practiceexamples

July 27, 2025

1. What is type casting? Give an example.

[1]:
```
x = "123"
y = int(x)
print(y, type(y))
```

```
123 <class 'int'>
```

2. Convert the following:

a. “123” to integer

[1]:
```
a = "123"
a_int = int(a)
print(a_int, type(a_int))
```

```
123 <class 'int'>
```

b. 25 to string

[3]:
```
b = 25
b_str = str(b)
print(b_str, type(b_str))
```

```
25 <class 'str'>
```

[ ]:
```
c. "3.14" to float
```

[4]:
```
c = "3.14"
c_float = float(c)
print(c_float, type(c_float))
```

```
3.14 <class 'float'>
```

[ ]:
```
3. What error occurs if you try int("abc")? Why? Correct it.
```

[5]:
```
# This will raise a ValueError because "abc" is not a number.
try:
    val = int("abc")
except ValueError as e:
    print("Error:", e)
```

```
Error: invalid literal for int() with base 10: 'abc'
```

[6]:
```
# Corrected version (example with valid numeric string):
val = int("456")
print(val)
```

```
456
```

4. How do you create a multi-line string in Python?

[7]:
```
multi_line = """This is a
multi-line string
in Python."""
print(multi_line)
```

```
This is a
multi-line string
in Python.
```

5. How do you include a quote inside a string?

[8]:
```
# Using escape character
quote_str = "He said, \"Python is awesome!\""
print(quote_str)
```

```
He said, "Python is awesome!"
```

[9]:
```
# OR using single quotes around the string
quote_str2 = 'He said, "Python is awesome!"'
print(quote_str2)
```

```
He said, "Python is awesome!"
```

6. Convert price = “99.99” to a float

[10]:
```
price = "99.99"
price_float = float(price)
print(price_float, type(price_float))
```

```
99.99 <class 'float'>
```

7. Create a string: I’m learning “Python” using escape characters

[11]:
```
sentence = "I\'m learning \"Python\""
print(sentence)
```

```
I'm learning "Python"
```

8. What is a raw string? What is its use?

[ ]:
```

```

1.Question: Write a Python program that takes a string as input and prints its length. Example Input: “Hello World” Expected Output: The length of the string is: 11

[2]:
```
input_string = input("Enter a string: ")
string_length = len(input_string)
print("The length of the string is:", string_length)
```

```
Enter a string:  Hello shubham
```

```
The length of the string is: 13
```

2.Question: Given the string data = “abcdefghijklmnop”, write Python code to: A. Get the first 5 characters. B. Get the last 3 characters. C. Get characters from index 2 to 7 (inclusive of 2, exclusive of 8). D. Get every second character. E. Reverse the string.

1 Given string

data = “abcdefghijklmnop”

[5]:
```
#A
data = "abcdefghijklmnop"
first_5 = data[:5]
print("A. First 5 characters:", first_5)
```

```
A. First 5 characters: abcde
```

[6]:
```
#B
data = "abcdefghijklmnop"
last_3 = data[-3:]
print("B. Last 3 characters:", last_3)
```

```
B. Last 3 characters: nop
```

[7]:
```
#C
data = "abcdefghijklmnop"
index_2_to_7 = data[2:8]
print("C. Characters from index 2 to 7:", index_2_to_7)
```

```
C. Characters from index 2 to 7: cdefgh
```

[8]:
```
#D
data = "abcdefghijklmnop"
every_second = data[::2]
print("D. Every second character:", every_second)
```

```
D. Every second character: acegikmo
```

[9]:
```
#E
data = "abcdefghijklmnop"
```

```
reversed_data = data[::-1]
print("E. Reversed string:", reversed_data)
```

```
E. Reversed string: ponmlkjihgfedcba
```

3.Question: If you have a string sentence = “This is a long sentence.”, how would you extract the word “long” using slicing?

[10]:
```
# sentence = "This is a long sentence."
sentence = "This is a long sentence."
word = sentence[10:14]
print("Extracted word:", word)
```

```
Extracted word: long
```

Index: 01234567890123456789012 3 Sentence: “This is a long sentence.” ↑ l(10) o(11) n(12) g(13)

4.Question: Write a Python program that takes a string as input and prints each character of the string on a new line using a for loop.

[11]:
```
user_input = input("Enter a string: ")
for char in user_input:
    print(char)
```

```
Enter a string:  Hello
```

```
H
e
l
l
o
```

5.Question: Modify the program in question 4 to also print the index of each character alongside the character.

[12]:
```
user_input = input("Enter a string: ")
for index, char in enumerate(user_input):
    print(f"Index {index}: {char}")
```

```
Enter a string:  Hello
```

```
Index 0: H
Index 1: e
Index 2: l
Index 3: l
Index 4: o
```

6.Question: You have the variables product = “Laptop” and price = 60,000. Use an f-string to print the following sentence: “The Laptop costs 60,000.”

[13]:
```
product = "Laptop"
price = 60000
```

```
print(f"The {product} costs {price:,}.")
```

```
The Laptop costs 60,000.
```

7.Question: Given the variables name = “vikas” and score = 85, use an f-string to print: “vikas’s score is 85.”

[14]:
```
name = "Shubham"
score = 85
print(f"{name}'s score is {score}.")
```

```
Shubham's score is 85.
```

8.Question: Using the .format() method, create the same output as in Question 6: “The Laptop costs 60,000.”

[15]:
```
product = "Laptop"
price = 60000

print("The {} costs {:,}.".format(product, price))
```

```
The Laptop costs 60,000.
```

9.Question: Write a Python program that takes a string as input and uses the enumerate() function to print each character along with its index.

[16]:
```
user_input = input("Enter a string: ")
for index, char in enumerate(user_input):
    print(f"Index {index}: {char}")
```

```
Enter a string:  Python

Index 0: P
Index 1: y
Index 2: t
Index 3: h
Index 4: o
Index 5: n
```

10.Question: Write a Python program that takes a string as input and uses the range() function along with a for loop to print each character of the string in reverse order of their index.

[18]:
```
user_input = input("Enter a string: ")
for i in range(len(user_input) - 1, -1, -1):
    print(f"Index {i}: {user_input[i]}")
```

```
Enter a string:  Shubham

Index 6: m
Index 5: a
Index 4: h
Index 3: b
```

```
Index 2: u
Index 1: h
Index 0: S
```

[ ]: