

Jakub Trzaskowski – sprawozdanie z zadań z pythona (zjazd 13.04.2024).

Poniżej przedstawiam screenshoty z przeprowadzonych pomyślnie komplikacji skryptów z ćwiczeń. Kody źródłowe znajdują się w poszególnych folderach:

- Python\_Data\_Types\_Dictionary
- Python\_Data\_Type\_List
- Python\_Data\_Type\_String
- Python\_Conditional\_Statements\_and\_loops

W każdym z folderów znajdują się również screenshoty w rozszerzeniu PNG. Z każdego z 4 wybranych przez Pana działów, wykonałem po 20 zadań.

Link do repozytorium github:

Rozwiązań zadań z rozdziału „Python\_Conditional\_Statements\_and\_loops”:

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Copyright: (c) Win_Jump 2024
# Licence: <your licence>
#
#Write a Python program to construct the following pattern, using a nested for loop.
#
#*
#* *
#* * *
#* * * *
#* * * *
#* *
#* *
#*
n = 5

for i in range(n):
    for j in range(i):
        print('* ', end="")
    print('')

for i in range(n, 0, -1):
    for j in range(i):
        print('* ', end="")
    print('')
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py zadanie\_4.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 23:1 Insert

19:43

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Name: module1
# Purpose:
#
# Author: Win_Jump
#
# Created: 13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence: <your Licence>
#
##Write a Python program to guess a number between 1 and 9.
##Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again.
import random
target_num, guess_num = random.randint(1, 10), 0
while target_num != guess_num:
    guess_num = int(input('zgaduj jaka to liczba od 1 do 10. : '))
print('Trafione!!')
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
1505, 1540, 1575, 1610, 1645, 1680, 1715, 1750, 1785, 1820, 1855, 1890, 1925, 1960, 1995, 2030, 2065, 2100, 2135, 2170
, 2205, 2240, 2275, 2310, 2345, 2380, 2415, 2450, 2485, 2520, 2555, 2590, 2625, 2660, 2695
>>> *** Remote Interpreter Reinitialized ***
wprowadz temperaturę, którą chcesz przeliczyć? (np., 45F, 102C itp.) : 45F
temperatura w celsjusz wynosi 7 stopni.
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 23:1 Insert

19:40

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python Interpreter

```
10 #-
11 #
12 #Write a Python program to convert temperatures to and from Celsius and Fahrenheit.
13 #[ Formula : c/5 = f-32/9 [ where c = temperature in celsius and f = temperature in fahrenheit
14 #Expected Output :
15 #60°C is 140 in Fahrenheit
16 #45°F is 7 in Celsius
17
18 temp = input("wprowadz temperaturę, którą chcesz przeliczyć? (np., 45F, 102C itp.) : ")
19
20 degree = int(temp[:-1])
21
22 i_convention = temp[-1]
23
24 if i_convention.upper() == "C":
25     # zamiana z celsusza na fahrenheita
26     result = int(round((9 * degree) / 5 + 32))
27     o_convention = "Fahrenheit"
28 elif i_convention.upper() == "F":
29     # zamiana z fahrenheita na celsjusza
30     result = int(round((degree - 32) * 5 / 9))
31     o_convention = "celsjusz"
32 else:
33     print("niepoprawne.")
34     quit()
35
36 print("temperatura w", o_convention, "wynosi", result, "stopni.")
```

zadanie\_1.py zadanie\_2.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
1505, 1540, 1575, 1610, 1645, 1680, 1715, 1750, 1785, 1820, 1855, 1890, 1925, 1960, 1995, 2030, 2065, 2100, 2135, 2170
, 2205, 2240, 2275, 2310, 2345, 2380, 2415, 2450, 2485, 2520, 2555, 2590, 2625, 2660, 2695
>>>
*** Remote Interpreter Reinitialized ***
wprowadz temperaturę, którą chcesz przeliczyć? (np., 45F, 102C itp.) : 45f
temperatura w celsjusz wynosi 7 stopni.
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready | Python 3.11 (64-bit) | Remote | 32:6 | Insert | 19:39

The screenshot shows the PyScripter IDE interface with the following details:

- Title Bar:** PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...
- Toolbar:** File, Edit, Search, View, Project, Run, Tools, Help.
- Code Editor:** Displays Python code for generating a list of numbers from 1500 to 2701 that are divisible by both 7 and 5. The code uses a for loop with an if condition to filter the range.

```
#-
# Name: module1
# Purpose:
#
# Author: Win_JumP
#
# Created: 13.04.2024
# Copyright: (c) Win_JumP 2024
# Licence: <your licence>
#
#-
#Write a Python program to find those numbers which are divisible by 7 and multiples of 5, between 1500 and 2701:
nl = []
for x in range(1500, 2701):
    if (x % 7 == 0) and (x % 5 == 0):
        nl.append(str(x))
print(','.join(nl))
```

- File Tab:** zadanie\_1.py
- Python Interpreter:** Shows the output of the interpreter for the generated list of numbers.

```
*** Remote Interpreter Reinitialized ***
1505,1540,1575,1610,1645,1680,1715,1750,1785,1820,1855,1890,1925,1960,1995,2030,2065,2100,2135,2170
,2205,2240,2275,2310,2345,2380,2415,2450,2485,2520,2555,2590,2625,2660,2695
>>>
*** Remote Interpreter Reinitialized ***
1505,1540,1575,1610,1645,1680,1715,1750,1785,1820,1855,1890,1925,1960,1995,2030,2065,2100,2135,2170
,2205,2240,2275,2310,2345,2380,2415,2450,2485,2520,2555,2590,2625,2660,2695
>>>
```

- Bottom Navigation:** Call Stack, Variables, Watches, Breakpoints, Output, Messages, Python Interpreter.
- Status Bar:** Python 3.11 (64-bit), Remote, 14:1, Insert.

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

20  
\*  
\* \*\*\*  
\* \*  
\* \*  
\*\*\*  
  
\*\*\*\*  
def print\_G():  
 pattern = [  
 " \*\*\* ",  
 "\* \* ",  
 "\* ",  
 "\* \*\*\* ",  
 "\* \* ",  
 "\* \* ",  
 " \*\*\* "  
 ]  
 for line in pattern:  
 print(line)  
  
40 print\_G()

zadanie\_16.py zadanie\_17.py zadanie\_18.py zadanie\_19.py zadanie\_20.py

Python Interpreter

```
*  
*  
***  
*  
*  
*****  
>>>  
*** Remote Interpreter Reinitialized ***  
***  
* *  
*  
* ***  
* *  
* *  
***  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 40:10 Insert



PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Python Interpreter

20     \* \*  
      \* \*  
      \*\*\*\*  
      ....  
30     def print\_D():  
      pattern = [  
          "\*\*\*\*\* ",  
          "\*    \*",  
          "\*    \*",  
          "\*    \*",  
          "\*    \*",  
          "\*    \*",  
          "\*\*\*\*\* "  
      ]  
      for line in pattern:  
          print(line)  
      # Wywołanie funkcji do wyświetlenia wzorca litery 'D'  
      print\_D()  
39

zadanie\_13.py zadanie\_14.py zadanie\_16.py zadanie\_17.py zadanie\_18.py

Python Interpreter

```
* *  
*****  
* *  
* *  
* *  
  
>>>  
*** Remote Interpreter Reinitialized ***  
*****  
* *  
* *  
* *  
* *  
* *  
*****  
>>>
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

1 \*\*\*  
2 \* \*  
3 \* \*  
4 \*\*\*\*\*  
5 \* \*  
6 \* \*  
7 \* \*  
8  
9 \*\*\*\*  
10 result\_str = ""  
11 for row in range(0, 7):  
12 for column in range(0, 7):  
13 if (((column == 1 or column == 5) and row != 0) or ((row == 0 or row == 3) and (column == 0 or column == 6)) or ((row == 1 or row == 4) and (column == 2 or column == 4)) or ((row == 2 or row == 5) and (column == 3 or column == 5))):  
14 result\_str = result\_str + "\*"  
15 else:  
16 result\_str = result\_str + " "  
17  
18 result\_str = result\_str + "\n"  
19  
20 print(result\_str)

zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_16.py zadanie\_17.py

Python Interpreter

Not a Valid Password  
>>>  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
200,202,204,206,208,220,222,224,226,228,240,242,244,246,248,260,262,264,266,268,280,282,284,286,288  
,400  
>>>  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
\*\*\*  
\* \*  
\* \*  
\*\*\*\*\*  
\* \*  
\* \*  
\* \*  
  
>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 36:18 | Insert |

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
# Created: 24.04.2024
# Copyright: (c) Win_Jump 2024
# Licence: <your Licence>
#-----
#zadanie 16
""" Write a Python program to find numbers between 100 and 400 (both included) where each digit is even.
"""
def all_digits_even(number):
    for digit in str(number):
        if int(digit) % 2 != 0:
            return False
    return True

# 401 w zakresie, gdyz liczenie w programowaniu zaczyna sie od 0 (jest pierwsza cyfra)
result = [str(num) for num in range(100, 401) if all_digits_even(num)]  
print(','.join(result))
```

zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_16.py

Python Interpreter

```
*** Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32. ***
*  
*** Remote Python engine is active ***
>>>  
*** Remote Interpreter Reinitialized ***
wprowadz stringaeoeoeo
litery 6
liczby 0
>>>  
*** Remote Interpreter Reinitialized ***
Input your passwordlolo
Not a Valid Password
>>>  
*** Remote Interpreter Reinitialized ***
200,202,204,206,208,220,222,224,226,228,240,242,244,246,248,260,262,264,266,268,280,282,284,286,288,400
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 25:24 | Insert |

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

30 if (len(p) < 6 or len(p) > 12):  
 break  
elif not re.search("[a-z]", p):  
 break  
elif not re.search("[0-9]", p):  
 break  
elif not re.search("[A-Z]", p):  
 break  
elif not re.search("[#\$@]", p):  
 break  
elif re.search("\s", p):  
 break  
else:  
  
 print("Valid Password")  
 x = False  
 break  
  
if x:  
 print("Not a Valid Password")  
50

zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_16.py

Python Interpreter

```
*** Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32. **  
*  
*** Remote Python engine is active ***  
>>>  
*** Remote Interpreter Reinitialized ***  
wprowadz stringaeoeoeo  
litery 6  
liczby 0  
>>>  
*** Remote Interpreter Reinitialized ***  
Input your passwordlolo  
Not a Valid Password  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 50:1 | Insert |

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Sample Data : Python 3.2  
Expected Output :  
Letters 6  
Digits 2

```
""" Write a Python program that accepts a string and calculates the number of digits and letters.  
Sample Data : Python 3.2  
Expected Output :  
Letters 6  
Digits 2  
"""  
20  
s = input("wprowadz stringa")  
d = l = 0  
26  
for c in s:  
    if c.isdigit():  
        d = d + 1  
    elif c.isalpha():  
        l = l + 1  
    else:  
        pass  
30  
    print("litery", l)  
    print("liczby", d)
```

zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py

Python Interpreter

```
*** Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32. **  
*  
*** Remote Python engine is active ***  
>>>  
*** Remote Interpreter Reinitialized ***  
wprowadz stringaeoeoeo  
litery 6  
liczby 0  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 26:22 Insert

Word Oznaczone gwiazdką ... PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditional...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

# Copyright: (c) Win\_Jump 2024  
# Licence: <your licence>  
#-----  
#Write a Python program that accepts a sequence of comma separated 4 digit binary numbers as input  
#Sample Data : 0100,0011,1010,1001,1100,1001  
#Expected Output : 1010

```
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20 for p in num:  
21     x = int(p, 2)  
22     if not x % 5:  
23         items.append(p)  
24  
25 print(','.join(items))
```

zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py

Python Interpreter

```
ValueError: invalid literal for int() with base 2: 'eoeo'  
>>>  
*** Remote Interpreter Reinitialized ***  
2  
Traceback (most recent call last):  
  File "C:\Users\kubit\Desktop\szkolne\studia\semestr_2\podstawy_programowania_s2  
\Python_Conditional_Statements_and_loops\zadanie_13.py", line 21, in <module>  
    x = int(p, 2)  
    ^^^^^^^^  
ValueError: invalid literal for int() with base 2: '2'  
>>>  
*** Remote Interpreter Reinitialized ***  
100  
  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 21:1 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditional...

The screenshot shows the PyScripter IDE interface with the following details:

- Title Bar:** PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...
- Toolbar:** File, Edit, Search, View, Project, Run, Tools, Help.
- Code Editor:** Displays Python code for reading lines from input and printing them in uppercase. The code includes comments about the creation date (13.04.2024), copyright ((c) Win\_Jump 2024), and license (<your licence>). It uses a while loop to read lines until an empty line is entered, then prints each line in uppercase.
- File List:** Shows tabs for zadanie\_8.py, zadanie\_9.py, zadanie\_10.py, zadanie\_11.py, and zadanie\_12.py (highlighted).
- Python Interpreter:** Shows the output of running the code. It prompts for the number of columns (wprowadz ilosc kolumn: 4) and then lists four lists of length 3. It then reinitializes the interpreter and prints nine '3' characters.
- Bottom Navigation:** Call Stack, Variables, Watches, Breakpoints, Output, Messages, Python Interpreter (highlighted).
- Status Bar:** Python 3.11 (64-bit), Remote, 15:1, Insert.
- System Tray:** Icons for battery, signal, volume, and clock (19:57).

The screenshot shows the PyScripter IDE interface with the following details:

- Title Bar:** PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona... (partially visible)
- Menu Bar:** File, Edit, Search, View, Project, Run, Tools, Help
- Toolbar:** Includes icons for Open, Save, Find, Replace, Run, Stop, and others.
- Code Editor:** Displays Python code for generating a multiplication table. The code uses nested loops to create a 2D list where each element is the product of its row and column indices. It includes comments for notes and test data.
- File List:** Shows open files: zadanie\_8.py, zadanie\_9.py, zadanie\_10.py, zadanie\_11.py (highlighted in orange), and module1.
- Python Interpreter:** Shows the output of a fizzbuzz interpreter and a remote interpreter reinitialization. The user inputs 5 rows and 4 columns, resulting in a 5x4 multiplication table.
- Bottom Navigation:** Call Stack, Variables, Watches, Breakpoints, Output, Messages, Python Interpreter (highlighted in orange).
- Status Bar:** Ready, Python 3.11 (64-bit), Remote, 26:31, Insert.
- System Tray:** Shows icons for battery, signal, and time (19:57).

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python FizzBuzz

```
· # Created: 13.04.2024
· # Copyright: (c) Win_Jump 2024
· # Licence: <your Licence>
10 #-----
·
13 #Write a Python program that iterates the integers from 1 to 50. For multiples of three print
#Sample Output :
· #fizzbuzz
· #1
· #2
· #fizz
· #4
· #buzz
20 · for fizzbuzz in range(51):
·     if fizzbuzz % 3 == 0 and fizzbuzz % 5 == 0:
·         print("fizzbuzz")
·         continue
·     elif fizzbuzz % 3 == 0:
·         print("fizz")
·         continue
·     else:
·         print(fizzbuzz)
zadanie_6.py zadanie_7.py zadanie_8.py zadanie_9.py zadanie_10.py
```

Python Interpreter

```
17
fizz
19
buzz
fizz
22
23
fizz
buzz
26
fizz
28
29
fizzbuzz
31
32
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 13:17 Insert

Exercise: Get Fi... PyScripter - C:\Users\k... 19:54

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

#-  
# Name: module1  
# Purpose:  
#  
# Author: Win\_Jump  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_Jump 2024  
# Licence: <your licence>  
#-  
#Write a Python program to get the Fibonacci series between 0 and 50.  
#Note : The Fibonacci Sequence is the series of numbers :  
#0, 1, 1, 2, 3, 5, 8, 13, 21, ....  
#Every next number is found by adding up the two numbers before it.  
#Expected Output : 1 1 2 3 5 8 13 21 34  
x, y = 0, 1  
while y < 50:  
 print(y)  
 x, y = y, x + y

zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***  
1  
1  
2  
3  
5  
8  
13  
21  
34  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 23:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

#-  
# Name: module1  
# Purpose:  
#  
# Author: Win\_Jump  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_Jump 2024  
# Licence: <your Licence>  
#-  
#Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.  
#Note : Use 'continue' statement.  
#Expected Output : 0 1 2 4 5  
for x in range(6):  
 if (x == 3 or x == 6):  
 continue  
 print(x, end=' ')  
print("\n")

zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py

Python Interpreter

```
Type of (0, -1) is <class 'tuple'>
Type of [5, 12] is <class 'list'>
Type of {'class': 'V', 'section': 'A'} is <class 'dict'>
>>>
*** Remote Interpreter Reinitialized ***
0 1 2 4 5

>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Run active module Python 3.11 (64-bit) Remote 21:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python Interpreter

```
10
11  #--#
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:    13.04.2024
18  # Copyright:  (c) Win_JumP 2024
19  # Licence:   <your licence>
20  #--#
21
22  #Write a Python program that prints each item and its corresponding type from the following List
23  #Sample List : dataList = [1452, 11.23, 1+2j, True, 'w3resource', (0, -1), [5, 12], {"class":'V'
24  dataList = [1452, 11.23, 1+2j, True, 'w3resource', (0, -1), [5, 12], {"class": 'V', "section": 'A'}
25
26  for item in dataList:
27      print("Type of", item, "is", type(item))
28
```

zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py

Type of 11.23 is <class 'float'>  
Type of (1+2j) is <class 'complex'>  
Type of True is <class 'bool'>  
Type of w3resource is <class 'str'>  
Type of (0, -1) is <class 'tuple'>  
Type of [5, 12] is <class 'list'>  
Type of {'class': 'V', 'section': 'A'} is <class 'dict'>  
>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 18:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Purpose:  
#  
# Author:      Win_JumP  
#  
# Created:     13.04.2024  
# Copyright:   (c) Win_JumP 2024  
# Licence:     <your licence>  
#-----  
#Write a Python program to count the number of even and odd numbers in a series of numbers  
#Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)  
#Expected Output :  
#Number of even numbers : 5  
#Number of odd numbers : 4  
  
numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)  
  
count_odd = 0  
count_even = 0  
  
for x in numbers:  
    if not x % 2:  
        count_even += 1  
    else:  
        count_odd += 1  
  
print("liczba parzysta:", count_even)  
print("liczba nieparzysta:", count_odd)
```

zadanie\_2.py zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 23:18 | Insert | 19:47

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Conditiona...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
#-
# Name: module1
# Purpose:
#
# Author: Win_Jump
#
# Created: 13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence: <your licence>
#-

#Write a Python program that accepts a word from the user and reverses it.
word = input("wprowadz slowo do odrocenia jego kolejnosci: ")

for char in range(len(word) - 1, -1, -1):
    print(word[char], end="")

print("\n")
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py zadanie\_4.py zadanie\_5.py

Python Interpreter

```
* *
* 
>>>
*** Remote Interpreter Reinitialized ***
wprowadz slowo do odrocenia jego kolejnosci: kuba
abuk

>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 19:1 Insert

Users\k...

Rozwiązań z działu „Python\_Data\_Type \_String”:

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 16:1 Insert

atnik PyScripter - C:\Users\k...

```
1 #-
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #-
11
12 ##Write a Python function to reverse a string if its length is a multiple of 4.
13 def reverse_string(str1):
14     if len(str1) % 4 == 0:
15         return ''.join(reversed(str1))
16     else:
17         return str1
18
19 print(reverse_string('abcd'))
20 print(reverse_string('python'))
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:    13.04.2024
# Copyright:  (c) Win_Jump 2024
# Licence:   <your licence>
#-----#
#
##Write a Python program to get the last part of a string before a specified character.
#https://www.w3resource.com/python-exercises
#https://www.w3resource.com/python
str1 = 'https://www.w3resource.com/python-exercises/string'
#
# Użyj metody rsplit() z '/' jako separatorem, aby podzielić ciąg od prawej strony,
# a [0] oznacza część przed ostatnim znakiem '/'. Następnie wydrukuj wynik.
print(str1.rsplit('/', 1)[0])
#
print(str1.rsplit('-', 1)[0])
```

zadanie\_15.py zadanie\_16.py zadanie\_17.py zadanie\_18.py zadanie\_19.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
https://www.w3resource.com/python-exercises
https://www.w3resource.com/python
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 20:1 Insert

atnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #--#
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your licence>
10 #
11
12 #Write a Python function to get a string made of the first three characters of a specified string
13 #Sample function and result :
14 def first_three(str):
15     if len(str) > 3:
16         return str[:3]
17     else:
18         return str
19
20 print(first_three('ipy'))
21 print(first_three('python'))
22 print(first_three('py'))
```

zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py zadanie\_18.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
ipy
pyt
py
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 19:1 Insert

atnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

10: #-----  
# Name: module1  
# Purpose:  
#  
# Author: Win\_JumP  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_JumP 2024  
# Licence: <your Licence>  
#-----  
#Write a Python function to get a string made of 4 copies of the last two characters of a specific string.  
#Sample function and result :  
#insert\_end('Python') -> onononon  
#insert\_end('Exercises') -> eseseses  
def insert\_end(str):  
 # Wyodrębnij ostatnie dwa znaki z ciągu wejściowego 'str' i zapisz je w zmiennej 'sub\_str'.  
 sub\_str = str[-2:]  
 return sub\_str \* 4  
# Wywołaj funkcję insert\_end z różnymi ciągami wejściowymi i wydrukuj wyniki.  
print(insert\_end('Python'))  
print(insert\_end('Exercises'))

zadanie\_13.py zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py

Python Interpreter

>>>  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
onononon  
eseseses  
>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 21:1 | Insert |

tnik PyScripter - C:\Users\k... 11:50

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 23:47 | Insert

Notepad 11:46

```
1 #--#
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:    <your licence>
10#--#
11
12#Write a Python function to insert a string in the middle of a string.
13#Sample function and result :
14#insert_string_middle('[]<>>', 'Python') -> [[Python]]
15#insert_string_middle('{}{}', 'PHP') -> {{PHP}}
16# Zdefiniuj funkcję o nazwie insert_string_middle, która przyjmuje dwa argumenty: 'str' (ciąg znaków) i 'word'.
17def insert_string_middle(str, word):
18    # Utwórz i zwróć nowy ciąg znaków poprzez konkatenerację pierwszych dwóch znaków 'str',
19    # a następnie 'word', i na końcu pozostałych znaków 'str' od trzeciego znaku.
20    return str[:2] + word + str[2:]
21
22# Wywołaj funkcję insert_string_middle z różnymi ciągami wejściowymi i słowami, a następnie wydrukuj wynik.
23print(insert_string_middle('[]', 'Python')) |
24print(insert_string_middle('{}', 'PHP')) |
25print(insert_string_middle('<>>', 'HTML'))
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_15.py

```
1  #-----
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:    <your licence>
10 #-----
11
12 # Write a Python function to create an HTML string with tags around the word(s).
13 # Sample function and result :
14 #add_tags('i', 'Python') -> '<i>Python</i>'
15 #add_tags('b', 'Python Tutorial') -> '<b>Python Tutorial </b>'
16 def add_tags(tag, word):
17     return "<%s>%s</%s>" % (tag, word, tag)
18
19 print(add_tags('i', 'Python'))
20
21 print(add_tags('b', 'Python Tutorial'))
```

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
<i>Python</i>
<b>Python Tutorial</b>
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 8:33 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python Interpreter

```
10
11  #-----
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:   13.04.2024
18  # Copyright: (c) Win_JumP 2024
19  # Licence:   <your licence>
20  #-----
21
22  ##Write a Python program that accepts a comma-separated sequence of words as input and prints the
23  #Sample Words : red, white, black, red, green, black
24  #Expected Result : black, green, red, white,red
25  # Poproś użytkownika o wprowadzenie ciągu słów oddzielonych przecinkami i zapisz go w zmiennej
26  items = input("napisz cos korzystajac z przecinkow ")
27
28  words = [word.strip() for word in items.split(",")]
29
30  unique_words = list(set(words))
31
32  sorted_words = sorted(unique_words)
33  result = ",".join(sorted_words)
34  print(result)
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Notepad PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ... 11:37

```
10
11  #-----
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:    13.04.2024
18  # Copyright:  (c) Win_JumP 2024
19  # Licence:    <your licence>
20  #-----
```

# Write a Python script that takes input from the user and displays that input back in upper and lower case.  
# Poproś użytkownika o podanie ulubionego języka i zapisz odpowiedź w zmiennej 'user\_input'.  
user\_input = input("twoj najulubienszy język to? ")  
  
# Wydrukuj wiadomość "My favorite Language is" po czym dodaj odpowiedź użytkownika z konwersją r  
print("moj najulubienszy język to " + user\_input.upper())  
  
# Wydrukuj wiadomość "My favorite Language is" po czym dodaj odpowiedź użytkownika z konwersją l  
print("moj najulubienszy język to " + user\_input.lower())

zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
twoj najulubienszy język to? c#
moj najulubienszy język to C#
moj najulubienszy język to  c#
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 19:47 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
10
11  # -----
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:   13.04.2024
18  # Copyright: (c) Win_JumP 2024
19  # Licence:   <your licence>
20  #-----
21
22  #. Write a Python program to count the occurrences of each word in a given sentence.
23 def word_count(str):
24     counts = dict()
25     words = str.split()
26
27     for word in words:
28         if word in counts:
29             counts[word] += 1
30         else:
31             counts[word] = 1
32
33     return counts
34
35 print(word_count('Trzaskowski Jakub testowanie zadanka eoeoeo'))
```

zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py

Python Interpreter

```
tzsosiauts
>>>
*** Remote Interpreter Reinitialized ***
{'Trzaskowski': 1, 'Jakub': 1, 'testowanie': 1, 'zadanka': 1, 'eoeoeo': 1}
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 10:78 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:   <your Licence>
#-----#
#
##Write a Python program to remove characters that have odd index values in a given string.
# funkcja przyjmująca 'str' ciąg znaków
def odd_values_string(str):
    result = ""

    # Iteruj przez indeksy (i) znaków w ciągu wejściowym 'str'.
    for i in range(len(str)):
        # Sprawdź, czy indeks (i) jest parzysty (czyli ma resztę 0 przy dzieleniu przez 2).
        if i % 2 == 0:
            # Jeśli indeks jest parzysty, dodaj znak na tym indeksie do ciągu 'result'.
            result = result + str[i]

    return result
print(odd_values_string('trzaskowski'))
print(odd_values_string('trzaskowskijakubtest'))
```

zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
tzsosi
tzsosiauts
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 25:1 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

10

```
-----  
# Name: module1  
# Purpose:  
#  
# Author: Win_JumP  
#  
# Created: 13.04.2024  
# Copyright: (c) Win_JumP 2024  
# Licence: <your Licence>  
#-----  
  
#Write a Python program to change a given string to a newly string where the first and last character are swapped.  
# zdefiniowanie funkcji, przejmującej 'str1' (ciąg znaków).  
def change_string(str1):  
    # zwrócenie 'str1', przestawiając jego ostatni znak na pierwszą pozycję  
    return str1[-1:] + str1[1:-1] + str1[:1]  
  
# Wywołanie funkcji  
print(change_string('kubatrzaskowski'))  
print(change_string('123123'))
```

zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py

Python Interpreter

```
>>>  
*** Remote Interpreter Reinitialized ***  
iubatrzaskowskk  
323121  
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 20:31 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
10
11  # ---
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:   13.04.2024
18  # Copyright: (c) Win_JumP 2024
19  # Licence:   <your Licence>
20  #---
21
22  # Write a Python program to remove the nth index character from a nonempty string.
23  # zdefiniowanie funkcji przyjmującej 'str' (ciąg znaków) i 'n' (indeks znaku do usunięcia).
24  def remove_char(str, n):
25      # Utwórz nowy ciąg znaków 'first_part', który zawiera wszystkie znaki od początku 'str' do indeksu 'n'.
26      first_part = str[:n]
27
28      # Utwórz nowy ciąg znaków 'last_part', który zawiera wszystkie znaki od znaku o indeksie 'n' do końca 'str'.
29      last_part = str[n+1:]
30
31      # zwrocenie
32      return first_part + last_part
33
34  # wywołanie funkcji i jej wypisanie.
35  print(remove_char('Python', 0))
36  print(remove_char('Python', 3))
37  print(remove_char('Python', 5))
```

zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 25:33 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
# Copyright: (c) Win_Jump 2024
# Licence: <your Licence>
#-----#
##Write a Python function that takes a List of words and return the longest word and the length.
##Sample Output:
##Longest word: Exercises
##Length of the Longest word: 9
## zdefiniowanie funkcji
def find_longest_word(words_list):
    # Utwórz pustą listę 'word_len' do przechowywania par (długość słowa, słowo).
    word_len = []

    # Iteruj przez każde słowo 'n' w liście 'words_list'.
    for n in words_list:
        # Dodaj krotkę zawierającą długość słowa i samo słowo do listy 'word_len'.
        word_len.append((len(n), n))

    # Posortuj listę 'word_len' na podstawie długości słów (rosnąco).
    word_len.sort()

    # zwrocenie
    return word_len[-1][0], word_len[-1][1]

## wywołanie funkcji i zapisanie wyniku
result = find_longest_word(["PHP", "Exercises", "Backend"])

## wypisanie słowa i jego długości
print("\nNajdłuższe słowo: ", result[1])
print("Długość najdłuższego słowa: ", result[0])
```

zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***

Najdłuższe słowo: Exercises
Długość najdłuższego słowa: 9
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 20:1 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

10 #-----

# Write a Python program to find the first appearance of the substrings 'not' and 'poor' in a given string. If 'not' appears before 'poor', replace 'not' with 'good'. Otherwise, leave the string as it is.

# Sample String : 'The lyrics is not that poor!'

# 'The lyrics is poor!'

# Expected Result : 'The lyrics is good!'

# 'The lyrics is poor!'

# definiowanie funkcji przyjmujacej 'str1'.

def not\_poor(str1):

# Znajdź indeks podcięgu 'not' w ciągu wejściowym 'str1' i przechowaj go w 'snot'.

snot = str1.find('not')

# Znajdź indeks podcięgu 'poor' w ciągu wejściowym 'str1' i przechowaj go w 'spoor'.

spoor = str1.find('poor')

# Sprawdź, czy 'poor' występuje po 'not' i zarówno 'not', jak i 'poor' są obecne w ciągu.

if spoor > snot and snot > 0 and spoor > 0:

# Zastąp podcięg od 'not' do 'spoor+4' (włącznie) podcięgiem 'good'.

str1 = str1.replace(str1[snot:(spoor+4)], 'good')

return str1

else:

# Jeśli warunki nie są spełnione, zwróć oryginalny ciąg 'str1'.

return str1

# wywołanie i wypisanie funkcji

print(not\_poor('The lyrics is not that poor!')) # Wynik: 'The lyrics is good!'

print(not\_poor('The lyrics is poor!')) # Wynik: 'The lyrics is poor!'

zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py

Python Interpreter

>>>

\*\*\* Remote Interpreter Reinitialized \*\*\*

The lyrics is good!

The lyrics is poor!

>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 30:10 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k... 11:21

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Copyright: (c) Win_Jump 2024
# Licence: <your licence>
#-----
#Write a Python program to add 'ing' at the end of a given string (Length should be at Least 3).
#Sample String : 'abc'
#Expected Result : 'abcing'
#Sample String : 'string'
#Expected Result : 'stringly'
# definiowanie funkcji, przyjmujacej 'str1'.
def add_string(str1):
    # Pobierz dlugosc ciagu wejsciowego 'str1' i przechowaj ją w zmiennej 'length'.
    length = len(str1)

    # Sprawdz, czy dlugosc 'str1' jest wieksza niz 2 znaki.
    if length > 2:
        # Jesli ostatnie trzy znaki 'str1' to 'ing', dodaj 'ly' na koncu.
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else:
            # dodanie inga na koniec.
            str1 += 'ing'

    # zwrocenie
    return str1

# Wywolanie i wypisanie funkcji
print(add_string('ab'))
print(add_string('abc'))
print(add_string('string'))
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

10

```
1. #-----
2. # Name:      module1
3. # Purpose:
4. #
5. # Author:     Win_JumP
6. #
7. # Created:   13.04.2024
8. # Copyright: (c) Win_JumP 2024
9. # Licence:   <your licence>
10. #-----
11.
12. ##Write a Python program to get a single string from two given strings, separated by a space and
13. #Sample String : 'abc', 'xyz'
14. #Expected Result : 'xyc abz'
15. # definiowanie funkcji chars_mix_up, która przyjmuje dwa argumenty 'a' i 'b'.
16. def chars_mix_up(a, b):
17.     # Utwórz nowy ciąg 'new_a', biorąc pierwsze dwa znaki z 'b' i łącząc je
18.     # z pozostałymi znakami z 'a' zaczynając od trzeciego znaku.
19.     new_a = b[:2] + a[2:]
20.
21.     # Utwórz nowy ciąg 'new_b', biorąc pierwsze dwa znaki z 'a' i łącząc je
22.     # z pozostałymi znakami z 'b' zaczynając od trzeciego znaku.
23.     new_b = a[:2] + b[2:]
24.
25.     # Połącz 'new_a', spację i 'new_b', aby utworzyć pojedynczy ciąg.
26.     return new_a + ' ' + new_b
27.
28. #wywołanie i wypisanie funkcji
29. print(chars_mix_up('abc', 'xyz')) # Wynik: 'xyc abz'
```

zadanie\_2.py zadanie\_1.py zadanie\_3.py zadanie\_4.py zadanie\_5.py

Python Interpreter

```
resta$t
>>>
*** Remote Interpreter Reinitialized ***
xyc abz
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 16:24 Insert

atnik PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Name: module1
# Purpose:
#
# Author: Win_JumP
#
# Created: 13.04.2024
# Copyright: (c) Win_JumP 2024
# Licence: <your Licence>
10 #-----

#Write a Python program to get a string from a given string where all occurrences of its first character are '$'.
#Sample String : 'restart'
#Expected Result : 'resta$t'
# Definiowanie funkcji change_char, przyjmującej 'str1'.
def change_char(str1):
    # pobranie i przechowanie pierwszego znaku.
    char = str1[0]

    # zamiana wystąpień znaku 'char' na '$' w ciągu 'str1'.
    str1 = str1.replace(char, '$')

    # rekonstrukcja ciągu 'str1', umieszczając oryginalny znak 'char' jako pierwszy znak,
    # a następnie zmodyfikowany ciąg rozpoczynający się od drugiego znaku.
    str1 = char + str1[1:]

    # zwrocenie
    return str1

#wywylanie funkcji i wypisanie wyniku
print(change_char('restart')) # Wynik: 'resta$t'
```

zadanie\_2.py zadanie\_1.py zadanie\_3.py zadanie\_4.py

Python Interpreter

```
resta$t
>>>
*** Remote Interpreter Reinitialized ***
resta$t
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 23:45 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:    13.04.2024
# Copyright:  (c) Win_Jump 2024
# Licence:   <your licence>
#-----#
#
#Write a Python program to get a string made of the first 2 and last 2 characters of a given string.
#Sample String : 'w3resource'
#Expected Result : 'w3ce'
#Sample String : 'w3'
#Expected Result : 'w3w3'
#Sample String : ' '
#Expected Result : Empty String
# Zdefiniowanie funkcji string_both_ends, przyjmującej argument, 'str'.
def string_both_ends(str):
    # Sprawdzenie czy długość ciągu wejściowego 'str' jest mniejsza niż 2 znaki.
    if len(str) < 2:
        # Jeżeli ciąg jest krótszy niż 2 znaki, zwróć pusty串 of znaków.
        return ''

    # Jeżeli ciąg ma co najmniej 2 znaki, połącz pierwsze dwa znaki
    # i ostatnie dwa znaki ciągu wejściowego, a następnie zwróć wynik.
    return str[0:2] + str[-2:]

# wywołania funkcji
print(string_both_ends('w3resource')) # Wynik: 'w3ce'
print(string_both_ends('w3'))         # Wynik: 'w3w3'
```

zadanie\_2.py zadanie\_1.py zadanie\_3.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
w3ce
w3w3

>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 12:34 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k... 11:02

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Purpose:  
#  
# Author:      Win_Jump  
#  
# Created:    13.04.2024  
# Copyright:  (c) Win_Jump 2024  
# Licence:    <your Licence>  
#-----  
#Write a Python program to count the number of characters (character frequency) in a string.  
#Sample String : google.com'  
#Expected Result : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}  
# Zdefiniuj funkcję o nazwie char_frequency, która przyjmuje jeden argument, str1.  
def char_frequency(str1):  
    # Zainicjuj pusty słownik o nazwie 'dict' do przechowywania częstości występowania znaków.  
    dict = {}  
  
    # Iteruj przez każdy znak 'n' w ciągu wejściowym str1.  
    for n in str1:  
        # Pobierz klucze (unikalne znaki) w słowniku 'dict'.  
        keys = dict.keys()  
  
        # Sprawdź, czy znak 'n' jest już kluczem w słowniku.  
        if n in keys:  
            # Jeśli 'n' jest już kluczem, zwiększ jego wartość (częstość) o 1.  
            dict[n] += 1  
        else:  
            # Jeśli 'n' nie jest kluczem, dodaj go do słownika z częstością 1.  
            dict[n] = 1  
  
    # Zwróć słownik zawierający częstość występowania każdego znaku w ciągu wejściowym.  
    return dict  
  
# Wywołaj funkcję char_frequency z argumentem 'google.com' i wydrukuj wynik.  
print(char_frequency('google.com'))
```

zadanie\_2.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***  
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}  
=>>> *** Remote Interpreter Reinitialized ***  
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 35:1 | Insert | 10:56

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type ...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #--#
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:   <your Licence>
10#
11#
12#1. Write a Python program to calculate the length of a string.
13# Zdefiniuj funkcję o nazwie string_length, która przyjmuje jeden argument, str1.
14def string_length(str1):
15    # Zainicjuj zmienną count na 0, aby śledzić długość ciągu znaków.
16    count = 0
17
18    # Iteruj przez każdy znak w ciągu wejściowym str1.
19    for char in str1:
20        # Dla każdego napotkanego znaku zwiększ count o 1.
21        count += 1
22
23    # Zwróć ostateczny count, który reprezentuje długość ciągu wejściowego.
24    return count
25
26
27# Wywołaj funkcję string_length z argumentem 'w3resource.com' i wydrukuj wynik.
28print(string_length('TrzaskowskiJakub'))
```

zadanie\_2.py zadanie\_1.py

Python Interpreter

```
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}
>>>
*** Remote Interpreter Reinitialized ***
16
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 23:76 Insert

\*Bez tytułu — Notatnik PyScripter - C:\Users\k...

Rozwiązań z działu „Python\_Data\_Type\_List”:

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #-
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:   13.04.2024
8 # Copyright: (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #-
11
12 #Write a Python program to calculate the difference between the two lists.
13 list1 = [1, 3, 5, 7, 9]
14 list2 = [1, 2, 4, 6, 7, 8]
15
16 diff_list1_list2 = list(set(list1) - set(list2))
17 diff_list2_list1 = list(set(list2) - set(list1))
18 total_diff = diff_list1_list2 + diff_list2_list1
19
20 print(total_diff)
```

zadanie\_16.py zadanie\_17.py zadanie\_18.py zadanie\_19.py zadanie\_20.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
0 10
1 25
2 35
3 418
4 98
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 9:30 | Insert | 16:56

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

#-----  
# Name: module1  
# Purpose:  
#  
# Author: Win\_Jump  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_Jump 2024  
# Licence: <your licence>  
#-----  
  
##Write a Python program to calculate the difference between the two lists.  
list1 = [1, 3, 5, 7, 9]  
list2 = [1, 2, 4, 6, 7, 8]  
  
diff\_list1\_list2 = list(set(list1) - set(list2))  
diff\_list2\_list1 = list(set(list2) - set(list1))  
total\_diff = diff\_list1\_list2 + diff\_list2\_list1  
  
print(total\_diff)

zadanie\_15.py zadanie\_16.py zadanie\_17.py zadanie\_18.py zadanie\_19.py

Python Interpreter

```
[4, 5, 6), (4, 6, 5), (5, 4, 6), (5, 6, 4), (6, 4, 5), (6, 5, 4)]  
=>>>  
*** Remote Interpreter Reinitialized ***  
[9, 3, 5, 8, 2, 4, 6]  
=>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 13:14 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py zadanie\_18.py

```
10
11  #-----
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:   13.04.2024
18  # Copyright: (c) Win_JumP 2024
19  # Licence:   <your licence>
20  #-----
21
22  # Write a Python program to generate all permutations of a List in Python.
23  import itertools
24
25  print(list(itertools.permutations([4, 5, 6])))
```

Python Interpreter

```
[4, 5, 6], [4, 6, 5], [5, 4, 6], [5, 6, 4], [6, 4, 5], [6, 5, 4]
>>>
*** Remote Interpreter Reinitialized ***
[4, 5, 6], [4, 6, 5], [5, 4, 6], [5, 6, 4], [6, 4, 5], [6, 5, 4]
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 15:43 Insert

PyScripter - C:\Users\k... 16:53

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

#Write a Python program to check if each number is prime in a given List of numbers. Return True or False  
#Sample Data:  
#[0, 3, 4, 7, 9] -> False  
#[3, 5, 7, 13] -> True  
#[1, 5, 3] -> False

```
def test(nums):
    return all(is_prime(i) for i in nums)

def is_prime(n):
    if n == 1:
        return False
    elif n == 2:
        return True
    else:
        for x in range(2, n):
            if n % x == 0:
                return False
        return True

nums = [0, 3, 4, 7, 9]
```

zadanie\_13.py zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
Original list of numbers:
[0, 3, 4, 7, 9]
Check if each number is prime in the said list of numbers:
False

Original list of numbers:
[3, 5, 7, 13]
Check if each number is prime in the said list of numbers:
True

Original list of numbers:
[1, 5, 3]
Check if each number is prime in the said list of numbers:
False
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 11:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #-----
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:   13.04.2024
8  # Copyright: (c) Win_JumP 2024
9  # Licence:   <your Licence>
10 #
11
12 # Write a Python program to generate and print a list of the first and last 5 elements where the
13 # Define a function named printValues
14 def printValues():
15     l = list()
16     for i in range(1, 21):
17         l.append(i**2)
18     print(l[:5])
19     print(l[-5:])
20
21 printValues()
```

zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_15.py zadanie\_16.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
[1, 4, 9, 16, 25]
[256, 289, 324, 361, 400]
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 15:15 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:   13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence:   <your licence>
#-----#
#15. Write a Python program to shuffle and print a specified list.
from random import shuffle

color = ['zielony', 'czerwony', 'czarny', 'biały', 'żółty', 'pomaranczowy']

shuffle(color)

print(color)
```

zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_15.py

Python Interpreter

```
[5, 3, 21]
>>>
*** Remote Interpreter Reinitialized ***
['biały', 'czerwony', 'zielony', 'pomaranczowy', 'czarny', 'żółty']
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Run active module | Python 3.11 (64-bit) | Remote | 15:74 | Insert | | |

PyScripter - C:\Users\kub...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#-
# Created:   13.04.2024
# Copyright: (c) Win_JumP 2024
# Licence:   <your licence>
#-
# Write a Python program to print the numbers of a specified list after removing even numbers from it.
num = [5, 3, 12, 21, 34, 50, 64]

num = [x for x in num if x % 2 != 0]
print(num)
```

zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py

Python Interpreter

```
[5, 3, 21]
>>>
*** Remote Interpreter Reinitialized ***
[5, 3, 21]
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 6:2 Insert

PyScripter - C:\Users\kub...



PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:   13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence:   <your licence>
#-----
#Write a Python program to print a specified List after removing the 0th, 4th and 5th elements.
#Sample List : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
#Expected Output : ['Green', 'White', 'Black']
# Create a list 'color' with several color strings
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

color = [x for i, x in enumerate(color) if i not in (0, 4, 5)]
print(color)
```

zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py

Python Interpreter

```
None
>>>
*** Remote Interpreter Reinitialized ***
['Green', 'White', 'Black']
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 16:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #--#
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your licence>
10 #
11
12 # Write a Python function that takes two lists and returns True if they have at least one common
13 # Define a function called 'common_data' that takes two lists, 'list1' and 'list2', as input
14 def common_data(list1, list2):
15     result = False
16
17     for x in list1:
18         for y in list2:
19             if x == y:
20                 result = True
21                 return result
22
23 print(common_data([1, 2, 3, 4, 5], [5, 6, 7, 8, 9]))
24 print(common_data([1, 2, 3, 4, 5], [6, 7, 8, 9]))
```

zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 21:30 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:    <your Licence>
#-----#
#
# Write a Python program to find the List of words that are Longer than n from a given List of words
def long_words(n, str):
    word_len = []

    txt = str.split(" ")

    for x in txt:
        if len(x) > n:
            word_len.append(x)

    return word_len

print(long_words(3, "Trzaskowski Jakub test apki eoe ed"))
```

zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py

Ready Python 3.11 (64-bit) Remote 24:56 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:   13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence:   <your licence>
#-
#
#Write a Python program to clone or copy a list.
original_list = [23, 31, 44, 13, 100]

new_list = list(original_list)

print(original_list)

print(new_list)
```

zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready | Python 3.11 (64-bit) | Remote | 13:37 | Insert | 16:40

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:   13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence:   <your licence>
#-----#
#
#Write a Python program to check if a List is empty or not.
l = []
if not l:
    print("List is empty")

```

zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py

{40, 10, 80, 50, 20, 60, 30}  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
List is empty

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 17:1 Insert

PyScripter - C:\Users\kub...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Open Save Print Find Replace Go Back Forward Search Run Stop Run All Run Selection Run Line Python Help

```
1  #-----
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:   13.04.2024
8  # Copyright: (c) Win_JumP 2024
9  # Licence:   <your Licence>
10 #-----
11
12 # Write a Python program to remove duplicates from a list.
13 a = [10, 20, 30, 20, 10, 50, 60, 40, 80, 50, 40]
14
15 dup_items = set()
16 uniq_items = []
17
18 for x in a:
19     if x not in dup_items:
20         uniq_items.append(x)
21         dup_items.add(x)
22
23 print(dup_items)
24
```

zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py

Python Interpreter

```
[2, 1], [1, 2], [2, 3], [4, 4], [2, 5]
>>>
*** Remote Interpreter Reinitialized ***
{40, 10, 80, 50, 20, 60, 30}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 24:1 Insert

PyScripter - C:\Users\k... 16:37

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module2
# Purpose:
#
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:   <your Licence>
#-----#
#
##Write a Python program to get a List, sorted in increasing order by the last element in each tuple
#Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]
#Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]
def last(n):
    return n[-1]

def sort_list_last(tuples):
    return sorted(tuples, key=last)

print(sort_list_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))
```

zadanie\_2.py zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py

Python Interpreter

```
2
>>>
*** Remote Interpreter Reinitialized ***
[(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 22:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 22:50 Insert

PyScripter - C:\Users\k...

```
1 #-
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #-
11
12 ##Write a Python program to count the number of strings from a given List of strings. The string
13 #Sample List : ['abc', 'xyz', 'aba', '1221']
14 #Expected Result : 2
15 def match_words(words):
16     ctr = 0
17     for word in words:
18         if len(word) > 1 and word[0] == word[-1]:
19             ctr += 1
20     return ctr
21
22 print(match_words(['abc', 'xyz', 'aba', '1221']))
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#
# Created:   13.04.2024
# Copyright: (c) Win_JumP 2024
# Licence:   <your licence>
#-
#
#Write a Python program to get the smallest number from a list.
def smallest_num_in_list(list):
    min = list[0]
    for a in list:
        if a < min:
            min = a
    return min

print(smallest_num_in_list([3, 5, -5, 10]))
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py zadanie\_4.py

Python 3.11 (64-bit) | Remote | 20:40 | Insert | 16:33

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 11:1 Insert

PyScripter - C:\Users\k...

```
1 #-----
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:   13.04.2024
8 # Copyright: (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #-----
11
12 ##Write a Python program to get the largest number from a List.
13 def max_num_in_list(list):
14     max = list[0]
15     for a in list:
16         if a > max:
17             max = a
18     return max
19
20 print(max_num_in_list([1, 2, -8, 0]))
```

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

```
1 #-----#
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:   13.04.2024
8 # Copyright: (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #-----
11
12 ##Write a Python program to multiply all the items in a List.
13 def multiply_list(items):
14     tot = 1
15     for x in items:
16         tot *= x
17     return tot
18
19 #wywolanie funkcji
20 print(multiply_list([4, 22, -8]))
```

zadanie\_1.py zadanie\_2.py

Python Interpreter

```
-16
>>>
*** Remote Interpreter Reinitialized ***
-704
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 18:1 Insert

PyScripter - C:\Users\k...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Configuration Run Output Messages Python Interpreter

#-----  
# Name: module1  
# Purpose:  
#  
# Author: Win\_JumP  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_JumP 2024  
# Licence: <your Licence>  
#-----  
  
##Write a Python program to sum all the items in a List.  
def sum\_list(items):  
 sum\_numbers = 0  
 for x in items:  
 sum\_numbers += x  
 return sum\_numbers  
  
print(sum\_list([2, 3, -4]))

zadanie\_1.py

Python Interpreter

-5  
>>> \*\*\* Remote Interpreter Reinitialized \*\*\*  
-1  
>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 19:25 | Insert |

PyScripter - C:\Users\k...

Rozwiązań z działu „Python\_Data\_Types\_Dictionary”

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:    13.04.2024
# Copyright:  (c) Win_Jump 2024
# Licence:    <your licence>
#-----#
#
#Write a Python program to print all distinct values in a dictionary.
#Sample Data : [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]
#Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}
L = [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]

print("oryginalna lista: ", L)

u_value = set(val for dic in L for val in dic.values())
print("unikalne wartosci: ", u_value)
```

zadanie\_16.py zadanie\_17.py zadanie\_18.py zadanie\_19.py zadanie\_20.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
>>>
*** Remote Interpreter Reinitialized ***
oryginalna lista: [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]
unikalne wartosci: {'S005', 'S002', 'S007', 'S001', 'S009'}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 22:1 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik

The screenshot shows the PyScripter IDE interface. The main window displays a Python script named `zadanie_19.py`. The code is as follows:

```
#-
# Name: module1
# Purpose:
#
# Author: Win_JumP
#
# Created: 13.04.2024
# Copyright: (c) Win_JumP 2024
# Licence: <your licence>
#-
#
# Write a Python program to combine two dictionary by adding values for common keys.
#d1 = {'a': 100, 'b': 200, 'c': 300}
#d2 = {'a': 300, 'b': 200, 'd': 400}
#Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

from collections import Counter

d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}

d = Counter(d1) + Counter(d2)

print(d)
```

Below the code editor, the tabs show other files: `zadanie_15.py`, `zadanie_16.py`, `zadanie_17.py`, `zadanie_18.py`, and `zadanie_19.py` (which is currently selected). The status bar at the bottom indicates the Python version is 3.11 (64-bit).

A separate window titled "Python Interpreter" is open, showing the following session:

```
'Czeresnia', 'klaśa': ['V'], 'przedmioty': ['dzwonek']}
>>>
*** Remote Interpreter Reinitialized ***
Słownik jest pusty
>>>
*** Remote Interpreter Reinitialized ***
Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
>>>
```

The Python Interpreter tab is highlighted in orange.

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py zadanie\_18.py

```
10
11  #--#
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:    13.04.2024
18  # Copyright:  (c) Win_JumP 2024
19  # Licence:    <your licence>
20  #--#
21
22  #Write a Python program to check if a dictionary is empty or not.
23  my_dict = {}
24
25  if not bool(my_dict):
26      print("Słownik jest pusty")
```

Python Interpreter

```
{'id1': {'imie': ['Klaudia'], 'klasa': ['V'], 'przedmioty': ['matematyka', 'angielski', 'polski', 'geografia']}, 'id2': {'imie': ['Eleonora'], 'klasa': ['VI'], 'przedmioty': ['matematyka', 'angielski', 'polski']}, 'id3': {'imie': ['Teodora'], 'klasa': ['X'], 'przedmioty': ['polski', 'geografia']}, 'id4': {'imie': ['Czeresnia'], 'klasa': ['V'], 'przedmioty': ['dzwonek']}}

>>>
*** Remote Interpreter Reinitialized ***
Słownik jest pusty
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 17:1 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik 18:50

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#--#
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:    <your licence>
#--#
#
#Write a Python program to remove duplicates from the dictionary.
student_data = {
    'id1': {
        'imie': ['Klaudia'],
        'klasa': ['V'],
        ' przedmioty': ['matematyka, angielski, polski, geografia']
    },
    'id2': {
        'imie': ['Eleonora'],
        'klasa': ['VI'],
        ' przedmioty': ['matematyka, angielski, polski']
    },
    'id3': {
        'imie': ['Teodora'],
        'klasa': ['X'],
        ' przedmioty': ['polski, geografia']
    }
}
{'x': 'czerwony', 'y': 'zolty', 'z': 'zielony'}
>>>
*** Remote Interpreter Reinitialized ***
{'id1': {'imie': ['Klaudia'], 'klasa': ['V'], ' przedmioty': ['matematyka, angielski, polski, geografia']}, 'id2': {'imie': ['Eleonora'], 'klasa': ['VI'], ' przedmioty': ['matematyka, angielski, polski']}, 'id3': {'imie': ['Teodora'], 'klasa': ['X'], ' przedmioty': ['polski, geografia']}, 'id4': {'imie': ['Czeresnia'], 'klasa': ['V'], ' przedmioty': ['dzwonek']}}
>>>
```

zadanie\_13.py zadanie\_14.py zadanie\_15.py zadanie\_16.py zadanie\_17.py

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_Jump
#
# Created:   13.04.2024
# Copyright: (c) Win_Jump 2024
# Licence:   <your licence>
#-----#
#
#Write a Python program to get a dictionary from an object's fields.
class dictObj(object):
    def __init__(self):
        self.x = 'czerwony'
        self.y = 'zolty'
        self.z = 'zielony'

    def do_nothing(self):
        pass

test = dictObj()
print(test.__dict__)
```

zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_15.py zadanie\_16.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 25:1 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
#-
# Name:      module1
# Purpose:
# 
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:   <your licence>
#-

#Write a Python program to get the maximum and minimum values of a dictionary.
my_dict = {'x': 400, 'y': 5474, 'z': 521}

key_max = max(my_dict.keys(), key=(lambda k: my_dict[k]))

key_min = min(my_dict.keys(), key=(lambda k: my_dict[k]))

print('maxymalna wartosc: ', my_dict[key_max])
print('minimalna wartosc: ', my_dict[key_min])
```

zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py zadanie\_15.py

Python Interpreter

```
czarny: #000000
czerwony: #FF0000
zielony: #008000
>>>
*** Remote Interpreter Reinitialized ***
maxymalna wartosc: 5474
minimalna wartosc: 400
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 13:41 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik 18:40

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #-----
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your Licence>
10 #
11
12 #Write a Python program to sort a given dictionary by key
13 color_dict = {
14     'czarny': '#000000',
15     'zielony': '#008000',
16     'biały': '#FFFFFF'
17 }
18
19 for key in sorted(color_dict):
20     print("%s: %s" % (key, color_dict[key]))
```

zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py zadanie\_14.py

Python Interpreter

```
{'czarny': '#000000', 'zielony': '#008000', 'niebieski': '#0000FF'}
>>>
*** Remote Interpreter Reinitialized ***
biały: #FFFFFF
czarny: #000000
czarny: #FF0000
zielony: #008000
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 15:26 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik 18:38

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #-
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:   <your licence>
10 #
11
12 #Write a Python program to map two Lists into a dictionary.
13 keys = ['czerwony', 'zielony', 'niebieski']
14
15 values = ['#FF0000', '#008000', '#0000FF']
16
17 color_dictionary = dict(zip(keys, values))
18
19 print(color_dictionary)
20
```

zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py zadanie\_13.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
{'b': 2, 'c': 3, 'd': 4}
>>>
*** Remote Interpreter Reinitialized ***
{'czerwony': '#FF0000', 'zielony': '#008000', 'niebieski': '#0000FF'}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready | Python 3.11 (64-bit) | Remote | 19:1 | Insert |

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #--#
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:   <your Licence>
10#
11#
12#Write a Python program to remove a key from a dictionary.
13myDict = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
14
15print(myDict)
16
17if 'a' in myDict:
18    del myDict['a']
19
20print(myDict)
```

zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py zadanie\_12.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 19:1 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik 18:34

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #-
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:    13.04.2024
8 # Copyright:  (c) Win_JumP 2024
9 # Licence:    <your licence>
10 #-
11
12 #Write a Python program to multiply all the items in a dictionary. ()
13 my_dict = {'data1': 210, 'data2': -77, 'data3': 777}
14
15 result = 1
16
17 for key in my_dict:
18     result = result * my_dict[key]
19
20 print(result)
```

zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py zadanie\_11.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 20:14 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik 18:31

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #--#
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your licence>
10 #
11 #
12 #
13 #Write a Python program to sum all the items in a dictionary.
14 my_dict = {'data1': 210, 'data2': -77, 'data3': 777}
15
16 result = sum(my_dict.values())
17 print(result)
```

zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py zadanie\_10.py

niebieski powiązany z 3  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
293  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
910  
>>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 13:52 | Insert |

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik 18:29

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py zadanie\_9.py

```
#-----#
# Name:      module1
# Purpose:
#
# Author:     Win_JumP
#
# Created:    13.04.2024
# Copyright:  (c) Win_JumP 2024
# Licence:   <your Licence>
#-----#
#
##Write a Python program to iterate over dictionaries using for Loops.
d = {'zcerwony': 1, 'zielony': 2, 'niebieski': 3}
for color_key, value in d.items():
    print(color_key, 'powiązany z ', d[color_key])
```

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
{'a': 100, 'b': 200, 'x': 300, 'y': 200}
>>>
*** Remote Interpreter Reinitialized ***
zcerwony powiązany z 1
zielony powiązany z 2
niebieski powiązany z 3
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 17:1 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #-
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:   13.04.2024
8  # Copyright: (c) Win_JumP 2024
9  # Licence:   <your licence>
10 #
11
12 #Write a Python script to merge two Python dictionaries.
13 d1 = {'a': 100, 'b': 200}
14
15 d2 = {'x': 300, 'y': 200}
16
17 d = d1.copy()
18
19 d.update(d2)
20
21
22 print(d)
```

zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py zadanie\_8.py

Python Interpreter

```
>>>
*** Remote Interpreter Reinitialized ***
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14:
196, 15: 225}
>>>
*** Remote Interpreter Reinitialized ***
{'a': 100, 'b': 200, 'x': 300, 'y': 200}
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 22:1 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #-----
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your Licence>
10 #
11
12 #Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys
13 #Sample Dictionary
14 #{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
15 d = dict()
16
17 for x in range(1, 16):
18     d[x] = x ** 2
19
20 print(d)
21
```

zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py zadanie\_7.py

Python Interpreter

```
*** Remote Interpreter Reinitialized ***
Input a number 4
{1: 1, 2: 4, 3: 9, 4: 16}
>>>
*** Remote Interpreter Reinitialized ***
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 21:1 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

zadanie\_2.py zadanie\_3.py zadanie\_4.py zadanie\_5.py zadanie\_6.py

```
10
11  #--#
12  # Name:      module1
13  # Purpose:
14  #
15  # Author:     Win_JumP
16  #
17  # Created:    13.04.2024
18  # Copyright:  (c) Win_JumP 2024
19  # Licence:    <your licence>
20  #--#
21
22  #Write a Python script to generate and print a dictionary that contains a number (between 1 and
23  #Sample Dictionary ( n = 5 ) :
24  #Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
25  n = int(input("Input a number "))
26
27  d = dict()
28
29  for x in range(1, n + 1):
30      d[x] = x * x
31
32  print(d)
```

Python Interpreter

```
x -> 60
y -> 10
z -> 50
>>>
*** Remote Interpreter Reinitialized ***
Input a number 4
{1: 1, 2: 4, 3: 9, 4: 16}
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 22:10 Modified Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1  #--#
2  # Name:      module1
3  # Purpose:
4  #
5  # Author:     Win_JumP
6  #
7  # Created:    13.04.2024
8  # Copyright:  (c) Win_JumP 2024
9  # Licence:   <your licence>
10 #
11
12 #Write a Python program to iterate over dictionaries using for Loops.
13 d = {'x': 60, 'y': 10, 'z': 50}
14
15 for dict_key, dict_value in d.items():
16     print(dict_key, '->', dict_value)
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py zadanie\_4.py zadanie\_5.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) Remote 13:30 Insert

PyScripter - C:\Users\k... \*Bez tytułu — Notatnik 18:21

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
1 #-----#
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:     Win_JumP
6 #
7 # Created:   13.04.2024
8 # Copyright: (c) Win_JumP 2024
9 # Licence:   <your Licence>
10 #-----
11
12 #Write a Python script to check whether a given key already exists in a dictionary.
13 d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
14
15 def is_key_present(x):
16     if x in d:
17         print('Key is present in the dictionary')
18     else:
19         print('Key is not present in the dictionary')
20
21 is_key_present(5)
22
23 is_key_present(9)
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py zadanie\_4.py

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Ready Python 3.11 (64-bit) Remote 19:30 Insert

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type... \*Bez tytułu — Notatnik 18:16

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

Python Interpreter

```
# Purpose:  
#  
# Author: Win_JumP  
#  
# Created: 13.04.2024  
# Copyright: (c) Win_JumP 2024  
# Licence: <your licence>  
#-----  
#Write a Python script to concatenate the following dictionaries to create a new one.  
#  
#Sample Dictionary :  
#dic1={1:10, 2:20}  
#dic2={3:30, 4:40}  
#dic3={5:50,6:60}  
#Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}  
  
#utworzenie 4 slownikow, z czego ostatni jest pusty:  
dic1 = {1: 10, 2: 20}  
dic2 = {3: 30, 4: 40}  
dic3 = {5: 50, 6: 60}  
dic4 = {}  
  
#iteracja z uzykiem petli:  
for d in (dic1, dic2, dic3):  
    dic4.update(d)  
  
print(dic4)
```

zadanie\_1.py zadanie\_2.py zadanie\_3.py

Python Interpreter

```
>>>  
*** Remote Interpreter Reinitialized ***  
{0: 10, 1: 20}  
{0: 10, 1: 20, 2: 30}  
>>>  
*** Remote Interpreter Reinitialized ***  
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 29:1 | Insert | 18:09

PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...

File Edit Search View Project Run Tools Help

File Explorer Task List Run Output Messages Python Interpreter

#-----  
# Name: module1  
# Purpose:  
#  
# Author: Win\_Jump  
#  
# Created: 13.04.2024  
# Copyright: (c) Win\_Jump 2024  
# Licence: <your licence>  
#-----  
  
#Write a Python script to add a key to a dictionary.  
#Sample Dictionary : {0: 10, 1: 20}  
#Expected Result : {0: 10, 1: 20, 2: 30}  
#tworzenie słownika  
d = {0: 10, 1: 20}  
  
print(d)  
  
#dodanie zawartości do słownika  
d.update({2: 30})  
  
print(d)|

zadanie\_1.py zadanie\_2.py

Python Interpreter

Original dictionary : {1: 2, 3: 5, 7: 3, 2: 1, 10: 0}  
Dictionary in ascending order by value : [(10, 0), (2, 1), (1, 2), (7, 3), (3, 5)]  
Dictionary in descending order by value : {3: 5, 7: 3, 1: 2, 2: 1, 10: 0}  
->>>  
\*\*\* Remote Interpreter Reinitialized \*\*\*  
{0: 10, 1: 20}  
{0: 10, 1: 20, 2: 30}  
->>>

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

Python 3.11 (64-bit) | Remote | 23:9 | Insert |  
Users\k...

The screenshot shows the PyScripter IDE interface with the following details:

- Title Bar:** PyScripter - C:\Users\kubit\Desktop\szkolne\studia\semestr\_2\podstawy\_programowania\_s2\Python\_Data\_Type...
- Toolbar:** File, Edit, Search, View, Project, Run, Tools, Help.
- Code Editor:** The file "zadanie\_1.py" contains Python code to sort a dictionary by value. The code includes comments explaining the purpose and creation date, and imports the operator module. It defines a dictionary d, prints it, sorts it in ascending order, prints it again, sorts it in descending order, and prints it once more.

```
#-
# Name:         module1
# Purpose:
#
# Author:        Win_JumP
#
# Created:      13.04.2024
# Copyright:    (c) Win_JumP 2024
# Licence:      <your Licence>
10
#-
#
#Write a Python script to sort (ascending and descending) a dictionary by value.
# biblioteka operator umozliwia sortowanie
import operator
15
d = {1: 2, 3: 5, 7: 3, 2: 1, 10: 0}
.
print('Original dictionary : ',d)
.
20 sorted_d = sorted(d.items(), key=operator.itemgetter(1))
.
print('Dictionary in ascending order by value : ',sorted_d)
.
sorted_d = dict( sorted(d.items(), key=operator.itemgetter(1), reverse=True))
.
print('Dictionary in descending order by value : ',sorted_d)
.
```

- Python Interpreter:** Shows the execution of the script. The output is:

```
***  
*** Remote Python engine is active ***  
>>>  
*** Remote Interpreter Reinitialized ***  
Original dictionary : {1: 2, 3: 5, 7: 3, 2: 1, 10: 0}  
Dictionary in ascending order by value : [(10, 0), (2, 1), (1, 2), (7, 3), (3, 5)]  
Dictionary in descending order by value : {3: 5, 7: 3, 1: 2, 2: 1, 10: 0}  
>>>
```
- Bottom Navigation:** Call Stack, Variables, Watches, Breakpoints, Output, Messages, Python Interpreter.
- Status Bar:** Python 3.11 (64-bit), Remote, 15:1, Insert.