

Języki skryptowe - Python

Lista 1 - październik

Punktacja:

- 80-100: bdb
 - 70-79: db+
 - 60-69: db
 - 50-59: dst+
 - 40-49: dst
-

Zadanie 1 (1 pkt)

Użyj trybu interaktywnego jako kalkulatora, aby wyliczyć następujące wyrażenia:

- suma liczb: 2, 3, 5, -8
- średnia liczb: 2, 3, 5, -8
- $\cos(\pi)$

wskazówka:

```
import math
dir(math)
help(math.cos)
```

- $\sin(x)$, gdzie $x = 2/3 * \pi$
- $\sin(x)$, gdzie $x = 2/3 * \pi - \pi$

Zadanie 2 (1 pkt)

W trybie interaktywnym wykonaj następującą serię poleceń:

```
2 * 2
_ * 3
_ * 4
_ * 5
```

Jakie jest znaczenie zmiennej `_`? Czy ułatwiłoby to wyliczenie punktu 2 z zadania 1?

Zadanie 3 (1 pkt)

Zbadaj (i wyjaśnij) działanie funkcji wbudowanej `round`, np.

```
help(round)
round(2.5)
round(2.51)
round(-2.5)
round(-2.51)
```

Zadanie 4 (1 pkt)

- W Pythonie 2 sprawdź wynik następujących działań:

```
import sys # informacje systemowe

x = sys.maxint # użyj help(sys), aby sprawdzić definicję maxint
type(x)

x += 1
type(x)
```

- Powtórz to samo w Pythonie 3. Korzystając z dowolnych źródeł (innych niż prowadzący lub koleżanki / koledzy z grupy) wyjaśnij otrzymany rezultat.

Wskazówka: jeśli dysponujesz tylko interpreterem Pythona 3, możesz na potrzeby tego zadania użyć konsoli online, np. <https://repl.it/languages/python>

Zadanie 5 (2 pkt)

- Korzystając z wbudowanej dokumentacji znajdź funkcję, która wyznacza licznik i mianownik ułamka zwykłego dowolnej liczby zmiennoprzecinkowej.

Wskazówka: `help(float)`

- Przetestuj jej działanie na kilku dowolnych przykładach.
- Przetestuj jej działanie na `math.pi`. Czy jesteś w stanie to wyjaśnić (π jest niewymierna)? Jeśli nie, to zapoznaj się z np. https://pl.wikipedia.org/wiki/Liczba_zmiennoprzecinkowa i sprawdź `sys.float_info`
- Wykonaj następujące obliczenia i wyjaśnij otrzymany wynik:

```
x = sys.float_info.max
x
2 * x
```

Zadanie 6 (1 pkt)

Zdefiniuj dwie zmienne typu tekstowego i jedną całkowitą, np.

```
a = "jeden"
b = "dwa"
c = 3
```

Wykonaj następujące polecenia:

```
a * b
a + b
a * c
a + c
a + str(c)
```

Zadanie 7 (1 pkt)

Korzystając z dokumentacji klasy *string* (`help(str)`) zapoznaj się z definicjami funkcji: `strip()`, `isnumeric()` oraz `rjust()`. Zademonstruj ich działanie na dowolnych przykładach.

Zadanie 8 (1 pkt)

Napisz skrypt, który wykona następujące czynności:

- zmiennej `a` przypisze wartość 5
- zmiennej `b` przypisze wartość 3
- zmiennej `P` przypisze wartość `a*b`
- wypisze na ekranie "Pole prostokąta o bokach `a` i `b` wynosi `P`." (gdzie w miejsce zmiennych zostaną wstawione odpowiednie wartości).

Uruchom program w konsoli. Następnie edytuj skrypt zmieniając długości boków i uruchom go raz jeszcze.

Zadanie 9 (2 pkt)

- Stwórz krotkę (*tuple*) zawierającą pięć cyfr: 0, 1, 2, 3, 4 oraz pięć literalów słownych: "pięć", "sześć", "siedem", "osiem", "dziewięć".
- Wydrukuj na ekranie trzy pierwsze elementy.
- Wydrukuj na ekranie 2 ostatnie elementy.
- Wydrukuj co drugi element (zaczynając od drugiego).

- Korzystając z funkcji *len* sprawdź ilość elementów w krotce oraz długość przedostatniego elementu.
- Niech *x* oznacza nazwę krotki. Wykonaj:
 - *x[:5] + (5, 6) + x[-3:]*
 - *x[:5], (5, 6), x[-3:]*
 - porównaj (i wyjaśnij) otrzymane wyniki
- Dodaj pusty literal słowny na koniec krotki. Czy możesz skorzystać z funkcji *append* (wyjaśnij)?

Zadanie 10 (1 pkt)

- Stwórz listę studentów: Kasia, Basia, Marek, Darek.
- Korzystając z funkcji *append* dodaj do listy Józka.
- Korzystając z funkcji *extend* dodaj do listy Anię i Basię.
- Posortuj alfabetycznie studentów.
- Wypisz na ekranie:
 - czwartego studenta na liście
 - dwóch pierwszych studentów na liście
 - dwóch ostatnich studentów na liście
- Korzystając z funkcji *remove* usuń wszystkie Basie.
- Korzystając z funkcji *len* sprawdź ilość studentów.
- Z ostatecznej listy studentów utwórz krotkę.

Zadanie 11 (1 pkt)

- Korzystając z *range* utwórz listę zawierającą wszystkie wielokrotności liczby 3 mniejsze od 100.
- Korzystając z *del* usuń co trzeci element (zaczynając od piątego).
- Sprawdź definicję funkcji wbudowanej *sum* (*help(sum)*). Wykorzystaj ją oraz funkcję *len*, aby wyliczyć średnią arytmetyczną otrzymanej listy.

Zadanie 12 (2 pkt)

- Stwórz krotkę: ('a', 'b', 'c', 'd').
- Zapoznaj się z dokumentacją funkcji *str.join*.

- Wykonaj następujące polecenia (gdzie `x` to zmienna wskazująca na krotkę):

```
"".join(x)
" ".join(x)
", ".join(x)
```

- Korzystając z funkcji `join` wydrukuj (jednolinijkową komendą) na ekranie 100 zer oddzielonych tabulacjami.

Zadanie 13 (3 pkt)

- Stwórz obiekt typu `str`, który przechowuje tekst ślubowania studenta:

```
slubowanie = """
wstępując do wspólnoty akademickiej Uniwersytetu Wrocławskiego, ślubuję uroczyście:
- zdobywać wiedzę i umiejętności,
- postępować zgodnie z prawem, tradycją i dobrymi obyczajami akademickimi,
- dbać o dobre imię Uniwersytetu Wrocławskiego i godność studenta.
"""
```

- W interpreterze sprawdź wynik:

```
>>> print(slubowanie)
>>> slubowanie
>>> slubowanie[0]
```

- Popraw zmienną `slubowanie`, aby tekst zaczynał się wielką literą.
- Korzystając z funkcji `count` sprawdź, ile razy występuje spójnik “i”.
- Korzystając z funkcji `count` sprawdź, ile razy występuje litera “i”.
- Korzystając z `in` sprawdź, czy słowo “Uniwersytet” występuje w tekście.
- Korzystając z funkcji `str.split`:
 - stwórz listę wyrazów występujących w tekście (30 słów => 30 elementów)
 - stwórz listę, której każdy element odpowiada jednej linijce tekstu (4 linie => 4 elementy)

Zadanie 14 (2 pkt)

- Korzystając z funkcji `sys.getsizeof` sprawdź, ile pamięci zajmuje:
 - 0
 - `2**100`
 - `2**1000`

- Sprawdź, ile pamięci zajmują: *True* i *False*. Czy jest to wynik, którego się spodziewałeś?
- Zapoznaj się z dokumentacją funkcji *isinstance*.
- Wykonaj następujące polecenia:

```
isinstance(0, int)
isinstance(0, float)
isinstance(0.0, float)
isinstance(True, bool)
isinstance(True, int)
```

- Wyjaśnij rozmiar *True* i *False*.

Zadanie 15 (2 pkt)

- Zapoznaj się z dokumentacją funkcji *id*.
- Wykonaj następujące polecenia:

```
x = 2
y = x
print(x, y, id(x), id(y))
y = 3
print(x, y, id(x), id(y))
```

- Wykonaj następujące polecenia:

```
x = [1,2]
y = x
print(x, y, id(x), id(y))
y[1] = 3
print(x, y, id(x), id(y))
```

- Wyjaśnij, dlaczego modyfikacja listy *y* zmienia wartość listy *x*, ale nie dzieje się tak w przypadku *int*.

Zadanie 16 (1 pkt)

Niech `even = range(2, 100, 2)`. Korzystając z operatora *splat* (*):

- przypisz trzy pierwsze elementy zmiennym *a*, *b*, *c*, a pozostałe zmiennej *d*
- przypisz trzy pierwsze elementy zmiennym *a*, *b*, *c*, a pozostałe zmiennej *-*
- stwórz zmienne *start* i *end*, które odpowiednio przyjmą wartość pierwszego i ostatniego elementu

- stwórz nową listę, która będzie zawierać wszystkie elementy oprócz pierwszego i ostatniego

Zadanie 17 (1 pkt)

Wykorzystaj listę składaną (*list comprehension*), aby stworzyć sekwencję kwadratów liczb naturalnych mniejszych od 100. Następnie (korzystając z *enumerate*) wydrukuj na ekranie:

```
1 -> 1
2 -> 4
3 -> 9
.
.
.
```

Zadanie 18 (1 pkt)

Przeanalizuj poniższy kod:

```
#!/usr/bin/env python

i = 0

# drukujemy wszystkie liczby parzyste mniejsze od 10
while i < 10:
    if i % 2:      # reszta z dzielenia != 0 -> True
        continue # pomiń liczby nieparzyste
    else:
        print(i) # drukuj liczby parzyste

    i += 1 # zwiększ i o jeden
```

- Czy skrypt będzie działał zgodnie z założeniami? Jeśli nie, to napraw go.

Zadanie 19 (1 pkt)

Uzupełnij skrypt o brakujące fragmenty:

```
#!/usr/bin/env python

# lista zakupów
grocery = ['jajka', 'mleko', 'chleb', 'masło', 'piwo']
# ilość sztuk
```

```

n_items = []
# zakazane produkty
prohibited = ['wódka', 'piwo', 'wino']

# w pętli pytamy użytkownika, ile sztuk danego produktu chce kupić
for product in groceries:
    # wydrukuj na ekranie komunikat: "Produkt [nazwa produktu]: sztuk = "
    # pobierz liczbę od użytkownika i zapisz w n_items
    # pomiń produkty zakazane (i automatycznie przyjmij ilość = 0)

# drukujemy listę zakupów

print("{:-^50}".format("Lista zakupów"), end="\n\n")

# w pętli wydrukuj: [lp]. [nazwa produktu]: [ilość]
# czyli np.: 1. jajka: 5 itd.

```

Zadanie 20 (3 pkt)

Napisz skrypt, który:

- losuje liczbę całkowitą mniejszą od 100 (`help(random.randint)`)
- pyta użytkownika o odgadnięcie liczby
- informuje użytkownika, czy podana przez niego liczba jest:
 - dużo mniejsza (różnica > 50)
 - mniejsza (różnica > 10)
 - trochę mniejsza
 - trochę większa
 - większa (różnica > 10)
 - dużo większa (różnica > 50)
- program się kończy, gdy użytkownik odgadnie wylosowaną liczbę

Zadanie 21 (4 pkt)

Poniższy skrypt narysuje kwadrat:

```

#!/usr/bin/env python

import turtle

length = eval(input("Podaj długość boku: "))
n_sides = 4 # ilość boków

turtle.speed(20) # ustal prędkość żółwia

```



```

# powtórz n_sides razy
for i in range(n_sides):
    turtle.forward(length) # narysuj linię o danej długości
    turtle.right(90)        # obróć się w prawo o dany kąt

turtle.mainloop() # nie zamykaj okna po narysowaniu

```

- zmodyfikuj go tak, aby narysował trójkąt równoboczny (1 pkt)
- zmodyfikuj go tak, aby narysował sześciokąt foremny (2 pkt)
- zmodyfikuj go tak, aby narysował wielokąt foremny, którego liczba boków podana jest przez użytkownika (3 pkt)
- zmodyfikuj go tak, aby wielokąt rysowany był N razy (N podane przez użytkownika); każdy kolejny obrócony o odpowiedni kąt (aż do wykonania pełnego kąta) (4 pkt); poniżej przykład dla 50 kwadratów:

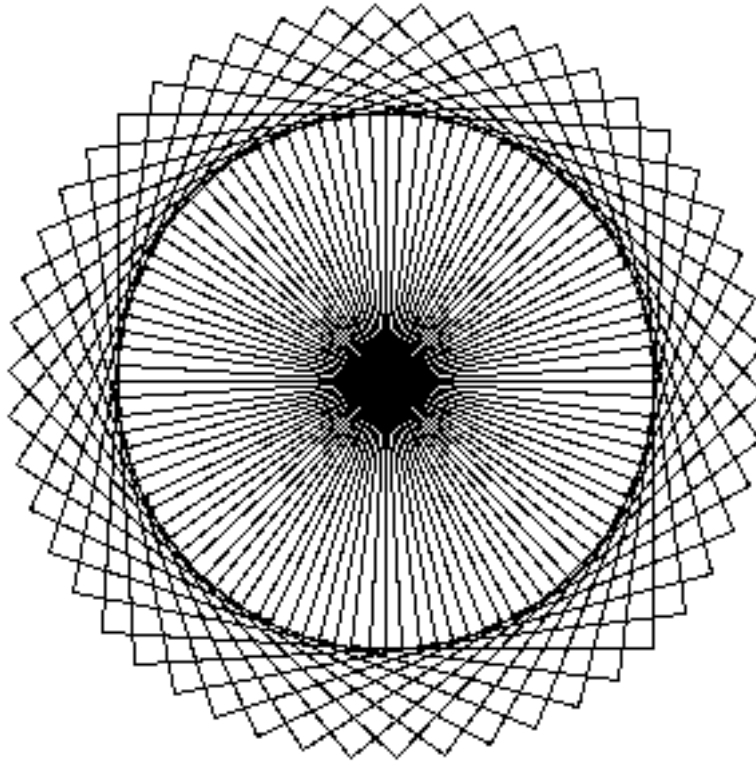


Figure 1: turtle example

Zadanie 22 (3 pkt)

Stwórz słownik, który przyporządkuje pięciu różnym produktom ich cenę. Następnie:

- w pętli wydrukuj na ekranie listę produktów z ceną
- policz średnią cenę produktu
- dodaj nowy produkt
- jak się zmieniła średnia cena?
- napisz funkcję, która liczy średnią cenę
- usuń produkt
- policz średnią cenę

Zadanie 23 (3 pkt)

- Napisz funkcję, która znajduje mniejszą liczbę z dwóch podanych. (*bez korzystania z wbudowanej funkcji min*)
- Wykorzystując funkcję z pierwszego punktu, napisz kolejną, która z podanych liczb (ilość dowolna) znajduje najmniejszą.

Zadanie 24 (2 pkt)

Napisz funkcję, która wypisze na ekranie n pierwszych wyrazów ciągu Fibonacciego.

Zadanie 25 (3 pkt)

Napisz program, który pobiera od użytkownika współczynniki trójkianu kwadratowego, a następnie podaje jego rozwiązania.

Uwaga: Rozłóż program na mniejsze funkcje.

Zadanie 26 (3 pkt)

- Napisz funkcję, która przyjmuje dowolną liczbę argumentów pozycyjnych oraz kluczowych.
- Niech funkcja drukuje argumenty pozycyjne w formie listy:

```
1 -> wartość pierwszego argumentu
2 -> wartość drugiego argumentu
.
.
.
```

- Niech funkcja drukuje argumenty kluczowe w formie listy:

nazwa (klucz) -> wartość

.

- Stwórz listę oraz słownik
- Przekaż pojedyncze elementy listy jako kolejne argumenty pozycyjne, a słownik jako kolejne argumenty kluczowe swojej funkcji.

Zadanie 27 (2 pkt)

Napisz program, który liczy liczbę dni między dwoma datami (*wsk. date-time.date*).

Zadanie 28 (2 pkt)

Napisz program, który pobiera od użytkownika średnicę okręgu a następnie drukuje na ekranie jego obwód i pole. *Uwaga: program powinien się upewnić, że podano liczbę i w razie potrzeby pytać do skutku.*

Zadanie 29 (2 pkt)

Napisz program, który pobiera od użytkownika literę i wyświetla komunikat z informacją, czy podano spółgłoskę czy samogłoskę.

Zadanie 30 (2 pkt)

Napisz funkcję, która przekształca podaną listę na string.

Zadanie 31 (3 pkt)

Napisz program, który liczy dystans między dwoma punktami (x_1, y_1) i (x_2, y_2) . Współrzędne punktów pobierane od użytkownika.

Zadanie 32 (5 pkt)

Napisz program do konwersji między kartezjańskim a sferycznym układem współrzędnych. Program powinien:

- na starcie dać wybór, w którą stronę będzie dokonywana konwersja
- pobrać odpowiednie współrzędne od użytkownika
- upewnić się, że podane dane mają sens
- wydrukować nowe współrzędne

Zadanie 33 (5 pkt)

Napisz program do analizy ruchu ciała w rzucie ukośnym. Program powinien:

- pobierać od użytkownika wartość prędkości początkowej i kąt
- drukować na ekranie poniższe informacje:
 - czas lotu
 - maksymalną wysokość
 - zasięg
- następnie w pętli pytać użytkownika o podanie czasu t i wyświetlać położenie punktu $(x(t), y(t))$
- pętla powinna się kończyć, gdy użytkownika poda $t = 0$

Zadanie 34 (2 pkt)

Napisz funkcję `generate_password(n)`, która zwraca losowo wygenerowane hasło o długości n . Hasło powinno składać się z liter i cyfr.

wskazówka:

```
import string
string.ascii_letters
string.digits
```

Zadanie 35 (5 pkt)

Dany jest szyfr, który zamienia samogłoski wg klucza:

```
klucz = {"a": "y", "e": "i", "i": "o", "o": "a", "y": "e"}
```

Czyli wszystkie a zamienia na y itd. Np. tekst:

```
to jest moj tekst
```

po przepuszczeniu przez szyfr wyglądać będzie następująco:

```
ta jist maj tikst
```

Napisz funkcję do szyfrowania i deszyfrowania tekstu. Dla uproszczenia przyjmij, że szyfrowane są tylko małe litery.

Za pomocą napisanej funkcji rozszyfruj:

```
noi zypamnoj a dakumintycjo
```

wskazówka: do deszyfrowania można użyć:

```
dekucz = {v: k for k, v in klucz.items()}
```

Zadanie 36 (3 pkt)

Napisz program, który sprawdza, czy podane przez użytkownika zdanie (załóż, że podany ciąg znaków jest zdaniem) jest pangramem (zawiera wszystkie litery alfabetu - można pominąć polskie znaki). Przetestuj działanie na:

```
The quick brown fox jumps over the lazy dog.
False, I see false, this exam I may pass.
```

wskazówka:

```
import string
string.ascii_lowercase
```

Zadanie 37 (3 pkt)

Napisz funkcję, która liczy i zwraca wartość wielomianu w zadanym punkcie:

$$w(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 \dots + a_n \cdot x^n$$

Współczynniki a_0, a_1, \dots, a_n oraz x powinny być przyjmowane jako argumenty funkcji.

Zadanie 38 (2 pkt)

Napisz program, który sprawdza, czy podane przez użytkownika słowo / zdanie jest palindromem (wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej).

Zadanie 39 (3 pkt)

Napisz funkcję, która rozkłada podaną liczbę na czynniki pierwsze. Następnie wykorzystaj tę funkcję do wyznaczenia największego wspólnego dzielnika oraz najmniejszej wspólnej wielokrotności dwóch liczb podanych przez użytkownika.

Zadanie 40 (4 pkt)

Napisz program do obsługi ruchu jednostajnie przyspieszonego. Program powinien:

- pobrać od użytkownika wartości: drogi początkowej, prędkości początkowej oraz przyspieszenia
- narysować wykresy (korzystając np. z `matplotlib`) zależności drogi i prędkości od czasu