

Zadania podstawy pythona

Maciej Puchała

5 lipca 2023

Funkcja print()

1. Zadanie 1.

Napisz program wypisujący dowolne dwa komunikaty na ekranie.

2. Zadanie 2.

Przypisz tekst "Hello World!" do zmiennej `my_text` oraz wypisz ją na ekranie.

```
1 my_text=""
2 print(my_text)
```

3. Zadanie 3.

Funkcja `print()` może przyjmować wiele argumentów różnego typu. Przetestuj jej możliwość podając wiele argumentów różnych typów np. tekst, liczby itp.

Zmienne

4. Zadanie 4.

Przypisz do zmiennej `glass_of_water` wartość 3.

```
1 glass_of_water=
2
3 print("I drank", glass_of_water, "glasses of water today.")
```

5. Zadanie 5.

Zobacz co się stanie gdy do naszej zmiennej `glass_of_water` przypiszemy nową wartość.

```
1 #Fill the print function so it prints glass_of_water
2
3 glass_of_water=3
4
5 glass_of_water=glass_of_water + 1
6
7
8 print()
```

Typy danych

6. Zadanie 6.

Sprawdźmy różne typy danych dostępne w pythonie.

a) stwórz zmienną z liczbą całkowitą. Następnie wypisz ją na ekranie razem z jej typem (użyj funkcji `type()`)

b) stwórz zmienną z liczbą zmiennoprzecinkową np. zaokrągloną liczbą PI. Następnie wypisz ją na ekranie razem z jej typem (użyj funkcji `type()`)

c) stwórz zmienną z tekstem. Następnie wypisz ją na ekranie razem z jej typem (użyj funkcji `type()`)

d) do jednej z stworzonych zmiennych przypisz wartość innego typu i zobacz co się stanie.

e)stwórz zmienną z wartością logiczną. Następnie wypisz ją na ekranie razem z jej typem(użyj funkcji type())

```
1 #Assign True or False to the variable below then print it.
2
3 staying_alive=
4 print(staying_alive)
```

Konwersja typów

7. Zadanie 7.

Przekonwertuj tekst(string) na liczbę całkowitą(int).

```
1 #my_grade variable is a string (because it's in quotes). On line 9, convert it to an integer.
2
3 my_grade="10"
4
5 answer_5=
6
7
8 print(answer_5)
```

8. Zadanie 8.

Przekonwertuj liczbę zmiennoprzecinkową(float) na liczbę całkowitą(int).

```
1 #my_temp variable is a float (because it has decimals). convert it to an integer.
2
3 my_temp=97.70
4
5 answer_6=
6
7
8 print(answer_6)
```

9. Zadanie 9.

Przekonwertuj liczbę zmiennoprzecinkową(float) na tekst(string).

```
1 #my_temp variable is a float (because it has decimals). convert it to an integer.
2
3 my_temp=97.70
4
5 answer_6=
6
7
8 print(answer_6)
```

Operatory

10. Zadanie

Operator przypisania. Przypisz do zmiennej lst liste kolorów ["red","green","blue"]

```
1
2 lst=
3
4
5 print(lst)
```

11. Zadanie

Operator dzielenia. Przypisz wynik dzielnia zmiennej a przez zmienną b do zmiennej result.

```
1 a=10
2 b=3
3
4
5 result=
6
7
8 print(result)
```

12. Zadanie

Operator przypisania z dodawaniem `+=`. Za pomocą operatora `+=` dodaj wartość 100 do zmiennej `speed`.

```
1 speed=750
2
3
4
5 print(speed)
```

13. Zadanie

Operator modulo(reszta z dzielnia) `%`. Za pomocą operatora modulo przypisz reszta z dzielenia 1000/400 do zmiennej `remainder`.

```
1 remainder=
2
3
4
5
6 print(remainder)
```

14. Zadanie

Operator potęgowania `**`. Za pomocą operatora potęgowania przypisz do zmiennej `p_result` kwadrat liczby 11111

```
1 p_result=
2
3
4
5
6
7 print(p_result)
```

15. Zadanie

Operator porównania `==`. Za pomocą operatora porównania sprawdź czy wartość zmiennej `"a"` jest równa zmiennej `"b"`

```
1 a = 450
2 b = 500
3
4 is_equal =
5
6 print(is_equal)
```

Operacje na napisach

16. Zadanie

Do zmiennej napis przypisz wartość `"It's always darkest before dawn."`

```
1 napis=""
2
3 print(napis)
```

17. Zadanie

Poniżej mamy dwie zmienne z napisami. połącz je ze sobą w jeden napis a następnie wyświetl. zrób to na co najmniej 2 sposoby.

```
1 witaj = "Witaj "
2 imie = "Jozek"
```

18. Zadanie

Używając pierwszej, drugiego i ostatniego znaku napisu ze zmiennej napis stwórz nowy napis.

```
1 napis="It's always darkest before dawn."
2
3 #Type your answer here.
4
5 ans_1=
6
7 print(ans_1)
```

19. Zadanie

Zamień znak "." na znak "!"

```
1 napis="It's always darkest before dawn."
2
3 #Type your code here.
4
5
6
7 print(napis)
```

20. Zadanie

Nadpisz zmienną napis do samej siebie tak aby wszystkie znaki były małymi literami.

```
1 napis="EVERY Strike Brings Me Closer to the Next Home run."
2 # Type your code here.
3
4
5
6
7 print(napis)
```

21. Zadanie

Nadpisz zmienną napis do samej siebie tak aby wszystkie znaki były WIELKIMI literami.

```
1 napis="don't stop me now."
2 # Type your code here.
3
4
5
6 print(napis)
```

22. Zadanie

Czy napis zaczyna się od litery "A"?

przypisz wartość logiczną odpowiedzi do zmiennej ans_1.

```
1 napis="There are no traffic jams along the extra mile."
2 # Type your code here.
3
4 ans_1=
5
6
7 print(ans_1)
```

23. Zadanie

Czy napis kończy się "."?

przypisz wartość logiczną odpowiedzi do zmiennej ans_1.

```
1 napis="There are no traffic jams along the extra mile."
2 # Type your code here.
3
4
5 ans_1=
6
7
8 print(ans_1)
```

24. Zadanie

a) używając metody `.index()`, znajdź indeks na którym znajduje się znak "v".

```
1 napis="The best revenge is massive success."
2
3 #   Type your code here.
4
5 ans_1=
6
7
8 print(ans_1)
```

b) Używając metody `.find()` znajdź indeks na którym znajduje się znak "m".

```
1 napis="The best revenge is massive success."
2
3 #   Type your code here.
4
5 ans_1=
6
7
8 print(ans_1)
```

c) Jaka jest różnica między tymi dwoma metodami?(spróbujcie znaleźć znak którego nie ma w napisie)

25. Zadanie

Który znak występuje częściej w napisie? "a" czy "o"?

```
1 napis="People often say that motivation doesn't last. Well, neither does bathing. That's why we
   recommend it daily."
2 #   Type your code here.
3
4 ans_1=
5
6 ans_2=
7
8
9 print("count of a is: ", ans_1, " count of o is: ", ans_2)
```

26. Zadanie

Wypisz długość napisu ze zmiennej `napis`.

```
1 napis="1.975.000"
2
3 #   Type your code here:
4
5 ans_1=
6
7
8 print(ans_1)
```

Cięcie(Slicing)

27. Zadanie

a) Przekrój słowo ze zmiennej `wrd` do pierwszego wystąpienia litery "a"(Tosc).

b) Przekrój słowo `wrd` tak aby otrzymać "cana".

c) Przekrój `wrd` tak aby otrzymać napis "can".

d) Przekrój słowo `wrd` tak aby otrzymać co drugi znak.

e) Przekrój słowo `wrd` tak aby otrzymać co drugi znak bez pierwszego i ostatniego znaku.

f) Czy możesz przeciąć słowo `wrd` tak aby było w odwrotnej kolejności bez używania metody `reverse`?(anacsoT)

```
1 wrd="Toscana"
2 #Type your answer here.
3
4 ans_1=
```

```
5  
6  
7 print(ans_1)
```

Struktury danych

28. Zadanie 10.

a) Stwórz pustą listę oraz wypisz jej typ.

```
1 gift_list=  
2  
3 answer_1=  
4  
5  
6 print(answer_1)
```

b) Stwórz pusty słownik oraz wypisz jego typ.

```
1 #Create an empty dictionary on line 3 and assign its type on line 4  
2  
3 grocery_items=  
4  
5 answer_2=  
6  
7  
8 print(answer_2)
```

c) Stwórz pustą krotkę oraz wypisz jej typ.

```
1 #Create an empty tuple on line 3 and assign its type on line 4  
2  
3 bucket_list=  
4  
5 answer_2=  
6  
7  
8 print(answer_2)
```

29. Zadanie 11.

a) Stwórz listę z wartościami oraz ją wypisz.

b) Stwórz słownik z wartościami oraz go wypisz.

c) Stwórz krotkę z wartościami oraz ją wypisz.

Listy

30. Zadanie 12.

Przypisz pierwszy element z listy do zmiennej `answer_1`.

```
1 lst=[11, 100, 99, 1000, 999]  
2 answer_1=  
3  
4 print(answer_1)
```

31. Zadanie 13.

Wypisz drugi element listy bezpośrednio w funkcji `print`.

```
1 lst=[11, 100, 101, 999, 1001]  
2  
3 print()
```

32. Zadanie 14.

Przypisz ostatni element listy do zmiennej `answer_1`.

```
1 lst=[11, 100, 101, 999, 1001]
2 #Type your answer here.
3
4 answer_1=
5
6 print(answer_1)
```

33. Zadanie

Metoda `.append` pozwala dodawać elementy do listy. Dodaj do listy napis "pajamas" za pomocą metody `append`.

```
1 gift_list=['socks', '4K drone', 'wine', 'jam']
2 # Type your code here.
3
4
5
6 print(gift_list)
```

34. Zadanie 15.

W listach może znajdować się wiele różnych typów danych, możesz nawet dodać nową listę jako element list. Nazywane to jest zagnieżdżaniem danych.

W 3 linii dodaj listę ["socks", "tshirt", "pajamas"] na koniec listy `gift_list`.

```
1 gift_list=['socks', '4K drone', 'wine', 'jam']
2 # Type your code here.
3
4
5
6 print(gift_list)
```

35. Zadanie

Metoda `.insert()` pozwala wstawić element w konkretnym miejscu listy (na konkretnym indeksie).

Za pomocą metody `.insert()` dodaj napis "slippers" na 3 miejscu listy.

```
1 gift_list=['socks', '4K drone', 'wine', 'jam']
2 # Type your code here.
3
4
5
6 print(gift_list)
```

36. Zadanie

Przy pomocy metody `.index()` można znaleźć numer indeksu na którym znajduje się dany element.

Do zmiennej `answer_1` przypisz numer indeksu na którym znajduje się wartość 8679.

```
1 lst=[55, 777, 54, 6, 76, 101, 1, 2, 8679, 123, 99]
2 # Type your code here.
3
4 answer_1=
5
6 print(answer_1)
```

37. Zadanie

Używając metody `remove` usuń ostatni element z listy.

```
1 lst=[55, 777, 54, 6, 76, 101, 1, 2, 8679, 123, 99]
2 # Type your code here.
3
4 lst.remove(99)
5
6
7 print(lst)
```

38. Zadanie

Używając metody `.reverse()` odwróć kolejność elementów w liście.

```
1 lst=[55, 777, 54, 6, 76, 101, 1, 2, 8679, 123, 99]
2 # Type your code here.
3
4
5
6 print(lst)
```

39. Zadanie

Używając metody `.count()` sprawdź ile razy 6 występuje w liście.

```
1 lst=[55, 6, 777, 54, 6, 76, 101, 1, 6, 2, 6]
2 # Type your code inside print() function.
3
4 answer_1=
5
6 print(answer_1)
```

40. Zadanie

Wypisz sumę wszystkich elementów z listy `lst`.

```
1 lst=[55, 6, 777, 54, 6, 76, 101, 1, 6, 2, 6]
2
3 # Type your code on line 4:
4 answer_1=
5
6 print(answer_1)
```

41. Zadanie

a)Znajdź najmniejszy element listy.

b)znajdź największy element listy

```
1 lst=[55, 6, 777, 54, 6, 76, 101, 1, 6, 2, 6]
2
3 # Type your code on line 4:
4 answer_1=
5
6 print(answer_1)
```

Krotki

42. Zadanie

Wykonaj zadania 29 - 40 zamieniając listy na krotki.

Słowniki

43. Zadanie

Słowniki nie mają indeksów więc odnoszenie się do pierwszego lub ostatniego elementu nie jest poprawne. Zamiast tego słowniki posiadają klucze a my możemy używać kluczy do pozyskiwania z nich wartości.

a) kiedy urodził się Plato?

b) Zmień datę urodzenia Plato z B.C. 427 na B.C. 428.

```
1 dict={"name": "Plato", "country": "Ancient Greece", "born": -427, "teacher": "Socrates", "student": "
  Aristotle"}
2 answer_1=
3
4 print(answer_1)
```


44. Zadanie

Słowniki także mogą zawierać zagnieżdżone dane. Dodaj do słownika klucz "work" z poniższą listą.
"work": ["Apology", "Phaedo", "Republic", "Symposium"]

```
1 dict={"name": "Plato", "country": "Ancient Greece", "born": -427, "teacher": "Socrates", "student": "Aristotle"}
2
3 #Type your answer below.
4
5
6 print(dict)
```

45. Zadanie

Dodaj 2 do wartości "son's height".

```
1 dict={"son's name": "Lucas", "son's eyes": "green", "son's height": 32, "son's weight": 25}
2
3 #Type your answer below.
4
5
6 print(dict)
```

46. Zadanie

Przy pomocy metody .items() wygeneruj listę krotek gdzie w każdej krotce będzie klucz i wartość.

```
1 dict={"son's name": "Lucas", "son's eyes": "green", "son's height": 32, "son's weight": 25}
2
3 #Type your answer below.
4 ans_1= dict.items()
5
6 print(ans_1)
```

47. Zadanie

Zapoznaj się z metodą .get() i wykonaj następujące zadania:

a) wypisz wartość dla klucza "son's eyes"

b) Spróbuj odczytać wartość dla klucza "son's age" a jeżeli klucz nie istnieje zwróć wartość "2"

```
1 dict = {"son's name": "Lucas", "son's eye color": "green", "son's height": 32, "son's weight": 25}
2
3 #Type your answer inside the print.
4 ans_1=
5
6 print (ans_1)
```

48. Zadanie

Metoda .clear() służy do czyszczenia słownika. Wypróbuj ją.

```
1 dict={"son's name": "Lucas", "son's eye color": "green", "son's height": 32, "son's weight": 25}
2
3 #clear the dictionary here then print it.
4
5
6 print(dict)
```

49. Zadanie

Za pomocą funkcji len() sprawdź ile kluczy znajduje się w słowniku.

```
1 dict={"son's name": "Lucas", "son's eye color": "green", "son's height": 32, "son's weight": 25}
2
3 #Write your answer here.
4 ans_1=
5
6 print(ans_1)
```

Wprowadzanie danych

50. Zadanie

Używając funkcji input() poproś użytkownika o imię.

```
1 #Type your answer here.
2
3 ans_1=
4
5 print("Hello!, " + ans_1)
```

51. Zadanie

Poproś użytkownika o dane numeryczne np. wiek.

```
1 #Type your code here.
2 ans_1=
3
4
5 print(type(ans_1))
```

52. Zadanie

Poproś użytkownika o wpisanie obecnego roku i wypisz rok za 50 lat.

53. Zadanie

Stwórz konwerter który zapyta o ilość dni i przekonwertuje je na lata.
załóżmy, że rok ma zawsze 365 dni.

```
1 # Type your answer here.
2
3 message=
4
5 result=
6
7
8 print(result)
```

54. Zadanie

Stwórz konwerter który przeliczy mile na kilometry.
załóżmy, że rok ma zawsze 365 dni.

```
1 # Type your answer here.
2
3 message=
4
5 result=
6
7
8 print(result)
```

Instrukcje sterujące

55. Zadanie

Napisz program który przyjmie od użytkownika imię i jeżeli będzie to "Bond"to wypisze w konsoli "Welcome on board 007."w przeciwnym wypadku wypisze "Good morning NAME"(NAME zamień na wprowadzone imie.)

56. Zadanie

Napisz program który sprawdzi czy wprowadzona liczba jest parzysta. i Jeżeli jest wypisze "Parzysta" w przeciwnym wypadku wypisze "nieparzysta".

Pętle

57. Zadanie

Napisz pętlę while która doda do siebie wszystkie liczby od 0 do 100 włącznie.

```
1 counter=0
2 total=0
3
4 #Construct your while loop here.
5
6
7
8
9
10 print(total)
```

58. Zadanie

Używając pętli while, funkcji len(), oraz instrukcji warunkowej sprawdź czy w danej liście znajduje się wartość 100 jeżeli tak to przypisz do zmiennej my_message wiadomość na jakim indeksie się znajduje np. "There is a 100 at index no: 5".

```
1 lst=[10, 99, 98, 85, 45, 59, 65, 66, 76, 12, 35, 13, 100, 80, 95]
2
3 my_message=""
4
5 #Type your code here.
6
7
8
9
10 print(my_message)
```

59. Zadanie

Używając pętli while i instrukcji warunkowych napisz program który stworzy nową listę napisów bez napisów pustych "".

```
1 lst1=["Joe", "Sarah", "Mike", "Jess", "", "Matt", "", "Greg"]
2
3
4 #Type your code here.
5
6
7
8
9
10
11
12 print(name_adder(lst1))
```

60. Zadanie

używając pętli for stwórz program który wypisze każdy element z listy w osobnym wierszu.

```
1
2 lst=["koala", "cat", "fox", "panda", "chipmunk", "sloth", "penguin", "dolphin"]
3 #Type your answer here.
```

61. Zadanie

używając pętli for stwórz program który wypisze tekst "Hello!" + każde imię z listy tj. "Hello!, Sam"

```
1
2 st=["Sam", "Lisa", "Micha", "Dave", "Wyatt", "Emma", "Sage"]
3 Type your code here.
```

62. Zadanie

Stwórz licznik do którego w każdym obiegu pętli będziesz dodawać 1. Czy wiesz ile razy wykona się pętla?

```

1 str="Civilization"
2
3 c=0
4 for i in str:
5     #Type your answer here.
6
7
8 print(c)

```

63. Zadanie

Używając pętli for oraz metody .append() dodaj przedrostek Dr. d każdego elementu lst1.

```

1 lst1=["Phil", "Oz", "Seuss", "Dre"]
2 lst2=[]
3 #Type your answer here.
4
5
6
7
8 print(lst2)

```

64. Zadanie

Używając pętli for oraz metody .append() do listy lst2 dodaj kwadrat każdego elementu z lst1.

```

1 lst1=[3, 7, 6, 8, 9, 11, 15, 25]
2 lst2=[]
3 #Type your answer here.
4
5
6
7 print(lst2)

```

65. Zadanie

Używając pętli for napisz program który do listy lst2 doda tylko dodatnie liczby z listy lst1.

```

1 lst1=[111, 32, -9, -45, -17, 9, 85, -10]
2 lst2=[]
3 #Type your answer here.
4
5
6
7 print(lst2)

```

66. Zadanie

Używając pętli for napisz program który, do listy lst doda wartość ze słownika dict -1000, jeżeli wartość w słowniku jest większa od 1000, np jeżeli wartość w słowniku jest równa 1500 to do listy powinien zostać dodany element 500.

```

1 dict={"z1":900, "t1": 1100, "p1": 2300, "r1": 1050, "k1": 3200, "g1": 400}
2 lst=[]
3 #Type your answer here.
4
5
6
7 print(lst)

```

67. Zadanie

Napisz program który do listy lst2 doda typ dla każdego elementu z listy lst1. np. lst1 = ["hi",1] to lst2 = [type str, type int]

```

1 lst1 = [3.14,66,"Teddy Bear",True,[],{}]
2 lst2=[]
3 #Type your answer here.
4
5
6
7 print(lst2)

```

List Comprehension

68. Zadanie

Stwórz identyczną listę przy pomocy list comprehension. czyli lst1 powinna być taka sama jak lst2

```
1 lst1=[1,2,3,4,5]
2
3 #Type your answer here.
4
5 lst2=
6
7 print(lst2)
```

69. Zadanie

Stwórz listę z elementów od 1200 do 2000 z krokiem 130 używając list comprehension.

```
1 #Type your answer here.
2
3 rng=
4
5 lst=[]
6
7 print(lst)
```

70. Zadanie

Użyj list comprehension aby stworzyć nową listę ale dodaj 6 do każdego elementu listy lst1, np. lst1 = [1,2,3]
lst2 = [7,8,9]

```
1
2 #Type your answer here.
3
4 lst1=[44,54,64,74,104]
5
6 lst2=[]
7
8
9 print(lst2)
```

71. Zadanie

Użyj list comprehension aby stworzyć nową listę z kwadratów każdego elementu listy lst1 ale tylko jeżeli jest on większy od 50. np. lst1=[1,2,10] to lst2 = [100]

```
1
2 lst1=[2, 4, 6, 8, 10, 12, 14]
3
4 #Type your answer here.
5
6 lst2=[]
7
8
9 print(lst2)
```

72. Zadanie(bonus)

Mamy słownik w którym kluczem są typy pojazdów a wartością ich waga. Stwórz listę pojazdów których waga jest mniejsza od 5000 kg, a także ich nazwy będą w liście w wielkich literach.

```
1 dict={"Sedan": 1500, "SUV": 2000, "Pickup": 2500, "Minivan": 1600, "Van": 2400, "Semi": 13600, "
   Bicycle": 7, "Motorcycle": 110}
2
3 #Type your answer here.
4
5 lst=[]
6
7 print(lst)
```

Mini projekty

73. Wisielec

- a) stwórz zmienną na słowa z których program będzie mógł wybrać jedno (na początku może to być stałe jedno słowo)
- b) stwórz zmienną która będzie przechowywała liczbę szans np. 5 po 5 niepoprawnych zgadnięciach litery gra się kończy.
- c) stwórz główną pętlę gry która będzie działała tak długo aż liczba szans się nie wyczerpie lub słowo nie zostanie w pełni odkryte.
- d) wewnątrz pętli poproś o podanie litery, sprawdź czy litera znajduje się w słowie i wypisz miejsca w których się znajduje np. " _ _ _ a _ _ _ a _ _ _ " jeżeli się nie znajduje zmniejsz liczbę szans o 1, na końcu sprawdź czy słowo nie jest kompletne i jeżeli jest to zakończ grę.

74. Kółko i krzyżyk

- a) Stwórz zmienną(listę) która będzie przechowywać dane dla każdej komórki planszy czyli o,x lub pusty napis. A także stwórz zmienną która będzie przechowywać symbol obecnego gracza o lub x
- b) narysuj pustą planszę do gry w konsoli za pomocą funkcji print.
- c) Stwórz główną pętlę gry, która będzie działała aż jeden z graczy nie wygra lub nie skończy się miejsce na planszy(remis).
- d) w pętli poproś gracza o wpisanie pozycji w której chce wstawić swój symbol. wstaw symbol w odpowiednie miejsce listy. następnie wyrysuj planszę z naniesionymi znakami. sprawdź czy gracz nie wygrał lub nie skończyło się miejsce na planszy. i zmień gracza.
- e) poinformuj o zwycięstwie lub remisie i zakończ pętlę

75. Konsolowa symulacja bazy danych (np. baza wydatków, baza studentów z ocenami itp.)

- a) stwórz kolekcję która będzie imitować bazę danych wybierz która będzie ci najbardziej pasować.(lista, słownik, krotka?, zbiór?)
- b) Stwórz pętlę w której zapytasz użytkownika co chce zrobić.np, dodać nowy wpis, usunąć wpis, edytować wpis itp.
- c) następnie w zależności od wcześniejszego wyboru zapytasz o dane do wpisania lub o wiersz do usunięcia lub zmiany.(jak będzie to wyglądać zależy jakiego typu kolekcję wybierzesz.)
- d) po wykonaniu operacji powinniśmy wrócić do 1 menu.
- e) umożliwicie także wyjście z całego programu przez użytkownika np poprzez wpisanie exit lub odpowiedniego wyboru z opcji w 1 menu.

Bonus można przed samym zamknięciem zapisać baze do pliku. (tego jeszcze nie było na zajęciach) jak ktoś by bardzo chciał to zrobić to tutaj jest link jak otwierać pliki i do nich pisać w pythonie: [Operacje na plikach](#)

Funkcje

Moduły

Programowanie obiektowe