

Informatyka II

Podstawy języka Python

Cel

Celem ćwiczenia jest opanowanie podstaw języka Python w wersji 3.6.1.

Zadanie 1

Uruchom przeglądarkę internetową. Otwórz stronę znajdującą się pod adresem <https://repl.it/languages/python3>. Zapoznaj się z środowiskiem on-line.

Możesz również skorzystać z własnego komputera lub ze środowisk obecnych na komputerach w laboratorium. Upewnij się, że Twoje środowisko obsługuje język Python w wersji 3.6.1 lub nowszej.

Zadanie 2

Przepisz następujący kod do interpretera języka Python:

```
imie = input()
nazwisko = input('Podaj nazwisko: ')
print("Cześć " + imie + " " + nazwisko)
```

Deklarując zmienne w języku Python nie określamy ich typu. Każdy poprawny identyfikator, który nie jest słowem kluczowym oraz nie był wcześniej zdefiniowany w inny sposób, traktowany jest jako zmienna.

Funkcja *input* pozwala pobrać dane z klawiatury i zapisać je do zmiennej. Podanie opcjonalnego parametru powoduje wyświetlenie tekstu na ekranie.

Funkcja *print* pozwala wypisać ciąg znaków (lub inny typ, który można skonwertować do ciągu znaków) na ekran.

Aby uruchomić program w środowisku on-line, wykonaj następujące kroki:

- Krok 1. Przepisz lub skopiuj kod z listingu
- Krok 2. Wciśnij przycisk „run”. Znajduje się on na środku okna, u góry
- Krok 3. Po uruchomieniu programu wpisz swoje imię i zatwierdź klawiszem ENTER

Zadanie 3

Uruchom następujący program napisany w języku Python:

```
wiek = int(input('podaj wiek: '))
rok = int(input('podaj aktualny rok: '))
print("Urodziłeś się w "+str(rok-wiek)+" roku.")
```

Do zamiany ciągu znaków na wartość całkowitoliczbową można użyć funkcji *int*. Do wykonania konwersji odwrotnej służy funkcja *str*. Do konwersji na tym zmiennoprzecinkowy można użyć funkcji *float*.

Zadanie opcjonalne (na plusa):

Mając dwie zmienne - *imie* i *nazwisko* (jak w zadaniu 2), utwórz zmienną *daneOsobowe*, która łączy nazwisko, spację i imię przechowywane we wcześniej wymienionych zmiennych.

Zadanie 4

Uruchom następujący program napisany w języku Python:

```
imie = input('Podaj imię: ')
if imie[-1]=="a":
    print("Nosisz imię żeńskie!")
else:
    print("Nosisz imię męskie!")
```

W języku Python instrukcja warunkowa *if* ma następującą postać:

If [warunek]:

Dwukropek kończący instrukcję jest wymagany i jego pominięcie skutkuje błędem (W przypadku środowiska repl.it, będzie to błąd „*SyntaxError: invalid syntax*”). To samo tyczy się dwukropka po instrukcji *else*. Pomiędzy *if* i *else* można wprowadzić dodatkowy blok: *elif*. Jego użycie jest tożsame z zastosowaniem instrukcji *else* i *if* bezpośrednio po sobie.

Warunki można łączyć za pomocą słów kluczowych *and* („i”) oraz *or* („lub”), a także zaprzeczać ich wynik za pomocą słowa kluczowego *not* („nie”).

W języku Python nie ma żadnych znaków ograniczających blok. Zamiast tego stosuje się wcięcia (tabulator lub 3 następujące po sobie spacje).

Aby dostać się do konkretnego znaku w ciągu znaków, można zastosować operator indeksowania. Argument wskazuje numer znaku, licząc od 0. Ujemne wartości argumentu (jak w przykładzie) wskazują znaki od końca ciągu.

Zadanie opcjonalne (na plusa):

Zmień kod programu w taki sposób aby imiona „Barnaba” i „Kosma” były uznawane za imiona męskie.

Zadanie 5

Uruchom następujący program napisany w języku Python:

```
liczba = int(input('Podaj liczbę: '))
silnia = 1
for i in range(1,liczba+1,1):
    silnia *= i
print("Silnia: "+str(silnia))
```

Tak jak po instrukcji warunkowej, tak i po pętlach należy umieścić znak dwukropka.

Wbudowana funkcja *range* generuje listę liczb, które spełniają pewne kryteria. Pierwszy argument określa wartość początkową, drugi argument wartość końcową, a trzeci argument określa wartość, o którą będzie się różnić kolejny element od poprzedniego. Trzeci argument jest opcjonalny i domyślne jego wartość wynosi 1.

Możliwe jest też ominięcie pierwszego argumentu. Jeżeli do funkcji *range* podany zostanie tylko jeden argument, wartość początkowa zostanie ustalona na 0, krok na 1, a wartość maksymalna zostanie ustalona na taką, którą wskazuje podany argument.

W języku Python istnieje również pętla *while*. Uruchom następujący program napisany w języku Python:

```
liczba = 0
while liczba != 42:
    liczba = int(input("Podaj odpowiedź: "))
    print("Dobra odpowiedź!")
```

Zadanie 6

Uruchom następujący program napisany w języku Python:

```
lista = [1, 2, "tekst", 3.14]
lista.append("nowy element")
lista.extend([3,4,5])
lista.insert(0,"nowy pierwszy element")
lista.remove(4)

for x in lista:
    print(x)
```

Domyślnie język Python nie posiada tablic, a listy. Listy w języku Python nie są ograniczone do jednego rodzaju danych, w jednej liście przechowywać mogą znajdować się elementy o różnych typach.

Aby dodać pojedynczy element do istniejącej listy należy użyć funkcji *append*. W celu dołączenia innej listy na koniec listy już istniejącej, należy użyć funkcji *extend*.

Za pomocą funkcji *insert* można wstawić element w konkretnej pozycji.

Metoda *remove* pozwala usunąć konkretny element. Argumentem metody jest element, który chcemy usunąć. Jeżeli element nie znajduje się w liście, zostanie wyrzucony wyjątek *valueError*.

Do funkcji *print* można bezpośrednio przekazać listę. Wszystkie jej elementy zostaną wtedy wypisane w jednej linii.

Zadanie 7

Napisz w języku Python program który wypisze na ekranie tabliczkę mnożenia (od 1*1 do 10*10). Użyj pętli. Wynik powinien być wypisany w postaci macierzy, jak poniżej:

1	2	3
2	4	6
3	6	9

Zadanie 8

Napisz w języku Python program który prosi użytkownika o podanie liczb. Jeżeli użytkownik poda liczbę inną od 0, program prosi o podanie kolejnej liczby, dopóki użytkownik nie poda 0. Kiedy użytkownik poda liczbę 0, program wypisuje średnią dotychczas podanych liczb (nie licząc zera!).

Zadanie 9

Napisz w języku Python program który prosi użytkownika o podanie pewnej liczby całkowitej *N*. Następnie program prosi użytkownika o wpisanie imion, nazwisk i ocen *N* studentów. Gdy użytkownik skończy podawać dane, program powinien wypisać listę wszystkich studentów wraz z ocenami.

Zadanie 10

Na wzór bazy studentów z poprzedniego ćwiczenia, zaprojektuj bazę samochodów. Baza powinna przechowywać markę, model i rok produkcji. Przed wypisaniem zawartości bazy, posortuj ją względem lat produkcji, rosnąco.