Astronomiczny Kurs Studencki
cz. I - Wprowadzenie

Paweł Biernacki

1 marca 2012 r.

Plan prezentacji

Co i jak?

- twórca: Guido van Rossum
- multiparadygmat - obiektowość, imperatywność, funkcyjność
- nazwa pochodzi od Monty Python's Flying Circus
- filozofia: 'fun to use'
- wersja: - obecnie dwie szeroko używane: 2.7 i 3.2

Czemu python?

- prosty w użyciu
- dość intuicyjna składnia
- duże wsparcie w postaci zewnętrznych bibliotek
- brak kompilowania
- możliwość uruchamiania zewnętrznych programów/kodów
- pełna dokumentacja z przykładami oraz rozbudowane wsparcie
- stabilność i wieloplatformowość

Dlaczego nie python i co zamiast niego?

Czemu nie python?

- wolniejszy od języków kompilowanych takich jak C/C++ czy Fortran
- brak dobrego wsparcia dla użycia wielu rdzeni

Co zamiast pythona?

- C/C++, Java, Fortran, IDL - Twój wybór ;)

ipython

Czym jest ipython?

Jest to dynamiczna konsola pythona, która posiada:

- pamięć wykonywanych poleceń,
- pełny dostęp do pomocy (`help`),
- dopełnianie nazw (poprzez naciśnięcie *tab*),
- dostęp do poleceń systemowych (np. `!ls`),
- autoskładnia i autonawiasowanie

Porównanie

- C:

```c
#include <stdio.h>

int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

Porównanie

- C:

```c
#include <stdio.h>

int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

- Python:

```python
print('Hello, world!')
```

Program - hello.py

```
1 print('Hello, world!')
```

Powyższy program musi być wywołany w terminalu w następujący sposób:
`python hello.py`
Jeśli umieścimy tak zwany 'shebang' to nasz kod będzie wyglądał następująco:

```
1 #!/usr/bin/env python # <- to jest shebang
2
3 print('Hello, world!')
```

i wtedy możemy do wykonać w normalny dla systemów unixowych sposób, tj.
`./hello.py`

Zmienne

W odróżnieniu do C w pythonie nie trzeba deklarować zmiennych

- C:

```
1 int calkowita;
2 calkowita = 3;
3 // int calkowita = 3;
4
5 double zmiennoprzecinkowa;
6 zmiennoprzecinkowa = 3.14;
7 // double zmiennoprzecinkowa = 3.14;
```

- Python:

```
1 calkowita =  3
2 zmiennoprzecinkowa = 3.14
```

Typy zmiennych

W pythonie możemy używać następujących wbudowanych typów danych:

Typ	Przykład
str	'eggs' or "spam"
bytearray	bytearray(b'Some ASCII')
bytes	b'Some ASCII'
list	[4.0, 'string', True]
tuple	(4.0, 'string', True)
set, frozenset	{4.0, 'string', True} lub frozenset([4.0, 'string', True])
dict	'key1': 1.0, 3: False
int	42
float	3.1415927
complex	3+2.7j
bool	True lub False

Problemy ze zmiennymi

$22/7$ jest pewnym przybliżeniem liczby π.
Załóżmy sobie następującą sytuację:

```
1 pi = 22/7
2 print(pi)
```

Wynikiem będzie (!): 3. Dlaczego?

Problemy ze zmiennymi

$22/7$ jest pewnym przybliżeniem liczby π.
Załóżmy sobie następującą sytuację:

```
1 pi = 22/7
2 print(pi)
```

Wynikiem będzie (!): 3. Dlaczego?
Możemy temu zapobiec w następujący sposób:

```
1 pi = 22.0/7 # lub  pi = 22/7.0 lub  pi = 22.0/7.0
2 print(pi)
```

Takie wyrażenie da nam następujący wynik: 3.142857142857143

Trochę o liczbach...

W pythonie, liczby są definiowane jako obiekty o specjalnych właściwościach i operatorach.
Predefiniowane funkcje działające na liczbach to:

- abs(x)
- divmod(x,y)
- pow(x,y[,z])
- round(x,y)

Operatory

```
1 x = 10.0
2 y = 2.0
3 print x + y
4 print x - y
5 print x * y
6 print x / y
7 print x % y
8 print x**y
```

Także:

```
1 a = a + 1 # lub:
2 a += 1
```

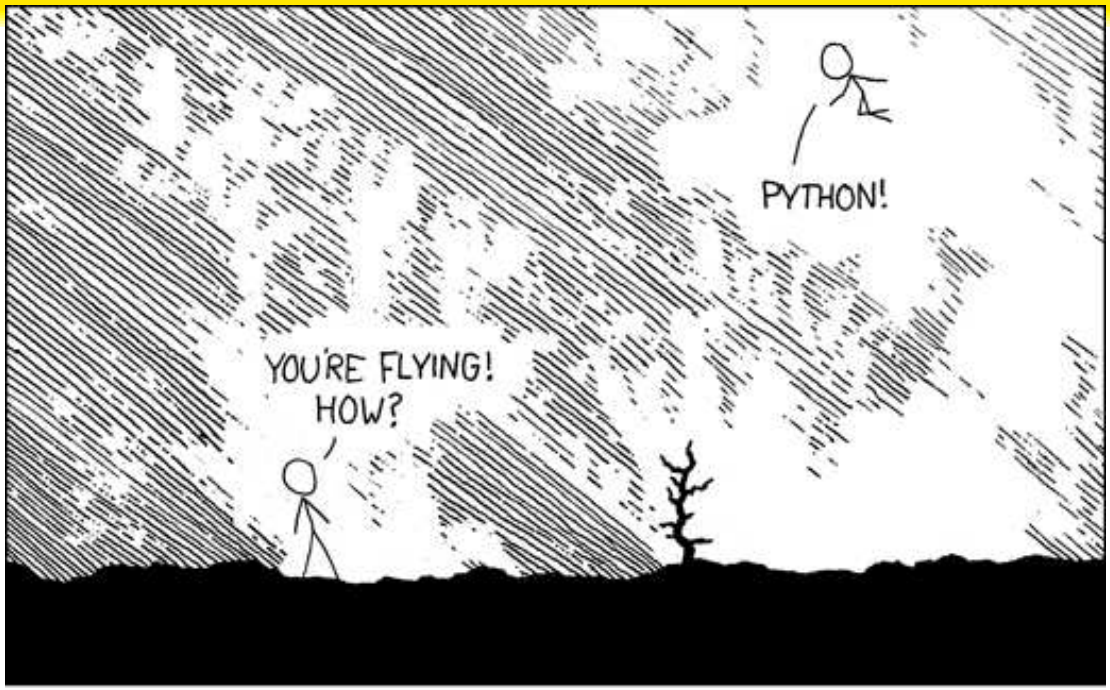
ale nie znane z C:

```
1 a++;
```

Dwa bardzo podstwowe źródła

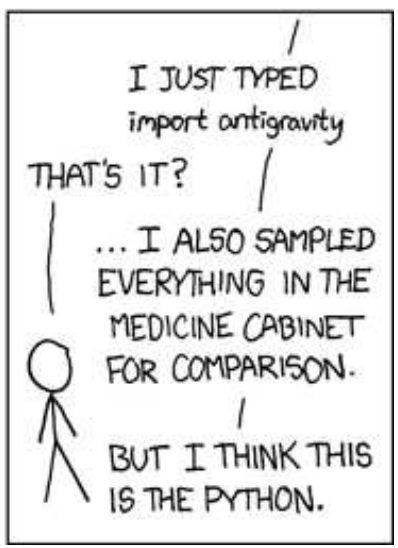
- `http://docs.python.org/tutorial/`
- `http://www.python.org/doc/`

A na koniec...


PYTHON!
YOU'RE FLYING!
HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.