

python.py - D:/python.py (3.12.0)

File Edit Format Run Options Window Help

```
# 1. Greeting function
```

```
def Greetings(name):
```

```
    print(f"Welcome to SMIT training center, {name}")
```

```
Greetings("Ahsan")
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 15:34:56) [AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more

>>>

= RESTART: D:/python.py

>>>

===== RESTART: D:/python.py

Welcome to SMIT training center, Ahsan

>>>

|

File Edit Format Run Options Window Help

: 2. Check if number is positive, negative or zero

```
def check_number(num):  
    if num > 0:  
        print("Positive number")  
    elif num < 0:  
        print("Negative number")  
    else:  
        print("Zero")
```

```
check_number(5)
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0f1  
AMD64) on win32

Type "help", "copyright", "cre

>>>

= RESTART: D:/python.py

>>>

===== ]

Positive number

>>>

File Edit Format Run Options Window Help

```
# 3. Return larger of two numbers
```

```
def larger_number(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
print(larger_number(5, 6))
```

 IDLE Shell 3.12.0

File Edit Shell Debug Options Win

Python 3.12.0 (tags/v3.12.0: AMD64) on win32

Type "help", "copyright",

>>>

= RESTART: D:/python.py

>>>

=====

6

>>>

File Edit Format Run Options Window Help

# 4. Return largest of three numbers using if-else

```
def largest_number(a, b, c):  
    if a >= b and a >= c:  
        return a  
    elif b >= a and b >= c:  
        return b  
    else:  
        return c  
print(largest_number(4,5,6))
```

IDLE Shell 3.12.0

File Edit Shell Debug Options

Python 3.12.0 (tags/v3  
AMD64) on win32  
Type "help", "copyrigh

>>>

= RESTART: D:/python.p  
6

>>>

File Edit Format Run Options Window Help

```
# 5. Determine age category
```

```
def age_category(age):
```

```
    if age < 18:
```

```
        return "Minor"
```

```
    elif age >= 18 and age < 60:
```

```
        return "Adult"
```

```
    else:
```

```
        return "Senior Citizen"
```

```
print(age_category(16))
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Win

Python 3.12.0 (tags/v3.12.0

AMD64) on win32

Type "help", "copyright",

>>>

= RESTART: D:/python.py

Minor

>>>

|

File Edit Format Run Options Window Help

# 6. Check if number is even or odd

```
def even_or_odd(n):  
    if n % 2 == 0:  
        print("Even number")  
    else:  
        print("Odd number")  
print(even_or_odd(6))
```

IDLE Shell 3.12.0

File Edit Shell Debug Optic

Python 3.12.0 (tags  
AMD64) on win32  
Type "help", "copyr

>>>

= RESTART: D:/pythc  
Even number  
None

>>>

python.py - D:/python.py (3.12.0)

File Edit Format Run Options Window Help

7. Return square of number

```
def square(num):  
    return num * num  
print(square(7))
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb  
AMD64) on win32

Type "help", "copyright", "cred

>>>

= RESTART: D:/python.py

49

>>>

|

File Edit Format Run Options Window Help

```
# 8. Compute area and circumference of circle
```

```
def circle_properties(radius):  
    pi = 3.14159  
    area = pi * radius * radius  
    circumference = 2 * pi * radius  
    return area, circumference  
print(circle_properties(4))
```

Python 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 20  
AMD64) on win32

Type "help", "copyright", "credits" or "licens

>>>

= RESTART: D:/python.py  
(50.26544, 25.13272)

>>> |



File Edit Format Run Options Window Help

```
# 9. Check pass/fail based on score
```

```
def check_pass(score):
```

```
    if score > 60:
```

```
        return "Pass"
```

```
    else:
```

```
        return "Fail"
```

```
print(check_pass(66))
```

 IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb1  
AMD64) on win32

Type "help", "copyright", "credi

>>>

= RESTART: D:/python.py

Pass

>>>

File Edit Format Run Options Window Help

```
# 10. Check if a number is prime (simple version)
```

```
def is_prime(num):
```

```
    if num < 2:
```

```
        return False
```

```
    # Check if num is divisible by any number from 2 to num-
```

```
    for i in range(2, num):
```

```
        if num % i == 0:
```

```
            return False # Not a prime number
```

```
    return True # Prime number
```

```
print(is_prime(4))
```

 IDLE Shell 3.12.0

File Edit Shell Debug Options

Python 3.12.0 (tags/v: AMD64) on win32  
Type "help", "copyright"

>>>

= RESTART: D:/python.1  
False

>>>

|

File Edit Format Run Options Window Help

```
# 11. Factorial using recursion
```

```
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n-1)  
print(factorial(5))
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 1 2023, 15:34:56) on win32  
Type "help", "copyright", "credits" or "quit()"

>>>

= RESTART: D:/python.py

120

>>>

|

File Edit Format Run Options Window Help

```
# 1. Create and display list elements
```

```
student = ["Tahir", 44, "AI and Data Science", True]
```

```
for item in student:
```

```
    print(item)
```

 IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, AMD64) on win32

Type "help", "copyright", "credits" or "license()"

>>>

= RESTART: D:/python.py

Tahir

44

AI and Data Science

True

>>>

```
File Edit Format Run Options Window Help
# 2. Separate data types into different lists
student = ["Tahir", 44, "AI and Data Science", True]

strings = []
numbers = []
booleans = []

for item in student:

    if type(item) == str:
        strings.append(item)

    elif type(item) == bool:
        booleans.append(item)

    elif type(item) == int or type(item) == float:
        numbers.append(item)

print("Strings:", strings)
print("Numbers:", numbers)
print("Booleans:", booleans)
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, AMD64) on win32  
Type "help", "copyright", "credits" or "license()"

>>>

= RESTART: D:/python.py  
Strings: ['Tahir', 'AI and Data Science']  
Numbers: [44]  
Booleans: [True]

>>>

File Edit Format Run Options Window Help

# 3. Fruit list manipulation

```
fruits = ["apple", "raspberry", "pineapple", "cherry"]
```

# a. Check if "apple" is in list and get index

```
if "apple" in fruits:
```

```
    print("Apple found at index:", fruits.index("apple"))
```

# b. Replace "raspberry" and "pineapple" with "orange"

```
fruits[1:3] = ["orange"]
```

```
print(fruits)
```

# c. Insert "apricot" at index 2

```
fruits.insert(2, "apricot")
```

```
print(fruits)
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023)

AMD64) on win32

Type "help", "copyright", "credits" or "license()"

>>>

= RESTART: D:/python.py

Apple found at index: 0

['apple', 'orange', 'cherry']

['apple', 'orange', 'apricot', 'cherry']

>>>

```
# 3. Fruit list manipulation
fruits = ["apple", "raspberry", "pineapple", "cherry"]

# a. Check if "apple" is in list and get index
if "apple" in fruits:
    print("Apple found at index:", fruits.index("apple"))

# b. Replace "raspberry" and "pineapple" with "orange"
fruits[1:3] = ["orange"]
print(fruits)

# c. Insert "apricot" at index 2
fruits.insert(2, "apricot")
print(fruits)

# d. Extend with vehicles
fruits.extend(["car", "bike", "aeroplane"])

print("Updated List:", fruits)
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python.py
Apple found at index: 0
['apple', 'orange', 'cherry']
['apple', 'orange', 'apricot', 'cherry']
Updated List: ['apple', 'orange', 'apricot', 'cherry', 'car', 'bike', 'aeroplane
']
>>>
```

```
File Edit Format Run Options Window Help
Scores_list = [40, 89, 90, 89, 23, 90, 50]

unique_scores = []

for score in Scores_list:
    if score not in unique_scores:
        unique_scores.append(score)

unique_scores.sort()
unique_scores.reverse()

first_best = unique_scores[0]
second_best = unique_scores[1]

print("First Best Score:", first_best)
print("Second Best Score:", second_best)
```

```
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2021;
AMD64) on win32
Type "help", "copyright", "credits" or "license()"
>>>
= RESTART: D:/python.py
First Best Score: 90
Second Best Score: 89
>>>
```



File Edit Format Run Options Window Help

```
# 4. Get first and second best scores
```

```
Scores_list = [40, 89, 90, 89, 23, 90, 50]
```

```
unique_scores = []
```

```
for score in Scores_list:
    if score not in unique_scores:
        unique_scores.append(score)
```

```
unique_scores.sort()
unique_scores.reverse()
```

```
first_best = unique_scores[0]
second_best = unique_scores[1]
```

```
print("First Best Score:", first_best)
print("Second Best Score:", second_best)
```

 IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, AMD64) on win32

Type "help", "copyright", "credits"

>>>

= RESTART: D:/python.py

First Best Score: 90

Second Best Score: 89

>>>

|

File Edit Format Run Options Window Help

```
# 5. Remove values < 20 and sortnumbers = [32, 54, 66, 11, 77, 10, 90]
numbers = [32, 54, 66, 11, 77, 10, 90]
# a. Remove values less than 20
filtered = []
for num in numbers:
    if num >= 20:
        filtered.append(num)

# b. Sort the filtered list
# Ascending order
ascending = sorted(filtered)

# Descending order
descending = sorted(filtered, reverse=True)

# c. Calculate average
total = 0
for num in filtered:
    total += num

average = total / len(filtered)

# Print results
print("Filtered List (>=20):", filtered)
print("Ascending Order:", ascending)
print("Descending Order:", descending)
print("Average:", average)
```

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023; AMD64) on win32

Type "help", "copyright", "credits" or "license()" for more

>>>

= RESTART: D:/python.py

Filtered List (>=20): [32, 54, 66, 77, 90]

Ascending Order: [32, 54, 66, 77, 90]

Descending Order: [90, 77, 66, 54, 32]

Average: 63.8

>>>

File Edit Format Run Options Window Help

# 6. Gadget list manipulation

Gadgets = ["Mobile", "Laptop", 10.0, "Marker", 3, 10, "Speakers", "Camera", 310.25]

strings = []

numbers = []

for item in Gadgets:

if type(item) == str:

strings.append(item)

elif type(item) == int or type(item) == float:

numbers.append(item)

print("Strings:", strings)

print("Numbers:", numbers)

print("String Asc:", sorted(strings))

print("String Desc:", sorted(strings, reverse=True))

sorted\_by\_len = []

for word in sorted(strings, key=len):

sorted\_by\_len.append(word)

print("String by Length:", sorted\_by\_len)

print("Number Asc:", sorted(numbers))

print("Number Desc:", sorted(numbers, reverse=True))

IDLE Shell 3.12.0

File Edit Shell Debug Options Window Help

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64-bit AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: D:/python.py

Strings: ['Mobile', 'Laptop', 'Marker', 'Speakers', 'Camera']

Numbers: [10.0, 3, 10, 310.25]

String Asc: ['Camera', 'Laptop', 'Marker', 'Mobile', 'Speakers']

String Desc: ['Speakers', 'Mobile', 'Marker', 'Laptop', 'Camera']

String by Length: ['Mobile', 'Laptop', 'Marker', 'Camera', 'Speakers']

Number Asc: [3, 10.0, 10, 310.25]

Number Desc: [310.25, 10.0, 10, 3]

>>>

|