Język Python

- Stworzony przez holendra Guido van Rossum w 1991 roku
- Open Source
- Interpretowany, interaktywny, zorientowany obiektowo
- Przenośny
- Bogata biblioteka funkcji
- Dobra dokumentacja
- Łatwy do nauki, przejrzysty w zapisie
- Funkcje, moduły, klasy, pakiety
- Dynamiczne typowanie
- Automatyczne zarządzanie pamięcią
- Obsługa wyjątków
- Struktury wysokiego poziomu: zbiory, krotki, listy, słowniki

1

Język Python

Gdzie można używać:

- szybkie prototypowanie
- programowanie sieciowe (po stronie klienta i serwera)
- •programowanie ad hoc ("skryptowanie")
- w aplikacjach naukowych
- •język rozszerzeń
- przetwarzanie XML
- aplikacje bazodanowe
- aplikacje GUI
- edukacja

Kto używa języka Python?

- Google (various projects)
- NASA (several projects)
- NYSE (one of only three languages "on the floor")
- Industrial Light & Magic (everything)
- Yahoo! (Yahoo mail & groups)
- •i inni

Python w systemie Windows

```
7. Python Shell
File Edik Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:hddaarb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> 2**100
1267650600228229401496703205376
>>> print(3*"Tak! ")
Tak! Tak! Tak!
>>>
Ln:9Col:4
```

3

Python w systemach wbudowanych

Adafruit Metro M4

Mikrokontroler: ATSAMD51J19Częstotliwość zegara: 120 MHz

Pamięć RAM: 192 kB

Flash: 512 kBQSPI Flash: 2 MB

Interfejsy: UART, SPI, I2C,

Wbudowany NeoPixel

•10-bitowy równoległy kontroler do podłączenia kamery

•Porty I/O: 25, PWM: 16, wejścia analogowe: 8, wyjścia analogowe 2

•Wymiary: 72 x 54 mm, masa 20 g

•Cena 155.00 zł



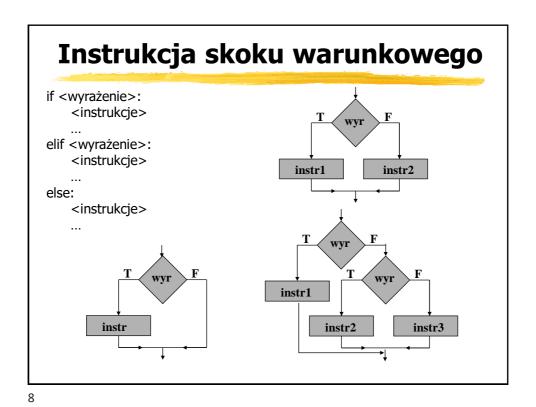
Operacje wejścia/wyjścia

```
>>> a = input("Jak masz na imie: ")
Jak masz na imie: Marek
>>> print(a)
Marek
>>> b = int(input("Podaj liczbę całkowitą: "))
Podaj liczbę całkowitą: 23
>>> print(b)
23
>>> c = float(input("Podaj liczbę rzeczywistą: "))
Podaj liczbę rzeczywistą: 0.1428
>>> print(c)
0.1428
```

6

Instrukcja przypisania

```
>>> a = 23
>>> a = "to jest napis"
>>> print(a)
to jest napis
>>> b, c = 6, 'kotek'
>>> print(b, c)
6 kotek
>>> c, b = b, c
>>> print(b, c)
kotek 6
>>> x = y = z = 0
```



Instrukcja pętli while

while <wyrażenie>:
 <instrukcja>
 ...

break – przerwanie pętli
continue – następna iteracja

q

Instrukcja pętli for

10

Typy danych w języku Python

Typy proste

- Logiczny (bool)
- Całkowity (int)
- Zmiennopozycyjny (float)
- Zespolony (complex)
- Napisowy (str)

Typy strukturalne

- Zbiór (set)
- Krotka (tuple)
- Lista (list)
- Słownik, tabela (dict)

Działania na typach prostych

- typ logiczny (bool) true false not or and
- typ całkowity (int)12 -21+ * // % **
- typ zmiennopozycyjny (float)12.3 2e-23+ * / **
- typ zespolony (complex)3+4j 3.0+4.0j 2j+ * / **

12

Funkcje wbudowane

abs(n) - wartość bezwzględna

chr(n) - znak o kodzie n ord(zn) - kod znaku

min(a,b,...) - najmniejsza z liczb max(a,b,...) - największa z liczb

round(x) - zaokrąglenie wartości len(s) - długość napisu, listy, krotki

Przykład:

x = abs(x)z = max(x,y)

Biblioteki funkcji

from math import sqrt
from math import * # niezalecane
import math # math.sqrt(x)

Funkcja	Znaczenie
ceil(<i>value</i>)	rounds up
cos (<i>value</i>)	cosine, in radians
floor(value)	rounds down
log(value)	logarithm, base e
log10 (<i>value</i>)	logarithm, base 10
max(value1, value2)	larger of two values
min(value1, value2)	smaller of two values
sin(<i>value</i>)	sine, in radians
sqrt(value)	square root

14

Kod ASCII

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	Р	`	р
1	SOH	DC1 XON	ļ	1	Α	Q	а	q
2	STX	DC2	"	2	В	R	b	r
3	ETX	DC3 XOFF	#	3	С	S	С	s
4	EOT	DC4	\$	4	D	Т	d	t
5	ENQ	NAK	%	5	E	U	е	u
6	ACK	SYN	&	6	F	V	f	٧
7	BEL	ETB	1	7	G	W	g	W
8	BS	CAN	(8	Н	Х	h	×
9	HT	EM)	9	- 1	Υ	i	У
Α	LF	SUB	*	:	J	Ζ	j	Z
В	VT	ESC	+	;	K	[k	{
С	FF	FS		<	L	١	- 1	-
D	CR	GS	-	=	M]	m	}
E	so	RS		>	N	Α	n	~
F	SI	US	1	?	0	_	0	del

Unicode

UTF-8

Znaki ASCII kodujemy za pomocą 1 bajta.

Alfabety: łaciński, grecki, armeński, hebrajski, arabski, koptyjski i cyrylica

kodujemy za pomocą 2 bajtów.

Kolejne znaki (m.in. alfabety chiński i japoński) kodowanych jest na 3 i 4 bajtach.

00000000 – 0000007F: 0*xxxxxx*

00000080 - 000007FF: 110*xxxxx* 10*xxxxxx*

00000800 - 0000FFFF: 1110*xxxx* 10*xxxxxx* 10*xxxxxx*

00010000 - 001FFFFF: 11110*xxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*

UCS-2

Wszystkie znaki zapisywane są za pomocą 2 bajtów. Kodowanie to pozwala na zapisanie tylko 65536 początkowych znaków Unikodu.

UCS-4

Wszystkie znaki zapisywane są za pomocą 4 bajtów.

16

Kodowanie "polskich" znaków

Znak	ISO 8859-2	Unicode	UTF-8
ą	161	261	196 133
ć	198	263	196 135
ę	202	281	196 153
ł	163	322	197 130
ń	209	324	197 132
ó	211	211	195 179
Ś	166	347	197 155
ź	172	378	197 186
Ż	175	380	197 188
Ą Ć	177	260	196 132
Ć	230	262	196 134
Ę	234	280	196 152
Ł	179	321	197 129
Ń	241	323	197 131
Ó	243	243	195 147
Ś Ź	182	346	197 154
	188	377	197 185
Ż	191	379	197 187

Liczby całkowite

Java	C/C++	zakres	rozmiar
byte	char	-128127	1
short	short int	-3276832767	2
int	int	-2147483648214748647	4
long	long int	-2^632^63-1	8

W języku Python 3.x zakres typu int jest ograniczony jedynie dostępną pamięcią.

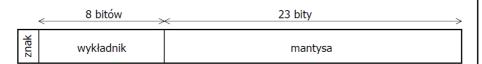
18

Liczby zmiennopozycyjne

Cel: Oddzielenie zakresu od dokładności

liczba = mantysa * 2 ^{cecha}

Przykład: liczba 4 bajtowa



Liczby zmiennopozycyjne

Java, C/C++

typ	zakres	dokładność	rozmiar
float	1.5E-45 3.4E38	7-8	4
double	5.0E-324 1.7E308	15-16	8

W języku Python typ float jest równoważny z typem double w języku C/C++.

20

Dokładność obliczeń

>>> 2.7+1.1

3.8000000000000003

Obliczenia iteracyjne

```
r = 3.0
p = 0.01
for i in range(0,100):
    q = p+r*p*(1-p)
    print(q)
    p = q;
# end for
```

iter	float (64bit)	wartość dokładna
1	0.039700000000000	0.03970000000000000000000000000000000000
2	0.154071730000000	0.1540717299999999999999999999999999999999
3	0.545072626044421	0.5450726260444212999999999999999999999999
4	1.288978001188801	1.28897800118880056414607744431892999999999
5	0.171519142109176	0.171519142109175610913273761193669501531848
 50	1.313996746606757	1.314489760648214812383780183964860775887564
51	0.076224636147602	0.074309050045856049321771955401870933940599
52	0.287467959122905	0.280670695427271580509655379893949005186949
53	0.901958353924756	0.886354663894201577758577930930197101949547
54	1.167246799055058	1.188544884955797172985886514749765353448385
55	0.581591926507394	0.516262709159421280638670823473078496363871
56	1.311620199093830	1.265469282031809506189821162818058704556637
57	0.085438156362505	0.257639616828927735003403861501470932678094
92	0.863051397010868	0.271467233262087610549464473055410611454272
93	1.217632446396239	0.864785556843432384355119764618126915551368
94	0.422643462034285	1.215580049398713266568656814984364161700898
95	1.154691360136162	0.429415628106318510122842974986646350734709
96	0.618829029025349	1.164469167439441909702911183666688004618360
97	1.326468014608027	0.589911344006446644012487937540734261222550
98	0.027319877097624	1.315659194663309877447649200395210188456166
99	0.107040381336610	0.069759429146912168060579275901754241874755
100	0.393788595636378	0.264438582722939495435219262057398282867365

typ float (64bit	$W = \frac{\left(\sqrt{1+u} - \sqrt{1-u}\right)}{u}$	$W = \frac{2}{\left(\sqrt{1+u} + \sqrt{1-u}\right)}$
0.100000000	1.001255501196379	1.001255501196378
0.010000000	1.000012500546898	1.000012500546907
0.001000000	1.000000125000011	1.000000125000055
0.000100000	1.00000001248891	1.00000001250000
0.000010000	1.00000000006551	1.00000000012500
0.00001000	1.00000000028755	1.00000000000125
0.00000100	1.00000000583867	1.00000000000001
0.00000010	1.00000005024759	1.00000000000000
0.00000001	1.000000082740371	1.00000000000000
wartości dokła	dne	
0.100000000	1.00125550119637747391	785450350122038359405315883
0.010000000	1.00001250054690722874	1466702747516135163757833258
0.001000000	1.00000012500005468753	3222658432008437730121685070
0.000100000	1.0000000125000000546	8875003222656271820068519554
0.000010000	1.0000000001250000000	0054687500003222656250218200
0.000001000	1.0000000000012500000	0000005468750000003222656250
0.00000100	1.00000000000000125000	000000000546875000000003222
0.00000010	1.00000000000000001250	0000000000000054687500000000
		2500000000000000005468750000

Operacje na napisach

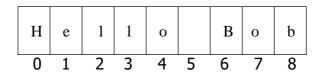
```
• "hello"+"world"
                       "helloworld"
                                        # concatenation
• "hello"*3
                       "hellohello" # repetition
• "hello"[0]
                       "h"
                                       # indexing
• "hello"[-1]
                       "o"
                                       # (from end)
• "hello"[1:4]
                       "ell"
                                       # slicing
len("hello")
                       5
                                       # size
• "hello" < "jello"
                       1
                                       # comparison
• "e" in "hello"
                        1
                                       # search
```

• "escapes: \n etc, \033 etc, \if etc"

• 'single quotes' """triple quotes""" r"raw strings"

25

Operacje na napisach



>>> greet[0:3]

'Hel'

>>> greet[5:9]

' Bob'

>>> greet[:5]

'Hello'

>>> greet[5:]

' Bob'

>>> greet[:]

'Hello Bob'

Funkcje przetwarzające napisy

- s.lower() małe litery
- s.upper() duże litery
- s.capitalize() wielka pierwsza litera
- s.title() wielkie pierwsze litery
- s.center(szer) formatowanie
- s.ljust(szer), s.rjust(szer) j.w.
- s.count(sub) liczba wystąpień sub
- s.find(sub) pierwsza pozycja na której występuje sub
- s.rfind(sub) ostatnia pozycja na której występuje sub
- s.split() lista napisów składowych

27

Typy danych w języku Python

Typy proste

- Logiczny (bool)
- Całkowity (int)
- Zmiennopozycyjny (float)
- Zespolony (complex)
- Napisowy (str)

Typy strukturalne

- Zbiór (set)
- Krotka (tuple)
- Lista (list)
- Słownik, tabela (dict)