Charakterystyka języka Python

M@я3k Pų𝐝€£kØ

Programowanie w Pythonie

Linie kodu

- Linie kodu są zakończone ENTER-em.
  - Nie używamy znaku średnika **;** jak w C++, Pascalu, Javie itp.

```
print ('Hello World')
```

- Jeśli linia kodu jest tak długa, że trzeba ją podzielić na więcej, należy użyć backslasha ****.

```
print\
('Hello World')
```

Wcięcia

- Wcięcia są traktowane jako początek bloku kodu.
- Często dwukropek pojawia się jako początek nowego bloku
- Wcięcia jako wydzielanie bloków kodu to cecha unikatowa wśród powszechnie stosowanych języków programowania.
  - W językach programowania wywodzących strukturę blokową od Algolu bloki kodu zaznaczane są klamrami lub słowami kluczowymi (C i Perl używają **{ }**, Pascal używa **begin** i **end**).
- Jednakże we wszystkich tych językach programiści stosują wcięcia, by wyróżnić wizualnie bloki w otaczającym kodzie.

Wcięcia

- Wcięcia są traktowane jako początek bloku kodu.

```
a=5
if a>3:
   if 4<5:
      print 2
   else:
      print 3
else:
   print 4
   print 5
```

- Dwukropek często pojawia się jako początek nowego bloku

```
if a>3:
          if 4<5:
class:
```

Komentarze

- Komentarz służy do opisu, co robi dany element kodu programu.
  - Zazwyczaj to skomentowanie mniej oczywistych partii kodu, informacja do czego służy dana funkcja itp..
  - Czasowe wyłączenie wiersza kodu chwilowo niewykonywanego
- Komentarz jest ignorowany przez kompilator i nie wpływa na działanie programu. Służy tylko człowiekowi.

Komentarze

- **Jednoliniowy**
- Oznaczany znakiem **#** (kompilator ignoruje wszystko co jest dalej). Zazwyczaj po # zostawia się jedną spację, by oddzielić tekst komentarza

```
print ('Hello World')    # tekst na ekranie
```

- **blokowy**
- Oznaczany ‘’’ i ‘’’, pomiędzy które wstawiony jest wieloliniowy komentarz

```
print ('Hello World')
'''
Instrukcja wyświetla
tekst na ekranie
'''
```

Komentarze - przykład

```
1  '''
2
3  Prosty program
4  wypisuje tekst na ekranie
5
6  '''
7  print ('Hello World')   #tekst na ekranie
8
```

Nazwy obiektów

- Istotna wielkość liter (A ≠ a)
- Nie mogą zaczynać się od cyfry
- Nazwy mogą zawierać wielki i małe litery, cyfry, znak podkreślenia

- **Poprawne**

```
Marek, Marek1,
Marek_2, mArEk5,
Marek_1_1a,
```

- **Niepoprawne**

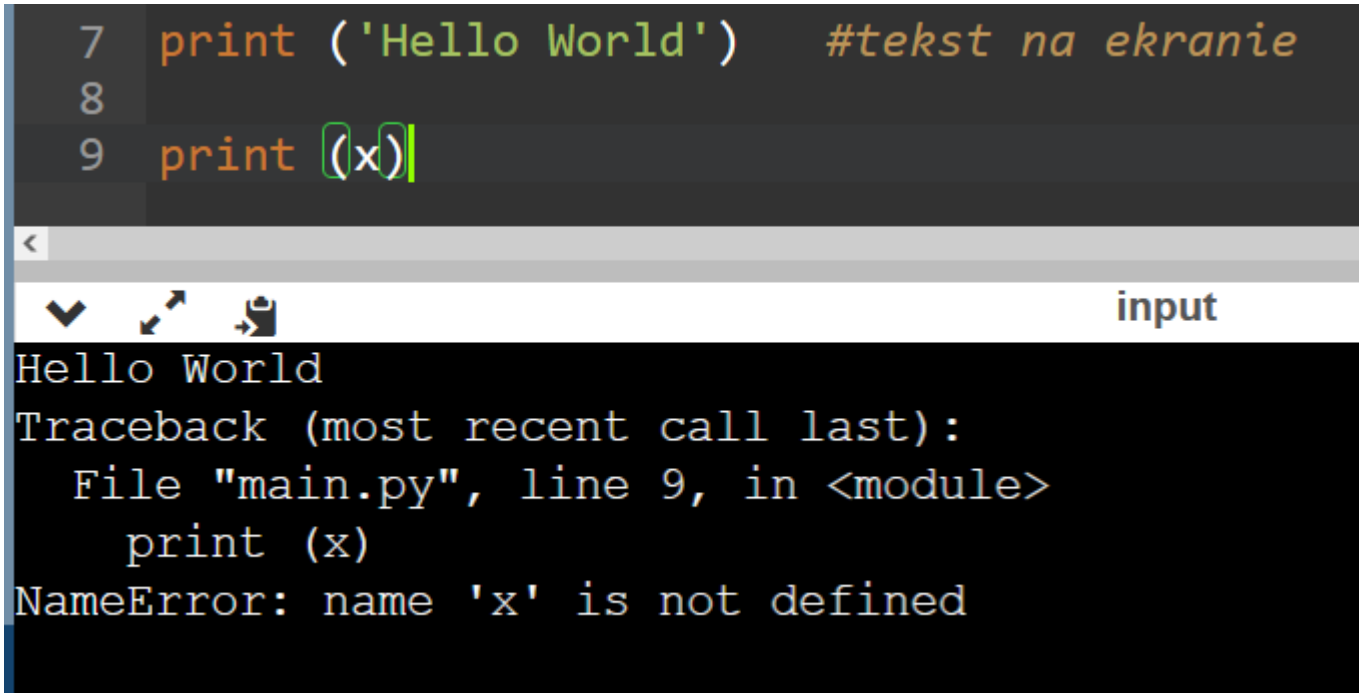
```
1marek,
marek^^&@,
#qW1_+==
```

Zastrzeżone słowa w języku python

- **and, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while**

Nieistniejące nazwy

- Przy próbie dostępu lub użycia nieistniejącej nazwy zostanie wyświetlony błąd



Typy i deklaracje danych

- W pythonie nie ma potrzeby deklarowania zmiennych. W momencie ich użycia są automatycznie deklarowanie.

```
x = 5
# deklaracja zmiennej x
```

- Nie trzeba też deklarować typu zmiennej. Zostaje on wymuszony w zależności od rodzaju danych

```
x = 5
# 5 to liczba całkowita → x to integer
```

Operatory

- Podstawienie = (`x = 11`)
- Porównanie == (`y == 0`)
- Działania na liczbach `+ - * / %`
- Działania na łańcuchach
  - `+` łączenie łańcuchów
  - `%` formatowanie łańcuchów
- Logiczne operatory (`and, or, not`)

Przyporządkowanie

- Operatorem przyporządkowania jest symbol =.

```
x = 5
```

- Możliwe jest połączenie go z innymi operatorami

```
y *= 3 + y
```

Równoważne

```
y = y * (3 + y)
```

```
x += 1
```

Równoważne

```
x = x + 1
```

- W pythonie nie ma odrębnego operatora inkrementacji / dekrementacji

```
x++ / x--
```

- są zabronione. Należy użyć `x = x + 1` / `x = x - 1`

Wielokrotne przyporządkowania

- **Wielokrotne przyporządkowania**

```
x, y = 2, 3
```

- Jest równoważne

```
x = 2
y = 3
```

- **Zamiana wartości zmiennych**

```
x, y = y, x
```

- Jest równoważne

```
z = x
x = y
y = z
```

Operatory

Operacje arytmetyczne				
x + y	x - y	x / y	x * y	x // y
x % y	-x	x ** y		
Operacje bitowe				
x|y	x^y	x&y	x<<n	x>>n
~x				
Operatory relacyjne				
<	<=	>	>=	==
!=	<>	is	is not	
Operacje logiczne				
x or y	x and y	not x		

Typy danych proste

null	Obiekt zwracany przez funkcje, które nie zwracają jawnie żadnej wartości. Nie obsługuje żadnych dodatkowych operacji.	Istnieje dokładnie jeden obiekt null pod nazwą wbudowaną None.
Integer	Liczba całkowita. Typ domyślny dla liczb	z = 4 / 2
Float	Liczba zmiennoprzecinkowa	pi = 3.141519
String	Napisy (łańcuch tekstowy)	s = ”Ala ma kota”
Bool	Typ logiczny	x = true
Complex	Liczba zespolona (część rzeczywista + urojona)	y = 2.2-3.3e-2j

Typy danych złożone

Lista	kontener danych przypominający tablicę z zachowaniem kolejności elementów. Modyfikowalna **x=[1, ‘x’, [4,(1,1-j)], {1:’b’}]**
Słownik	Zbiór par klucz-wartość. Kolejność elementów nie jest zachowana, a wartości są modyfikowalne. **x={(1,2,’a’):[3,’s’,8.2e-2], ‘x’:4j}**
Krotka	Niemodyfikowalna lista. Raz stworzona krotka nie może zmieniać swojej zawartości. **x=(2,3,’b’,(0,0),[1],{})**
Zbiory mutowalne (set)	Elementy w zbiorze nie mogą się powtarzać. **a=set([’x’,’y’,5,6])**
Zbiory niemutowalne (frozenset)	niezmienna wersja obiektu *zbiór* (set) **a=frozenset([7,’c’,set([‘0’,0]),11])**
Wektory i tablice (biblioteka array)	Macierze i tablice wielowymiarowe **>>>import array** **a = array.array(’f’,[1.0,3.14,6.66])**
Ułamki (biblioteka fractions)	Przechowuje licznik i mianownik **>>>from fractions import Fraction** **u = Fraction(3, 5) – Fraction(1, 5)**