



Jeżeli szykujesz się na przygodę z sympatycznym gadem, to nie będzie to takie proste. Nazwa Python pochodzi bowiem od serialu "Latający cyrk Monty Pythona", którego wielkim fanem jest twórca tego języka, holender Guido van Rossum. W czasie nauki, bardziej niż na styczność z wężem, nastaw się zatem na dozę brytyjskiego humoru.

Środowisko programistyczne **Python 3 + edytor IDLE.**



Pracę z instrukcją warto rozpocząć po obejrzeniu poniższego filmu:

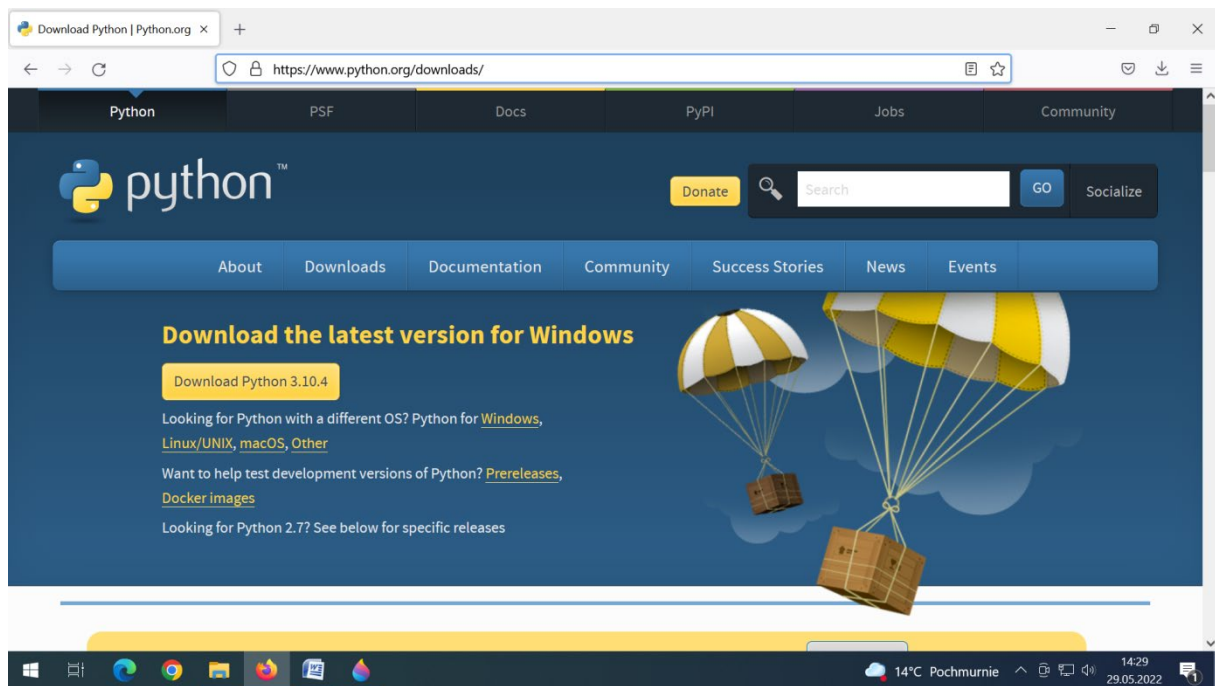
Kurs Python : 2. Instalacja oprogramowania

<https://www.youtube.com/watch?v=STawyS89fgM>

IDLE umożliwia tworzenie programów w języku Python, zapisywanie i uruchamianie. Zintegrowane środowisko programistyczne IDLE zawiera **powłokę Pythona(Shell)**, **edytor kodu źródłowego** i **interpreter** oraz inne narzędzia wspomagające programowanie.

Środowisko programistyczne języka Python możemy zainstalować ze strony:

<https://www.python.org/downloads/>

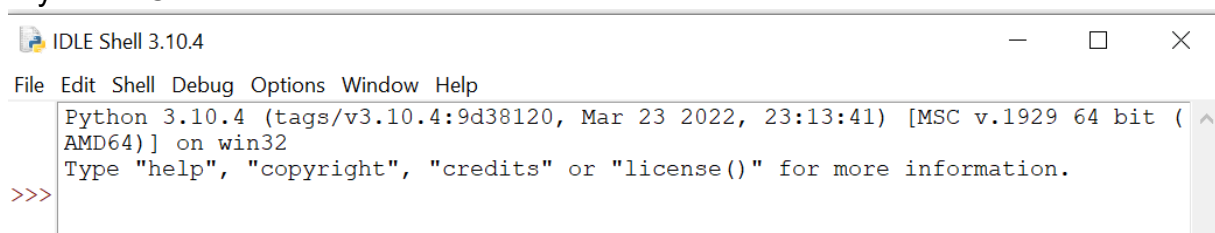


Aby uruchomić IDLE, należy z menu **Start** wybrać program **Python 3.10/IDLE (Python 3.10 64-bit)**. Otworzy się okno programu powłoka Pythona (Python Shell). Okno powłoki Pythona umożliwia pracę w dwóch trybach:

- interaktywnym;
- skryptowym.

Pisanie poleceń w trybie interaktywnym:

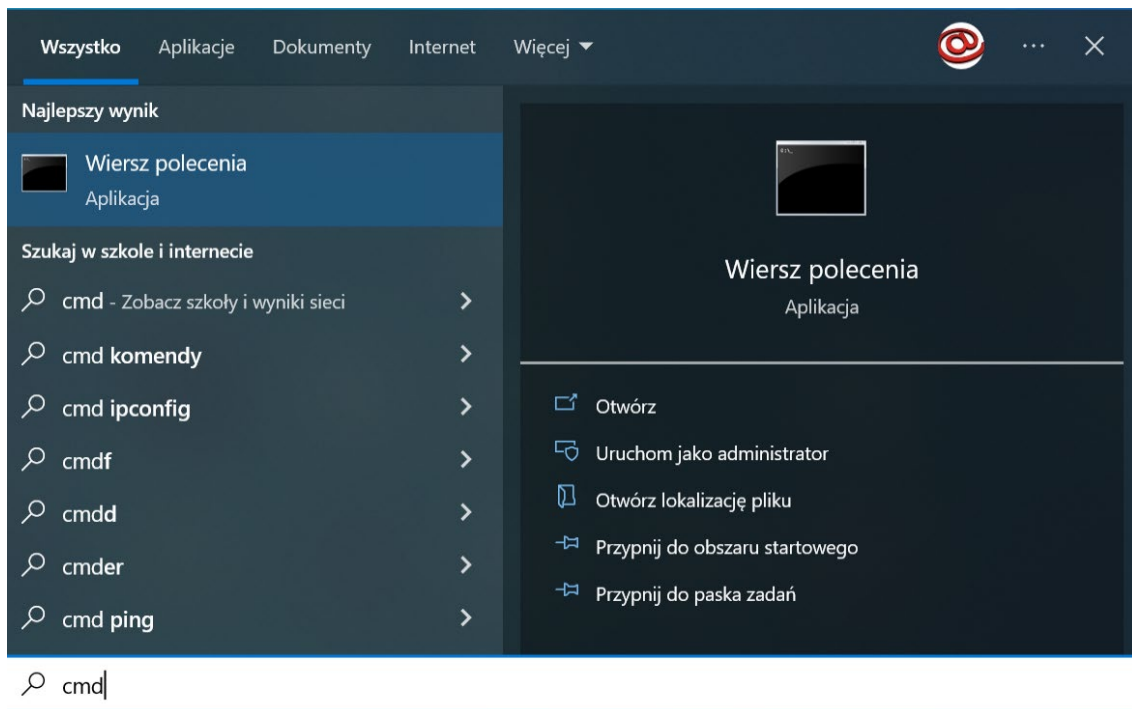
Powłoka Pythona uruchamia się w trybie interaktywnym. Po znaku zachęty **>>>** widzimy migający kursor. W tym miejscu możemy napisać instrukcję lub wyrażenie. po naciśnięciu klawisza **Enter** instrukcja zostanie wykonana, a wyrażenie obliczone. Trybem interaktywnego używaliśmy się także w innych językach np: Logo i Scratch czy też w C++.



Konsola pozwala nam wykonać kod i sprawdzić jego działanie.

Ćwiczenie 1:

Teraz sprawdźmy czy możemy uruchomić Python-a z poziomu Windowsowego wiersza poleceń. Aby to zrobić klikamy po prostu na przycisk **START**/wpisujemy komendę **cmd** i wybieramy **wiersz polecenia**.

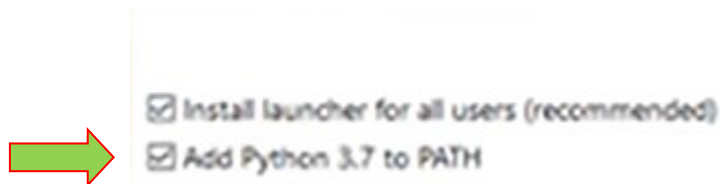


W wierszu polecenia wpisujemy komendę python i wciskamy ENTER.

```
Wiersz polecenia - python
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Andrzej>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Jeśli wyświetli Ci się komunikat o wersji Pythona to wszystko w porządku. W przypadku komunikatu o błędzie, prawdopodobnie zapomniałeś zaznaczyć ptaszka w oknie dodatnia ścieżki umożliwiającej uruchomienie Python-a z Windowsowego okna poleceń. Ponowna instalacja powinna pomóc.



Teraz sprawdźmy działanie trybu interaktywnego:

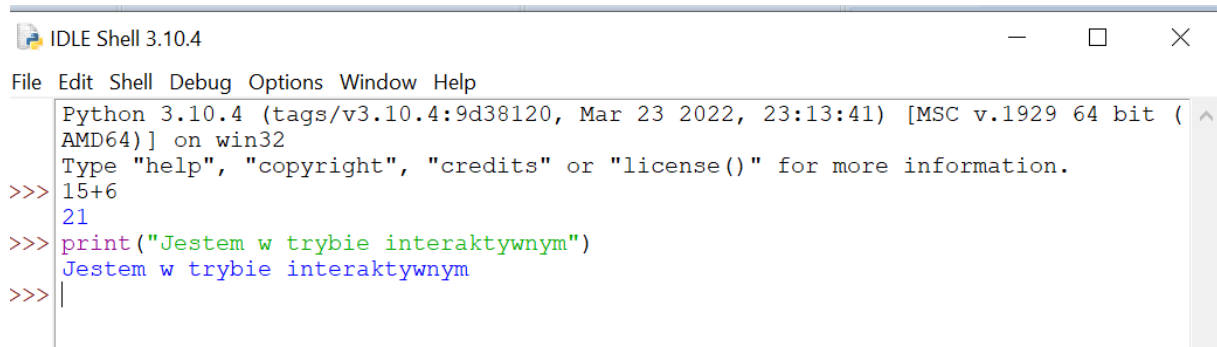
Ćwiczenie 2:

Uruchom środowisko IDLE.

W oknie powłoki Pythona sprawdź działanie trybu interaktywnego. Wpisz po znaku zachęty:

15 + 6 i naciśnij Enter,

print("Jestem w trybie interaktywnym") i naciśnij Enter.



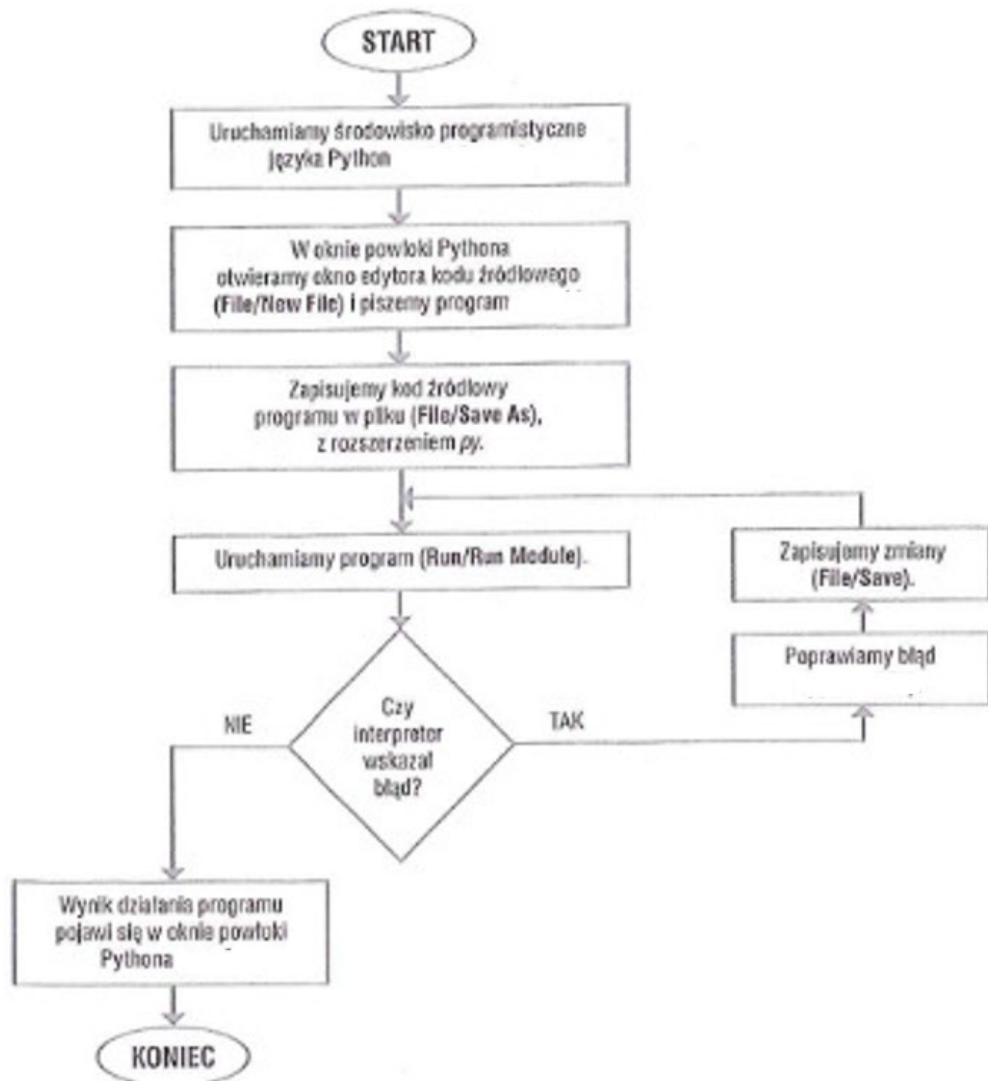
```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 15+6
21
>>> print("Jestem w trybie interaktywnym")
Jestem w trybie interaktywnym
>>> |
```

Pisanie programów w trybie skryptowym

W językach interpretowanych, takich jak Python, możemy napisać program składający się z wielu poleceń (tworzący tzw. skrypt), zapisać go w pliku tekstowym, a następnie uruchomić, używając interpretera. Mówimy wówczas, że pracujemy w trybie skryptowym.

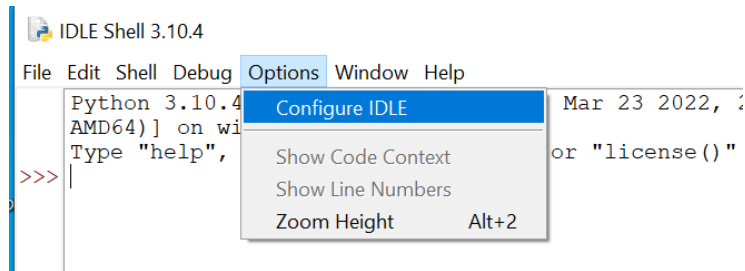
Do trybu skryptowego możemy przejść z trybu interaktywnego z okna powłoki Pythona. Program piszemy w edytorze kodu źródłowego, zapisujemy w pliku na dysku, a następnie uruchamiamy (poprzez wywołanie interpretera Pythona).

W języku Python możemy wyróżnić etapy tworzenia programu pokazane poniżej:



Podstawowe zasady pisania programu w języku Python

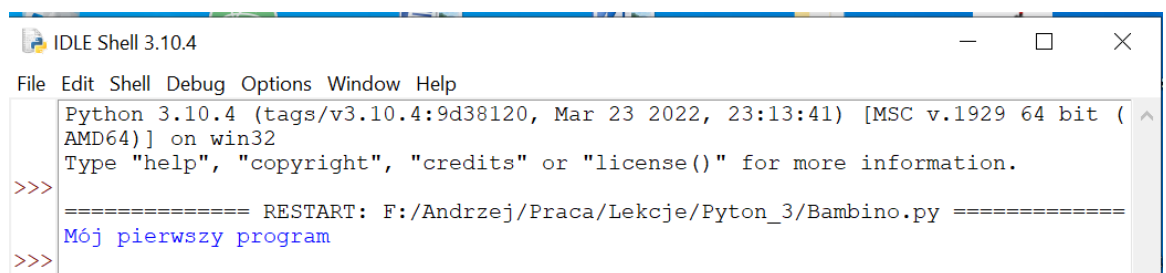
- W programowaniu w języku Python, jak w każdym języku, bardzo istotna jest prawidłowa kolejność zapisu poleceń.
- Instrukcje języka Python, podobnie jak innych języków programowania, realizują takie czynności, jak: wprowadzanie danych, wyprowadzanie wyników, wykonywanie obliczeń, sprawdzanie warunków czy powtarzanie operacji.
- Bloki instrukcji wyodrębnia się inaczej niż w innych językach. Nie stosuje się nawiasów ani słów kluczowych, tylko wcięcia, składające się z przynajmniej jednej spacji (**przyjęte jest wcięcie składające się z czterech spacji**).
- Komentarze rozpoczyna się znakiem #. Tekst po znaku # do końca wiersza jest ignorowany przez interpreter.
- Na zrzucie poniżej pokazałem jak można zmieniać rozmiar czcionki:



Ćwiczenie 3

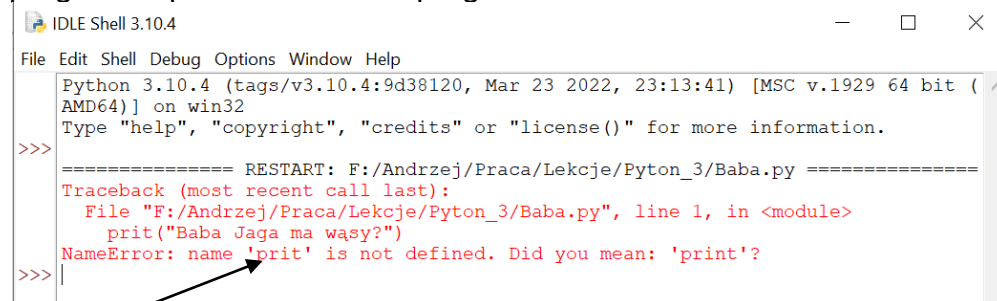
Tworzenie programu w języku Python

1. Utwórz nowy plik źródłowy i napisz program wyświetlający na ekranie napis „Mój pierwszy program”.
2. Zapisz program w pliku pod nazwą Bambino.
3. Uruchom program.
4. Jeśli interpreter wykrył błędy, popraw je. Zapisz plik pod tą samą nazwą i ponownie uruchom.



W trakcie wykonywania programu interpreter rozpoznaje polecenia języka Python i zamienia je na instrukcje wewnętrznego języka procesora. Po wykonaniu programu wyniki pojawiają się na wybranym urządzeniu zewnętrznym, np. na ekranie monitora (tu w oknie powłoki Pythona).

W odróżnieniu od kompilatora, interpreter próbuje wykonać program instrukcja po instrukcji. Jeżeli w programie wystąpi błąd, interpreter przerwie wykonanie programu i wyświetli komunikat o błędzie. Należy poprawić wskazany błąd, ponownie zapisać program w pliku i uruchomić program.



Komunikat interpretera informuje o nieznanej nazwie prit. Podpowiada prawidłową składnię polecenia: print.

Stosowanie zmiennych

Chcemy obliczyć iloczyn dwóch liczb całkowitych. Jak napisać w języku Python program umożliwiający wprowadzenie dwóch liczb całkowitych z

klawiatury, obliczający ich iloczyn i wyprowadzający wynik obliczeń na ekran?

W wielu językach programowania (np. w języku C++) przed użyciem zmiennej należy ją zadeklarować, czyli określić jej nazwę i typ. **W języku Python nie deklaruje się zmiennych.** Nie musimy wcześniej określać typu zmiennej, czyli rodzaju przechowywanych w niej danych (np. liczby całkowite, rzeczywiste, znaki).

Zasady dotyczące nazw zmiennych w języku Python

1. Wielkie i małe litery w nazwach są traktowane odmiennie (np. suma i Suma będą oznaczać różne zmienne). Pisząc program, należy zwracać uwagę na poprawne używanie małych i wielkich liter.
2. W nazwach zmiennych powinno się używać liter, znaku podkreślenia i cyfr. Nazwa nie może zaczynać się od cyfry. Przyjęte jest stosowanie małych liter i niestosowanie polskich liter.
3. W nazwach zmiennych nie wolno stosować spacji. W przypadku nazw kilkuczłonowych zamiast spacji stosujemy znak podkreślenia.
4. Wskazane jest tworzyć nazwy zmiennych, które określają znaczenie danej zmiennej, np. suma, liczba elementów.

Interpreter określa typ zmiennej na podstawie rodzaju danych, jaki jej przypiszemy. **Nie ustanawia jednak typu zmiennej na stałe**, ponieważ przy kolejnym przypisaniu może się on zmienić. Na przykład, jeśli na początku programu przypiszemy zmiennej liczba wartość całkowitą (27), to jej typ jest całkowity. Jeśli jednak w dalszej części programu przypiszemy jej wartość rzeczywistą (np. 81,8, którą w języku Python zapisujemy z kropką: 81.8), to zmienna liczba będzie typu rzeczywistego.

Nadawanie zmiennym wartości

Zmiennej stosowanej w programie możemy nadać konkretną wartość za pomocą **instrukcji przypisania**. instrukcji przypisania zmiennej podanej po lewej stronie instrukcji zostanie obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji.

Instrukcja przypisania	zmienna = wyrażenie
------------------------	---------------------

Wyrażenie to m.in.: konkretna wartość, zmienna, wyrażenie algebraiczne.

Przykład	Opis
rok = 2022	Zmiennej rok przypisujemy wartość, która jest liczbą całkowitą, czyli zmienna rok będzie typu całkowitego (int).
x = 6.24	Zmiennej x przypisujemy wartość, która jest liczbą rzeczywistą, czyli zmienna x będzie typu rzeczywistego (float).

<code>z = y</code>	Zmiennej z przypisujemy wartość zmiennej y.
<code>obwod = 2 * a + 2 * b</code>	Zmiennej obwod przypisujemy wartość wyrażenia $2 \cdot a + 2 \cdot b$; jeśli wynik obliczeń będzie liczbą całkowitą, interpreter określi dla zmiennej obwod typ całkowity; jeśli wynik będzie liczbą dziesiętną, interpreter określi typ zmiennej jako rzeczywisty.
<code>liczba = 12</code> <code>liczba = 35.4</code>	Zmiennej liczba przypisujemy wartość 12 (która jest liczbą całkowitą), a w drugiej instrukcji - wartość 35,4 (która jest liczbą rzeczywistą); poprzednia wartość zmiennej liczba zostanie usunięta, a interpreter określi dla zmiennej liczba typ rzeczywisty.

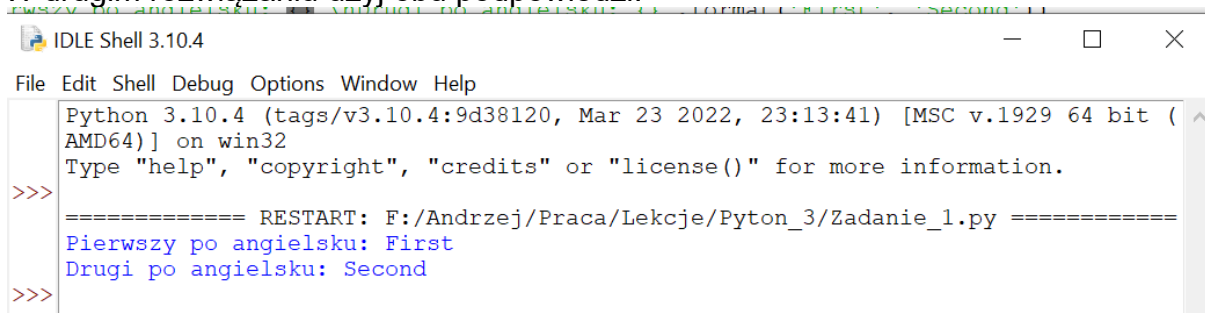
Zadanie 1

Umiesz już wykonywać napisy w Pythonie. Tym razem zadanie wymagające poszukania odpowiedzi. Jak uzyskać poniższe napisy za pomocą jednego polecenia „print”?

Zadanie rozwiąż dwoma sposobami:

W pierwszym rozwiązaniu użyj tylko pierwszej podpowiedzi.

W drugim rozwiązaniu użyj obu podpowiedzi.



```

Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:/Andrzej/Praca/Lekcje/Pyton_3/Zadanie_1.py =====
Pierwszy po angielsku: First
Drugi po angielsku: Second
>>>

```

Podpowiedzi:

- 1.Podobnie jak w C++ możemy użyć: `\n` – znaku nowej linii;
- 2.W nowym formatowaniu miejsca wstawiania oznaczamy przez klamry {}, a napis i argumenty łączymy przez `.format()`.

Zadanie 2

Teraz możemy wykonać zadanie 1 z pierwszej części instrukcji do C++.

Napisz program opisujący zakresy wartości przedstawionych w tabeli (Zakres wartości typów danych – cz.2 instr. C++) umieszczonej w instrukcji w formie tabelarycznej (bez ramek). Wpisujesz z tabeli numery wierszy nr_z_dziennika+4. Np. Uczeń o numerze 7 w dzienniku wpisuje 7 ,11, 15, 19, 23, 27, 3.


```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\Andrzej\Praca\Lekcje\Python_3\Zadanie_2.py =====
Lp.      Typ      Typowy rozmiar w bajtach      rozmiar
7         signed short      2 bajty      od -32768 do 32767
11        Float          4 bajty      3.4 E- 38 do 3.4e+38
15        unsigned short int  2 bajty      0 do 65535
19        unsigned long long   8 bajtów     Od 0 do 18446744073709551615
23        Long              4 bajty      -2147483648 do 2147483647
27        Signed long long    8 bajtów     Od -9223372036854775808 do 92233720368547758
3         unsigned short      2 bajty      0 do 65535
>>>

```

Zmiennej stosowanej w programie możemy również nadać wartość **za pomocą instrukcji przypisania**, wprowadzając wartość z klawiatury w trakcie działania programu. W tym celu stosujemy instrukcję wejścia — funkcję **input()**.

Wprowadzenie danych z klawiatury ---- zmienna = input (tekst_zachęty)

Argumentem funkcji input() (podanym w nawiasach) jest tekst (ciąg znaków, łańcuch). Tekst ten pojawi się na ekranie jako swego rodzaju „zaproszenie” do wpisania ciągu znaków z klawiatury. Jeśli użyjemy funkcji input() w instrukcji przypisania, wpisany przez użytkownika ciąg znaków zostanie zapamiętany w zmiennej podanej po lewej stronie.

Aby dana wprowadzona za pomocą instrukcji input() została zapamiętana jako : liczba, musimy dodać instrukcję, która zamieni ciąg znaków na liczbę: int() — w przypadku liczb całkowitych, float() — w przypadku liczb rzeczywistych.

Przykład	Opis
X= input("Wprowadź liczbę: ")	Jeśli wpiszemy z klawiatury liczbę 123, w zmiennej X zostanie zapamiętany ciąg znaków „123” a nie liczba 123.
Y= int(input("Wprowadź liczbę: "))	Jeśli wpiszemy z klawiatury liczbę 123, w zmiennej Y zostanie zapamiętana liczba całkowita 123.
srednia = float(input("Podaj średnią ocen : "))	Jeśli wpiszemy z klawiatury liczbę 3.3, w zmiennej srednia zostanie zapamiętana liczba rzeczywista 3,3.

`miasto = input("Wprowadź
nazwę miasta: ")`

Jeśli wpisujemy z klawiatury nazwę miasta „Gdansk”, w zmiennej zostanie zapamiętany ciąg „Gdansk”.

Wykonywanie obliczeń

W języku Python obliczenia możemy zapisywać, korzystając z instrukcji przypisania. W obliczeniach stosujemy podstawowe operatory arytmetyczne:

+ (dodawania),
- (odejmowania),
* (mnożenia),
// (dzielenia całkowitego),
/ (dzielenia zmiennoprzecinkowego)
% (reszty z dzielenia dwóch liczb całkowitych).

Wyprowadzanie komunikatów i wyników na ekran monitora

Dotychczas wyprowadzaliśmy komunikaty na ekran monitora, używając polecenia `print`. Za pomocą tego polecenia można również wyprowadzać wyniki obliczeń.

Wyprowadzanie wyników i komunikatów na ekran --- `print (wartość)`

W języku Python `print()` jest funkcją, która wyświetla na ekranie wartość zapisaną w nawiasach i umieszcza kursor w nowym wierszu.

Wartość umieszczana wewnątrz nawiasów to np. zmienna, wyrażenie, a także napis. Napis umieszczany w programie, który ma zostać wyświetlony na ekranie monitora, musi być ujęty w górne cudzysłowy podwójne `" "` lub pojedyncze `' '`. Funkcja `print()` pozwala łączyć wyświetlanie napisów, wyrażeń i zmiennych. Poszczególne wartości, występujące jako parametry funkcji `print()`, należy rozdzielić przecinkami.

Przykład łączenia napisów i zmiennych (`()`zmienna a została uprzednio wprowadzona):

`print ("Dla boku o długości ", a, "pole kwadratu wynosi: ", a *a)`

Zadanie 3

Wykonaj program działania kalkulatora, który po wczytaniu dwóch liczb naturalnych dodatnich w jednej instrukcji poda wynik:

- Suma liczb
- Różnica liczb
- Iloczyn liczb
- Iloraz liczb
- Reszta z dzielenia

Zmienne a i b zapisz z trzema literami nazwiska np.: a_raw.

Wyświetlanie wyniku ilorazu ogranicz do dwóch cyfr po przecinku.

Wygląd ekranu:

Np.

Podaj a=

Podaj b=

Wynik działania programu(wygląd ekranu):

suma=5

różnica=-1

iloczyn=6

iloraz=0,67

reszta z dzielenia=2

```

===== RESTART: F:\Andrzej\
Podaj liczbe a: 2

Podaj liczbe b: 3

suma = 5

roznica = -1

iloczyn = 6

iloraz = 0.67

reszta_z_dzielenia = 2
>>> |

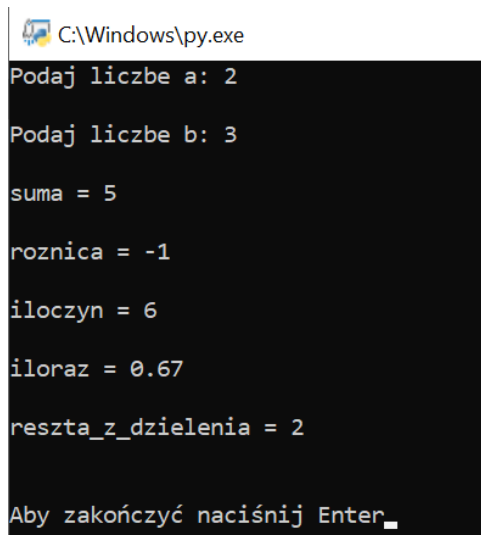
```

W większości języków programowania, w tym w języku Python, część dziesiętną liczby zapisujemy po kropce, a nie po przecinku, jak na matematyce. W ten sam sposób liczby dziesiętne są również wyświetlane na ekranie.

Program możemy również uruchomić, klikając dwukrotnie nazwę pliku z zapisanym programem w Eksploratorze plików — otworzy się okno, w którym zobaczymy wynik działania programu. Aby program został wykonany, niezbędny jest interpreter Pythona.

Aby okno programu nie zamknęło się automatycznie po wyświetleniu napisu, możemy napisać na końcu programu instrukcję oczekiwania na naciśnięcie klawisza Enter:

`input("\n\n Aby zakończyć, naciśnij Enter")`



```

C:\Windows\py.exe
Podaj liczbe a: 2

Podaj liczbe b: 3

suma = 5

roznica = -1

iloczyn = 6

iloraz = 0.67

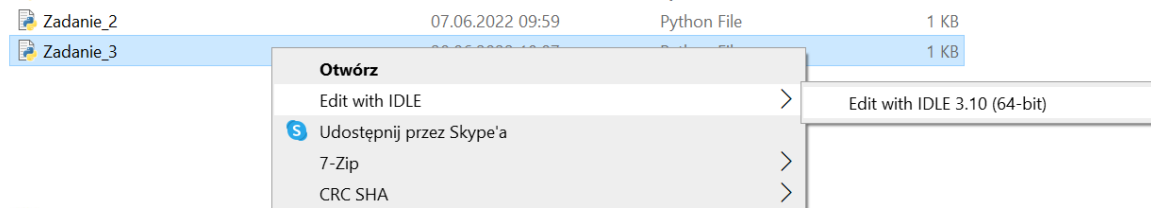
reszta_z_dzielenia = 2

Aby zakończyć naciśnij Enter_

```

W tym przypadku wartość zwracana przez funkcję `input()` jest pomijana (nie jest przypisywana żadnej zmiennej). Umieszczenie zapisu oznacza `\n\n` oznacza wstawienie dwóch pustych wierszy przed komunikatem.

Jeśli zamknęliśmy okno powłoki Pythona, a chcemy edytować program, można z menu kontekstowego nazwy pliku w oknie Eksploratora plików wybrać polecenie Edit with IDLE/ Edit with 3.10



Środowisko programistyczne Pythona IDLE udostępnia dwa tryby pracy:

- ❖ **interaktywny**, w którym możemy napisać jedno polecenie i zatwierdzić je klawiszem Enter; jeśli polecenie będzie poprawne, zostanie od razu wykonane,
- ❖ **skryptowy**, w którym możemy napisać program, zapisać go w pliku, a następnie uruchomić; jeśli program będzie poprawny, zostanie wykonany.

Aby utworzyć program w języku Python, piszemy kod źródłowy w edytorze kodu źródłowego, zapisujemy program w pliku i uruchamiamy za pomocą interpretera Pythona.

W języku Python typ wartości, która jest przypisana zmiennej, określa typ tej zmiennej.

Zmiennej możemy nadać wartość za pomocą instrukcji przypisania. **Po lewej stronie znaku = podajemy nazwę zmiennej, a po prawej: - konkretną wartość**

lub

funkcję input(), dzięki której użytkownik wprowadzi tę wartość z klawiatury po uruchomieniu programu

lub - **wyrażenie, którego wartość zostanie obliczona w czasie działania programu.**