

Po co programować?

Rozdział 1



Python dla wszystkich
www.py4e.pl



Mowa węży (ang. Parseltongue) – język węży i tych, którzy potrafią się z nimi porozumiewać. Osoba znająca mowę węży jest nazywana Wężoustą. Jest to umiejętność bardzo rzadka i może być przekazywana z pokolenia na pokolenie. Prawie wszyscy znani Wężouści są spadkobiercami Salazara Slytherina.



na podstawie:

<https://harrypotter.fandom.com/pl/wiki/Wężousty>

Python jako język

Python – język interpretera Pythona i tych, którzy potrafią się z nim porozumiewać. Osoba znająca język Python jest nazywana Pythonistą. Jest to umiejętność bardzo rzadka i może być przekazywana z pokolenia na pokolenie. Prawie wszyscy znani Pythoniści używają oprogramowania stworzonego przez Guido van Rossuma.



Początkujący uczeń: Błędy składniowe

- Musimy nauczyć się języka Python, żeby móc mu przekazywać instrukcje. Na początku będziemy popełniać wiele błędów i mówić nieskładnie, jak małe dzieci.
- Kiedy popełniasz błędy, to komputer nie uważa, że to “słodziotkie”. Mówi “błąd składni” — ponieważ zna język, którego ty dopiero się uczysz. Python wydaje się nieczuły i okrutny.
- Pamiętaj, że jesteś istotą inteligentną i potrafisz się uczyć. Komputer jest szybką, ale prostą maszyną i nie potrafi się uczyć. Więc Tobie łatwiej będzie nauczyć się Pythona niż komputerowi polskiego...

Rozmawianie z Pythonem

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>>
```

← Co dalej?

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
```

```
>>> x = 1
>>> print(x)
1
>>> x = x + 1
>>> print(x)
2
>>> exit()
```

To dobry test, żeby potwierdzić, że masz poprawnie zainstalowanego Pythona. Zwróć uwagę, że `quit()` też kończy sesję interaktywną.

Jak się mówi?

```
name = input('Podaj nazwę pliku: ')
handle = open(name, 'r')

counts = dict()
for line in handle:
    line = line.lower()
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in list(counts.items()):
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Krótką “opowieść”
o zliczaniu słów w
pliku za pomocą
Pythona

```
python3 words.py
Podaj nazwę pliku: words.txt
w 5
```

Elementy Pythona

- Słownictwo – zmienne i słowa zastrzeżone (Rozdział 2)
- Struktura zdań – poprawne wzorce składni (Rozdziały 3-5)
- Struktura opowieści – budowanie programu z myślą o celu

Zastrzeżone słowa

Nie możesz używać słów zastrzeżonych jako nazw zmiennych / identyfikatorów

False	class	return	is	finally
None	if	for	lambda	continue
True	def	from	while	nonlocal
and	del	global	not	with
as	elif	try	or	yield
assert	else	import	pass	
break	except	in	raise	

Zdania lub linie

```
x = 2      ← Instrukcja przypisania
x = x + 2  ← Przypisanie wyrażenia
print(x)   ← Instrukcja "print"
```

Zmienna Operator Stała Funkcja

Programistyczne akapity

Skrypty Pythona

- Tryb interaktywny Pythona nadaje się do eksperymentów i 3- lub 4-liniowych programów.
- Większość programów jest o wiele dłuższych, więc zapisujemy je w plikach i mówimy Pythonowi, żeby wykonywał zawarte w nich polecenia.
- W pewnym sensie “dajemy Pythonowi skrypt”.
- Z reguły dodajemy “.py” jako rozszerzenie nazw plików zawierających polecenia Pythona.

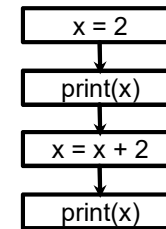
Tryb interaktywny kontra skrypt

- Tryb interaktywny
 - Pisziesz bezpośrednio do Pythona po jednej linii i dostajesz odpowiedź
- Skrypt
 - Zapisujesz sekwencje instrukcji (linii) do pliku, używając edytora tekstu i mówisz Pythonowi, żeby wykonał instrukcje z pliku

Kroki i przepływ programu

- Jak przepis albo instrukcja instalacji, program jest sekwencją kroków do wykonania w określonej kolejności.
- Niektóre są określone warunkowo – można je pominąć.
- Czasami jeden lub grupę kroków trzeba powtórzyć.
- Czasami zapisujemy listę kroków do wielokrotnego wykonania w razie potrzeby w różnych miejscach programu (Rozdział 4).

Kroki sekwencyjne



Program :

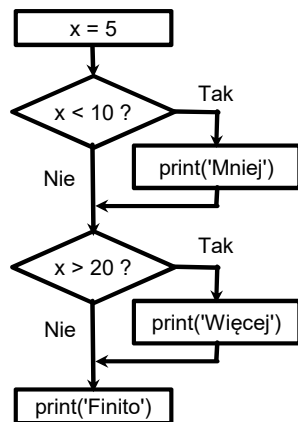
```

x = 2
print(x)
x = x + 2
print(x)
  
```

Wyjście

2
4

Podczas wykonywania program przechodzi od jednego kroku do następnego. Jako programiści wytyczamy "ścieżki", którymi ma przejść program.



Kroki warunkowe

Program :

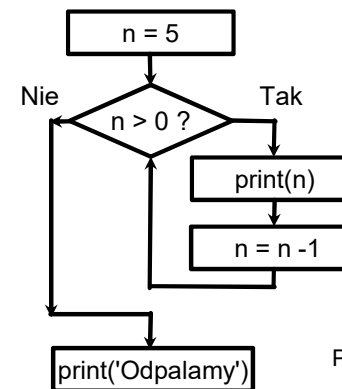
```

x = 5
if x < 10:
    print('Mniej')
if x > 20:
    print('Więcej')
print('Finito')
  
```

Wyjście:

Mniej
Finito

Powtarzane kroki



Program :

```

n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Odpalamy!')
  
```

Wyjście:

5
4
3
2
1
Odpalamy!

Pętle (powtarzane kroki) mają zmienne sterujące zmieniające się z każdym przejściem pętli.

```

name = input('Podaj nazwę pliku: ')
handle = open(name, 'r')
counts = dict()

for line in handle:
    line = line.lower()
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in list(counts.items()):
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)

```

Sekwencja
Powtarzanie
Warunek

```

name = input('Podaj nazwę pliku: ')
handle = open(name, 'r')
counts = dict()

for line in handle:
    line = line.lower()
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in list(counts.items()):
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)

```

Krótką "historia" w Pythonie o zliczaniu słów w pliku

Słowo używane, by odczytać dane od użytkownika

Zdanie o uaktualnianiu jednego z liczników

Akapit o tym, jak znaleźć największy element listy

Podsumowanie

- To był szybki przegląd Rozdziału 1
- Będziemy wracać do tych pojęć przez cały kurs
- Patrz na las, a nie drzewa



Podziękowania dla współpracowników



Copyright slajdów 2010 - Charles R. Severance
(www.dlcschuck.com) University of Michigan School of Information i open.umich.edu
dostępne na licencji Creative Commons Attribution 4.0. Aby zachować zgodność z
wymaganiami licencji należy pozostawić ten slajd na końcu każdej kopii tego
dokumentu. Po dokonaniu zmian, przy ponownej publikacji tych materiałów można
dodać swoje nazwisko i nazwę organizacji do listy współpracowników

Autorstwo pierwszej wersji: Charles Severance,
University of Michigan School of Information

Polska wersja powstała z inicjatywy Wydziału Matematyki
i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu

Tłumaczenie: Agata i Krzysztof Wierzbiccy, EnglishT.eu

... wstaw tu nowych współpracowników i tłumaczy