

1. # The following line won't run because of a syntax error
print("hi")
-->print("hi")

2. # Exercise 2

""" The following lines won't run properly,
even if the syntax error in the line above is corrected,
because of a run-time error """

```
print(hello)
-->hello = "Hello, world!"
print(hello)
```

3. # Display a string (greeting message) directly
-->print("Hello, welcome to the program!")

4. # Display the contents of a string variable
-->greeting_message = "Hello, welcome to the program!"
print(greeting_message)

5. # Display the string which contains single quotes
Ex: Indian's

```
-->print("Indian's")
```

6. # Display the string which contains Double Quotes
Ex: Students, "Welcome to SOIS".

```
-->print("""Students, "Welcome to SOIS". """)
```

6. Read two numbers in (user input) and store as num1 and num2, Calculate the sum, difference, product, Quotient, remainder, power

```
-->num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
```

```
sum_result = num1 + num2
difference_result = num1 - num2
product_result = num1 * num2
```

```
if num2 != 0:
```

```
    quotient_result = num1 / num2
    remainder_result = num1 % num2
```

```
else:
```

```
    quotient_result = 'undefined (division by zero)'
    remainder_result = 'undefined (division by zero)'
```

```
power_result = num1 ** num2
```

```
print(f"Sum: {sum_result}")
print(f"Difference: {difference_result}")
print(f"Product: {product_result}")
print(f"Quotient: {quotient_result}")
print(f"Remainder: {remainder_result}")
print(f"Power: {power_result}")
```

7. check the value of num1 is integer or not?

-->

```
input_str = input("Enter the first number: ")
```

```
try:
    num1 = float(input_str) # Use float to handle cases where input might be a decimal number
except ValueError:
    print("Invalid input. Please enter a numeric value.")
    exit()
```

```
if num1.is_integer():
    num1 = int(num1) # Convert num1 to an integer if it is an integer value
    is_integer = True
else:
    is_integer = False
```

```
num2_str = input("Enter the second number: ")
```

```
try:
    num2 = float(num2_str)
except ValueError:
    print("Invalid input. Please enter a numeric value.")
    exit()
```

```
sum_result = num1 + num2
difference_result = num1 - num2
product_result = num1 * num2
```

```
if num2 != 0:
    quotient_result = num1 / num2
    remainder_result = num1 % num2
else:
    quotient_result = 'undefined (division by zero)'
    remainder_result = 'undefined (division by zero)'
```

```
power_result = num1 ** num2
```

```
print(f"Sum: {sum_result}")
print(f"Difference: {difference_result}")
print(f"Product: {product_result}")
print(f"Quotient: {quotient_result}")
print(f"Remainder: {remainder_result}")
print(f"Power: {power_result}")
```

```
if is_integer:
    print("num1 is an integer.")
else:
```

```
print("num1 is not an integer.")
```

8. convert into integer

-->

```
input_str = input("Enter a number to convert to an integer: ")
```

```
try:
```

```
    num = int(input_str)
```

```
    print(f"The integer value is: {num}")
```

```
except ValueError:
```

```
    print("Invalid input. Please enter a valid integer.")
```

9. Find the datatype for the variable num1 and num2.

-->input_str1 = input("Enter the first number: ")

```
input_str2 = input("Enter the second number: ")
```

```
try:
```

```
    num1 = float(input_str1)
```

```
    num2 = float(input_str2)
```

```
except ValueError:
```

```
    print("Invalid input. Please enter numeric values.")
```

```
    exit()
```

```
print(f"Data type of num1: {type(num1)}")
```

```
print(f"Data type of num2: {type(num2)}")
```

```
if num1.is_integer():
```

```
    num1 = int(num1)
```

```
if num2.is_integer():
```

```
    num2 = int(num2)
```

```
print(f"Updated data type of num1: {type(num1)}")
```

```
print(f"Updated data type of num2: {type(num2)}")
```

10. read the float value from the user and print the number rounded to 2 decimal places

-->

```
input_str = input("Enter a floating-point number: ")
```

```
try:
```

```
    num = float(input_str)
```

```
except ValueError:
```

```
    print("Invalid input. Please enter a valid floating-point number.")
```

```
    exit()
```

```
rounded_num = round(num, 2)
```

```
print(f"The number rounded to 2 decimal places is: {rounded_num}")
```

11. read the float value from the user and print the absolute value

-->

```
input_str = input("Enter a floating-point number: ")
```

```
try:
```

```
    num = float(input_str)
```

```
except ValueError:
```

```
    print("Invalid input. Please enter a valid floating-point number.")
```

```
    exit()
```

```
absolute_value = abs(num)
```

```
print(f"The absolute value is: {absolute_value}")
```

12. Store different type values in the variable

String

numeric

complex

list

dictionary

set

tuple

-->String value: Hello, World! (Type: <class 'str'>)

Integer value: 42 (Type: <class 'int'>)

Float value: 3.14159 (Type: <class 'float'>)

Complex value: (2+3j) (Type: <class 'complex'>)

List value: [1, 2, 3, 4, 5] (Type: <class 'list'>)

Dictionary value: {'name': 'Alice', 'age': 30, 'city': 'Wonderland'} (Type: <class 'dict'>)

Set value: {1, 2, 3, 4, 5} (Type: <class 'set'>)

Tuple value: (1, 2, 3, 4, 5) (Type: <class 'tuple'>)

13. Find the data type for the above variables

-->Data type of string_value: <class 'str'>

Data type of numeric_int: <class 'int'>

Data type of numeric_float: <class 'float'>

Data type of complex_value: <class 'complex'>

Data type of list_value: <class 'list'>

Data type of dictionary_value: <class 'dict'>

Data type of set_value: <class 'set'>

Data type of tuple_value: <class 'tuple'>

14. # Display the number of letters in the string

```
greeting = "Welcome to Python Programming"
```

-->

```
greeting = "Welcome to Python Programming"
```

```
letter_count = 0
```

```
for char in greeting:
```

```
if char.isalpha():  
    letter_count += 1
```

```
print(f"The number of letters in the string is: {letter_count}")
```

15. read the first name and last name from the user and combine first name and last name. combine name and greeting message

```
-->  
first_name = input("Enter your first name: ")  
last_name = input("Enter your last name: ")  
  
full_name = first_name + " " + last_name  
greeting_message = f"Hello, {full_name}! Welcome to our platform."  
print(greeting_message)
```

16. Display the string with space

Ex: firstname lastname

```
-->  
first_name = input("Enter your first name: ")  
last_name = input("Enter your last name: ")  
  
full_name = first_name + " " + last_name  
  
print(f"Full name: {full_name}")
```

17. Display first two characters from the name

```
-->  
name = input("Enter your name: ")
```

```
first_two_chars = name[:2]
```

```
print(f"The first two characters of your name are: {first_two_chars}")
```

18. Display last three characters from the name

```
-->name = "ExampleName"  
last_three_characters = name[-3:]  
print(last_three_characters)
```

19. Display 3rd character to last character

```
-->name = "ExampleName"  
substring = name[2:] # Starts at index 2 (which is the 3rd character) and goes to the end  
print(substring)
```

20. Display 3rd to 5th character

```
-->name = "ExampleName"  
substring = name[2:5] print(substring)
```

21. Create a list of food with two elements.

```
-->food_list = ["Pizza", "Burger"]
print(food_list)
```

22. Add one more to the food list using .append()

```
-->food_list = ["Pizza", "Burger"]
food_list.append("Pasta")
print(food_list)
```

23. Add two more food strings to food using .extend()

```
-->food_list = ["Pizza", "Burger", "Pasta"]
food_list.extend(["Sushi", "Tacos"])
print(food_list)
```

24. Count total number of items in the list

```
-->food_list = ["Pizza", "Burger", "Pasta", "Sushi", "Tacos"]
total_items = len(food_list)
print(total_items)
```

25. Print the first two items in food using slicing notation

```
-->food_list = ["Pizza", "Burger", "Pasta", "Sushi", "Tacos"]
first_two_items = food_list[:2]
print(first_two_items)
```

26. Print the last item in food using index notation

```
-->food_list = ["Pizza", "Burger", "Pasta", "Sushi", "Tacos"]
last_item = food_list[-1]
print(last_item)
```

27. Debug: Program is to check the given number is odd or even

```
number = input("Enter a number: ")
x = str(number)/2
if x == 0
    print("The number is Even.")
else
    print("The number is Odd.")
```

```
-->number = int(input("Enter a number: "))
if number % 2 == 0:
    print("The number is Even.")
else:
    print("The number is Odd.")
```

28. Debug: Program is to convert centigrade to Fahrenheit

```
c = input("Enter temperature in Centigrade: ")
f = 9*(int(c)/5 +32)
print("Temperature in Fahrenheit is: ", f)
```

```
-->c = float(input("Enter temperature in Centigrade: "))
f = 9 * (c / 5) + 32
print("Temperature in Fahrenheit is:", f)
```

29. Debug:

```
int = int(input("Enter the count of numbers: "))
i = 0
summ= 0
for i in range(count):
    x = int(input("Enter an integer: "))
    sum = sum + x
    avg = sum/count
print("The average is: ", avg)
-->count = int(input("Enter the count of numbers: "))
total_sum = 0 for i in range(count):
    x = int(input("Enter an integer: "))
    total_sum += x
avg = total_sum / count
print("The sum is:", total_sum)
print("The average is:", avg)
```

30. Prove : strings is not mutable

lists are mutable

-->Strings are Not Mutable

In Python, strings are immutable. This means that once a string is created, it cannot be changed. Any operation that seems to modify a string will actually create a new string.

```
s = "hello"
try:
    s[0] = "H"
except TypeError as e:
    print("Error:", e)
```

OUTPUT

Error: 'str' object does not support item assignment

Lists are Mutable

In Python, lists are mutable. This means you can change their contents without creating a new list.

```
lst = [1, 2, 3, 4, 5]
lst[0] = 10
print(lst)
```

Output:

[10, 2, 3, 4, 5]