```
X - Assign values to variables of different types

********************************************************************************************************
name = "Alice"                                    # name is assigned a string value.
age = 28                                          # age is assigned an integer value.
height = 1.68                                     # height is assigned a float value.
is_student = True                                 # is_student is assigned a boolean value (True in this case).
grades = [80, 90, 85]                             # grades is assigned a list of integers.
person = {"name": "Bob", "age": 32}               # person is assigned a dictionary with two key-value pairs.

# Print out the variables
print("Name:", name)
#output: Name: Alice
print("Age:", age)
#output: Age: 28
print("Height:", height)
#output: Height: 1.68
print("Is student:", is_student)
#output: Is student: True
print("Grades:", grades)
#output: Grades: [80, 90, 85]
print("Person:", person)
#output: Person: {'name': 'Bob', 'age': 32}


# name is assigned a string value.
# age is assigned an integer value.
# height is assigned a float value.
# is_student is assigned a boolean value (True in this case).
# grades is assigned a list of integers.
# person is assigned a dictionary with two key-value pairs.

********************************************************************************************************
X - Basic functions (str , int , float )

********************************************************************************************************
name = "John"
age = 25
height = 1.75

# concatenate a string and an integer
greeting = "Hello, my name is " + name + " and I am " + str(age) + " years old."
print(greeting)
#output:Hello, my name is John and I am 25 years old.

# concatenate a string and a float
measurement = "My height is " + str(height) + " meters."
print(measurement)
#output: My height is 1.75 meters.

********************************************************************************************************
# Convert a number to an integer
num1 = 10.5
num1_int = int(num1)
print(num1_int)
#output: 10
********************************************************************************************************
# Convert a number to a float
num2 = 25
num2_float = float(num2)
print(num2_float)
#output: 25.0
********************************************************************************************************
I- build-in function for  a  list  in python

********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits) # Output: ["apple", "banana", "cherry", "orange"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
more_fruits = ["orange", "grape", "kiwi"]
fruits.extend(more_fruits)
print(fruits) # Output: ["apple", "banana", "cherry", "orange", "grape", "kiwi"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.insert(1, "orange")
print(fruits) # Output: ["apple", "orange", "banana", "cherry"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.remove("banana")
print(fruits) # Output: ["apple", "cherry"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
popped_fruit = fruits.pop(1)
print(popped_fruit) # Output: "banana"
print(fruits) # Output: ["apple", "cherry"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
banana_index = fruits.index("banana")
print(banana_index) # Output: 1
********************************************************************************************************
fruits = ["apple", "banana", "cherry", "banana"]
banana_count = fruits.count("banana")
print(banana_count) # Output: 2
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.sort()
print(fruits) # Output: ["apple", "banana", "cherry"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.reverse()
print(fruits) # Output: ["cherry", "banana", "apple"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits_copy = fruits.copy()
print(fruits_copy) # Output: ["apple", "banana", "cherry"]
********************************************************************************************************
fruits = ["apple", "banana", "cherry"]
fruits.clear()
print(fruits) # Output: []
********************************************************************************************************
fruits = ["apple", "banana", "orange", "mango"]
print(len(fruits))  # Output: 4
********************************************************************************************************
# Generate a list of numbers from 0 to 9
my_list = list(range(10))

# Generate a list of numbers from 1 to 10
my_list = list(range(1, 11))
```

```
# Generate a list of even numbers from 2 to 10
my_list = list(range(2, 11, 2))

# Generate a list of odd numbers from 1 to 9
my_list = list(range(1, 10, 2))
*********************************************************************************************************
# Define a list of numbers
my_list = [10, 20, 30, 40, 50]

# Get the sum of the numbers in the list
total = sum(my_list)  # Returns 150

# Get the maximum number in the list
largest = max(my_list)  # Returns 50

# Get the minimum number in the list
smallest = min(my_list)  # Returns 10
*********************************************************************************************************

X - some common functions for dictionaries in Python

*********************************************************************************************************

my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(len(my_dict)) # Output: 3

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(my_dict.keys()) # Output: dict_keys(['apple', 'banana', 'orange'])

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(my_dict.values()) # Output: dict_values([2, 3, 4])

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(my_dict.items()) # Output: dict_items([('apple', 2), ('banana', 3), ('orange', 4)])

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(my_dict.get("banana")) # Output: 3
print(my_dict.get("mango")) # Output: None
print(my_dict.get("mango", "Key not found")) # Output: Key not found

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
print(my_dict.pop("banana")) # Output: 3
print(my_dict) # Output: {'apple': 2, 'orange': 4}
print(my_dict.pop("mango", "Key not found")) # Output: Key not found

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3, "orange": 4}
my_dict.clear()
print(my_dict) # Output: {}

*********************************************************************************************************
my_dict = {"apple": 2, "banana": 3}
new_dict = {"orange": 4, "mango": 5}
my_dict.update(new_dict)
print(my_dict) # Output: {'apple': 2, 'banana': 3, 'orange': 4, 'mango': 5}

*********************************************************************************************************

X - operators in pythons

*********************************************************************************************************
Arithmetic operators:
a = 10
b = 5
print(a + b) # Output: 15
print(a - b) # Output: 5
print(a * b) # Output: 50
print(a / b) # Output: 2.0
print(a % b) # Output: 0
print(a ** b) # Output: 100000
print(a // b) # Output: 2
*********************************************************************************************************
Comparison operators:
a = 10
b = 5
print(a == b) # Output: False
print(a != b) # Output: True
print(a > b) # Output: True
print(a < b) # Output: False
print(a >= b) # Output: True
print(a <= b) # Output: False
*********************************************************************************************************
X - Assignment operators:
a = 10
b = 5
a += b # a = a + b
print(a) # Output: 15
a -= b # a = a - b
print(a) # Output: 10
a *= b # a = a * b
print(a) # Output: 50
a /= b # a = a / b
print(a) # Output: 10.0
a %= b # a = a % b
print(a) # Output: 0.0
a **= b # a = a ** b
print(a) # Output: 0.0
a //= b # a = a // b
print(a) # Output: 0.0
*********************************************************************************************************
X - Logical operators:
a = 10
b = 5
c = 7
print(a > b and b < c) # Output: True
print(a > b or b > c) # Output: True
print(not(a > b)) # Output: False
*********************************************************************************************************

X - Identity operators:
a = 10
b = 5
c = a
print(a is b) # Output: False
print(a is c) # Output: True
print(a is not b) # Output: True
********************************************************************************************************

X - Membership operators:
my_list = [1, 2, 3, 4, 5]
print(3 in my_list) # Output: True
print(6 not in my_list) # Output: True
********************************************************************************************************
X - Bitwise operators:
a = 10 # 1010
b = 5  # 0101
print(a & b) # Output: 0
print(a | b) # Output: 15
print(a ^ b) # Output: 15
print(~a)    # Output: -11
print(a << 1) # Output: 20
print(a >> 1) # Output: 5

********************************************************************************************************

X - if statement

********************************************************************************************************
#Checking if a number is positive or negative:
num = 5
if num >= 0:
    print("The number is positive")
else:
    print("The number is negative")

********************************************************************************************************
#Checking if a string is empty or not:
string = "Hello"
if string:
    print("The string is not empty")
else:
    print("The string is empty")

********************************************************************************************************
#Checking if a number is even or odd:
num = 4
if num % 2 == 0:
    print("The number is even")
else:
    print("The number is odd")

********************************************************************************************************
#Checking if a string contains a specific substring:
string = "Hello, world!"
if "world" in string:
    print("The string contains the word 'world'")
else:
    print("The string does not contain the word 'world'")

********************************************************************************************************
Checking if a number is within a certain range:
num = 10
if num in range(1, 11):
    print("The number is within the range of 1 to 10")
else:
    print("The number is outside the range of 1 to 10")

********************************************************************************************************
Checking if a variable is of a certain data type:
var = "Hello, world!"
if isinstance(var, str):
    print("The variable is a string")
else:
    print("The variable is not a string")
********************************************************************************************************
Checking multiple conditions using logical operators:

num = 7
if num > 0 and num % 2 == 1:
    print("The number is a positive odd number")
else:
    print("The number is not a positive odd number")

********************************************************************************************************
Checking if a list contains a specific value:
my_list = [1, 2, 3, 4, 5]
if 3 in my_list:
    print("The list contains the number 3")
else:
    print("The list does not contain the number 3")
********************************************************************************************************

X - Loops in python for loop while loop

********************************************************************************************************

# Example for loop
for i in range(5):
    print(i)

********************************************************************************************************
# Example while loop
x = 0
while x < 5:
    print(x)
    x += 1

********************************************************************************************************
# Example do-while loop
x = 0
while True:
    print(x)
    x += 1
    if x >= 5:
        break

********************************************************************************************************


X -  Advance functions

********************************************************************************************************
# Example lambda function
square = lambda x: x**2
```

```
print(square(5))  # Returns 25
```

****

```
# Example map function
my_list = [1, 2, 3, 4, 5]
squared_list = list(map(lambda x: x**2, my_list))
print(squared_list)  # Returns [1, 4, 9, 16, 25]
```

****

```
# Example filter function
my_list = [1, 2, 3, 4, 5]
even_list = list(filter(lambda x: x % 2 == 0, my_list))
print(even_list)  # Returns [2, 4]
```

****

```
# Example reduce function
from functools import reduce
my_list = [1, 2, 3, 4, 5]
product = reduce(lambda x, y: x * y, my_list)
print(product)  # Returns 120 (i.e., 1 * 2 * 3 * 4 * 5)
```

****

X - functions in python

****

Python function that takes two arguments and returns their sum:

```
def add_numbers(num1, num2):
    sum = num1 + num2
    return sum

result = add_numbers(2, 3)
print(result)  # This will output 5

# This will call the add_numbers function with the arguments 2 and 3, and assign the result to the variable result. Then, it will print the value of result, which is 5.
```

****

```
def add_numbers(*args):
    sum = 0
    for num in args:
        sum += num
    return sum

result = add_numbers(2, 3, 5)
print(result)  # This will output 10

# This will call the add_numbers function with the arguments 2, 3, and 5, and assign the result to the variable result. Then, it will print the value of result, which is 10. You can al
```

****