Nazwa implementacji: Nauka języka Python – zmienne

Autor: Piotr Fiorek

Opis implementacji: Poznanie pojęcia zmiennych ich typów oraz sposoby wczytywania danych z klawiatury.

Python jest dynamicznym językiem interpretowanym. „Interpretowany” oznacza, że kod, który napiszemy, możemy natychmiast wykonać bez potrzeby tłumaczenia kodu programistycznego na język maszynowy, czyli formę zrozumiałą przez komputer. Interpreter robi to wszystko za nas, czyli tłumaczy nasz kod oraz natychmiast go uruchamia. Interpreter może również być używany w trybie interaktywnym do testowania kawałków kodu. Właśnie od poznania interpretera zaczniemy naukę Pythona, ponieważ bez interpretera ani rusz. Większość programistów Pythona podczas programowania poza edytorem ma uruchomiony właśnie interpreter, aby na bieżąco testować kawałki swoich programów.

Interpreter uruchamiamy poleceniem

„python3”

. Trójka jest ważna, ponieważ używamy trzeciej wersji Pythona. Po uruchomieniu interpretera komputer powinien wypisać trzy linie tekstu, a w czwartej tak zwany znak zachęty, po którym wpisujemy komendy. Standardowym znakiem zachęty w interpreterze Pythona jest

„>>>”

.

Programując, zawsze operujemy na zmiennych. Zmienną są pudełkami trzymającymi różne informacje. „Dynamiczny” w opisie Pythona oznacza, że nie musimy tworzyć nowej zmiennej, zanim jej użyjemy. Pobawmy się zatem chwilę interpreterem i zaprzyjaźnijmy się z nim, ponieważ jak pisałem, jest on niezbędny każdemu programiście.

Przykłady:

```
>>> a = 3
>>> b = 5
>>> a + b
8
>>> c = 6.5
>>> d = -7
>>> a + c
9.5
>>> b – c
-1.5
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
>>> c + d

-0.5

>>> c – d

13.5

>>> 7/2

3.5

>>> 7//2

3

>>> 2*6

12

>>> 2**6

64

>>> kwiatek = 'stokrotka'

>>> kwiatek2 = ”róża”

>>> kwiatek + ' i ' + kwiatek2

'stokrotka i róża'
```

Jak widać, zmienne mogą mieć różne typy. Najpopularniejsze są typu liczbowe: całkowite („int” od angielskiego integer) oraz tzw. zmiennoprzecinkowe („float” od angielskiego floating point), czyli potocznie liczby z przecinkiem. Część całkowitą od ułamkowej oddzielamy kropką.

W przykładzie widać również napisy, czyli tzw. łańcuchy znakowe. Napisy zapisujemy pomiędzy pojedynczymi lub podwójnymi cudzysłowami. Nie ma znaczenia, których użyjemy, ale ważne, żeby z obu stron użyć tych samych. Python 3 nie ma również problemów z zapisem polskich znaków zarówno w tekstach, jak i nazwach zmiennych. Mimo to dobry zwyczaj mówi, aby w nazwach zmiennych używać tylko znaków z alfabetu łacińskiego. Jak widać w przykładzie, zmienne typu napisowego też można do siebie dodawać, co powoduje sklejanie ich ze sobą.

W Pythonie występują jeszcze inne typu danych, ale te podstawowe na razie nam wystarczą.

Będąc w interpreterze wykonaj:

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
zmienna1 = input(”Podaj imię: ”)

print(”Witaj”, zmienna1)
```

Te dwie instrukcje to już bardzo prosty program. Przy użyciu funkcji „input” (po angielsku wprowadzać), pobiera on do zmiennej to, co użytkownik wpisze z klawiatury, a następnie używając funkcji „print” (po angielsku drukować) wyświetla na ekranie kawałek tekstu oraz zawartość wcześniej utworzonej zmiennej.

Funkcje są to wcześniej zdefiniowane kawałki kodu, których możemy później użyć do wykonania określonej czynności, zamiast wpisywać ten sam kod po raz kolejny.

Rzeczy, które umieszczamy w okrągłych nawiasach po nazwach funkcji, nazywamy parametrami lub argumentami funkcji. Funkcje mogą przyjmować różne ilości argumentów – „input” przyjmuje tylko jeden argument typu napisowego (tekst, który pojawi się, prosząc nas o wprowadzenie danych; jeśli żaden tekst nie zostanie wpisany, żaden nie zostanie wyświetlony), a „print” przyjmuje dowolną ilość parametrów, zależnie od tego ile elementów chcemy wyświetlić. Parametry oddzielamy od siebie przecinkami.

Jak widać na przykładzie funkcji „input” niektóre funkcje pozostawiają coś po sobie. W tym przypadku funkcja „input” pozostawia po sobie to, co użytkownik wpisał z klawiatury, a my wrzucamy to do naszej zmiennej „zmienna1”. Kiedy funkcja pozostawia po sobie jakieś dane, mówimy, że funkcja zwraca dane.

Kolejny program zapiszemy już w pliku, aby prościej było go zmieniać oraz wykonywać wiele razy. W tym celu należy otworzyć prosty edytor tekstu, wpisać do niego instrukcje języka Python, a następnie zapisać z rozszerzeniem

„**.py** ”

. Aby uruchomić tak zapisany program, należy będąc w linii poleceń w tym samym katalogu, gdzie zapisaliśmy nasz plik wpisać:

```
python3 <nazwa-pliku>

Przy czym <nazwa-pliku>
```

Należy oczywiście zastąpić nazwą naszego pliku.

Zatem zmieńmy trochę nasz program, dodajmy do niego kilka linijek i zapiszmy go jako kalkulator.py:

```
zmienna1 = input(”Podaj 1 liczbę: ”) zmienna2 = input(”Podaj 2 liczbę: ”) wynik = zmienna1 + zmienna2 print(”Suma:”, wynik)
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Pierwsze dwie linie nie są nowe. W trzeciej linii dodajemy do siebie dwie zmienne i wynik zapisujemy w zmiennej „wynik”. W czwartej, tak jak wcześniej, wypisujemy tekst oraz wynik obliczenia.

Czy tak napisany kalkulator działa poprawnie? Jeśli się go uruchomi kilka razy i poda różne dane szybko można zauważyć, że wyniki nie są prawidłowe. Wynika to z tego, że funkcja „input” zwraca dane typu napisowego, a dodawanie danych tego typu to, jak widzieliśmy na pierwszej lekcji, po prostu sklejanie ich ze sobą. Aby rozwiązać ten problem i poprawić nasz kalkulator musimy zmienić typ naszych zmiennych na typ liczbowy. Robi się to przy użyciu funkcji „int”, która jako parametr przyjmuje zmienną do przekonwertowania. Zatem nasz program po poprawkach może wyglądać np. tak:

```
zmienna1 = input(”Podaj 1 liczbę: ”)

zmienna2 = input(”Podaj 2 liczbę: ”)

wynik = int(zmienna1) + int(zmienna2)

print(”Suma:”, wynik)
```

Pogrubioną czcionką zapisane są zmiany wprowadzone w naszym programie. W ten sposób zmienne, które wczytaliśmy z klawiatury przed dodaniem, zostają zamienione na cyfry.

Czy w tej chwili program działa poprawnie? Na pierwszy rzut oka wszystko jest jak trzeba. Jednak bardziej dociekliwi użytkownicy zauważą, że jeśli wpiszemy liczby z ułamkami (2.5, 3.33, itp.) to nasz kalkulator nie doda ich poprawnie i obetnie część dziesiętną. Wynika to z tego, że tak jak mówiłem w pierwszej lekcji „int” to liczny całkowite. Jeśli chcemy, aby nasz kalkulator poprawnie zachowywał się również dla liczb ułamkowych, należy zamiast funkcji „**int**” użyć funkcji „

float

”.

Pobawcie się i zmodyfikujcie kalkulator, aby odejmował, mnożył, dzielił i potęgował liczby (potęgowanie realizuje operator „**”).

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.