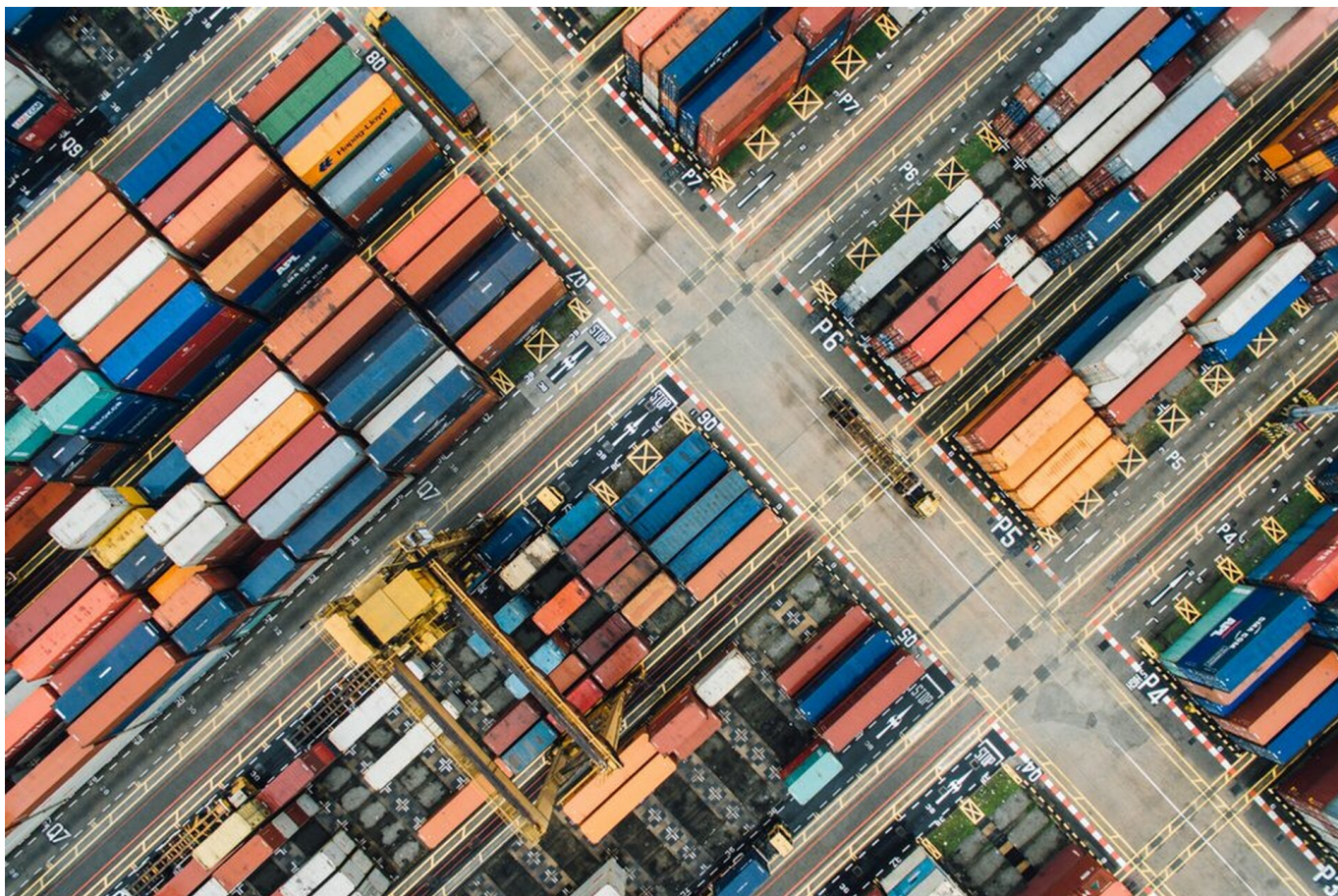




Podstawowe typy zmiennych w języku Python

- [Wprowadzenie](#)
- [Film samouczek](#)
- [Przeczytaj](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Podstawowe typy zmiennych w języku Python

Źródło: Chuttersnap, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Wiesz już, że programy wykorzystują pojemniki na dane – zmienne. Dowiedzieliśmy się tego z e-materiału [Podstawowe typy zmiennych](#).

Jaka jest ich specyfika w danym języku programowania?

W tym e-materiale zapoznamy się z podstawowymi typami zmiennych w języku Python.

Charakterystykę podstawowych typów zmiennych w innych językach programowania znajdziesz w e-materiałach:

- [Podstawowe typy zmiennych w języku C++](#),
- [Podstawowe typy zmiennych w języku Java](#).

Więcej zadań? Sięgnij do [Podstawowe typy zmiennych – ćwiczenia](#).

Twoje cele

- Przeanalizujesz różnice między typami zmiennych w języku Python.
- Dobierzesz do realizowanego algorytmu zmienne właściwego typu.
- Rozwiążesz proste problemy z zastosowaniem zmiennych w języku Python.

Film samouczek

Polecenie 1

Napisz program, który utworzy cztery zmienne oraz nada im przykładowe wartości.

Specyfikacja:

Dane:

- *imie* – zmienna typu `string`; informuje o imieniu użytkownika,
- *wiek* – zmienna typu `int`; przechowuje dane o wieku użytkownika,
- *srednia* – zmienna typu `float`; informuje o średniej ocen użytkownika,
- *czy_pelnoletni* – zmienna typu `bool`; zawiera informację, czy użytkownik ma więcej niż 18 lat.

Wynik:

Zmienne stworzone na podstawie specyfikacji, zainicjalizowane wybraną wartością początkową.

```
1 # Tutaj dodaj własny kod. Do wypisania zdania użyj funkcji  
2 # print()
```

Polecenie 2

Zapoznaj się z filmem i porównaj z nim swoje rozwiązanie.



Podstawowe typy zmiennych

Typy zmiennych w języku Python



Film dostępny pod adresem </preview/resource/R1HjcOO9Yinfo>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału podstawowe typy danych.

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w filmie samouczku.

Przeczytaj

Ważne!

Przedstawiony materiał stanowi rozszerzenie informacji, jakie zawarto w filmie. Zapoznaj się z nim, jeśli chcesz dowiedzieć się więcej na temat typów zmiennych w języku Python.

Wiesz już, że do przechowywania danych w programie wykorzystywane są **zmienne**. Różne języki programowania dysponują różnymi typami **danych**, jednak większość z nich jest bardzo podobna. Python oddaje nam do dyspozycji różne typy danych wbudowanych, m.in.:

Nazwa	Opis	Przykład
str	łańcuch znaków	'Adam Jurkiewicz'
int	liczba całkowita	2020
float	liczba zmiennoprzecinkowa	3.14151617
bool	wartość logiczna	True

Przykład 1

Oto przykład (w środowisku IDLE) przypisania wartości czterem zmiennym typów wbudowanych:

```
1 Python 3.6.7 (default, Oct 22 2018, 11:32:17)
2 [GCC 8.2.0] on linux
3 Type "help", "copyright", "credits" or "license()" for more inf
4 >>> typ_string = 'to jest napis - lubię Pythona'
5 >>> typ_int = 136
6 >>> typ_float = 3.14
7 >>> typ_boolean = True
```

Dla zainteresowanych

W języku Python minimalne i maksymalne wartości dla liczb float są różne w zależności od architektury procesora i implementacji języka Python. W przypadku procesorów 64-bitowych i najpopularniejszej implementacji CPython wynoszą odpowiednio $2.2250738585072014e-308$ do $1.7976931348623157e+308$, a do ich zapisu używane jest 8 bajtów.

Z kolei zmienne typu int w języku Python są liczbami o nieograniczonym zakresie, a w zasadzie ograniczonym wyłącznie rozmiarem dostępnej pamięci. Oznacza to, że możemy wykonywać operacje na dowolnie dużych liczbach całkowitych. Dzieje się tak

dzięki implementacji języka Python, która programowo pokonuje ograniczenia sprzętowe komputera. Warto jednak wspomnieć, że wewnętrznie dla procesorów 64-bitowych liczby całkowite są zapisywane na 8 bajtach (a więc największą wartością w tym zapisie jest $2^{63} - 1$, czyli 9223372036854775807).

Ważne!

W obiektach typu `float` część całkowitą od ułamkowej oddziela się za pomocą kropki, a nie przecinka.

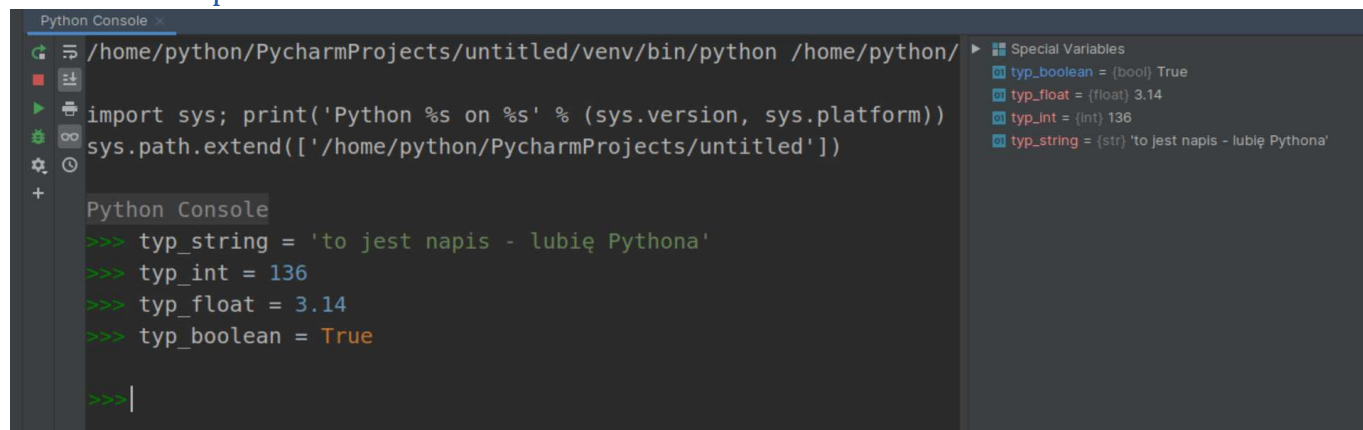
Ważne!

Wartości `True` oraz `False` należy zapisywać, zwracając uwagę na wielkość liter.

W języku Python rozróżniane są wielkie i małe litery.

Zapamiętaj: `true` nie oznacza w języku Python tego samego co `True`.

Przjrzyjmy się oknu Python Console programu PyCharm z wprowadzonymi danymi oraz modułowi [inspektor obiektów](#).



The screenshot shows the Python Console window in PyCharm. The console output displays the following code and its execution results:

```
>>> typ_string = 'to jest napis - lubię Pythona'
>>> typ_int = 136
>>> typ_float = 3.14
>>> typ_boolean = True
>>>
```

On the right side, the 'Special Variables' window is visible, showing the following variables and their values:

- `typ_boolean = {bool} True`
- `typ_float = {float} 3.14`
- `typ_int = {int} 136`
- `typ_string = {str} 'to jest napis - lubię Pythona'`

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Dane umieszczamy w zmiennych. Możemy wykonywać na nich wiele operacji za pomocą operatorów arytmetycznych oraz innych (więcej o tym dowiemy się w kolejnych materiałach). Teraz jednak spróbujemy wykonać proste działanie – dodawanie.

Polecenie 1

Zdefiniujmy dwie zmienne (dwa obiekty) typu `str`, dodajmy je do siebie i wyświetlmy wynik.

Polecenie 2

Zdefiniujmy dwie zmienne (dwa obiekty) typu `int`, dodajmy je do siebie i wyświetlmy wynik.

Polecenie 3

Zdefiniujmy dwie zmienne (dwa obiekty) typu `float`, dodajmy je do siebie i wyświetlmy wynik.

Polecenie 4

Zdefiniujmy zmienną (obiekt) typu `bool` i wyświetlmy ją.

Przykład 2

Zmienne typu `string` umożliwiają odwołanie się do pojedynczych znaków w przechowywanych napisach, używając [wyrażenia indeksującego](#). Oto przykład w środowisku IDLE:

```
1 Python 3.6.7 (default, Oct 22 2018, 11:32:17)
2 [GCC 8.2.0] on linux
3 Type "help", "copyright", "credits" or "license()" for more inf
4 >>> napis = 'Adam uczy się pilnie Pythona'
5 >>> napis[0] # pierwszy element sekwencji ma indeks [0]
6 'A'
7 >>> napis[1] # drugi element sekwencji
8 'd'
9 >>> napis[10] # jedenasty element
10 's'
11 >>> napis[10:13] # wycinek (ang. *slice*) - weź elementy od jed
12 'się'
```

Okno Python Console programu PyCharm z wprowadzonymi danymi oraz moduł inspektor obiektów prezentuje się następująco:

```
Python Console
/home/python/PycharmProjects/untitled/venv/bin/python /home/python/pycharm/h
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/python/PycharmProjects/untitled'])

Python Console
>>> napis = 'Adam uczy się pilnie Pythona'
>>> napis[0]
'A'
>>> napis[1]
'd'
>>> napis[10]
's'
>>> napis[10:13] # wycinek (ang. slice) - przydatna konstrukcja
'się'
>>>|
```

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ważne!

Obiekt typu `string` jest uporządkowanym ciągiem znaków, czyli sekwencją. Jest również [niezmienny](#).

Przykład 3

W przykładzie chcemy zmienić wartość czwartego znaku ciągu na „X”. Program zwróci jednak błąd, ponieważ – jak wspomnieliśmy – nie możemy zmienić części sekwencji znaków obiektu typu `string`; musimy zmienić cały obiekt (`TypeError: 'str' object does not support item assignment`).

Drugi błąd (`NameError: name 'obiekt_str' is not defined`) wynika z tego, że nie została zadeklarowana zmienna `obiekt_str`.

```
1 Python 3.6.7 (default, Oct 22 2018, 11:32:17)
2 [GCC 8.2.0] on linux
3 Type "help", "copyright", "credits" or "license()" for more inf
4 >>> zmienna_str = "Napis"
5 >>> zmienna_str[3] = "X"
6 Traceback (most recent call last):
7   File "<pyshell#4>", line 1, in <module>
8     zmienna_str[3] = "X"
9   TypeError: 'str' object does not support item assignment
10 >>> obiekt_str[4] = "X"
11 Traceback (most recent call last):
12   File "<pyshell#5>", line 1, in <module>
13     obiekt_str[4] = "X"
14   NameError: name 'obiekt_str' is not defined
15 >>>
```

Złożone typy danych

Omówione dotychczas zmienne (liczby, ciągi znaków i wartości logiczne) są prostymi (podstawowymi) typami danych. Język Python obsługuje również typy złożone. Są nimi np.:

Nazwa	Opis	Przykład
list	(lista) uporządkowany zbiór obiektów tego samego lub różnego typu bez ograniczeń dotyczących rozmiaru; elementy listy są zmienialne	[1, 'Python', 4.67]
tuple	(tupla lub krotka) – lista, której nie da się zmodyfikować (lista niezmiennalna)	(1, 'Python', 4.67)
dict	(słownik lub odwzorowanie) – zbiór obiektów różnego typu, skojarzonych z odpowiadającymi im kluczami; do obiektów odwołujemy się przez klucze, a nie przez pozycję w zbiorze (odwzorowania łączą klucze z wartościami, a wewnątrz słownika obiekty nie zachowują określonej kolejności); słownik jest zmienialny, ale klucze słownika muszą być niezmiennalne i niepowtarzalne	{1: 'Wartość 1'}
set	(zbiór) – kolekcja unikatowych, nieuporządkowanych i niezmiennalnych obiektów, obsługująca działania teorii zbiorów	{1, 4, 7, 8}

Polecenie 5

Innym typem złożonym są liczby zespolone. Poszukaj informacji na ich temat.

Przykład 4

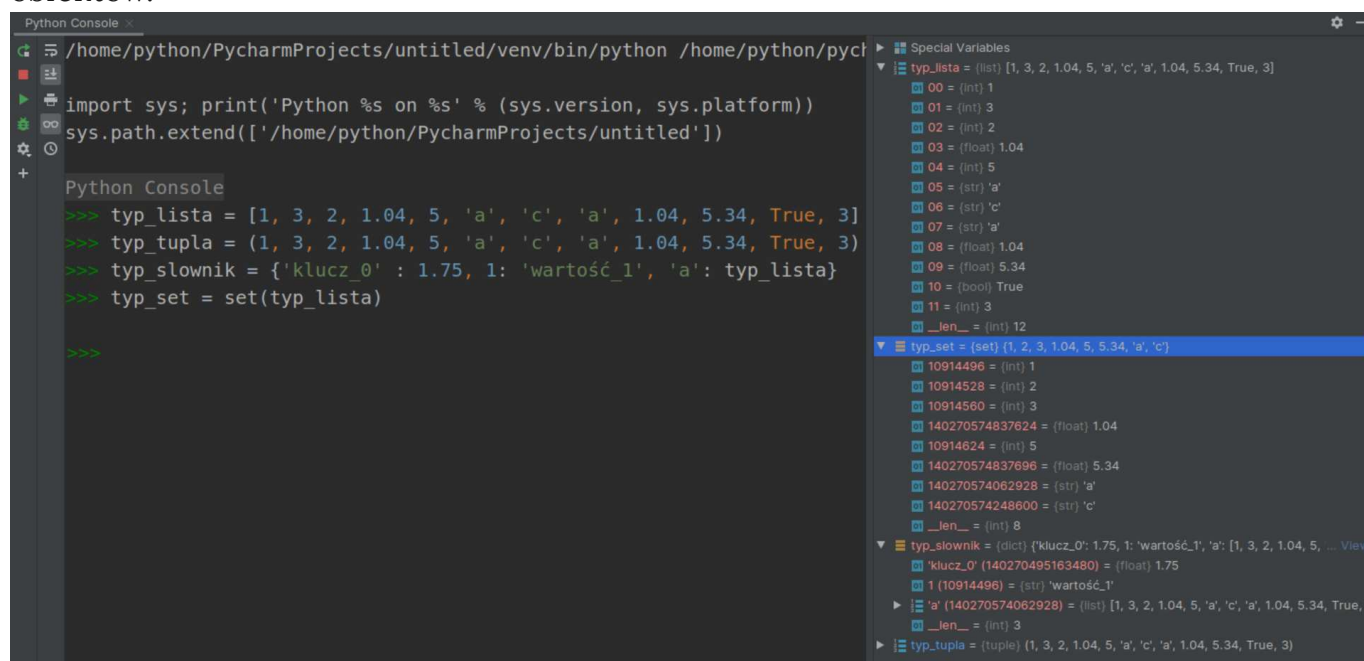
Oto przykłady zmiennych opisanych typów:

```
1 typ_lista = [1, 3, 2, 1.04, 5, 'a', 'c', 'a', 1.04, 5.34, True,
2 typ_tupla = (1, 3, 2, 1.04, 5, 'a', 'c', 'a', 1.04, 5.34, True,
3 typ_sownik = {'klucz_03' : 1.75, 1: 'wartość_1', 'a': typ_list
4 typ_set = set(typ_lista)
5
6 print(typ_set)
7 print(typ_lista)
8 print(typ_tupla)
```

Zauważmy:

- elementy zbioru (zestawu) przy każdym jego wywołaniu zostaną wyświetlone w losowej kolejności – skopiuj kod i sprawdź;
- elementy zestawu nie powtarzają się;
- w zbiorze nie pojawia się wartość `True` – dlaczego? W zbiorze jest już wartość jeden; program interpretuje wartość typu `bool` `True` jako jedynkę (analogicznie jest z wartością `False` – interpretuje ją jako 0), więc nie wyświetli jej po raz kolejny; podmień w liście wartość `True` na `False`, uruchom program i sprawdź, co się stanie.

Okno Python Console programu PyCharm z wprowadzonymi danymi oraz inspektor obiektów.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Przykłady zastosowania metod

Na zmiennych można wykonywać różne operacje, np. dodawania, mnożenia. Więcej informacji na ten temat znajdziesz w e-materiale [Operatory w języku Python](#). W języku Python mamy możliwość wykorzystania **metod**, czyli funkcji specyficznych dla obiektów konkretnego typu.

Python jest językiem programowania **typowanym dynamicznie**. Typy obiektów są **silne**; oznacza to, że na obiekcie można wykonać metody zdefiniowane dla konkretnego typu.

Ważne!

W języku Python nazwę metody oddziela się od nazwy obiektu znakiem kropki. Oto polecenie wywołania funkcji `metoda()` dla obiektu o nazwie `obiekt`:

```
1 obiekt.metoda()
```

Przykład 5

Przeanalizujmy przykład – chcemy poznać liczbę wystąpień litery „a” w zmiennej tekstowej:

```
1 Python 3.6.7 (default, Oct 22 2018, 11:32:17)
2 [GCC 8.2.0] on linux
3 Type "help", "copyright", "credits" or "license()" for more inf
4 >>> zmienna = "To jest długi tekst, a Python to świetny język p
5 >>> print(zmienna.count("a"))
6 4
7 >>>
```

Wynik wskazuje, że w podanym ciągu znaków litera „a” występuje cztery razy.

Dokumentacja w języku Python

Informacje na temat obiektów oraz odpowiadających im metod można otrzymać, korzystając z następujących funkcji:

- `dir(obiekt_do_sprawdzenia)` – wyświetla wszystkie metody przypisane do obiektu;
- `help(obiekt_do_sprawdzenia.nazwa_metody)` – wyświetla opis metody (tzw. *docstring*);
- `help(nazwa_funkcji)` – wyświetla opis funkcji (tzw. *docstring*).

Okno Python Console programu PyCharm z przykładowymi wywołaniami:

```
Python Console
/home/python/PycharmProjects/untitled/venv/bin/python /home/python/pycharm/helpers/pydev/p
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/python/PycharmProjects/untitled'])

Python Console
>>> napis = 'Obiekt tekstowy - string'
>>> type(napis)
<class 'str'>
>>> dir(napis)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '
>>> help(napis.center)
Help on built-in function center:

center(...) method of builtins.str instance
    S.center(width[, fillchar]) -> str

    Return S centered in a string of length width. Padding is
    done using the specified fill character (default is a space)

>>> napis.center(40, '*')
'*****Obiekt tekstowy - string*****'
>>>
```

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 6

Korzystając z funkcji `dir()` oraz `help()`, znajdź metody, które potrafią:

- sprawdzić, czy zmienna `str` zawiera tylko cyfry 0 – 9 typu `int`;
- zamienić podany ciąg znaków w zmiennej `str` na inny;
- podać liczbę wystąpień zadanego ciągu znaków w zmiennej `str`.

Ważne!

Przedstawione polecenia są jednym ze sposobów uzyskania pomocy w języku Python. W wierszu polecenia można również wpisać komendę: `pydoc -p 1234`, a następnie w przeglądarce wybrać adres `http://localhost:1234`. Dostępna będzie wówczas dokumentacja w wersji HTML.

Słownik

inspektor obiektów

w środowisku PyCharm moduł umożliwiający podgląd w czasie rzeczywistym obiektów, ich wartości oraz typu

niezmienialna

(ang. *immutable*) sekwencja, która jest niemodyfikowalna: nie można zmienić żadnego jej elementu; aby dokonać modyfikacji, trzeba zmienić cały obiekt

typowanie dynamiczne

określanie typu zmiennej w momencie przypisywania jej wartości; w języku Python nie deklaruje się typu zmiennej z góry (tak jak w języku C++); typ zmiennej może wielokrotnie zmieniać się w czasie działania programu

wyrażenie indeksujące

(ang. *slice*) zawarte w nawiasach kwadratowych wyrażenie, które pozwala określić wybraną część sekwencji; nazywane także **wycinkiem**; wycinek `[3:10]` oznacza: weź znaki od czwartego do dziesiątego – pamiętajmy, że w języku Python indeksujemy, licząc od 0, a więc pierwszy element sekwencji posiada indeks `[0]`

zmienialna

(ang. *mutable*) sekwencja, która jest modyfikowalna; oznacza to, że można zmienić dowolny jej element bez konieczności modyfikowania całego obiektu

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Połącz odpowiednie typy i przykładowe obiekty.

[2, 3, 56]	krotka - tuple
{1:2, 3:4}	łańcuch znaków - string
False	liczba zmiennoprzecinkowa - float
3.14	słownik - dict
(3, 4, 6)	zbiór - set
'napis'	liczba całkowita - int
12	logiczny fałsz
{1, 2, 3, 4}	lista - list

Ćwiczenie 2



Zaznacz wszystkie prawdziwe stwierdzenia.

☐

W obiektach typu `string` można odwołać się do ich poszczególnych elementów, używając wycinków (ang. *slice*).

☐

W języku Python wszystkie zmienne są obiektami.

☐

W języku Python tylko zmienne typu `tuple` są obiektami.

☐

`String` to obiekt zmienny (ang. *mutable*).

☐

`String` to obiekt niezmienny (ang. *immutable*).

Ćwiczenie 3



Uzupełnij zdania odpowiednimi nazwami funkcji.

W języku Python istnieje kilka funkcji, które służą do przedstawienia informacji na temat różnych obiektów. Funkcja `type()` pokazuje typ obiektu. Funkcja przedstawia jego metody, a funkcja pozwala zapoznać się z opisem konkretnej metody.

Ćwiczenie 4



Przeanalizuj kod i wskaż, jakiego typu będzie obiekt.

```
obiekt = 3 + 1.0
```

- ☐ Python zgłosi błąd: `TypeError: unsupported operand type(s) for +: 'int' and 'float'`
- ☐ typ float
- ☐ typ int

Ćwiczenie 5



Wpisz odpowiednie typy danych zgodnie z konwencją w języku Python.

1.0	"Python"	1	[1, 2, 3]	cTrue	("Python", "Linux")	{1: "Python"}
	str				tuple	

Ćwiczenie 6



Przeanalizuj kod i wskaż błędną odpowiedź.

```
wynik = int("kr", base=30)
```

- ☐ Otrzymamy wynik typu int.
- ☐ Wynikiem będzie liczba "627".
- ☐ Python zgłosi błąd: `ValueError: invalid literal for int() with base 30: 'kr'`.

Ćwiczenie 7



Uzupełnij kolumny odpowiednimi funkcjami lub danymi.

funkcja sprawdzająca typ	typ tekstowy	liczba 1 zapisana jako "float"
	string	

Ćwiczenie 8



Wskaż wszystkie operacje, które mogą zostać przeprowadzone w języku Python.

- ☐ `"a" * 12`
- ☐ `"Ala ma kota"[4].upper().center(5, "*")`
- ☐ `"Ala ma kota"[4].upper.center(5, "*")`
- ☐ `"a" / 12`

Dla nauczyciela

Autor: Zespół autorski [Contentplus.pl](https://contentplus.pl) sp. z o.o.

Przedmiot: Informatyka

Temat: Podstawowe typy zmiennych w języku Python

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

- 1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).
- 4) porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;
- 5) sprawdza poprawność działania algorytmów dla przykładowych danych.

- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz różnice między typami zmiennych w języku Python.
- Dobierzesz do realizowanego algorytmu zmienne właściwego typu.
- Rozwiążesz proste problemy z zastosowaniem zmiennych w języku Python.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał „Podstawowe typy zmiennych w języku Python”. Nauczyciel prosi o zapoznanie się z treściami w sekcji „Przeczytaj”.
2. Uczniowie szukają informacji na temat ograniczenia rozmiaru poszczególnych typów zmiennych w języku Python.

Faza wstępna:

1. Przedstawienie tematu zajęć oraz wspólne z uczniami ustalenie kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. Nauczyciel udziela wyjaśnień na temat ewentualnych wątpliwości po lekturze e-materiału.
2. Uczniowie w parach przygotowują programy rozwiązujące proste problemy np. przeliczanie jednostek, wyliczanie stref czasowych. Analizują, jakie typy danych należy zastosować.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-8 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązania ćwiczeń z programowania w języku Python.

Praca domowa:

1. Uczniowie rozwiązują zadanie: „Dana jest zmienna typu string o wartości : txt = „AlA Ma KoTa, A kOt Ma AlE – PyThOn JeSt CiEkAwYm JęZyKiEm”:
 - znajdź metodę tworzącą tekst składający się z wielkich liter (ang. *upper case*),
 - znajdź metodę tworzącą tekst składający się z małych liter (ang. *lower case*),
 - znajdź metodę zamieniającą litery 'A' na 'X'.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Nauczyciel może poprosić uczniów o to, żeby poszukali formularzy, jakie wypełniają w sieci i określili, jakie zmienne mogą być używane do ich przechowywania i dlaczego.