

# 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:

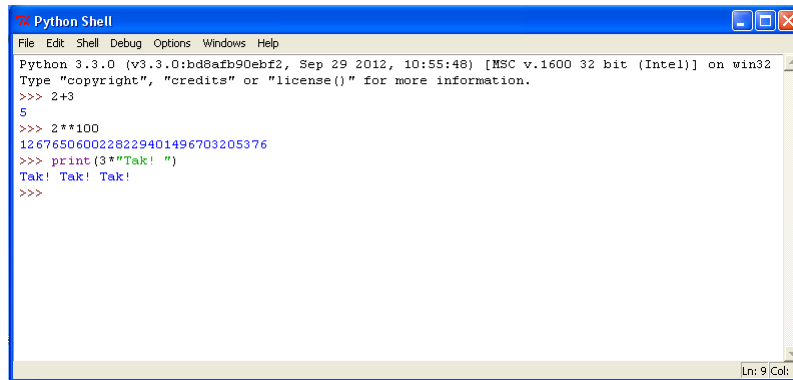
- szybkie prototypowanie
- programowanie sieciowe (po stronie klienta i serwera)
- programowanie ad hoc („skryptowanie”)
- w aplikacjach naukowych
- język rozszerzeń
- przetwarzanie XML
- aplikacje bazodanowe
- aplikacje GUI
- edukacja

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

2

# Python w systemie Windows



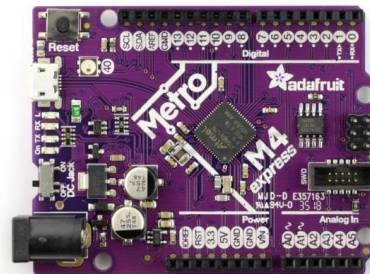
```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8af90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> 2**100
1267650600228229401496703205376
>>> print(3**"Tak! ")
Tak! Tak! Tak!
>>>
```

3

# Python w systemach wbudowanych

## Adafruit Metro M4

- Mikrokontroler: ATSAM51J19
- Częstotliwość zegara: 120 MHz
- Pamięć RAM: 192 kB
- Flash: 512 kB
- QSPI 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



5

## Operacje wejścia/wyjścia

```
>>> a = input("Jak masz na imię: ")
Jak masz na imię: 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

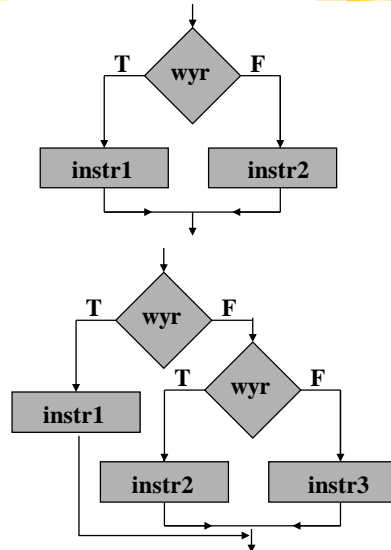
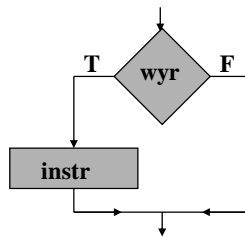
## Instrukcja przypisania

```
>>> 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
```

7

## Instrukcja skoku warunkowego

if <wyrażenie>:  
    <instrukcje>  
...  
elif <wyrażenie>:  
    <instrukcje>  
...  
else:  
    <instrukcje>  
...

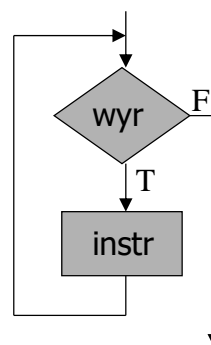


8

## Instrukcja pętli while

while <wyrażenie>:  
    <instrukcja>  
...

break – przerwanie pętli  
continue – następna iteracja



9

## Instrukcja pętli for

```
for <zmienna> in <sekwencja>:  
    <instrukcja>  
    ...
```

Przykład:

```
for x in range(1, 6):  
    print(x, "*", x, " = ", x*x)
```

```
1 * 1 = 1  
2 * 2 = 4  
3 * 3 = 9  
4 * 4 = 16  
5 * 5 = 25
```

10

## Typy danych w języku Python

Typy proste

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

Typy strukturalne

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

11

## 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)
```

13

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

# Kod ASCII

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

# Unicode

## UTF-8

Znaki ASCII kodujemy za pomocą 1 bajta.

Alfabet: łaciński, grecki, armeński, hebrajski, arabski, koptyjski i cyrylica kodujemy za pomocą 2 bajtów.

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

00000000 – 0000007F:	0xxxxxxx
00000080 – 000007FF:	110xxxxx 10xxxxxx
00000800 – 0000FFFF:	1110xxxx 10xxxxxx 10xxxxxx
00010000 – 001FFFFF:	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

## 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	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	182	346	197 154
Ź	188	377	197 185
Ż	191	379	197 187

17



# Liczby całkowite

Java	C/C++	zakres	rozmiar
byte	char	-128..127	1
short	short int	-32768..32767	2
int	int	-2147483648..214748647	4
long	long int	-2^63..2^63-1	8

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

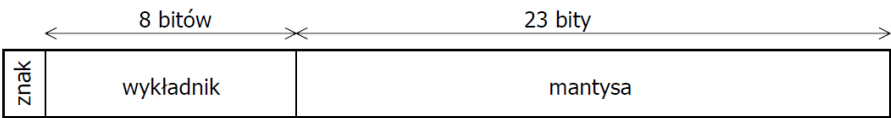
18

# Liczby zmiennopozycyjne

**Cel:** Oddzielenie zakresu od dokładności

$liczba = mantysa * 2^{cecha}$

**Przykład:** liczba 4 bajtowa



19

## 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
```

```
>>> 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1
0.9999999999999999
```

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
```

22



## Operacje na napisach

- "hello"+"world" "helloworld" # concatenation
- "hello"\*3 "hellohellohello" # 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

H	e	l	l	o		B	o	b
0	1	2	3	4	5	6	7	8

```
>>> greet[0:3]
'Hel'
>>> greet[5:9]
'Bob'
>>> greet[:5]
'Hello'
>>> greet[5:]
'Bob'
>>> greet[:]
'Hello Bob'
```

26

## 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)
- Zespólny (complex)
- Napisowy (str)

Typy strukturalne

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

28