

# Programowanie

dr inż. Krzysztof Dorywalski

A close-up photograph of a bright green snake with yellowish-orange markings on its head and body. The snake is coiled, with its head in the foreground, looking directly at the camera. The background is dark and out of focus.

# Wstęp do python

# Historia python?

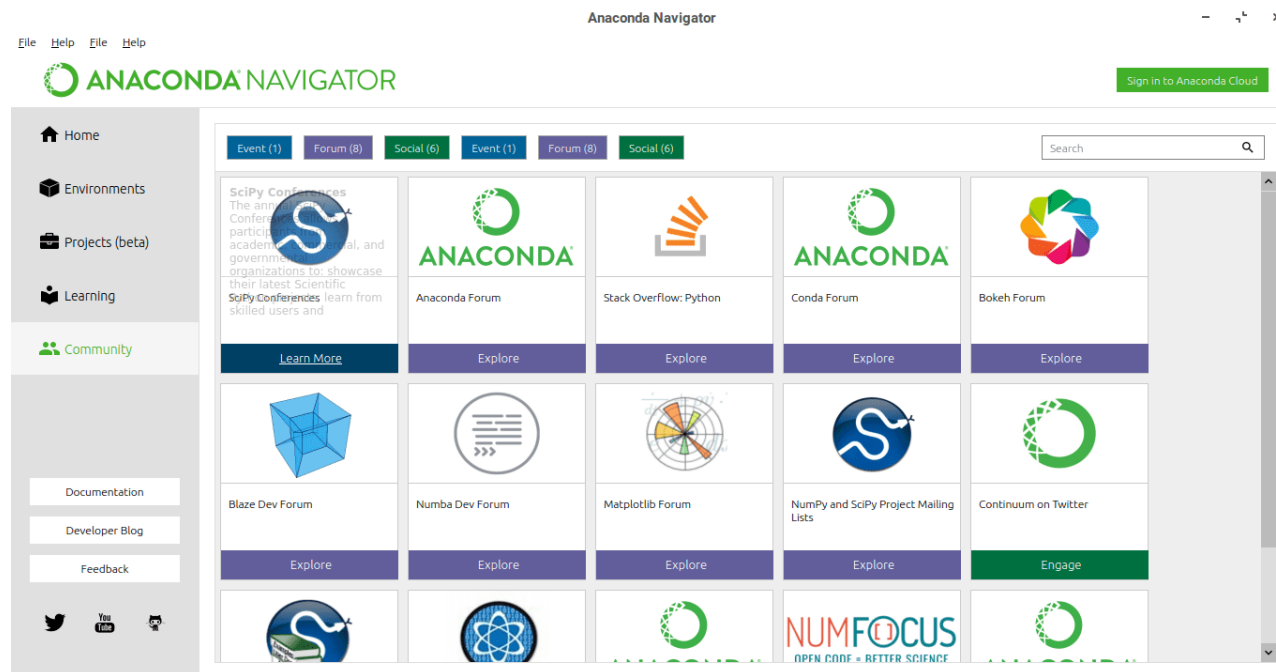
- Python powstał we wczesnych latach 90. dwudziestego wieku jako następca języka o nazwie ABC. Stworzył go Guido van Rossum z Stichting Mathematisch Centrum (CWI, patrz <http://www.cwi.nl/>) w Holandii. Guido pozostaje najważniejszym autorem Pythona, choć obecnie w pracach nad rozwojem projektu uczestniczy wiele innych osób
- W 1995 roku Guido kontynuował swoją pracę w organizacji Corporation for National Research Initiatives
- W maju 2000 roku Guido wraz z rdzeniem zespołu twórców Pythona przenieśli się do BeOpen.com, tworząc tam zespół BeOpen PythonLabs. W październiku tego samego roku zespół PythonLabs przeniósł się do firmy Digital Creations
- W 2001 roku została utworzona fundacja Python Software Foundation (PSF, patrz <http://www.python.org/psf/>) - nieochodowa organizacja, powstała przede wszystkim jako właściciel dorobku intelektualnego związanego z Pythonem. Firma Zope Corporation jest sponsorującym członkiem fundacji PSF.

# Dlaczego python?

- Język wysokiego poziomu
- Język obiektowy
- Struktura kodu – wcięcia
- Podstawowe narzędzie do obliczeń naukowych i inżynierskich
- Możliwość korzystania z bibliotek
- Język interpretowany
- Kwestia wydajności
- Dynamiczne typowanie
- Python 2.x a python 3.x

# Zaczynamy

# IDE



# Pierwszy program

## Zadanie 1:

Napisz skrypt, który wyświetli w 3 kolejnych liniach imiona:  
Rysiek, Zdzisiek, Krzysiek.



# Komentarz

- *#tekst* – komentarz jednowierszowy
- `'''` lub `"""` – komentarz wielowierszowy

# Typy danych

Ty obiektu	PRZYKŁAD
liczby	3.23, 101, 1e4
łańcuchy znaków	'tekst', "słowo", 'raz dwa trzy'
listy	[1, [2, 'trzy'], 4]
zbiory	set('abc'), {'a', 'b', 'c'}
wartości logiczne	wartości Boolean
liczby zespolone	1 = 4j

# Operatory arytmetyczne

Ty obiektu	PRZYKŁAD
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
%	reszta z dzielenia (modulo)
//	dzielenie całkowite

## Zadanie 2:

Napisz skrypt obliczający objętość kuli o zadanym przez użytkownika promieniu.

### Zadanie 3:

Napisz skrypt, który poprosi o podanie imienia i nazwiska użytkownika oraz przypisze imię i nazwisko do odpowiednich zmiennych. Dalej, skrypt wyświetli w konsoli powitanie. Zaproponuj, w jaki sposób moglibyśmy sprawdzić płeć użytkownika?

# Funkcje matematyczne – pakiet *math*

`math.sqrt(x)` - zwraca pierwiastek kwadratowy z liczby `x`.

`math.pow(x, y)` - podnosi liczbę `x` do potęgi `y`.

`math.exp(x)` - zwraca wartość wykładniczą liczby `x`.

```
mat import math
```

```
mat
```

```
mat print('pi to: %5.3f'% math.pi)
```

 ejszą lub równą `x`.

```
mat
```

 iszą lub równą `x`.

`math.sin(x)` - zwraca sinus kąta `x` wyrażonego w radianach.

`math.cos(x)` - zwraca cosinus kąta `x` wyrażonego w radianach.

`math.tan(x)` - zwraca tangens kąta `x` wyrażonego w radianach.

# Instrukcje warunkowe

```
if warunek:  
    polecenie_1  
    polecenie_2  
    itd.
```

# Instrukcje warunkowe

```
if warunek:  
    polecenie_1  
    polecenie_2  
else:  
    polecenie_3  
    polecenie_4  
    itd.
```



# Instrukcje warunkowe

```
if wyrażenie_warunkowe_1:  
    polecenie_1  
    polecenie_2  
    itd.  
elif wyrażenie_warunkowe_2:  
    polecenie_3  
    polecenie_4  
    itd.  
else:  
    polecenie_5  
    polecenie_6  
    itd.
```

# Instrukcje warunkowe

```
a = 5
b = 4
c = 3

if a < b:
    print(a)
elif b < c:
    print(b)
else:
    print(c)
```

# Instrukcje warunkowe

```
luty = 29 if rok_przestepny else 28
```

co odpowiada zwykłemu if/else

```
if rok_przestepny:  
    luty = 29  
else:  
    luty = 28
```

# Operatory porównań i logiczne

ZNAK	OPIS
==	sprawdzenie równości
!=	Nierówne
<>	Nierówne
>	większe
<	mniejsze
>=	większe równe
<=	mniejsze równe
and	koniunkcja
or	alternatywa
not	negacja

## Zadanie 4:

Napisz skrypt, który dla trzech liczb  $a$ ,  $b$ ,  $c$  wprowadzonych z klawiatury sprawdza, czy są to trójki pitagorejskie?

## Zadanie 5:

Napisz skrypt obliczający pierwiastki równania kwadratowego. Zastanów się, w jaki sposób napisać najbardziej wydajny program, zużywający jak najmniejsze zasoby

# Instrukcje warunkowe – przykład

Pierwsze prawo de Morgana brzmi: zaprzeczenie koniunkcji dwóch zdań  $\sim(p \wedge q)$  jest równoważne alternatywie zaprzeczeń tych zdań  $(\sim p) \vee (\sim q)$ .

```
a = True; b=True
```

```
lewa_str = not(a and b)
```

```
prawa_str = (not a) or (not b)
```

```
if lewa_str == prawa_str:
```

```
    print("Pierwsze prawo de Morgana jest OK")
```

```
else:
```

```
    print('WOW! Odkryłeś nowe prawo')
```

# Instrukcje warunkowe – porównanie tekstów

## Zadanie 6:

Napisz program, który korzystając z konstrukcji *if-else*, weryfikuje wprowadzone hasło i nazwę użytkownika.



# Pętle – iteracje: pętla *while*

```
while warunek:  
    polecenie1  
    polecenie2  
    itd.
```

# Pętle – iteracje: pętla *while*

```
a=0; b=10
```

```
while a<b:
```

```
    print(a, end=' ')
```

```
    a+=1
```

```
x='Siema'
```

```
while x:
```

```
    print(x, end=' ')
```

```
    x = x[1:]
```

## Zadanie 7:

Napisz program, który korzystając z instrukcji *while*, sumuje wszystkie liczby parzyste (nieparzyste) w przedziale od 1 do 100

# Instrukcje *break*, *continue*, *pass*, *else* w pętlach

- **break** – powoduje wyjście z najbliższej obejmującej daną instrukcję pętli
- **continue** – przechodzi na górę najbliższej obejmującej daną instrukcję pętli
- **pass** – nie wykonuje żadnej akcji
- **else w pętli** – wykonywana tylko wtedy, gdy pętla kończy się normalnie (bez instrukcji *break*)

# Instrukcja *break*

```
while True:
    imie = input('Podaj imię: ')
    if imie == 'stop':
        break
    print('Cześć ', imie)
```

# Instrukcja *continue*

```
x = 10
while x:
    x = x-1
    if x % 2 != 0:
        continue
    print(x, end=' ')
```

# Instrukcja *else* w pętli

```
y = 13
x = y//2

while x > 1:
    if y % 2 == 0:
        print(y, 'nie jest liczbą pierwszą, ma czynnik ', x)
        break;
    x -= 1
else:
    print(y, ' jest liczbą pierwszą')
```

# Pętle – iteracje: pętla *for*

```
for zmienna in [wartość1, wartość2 itd.]:  
    polecenie1  
    polecenie2  
    itd.
```



# Pętle – iteracje: pętla *for*

```
for i in range(10):  
    print(i, end=' ')
```

```
#przeglądanie listy  
lista = [5, 2, 1, 4, 7]  
  
for i in lista:  
    print(i)
```

# Pętle – iteracje: pętla *for* zagnieżdżona

*#for zagnieżdżony*

```
dane = ['aba', 101, (5, 6), 2.22]
test = [(5, 6), 3.22]
for klucz in test:
    for rzecz in dane:
        if rzecz == klucz:
            print(klucz, ' znaleziony')
            break
    else:
        print(klucz, ' nie znaleziony')
```

```
dane = ['aba', 101, (5, 6), 2.22]
test = [(5, 6), 3.22]
for klucz in test:
    if klucz in dane:
        print(klucz, ' znaleziony')
    else:
        print(klucz, ' nie znaleziony')
```

### Zadanie 8:

Napisz skrypt, który wyświetli liczby parzyste z zakresu od 0 do 10

### Zadanie 9:

Napisz skrypt, który wypisze co czwartą liczbę z zakresu od 1 do 50

### Zadanie 10:

Napisz skrypt, który znajdzie najmniejszą wartość na liście

### Zadanie 11:

Napisz skrypt, który poda sumę liczby aktualnej i poprzedniej z zakresu od 1 do 10

### Zadanie 12:

Napisz skrypt wyświetlający tabliczkę mnożenia dla iloczynu do 100. Pomyśl o wykorzystaniu podwójnej pętli oraz instrukcji *continue*

## Zadanie 13:

Napisz skrypt, który za pomocą instrukcji *for* znajduje największą i najmniejszą liczbę ze zbioru  $n$  wylosowanych liczb całkowitych, generowanych losowo w przedziale od 0 do 100 (w zadaniu  $n = 5$ ) oraz oblicza wartość średnią ze wszystkich wylosowanych liczb.

Porównaj obliczona wartość średniej dla, kolejno  $n = 50, 500, 5000$

Wskazówka: do losowania liczb pseudolosowych użyj modułu *random*

```
import random #dołączenie modułu  
random.randint(0, 100)
```

## Zadanie 14:

Napisz skrypt, do konwersji pomiędzy jednostkami temperatury. Program ma wyświetlać menu:

(1) C -> F

(2) F -> C

(k) koniec

Po wybraniu odpowiedniej opcji, wykonana zostanie odpowiednia operacja

## Zadanie 15:

Napisz skrypt sprawdzający, czy podane słowo jest palindromem

## Zadanie 16:

Napisz skrypt – grę, w którym celem jest odgadnięcie wylosowanej liczby w zakresie od 1 do 100. Jeśli podana liczba jest większa od wylosowanej, program powinien o tym poinformować. Podobnie w przypadku, gdy podana liczba jest mniejsza od wylosowanej



## Zadanie 17:

Napisz skrypt dokonujący analizy zadanego tekstu. Wynikiem analizy ma być częstość występowania poszczególnych liter alfabetu w tekście

## Zadanie 18:

Utwórz grę, w której komputer wybiera losowo słowo, które gracz musi odgadnąć. Komputer informuje gracza, ile liter znajduje się w wybranym słowie. Następnie gracz otrzymuje pięć szans na zadanie pytania, czy jakaś litera jest zawarta w tym słowie. Komputer może odpowiedzieć tylko „tak” lub „nie”. Potem gracz musi odgadnąć słowo.