythonie. Poradnik stylu PEP8:
<https://www.python.org/dev/peps/pep-0008/>
- Fajna książka z zadaniami do Pythona
https://www.brianheinold.net/python/A_Practical_Introduction_to_Python_Programming_Heinold.pdf

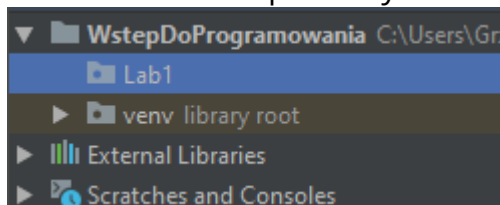
Zadanie 1

Stwórzmy nasz pierwszy program.

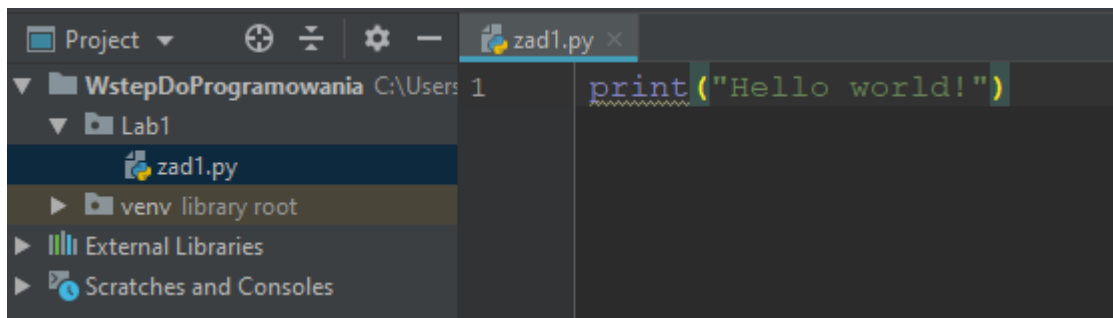
- Po uruchomieniu programu PyCharm i wstępnej konfiguracji stwórz projekt z nazwą przedmiotu:



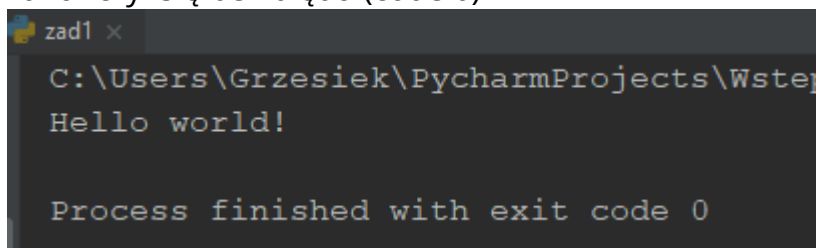
- Stwórz folder dla pierwszych laboratoriów, np. „Lab1”



- W tym folderze stwórz plik „zad1.py”. Następnie uzupełnij go instrukcją drukującą napis „Hello World!”



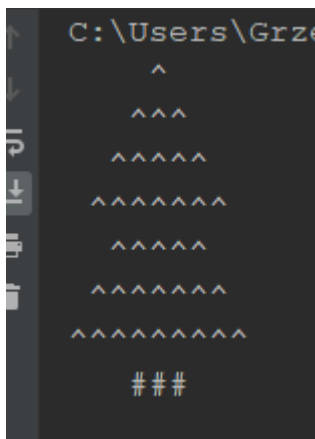
- Zapisz ten program. Uruchom ten program klikając na jego nazwę prawym przyciskiem myszy i wybierając komendę „Run” (z zieloną strzałką). Na dole w konsoli powinien pojawić się napis „Hello World” i informacja, że program zakończył się bez błędu (code 0)



Gratulacje! 😊 Pierwszy program, który wyświetla jedynie napis, już za nami.

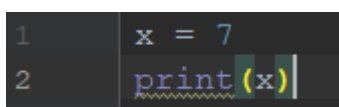
Zadanie 2

Stwórz program zad2.py, który wielokrotnie wykorzysta instrukcję print i wyświetli choinkę złożoną ze znaków ^ i # tak jak na obrazku poniżej.



Zadanie 3

W językach programowania wykorzystuje się zmienne, by przechowywać dane. Stwórzmy zmienną x, pod którą podstawimy liczbę 7. Następnie wyświetlimy x.



Co wyświetla się po uruchomieniu programu? Zapisz program jako zad3.py.

Zadanie 4

Przepisz poniższy program, zapisz go jako `zad4.py` i uruchom go. Przeanalizuj go linijka po linijce zwracając uwagę na następujące rzeczy:

- Ile zmiennych wykorzystaliśmy w programie?
- Do czego służą operatory: `+` `-` `*` `/` `?`
- Jak można wydrukować w jednej linijce tekst i wartość zmiennej razem?
- Gdzie stawiać spacje, by program napisany był ładnym stylem (PEP8)?

```
1  a = 7
2  b = 5
3  print(a)
4  print(a + 3)
5  print(a * 3)
6  print(a + b)
7  print(a / b)
8  print("Suma to a i b to: ", a + b)
9  print("a = ", a, ", b = ", b)
10 c = b - a
11 print(c)
```

Zadanie 5

Stwórz program `zad5.py`, który pod zmiennymi `a` i `b` będzie przechowywał dwie liczby (wpisz dowolne), i będzie obliczał sumę kwadratów tych liczb. Wynik powinien być wyświetlony w następujący sposób:

```
Liczba a = 2 ; liczba b = 3
Suma kwadratow tych liczb to 13 !
```

Zadanie 6

Stwórz program `zad6.py`, który w zmiennych przechowuje długość podstawy (`a`) trójkąta i wysokość padającą na tę podstawę (`h`) i obliczy i poda pole tego trójkąta.

Zadanie 7

Stwórz program `zad7.py`, który w zmiennych przechowuje boki prostokąta. Program obliczy i wyświetli pole oraz obwód tego prostokąta.

Zadanie 8

Pod zmienną można też przechowywać łańcuchy znaków. Stwórz i przetestuj poniższy program.

```
1  znaki = "abc hej!"
2
3  print("Znaki to: ", znaki)
4
```

Zadanie 9

Do czego służy specjalny argument „sep”? Stwórz i przetestuj poniższy program.

```
1 a = 3
2 print("Gdy do", a, "dodasz 1, to wyjdzie", a + 1)
3 print("Gdy do", a, "dodasz 1, to wyjdzie", a + 1, sep="")
4 print("Gdy do", a, "dodasz 1, to wyjdzie", a + 1, sep="-")
```

Zadanie 10

Stwórz program, który w zmiennej bombka przechowuje znak, np. literę „o”.

```
1 bombka = "o"
```

Program powinien wyświetlić choinkę z bombkami (umieść ze 4 bombki w różnych miejscach wewnątrz choinki). Wszystkie bombki w choince powinny być znakiem ze zmiennej bombka. W konsoli powinno wyświetlić się coś takiego:

```
  ^
  ^^^
 ^o^^^
 ^^^^o^
  ^^^^
 ^^o^^^^
 ^^^^o^^^
 ^^^^o^^^
  ###
```

Gdy użytkownik programu zmieni bombkę na inny znak, to program wyświetli inną choinkę, np.

```
1 bombka = "*"
  ^
  ^^^
 ^*^^^
 ^^^^*^
  ^^^^
 ^^*^^^^
 ^^^^*^^^
  ###
```

Zadanie 11

Niektórych instrukcji Pythona nie ma w jego „gołej” wersji i trzeba je importować ze specjalnych paczek. Paczka „math” zawiera wiele funkcji matematycznych. Przepisz i przetestuj poniższy program. Wyjaśnij linijka po linijce, co oznaczają jego funkcje.

```

1  import math
2
3  a = 16
4  b = math.sqrt(a)
5  print(b)
6
7  print(math.pi)
8  print(math.sin(0))
9  print(math.sin(math.pi/2))
10 print(math.floor(2.3))

```

Zadanie 12

Napisz program, który w zmiennej `a` przechowuje bok kwadratu. Program obliczy i wyświetli:

- Pole kwadratu.
- Obwód kwadratu.
- Przekątną kwadratu.

Zadanie 13

Rozwartość kąta można wyrazić w stopniach lub radianach. Napisz program, który demonstruje konwertowanie stopni na radiany, i vice versa.

Zadanie 14

Napisz program, który oblicza pole trójkąta. Podamy bardziej formalną specyfikację:

Wejście: bok `a`, bok `b` trójkąta i kąt `alfa` między nimi wyrażony w stopniach

Wyjście: pole trójkąta (wyświetlone na ekranie)

Warunki poprawności zadania: Zawsze pojawi się wynik. Zakładamy, że użytkownik poda dodatnie liczby dla boków, a kąt będzie mniejszy niż 180 stopni. Jeśli użytkownik poda liczby spoza zakresu, to wyświetlony wynik będzie oczywiście błędny.

Zadanie 15

Napisz program, który oblicza odległość pomiędzy dwoma punktami na płaszczyźnie:

Wejście: współrzędne punktu A: `x_a` i `y_a` oraz współrzędne punktu B: `x_b`, `y_b`

Wyjście: odległość pomiędzy punktami

Warunki poprawności zadania: Zawsze pojawi się wynik.