**Python. HW\_1.**

**Part\_1**

1) Создать переменную типа String

item = 1  
name = 'Irina'  
result = str(item) + ' ' + name

Hello, Irina!

2) Создать переменную типа Integer

print("Возраст:", age)  
count = 2  
print("Детей:", count)

print("Возраст:", age)  
count = 2  
print("Детей:", count)

Возраст: 38

Детей: 2

3) Создать переменную типа Float

height = 1.62  
weight = 78  
print("Рост:", height)  
print("Вес:", weight)

Рост: 1.62

Вес: 78

4) Создать переменную типа Bytes

#Тип данных bytes - это неизменяемые последовательности отдельных байтов  
b = bytes(10)  
print(type(b), b)

<class 'bytes'> b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

5) Создать переменную типа List

list = [True, 38, 3.14, 'Boroda', 78,5]  
second\_list = [123, 'Boroda']  
print(list) #Напечатает весь список  
print(list[0]) #Напечатает первый элемент списка  
print(list[1:3]) #Напечатает элементы списка со второго по третий  
print(list[2:]) #Напечатает элементы списка начиная с третьего  
print(second\_list \*2) #Напечатает удвоенный список  
print (list + second\_list) # Напечатает объединенные списки

[3.14, 'Boroda', 78, 5]

[123, 'Boroda', 123, 'Boroda']

[True, 38, 3.14, 'Boroda', 78, 5, 123, 'Boroda']

Process finished with exit code 0

6) Создать переменную типа Tuple

#Кортеж состоит из ряда значений, разделенных запятыми, заключенными в круглые скобки ( ( ) )  
my\_tuple = (True, 786, 3.14, 'Boroda', 70.2)  
second\_tuple = (123, 'Boroda')  
  
print(my\_tuple) # Печатает весь кортеж  
print(my\_tuple[0]) # Печатает первый элемент  
print(second\_tuple \* 2) # Печатает удвоенный кортеж  
print(my\_tuple + second\_tuple) # Печатает объединенные кортежи

(True, 786, 3.14, 'Boroda', 70.2)

True

(123, 'Boroda', 123, 'Boroda')

(True, 786, 3.14, 'Boroda', 70.2, 123, 'Boroda')

Process finished with exit code 0

7) Создать переменную типа Set

#Класс set (множество) представляет собой неупорядоченную коллекцию уникальных элементов  
num\_set = {1, 2, 3, 4, 5, 6}  
print(num\_set)

{1, 2, 3, 4, 5, 6}

8. Создать переменную типа Frozen set

#Для создания замороженного множества, мы используем метод frozenset()  
X = frozenset([1, 2, 3, 4, 5, 6])  
Y = frozenset([4, 5, 6, 7, 8, 9])  
print(X)  
print(Y)

frozenset({1, 2, 3, 4, 5, 6})

frozenset({4, 5, 6, 7, 8, 9})

9) Создать переменную типа Dict

# dict -Для создания словаря в Python необходимо передать последовательность  
# элементов внутри фигурных скобок {}, разделив их запятыми (,)  
my\_dict = {1: 'orange', 2: 'fruit'}  
print(my\_dict)

{1: 'orange', 2: 'fruit'}

10) Вывести в консоль все выше перечисленные переменные с добавлением типа данных.

#Вывести переменные в консоль с добавлением типов данных  
print("Hello, Irina!" , "Возраст:", age , "Рост:", height , list ,  
 second\_tuple , num\_set , X , Y , my\_dict , type(b), b)

Hello, Irina! Возраст: 38 Рост: 1.62 [True, 38, 3.14, 'Boroda', 78, 5] (123, 'Boroda') {1, 2, 3, 4, 5, 6} frozenset({1, 2, 3, 4, 5, 6}) frozenset({4, 5, 6, 7, 8, 9}) {1: 'orange', 2: 'fruit'} <class 'bytes'> b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

11) Создать 2 переменные String, создать переменную в которой сканкатенируете эти переменные. Вывести в консоль.

12) Вывести в одну строку переменные типа String и Integer используя “,” (Запятую)

13) Вывести в одну строку переменные типа String и Integer используя “+” (Плюс)