

Introduction to Programming SET A - AEC 01

Submitted by: Aishwarya M

Email: aishwarya.mukund@atriauniversity.edu.in

Submitted at: 9/19/2025, 9:55:57 AM

Question 1. With an example, differentiate between sep and end in the print() function.?

Answer:

Separate is the function in which you can separate the string/sentence with (" ") to get the output

Example :- sentence = ("to be or not to be")

```
print(sentence, sep(, ))
```

O/P = 'to', 'be', 'or', 'not', 'to', be'

end function is where you can add anything at the end of the word

Example :- word = cat

```
print(word, end="-")
```

O/P = c-a-t-

Question 2. What are lists in Python? With an example, explain why they are useful.?

Answer:

List is where you can store your values. It can be a string, integer, etc

Example :- groceries = ["apple", "banana", "chocolate"]

```
print(groceries)
```

here the output will print the list of groceries.

you can add, remove and sort anything from the lists

Question 3. Give any two examples of built-in functions and explain what they do.?

Answer:

Built in functions are pre-given by python itself

example :-

1) definition function(def):- it defines the function to make sure that the code is based on the def function

2) import function

Question 4. a. What are conditional statements in Python?

b. How do if, elif, and else work together in a program? Give an example.?

Answer:

a) Conditional Statements in python is the process in which whether the given statement is true or false.

Example :-The boolean function (bool) decides whether the given function is true or false.

It has 3 more conditions :-and, or, not

"and" checks if both the statement is true

"or" checks if one of the statement is true

"not" checks if both the statement is false

b) if statement :- tells if the statement is true.

elif statement = checks if the statement is true if the previous statement is false

else statement = says whether the conditional is true if both id and elif are false.

Example :-

```
num = 5
```

```
if num<0:
```

```
print ("The number is lesser than 5")
```

```
elif num ==0
```

```
print("The number is equal to 5")
```

```
else:
```

```
print("The number is greater than 5")
```

Question 5. Consider the function below:

```
def greet(name):  
    return "Hello " + name
```

- a. What is the name of the function?
- b. What are the parameters of the function?
- c. What is the return value when we call greet("Alice")?

Answer:

- a) The name of the function is def
- b) "Hello" is the parameter of the function
- c) "Alice" will be the return value which will go to the name.

**Question 6. words = ["abc"]
scale = 4.0
print(f"Scaled size: {len(words) * scale}")**

Predict and explain the output.?

Answer:

Explanation :-

Here, the word is "abc" which is a string. The scale of the word is 4.0. So, if we print

```
print(f"Scaled size :{len(words) * scale}")
```

as the formatted string is being used here to get the output more clearly, the output will come clearly.

O/P = 12.0 as the scale is a float value

Question 7. if x % 2 == 0:

```
print("Even")
```

```
else:
```

```
print("Odd")
```

a. What will be the output if x = 15?

b. Can you summarise the purpose of this code in 1-2 sentences?

Answer:

a) if the value of x is 15, then the remainder (as modulo function is used) will be 0 as the modulo function gives the remainder.

O/P = even

it will print even as the remainder is 0.

b) x = 15

if 15%2 == 0: (modulo function gives the remainder as the answer)

then it will print that the number is even. If it's not, then it'll print that the number is odd.

Question 8. for i in range(4):

```
for j in range(1):
```

```
print(i * j)
```

a. List all the values that i and j take in this nested loop.

b. What is the output of the code?

c. Suggest a change so that the entire output prints in a single line instead of multiple lines.?

Answer:

a) i = 0 1 2 3

j = 0

b) 0 0

0 1

0 2

0 3

4 outputs will be given

c) To print the entire output in one line instead of multiple lines, you can use a while loop

```
num = 4
```

```
num = 1
```

```
while i < 4 and i < 1:
```

```
print(i*j)
```

```
i = i + 1 (you can increment this as i +=1)
```

Question 9. def count_vowels(word):

```
count = 0
```

```
for ch in word:
```

```
if ch in "aeiou":
```

```
count += 1
```

```
return count
```

a. What will this function return for each of the following inputs: "Education", "aeiou", "rhythm"?

b. Explain why in is used instead of == in the above code.

c. Is this a good way to count vowels? Suggest a small improvement to this function.?

Answer:

a) "Education" will return to the def function

"aeiou" will return the ch as we are counting the vowels in this.

"rhythm" will return the count value.

b) in is used instead of the == as == is used for equality of the variable whereas in is used for string to find out whether in the word, the specific letter is there or not. We can even say how many times this letter occurred due to count function.

c) The function might not give the output that it should come. We can use the while loop in this as a for loop will have fixed number of iterations in it. While loop however will have a continuous output until we get the right answer for it.

Question 10. def is_prime(number):

```
start = 2  
while start  
print(start % 2)  
start = start + 1  
is_prime(3)  
is_prime(5)
```

a. Predict and explain the output.

b. This function clearly does not check for prime numbers. What does it actually do?

Answer:

a) is_prime has been defined here. So, the start number is 2. In a while loop, if start is less than the number given; then it will print(start%2). In this, the modulo function(%) is being used to show the remainder as the answer. (2%2=0) . Now, the start number will be added by 1.

is_prime(3) and is_prime(5); 3 and 5 are the prime numbers but they are not checked properly.

b) This function doesn't give the checking of prime numbers as it checks if the starting number is a prime number.

```
def is_prime(3)
```

```
start = 2
```

```
while start < number
```

```
print(start%2 == 0)
```

```
start = start + 1
```

O/P = 3%2 is 1

```
def is_prime(5)
```

```
start = 2
```

```
while start < number
```

```
print(start%2 == 0)
```

```
start = start + 1
```

O/P = 5%2 is 1

Here it actually finds out whether the given prime number is divisible by the start number if the start number is added by 1.

Question 11. num = -3

```
while num  
    print(num)
```

You expect this program to print integers between -3 and 3, but it does not work. What is the issue and how can you fix it?

Answer:

=> the num is -3 which is negative.

=> while the number -3 is less than 3, they will expect to print the number -3 but it wont.

=> the issue is that the number -3, will not be printed as in the while loop, the statement is true. You can fix it by using a for loop.

```
for i in range (-3, 3):
```

```
    print(i)
```

O/P= -3, -2, -1, 0, 1, 2

Question 12. The following Python code has an error:

```
score = input("Enter your score: ")  
print(age + 5)
```

What error will occur when this program runs? How would you correct the code so it works properly?

Answer:

An Indentation error will occur. This is because there is no gap after the print statement. And also age should be a variable.

code:-

```
age =18
```

```
score = input("Enter your score: ")
```

```
print(age + 5)
```

Question 13. Problem: Given a duration in minutes entered by the user, you need to convert it into hours and minutes. For example, 130 minutes = 2 hours and 10 minutes.

a. Which math operators do you need?

b. Explain how you would solve this problem in steps. (Code is not required.)?

Answer:

a) `math.ceil` (rounds up the number) and `math.floor` (rounds down the number) is required to solve this problem

b) Here the time 130 minutes is being converted to 2 hours and 10 minutes. So for this you can use these two functions because in minutes or hours, a float value will be there.

So just to get the nearest value, 2 math functions are used here.

You can also use concatenation to add the minutes and second statements together so that it will be easier to find out.

You can use an input value just to make sure just to write the statement. You can also use `int()` value for this as well as the numbers are integers

Question 14. Problem: Given a sentence typed by the user, you need to find the longest word in the sentence.

a. Do you need a loop? Why, or why not?

b. Explain how you would solve this problem in steps. (Code is not required.)?

Answer:

a) Yes you need a loop to find out which is the longest word in the sentence because you need to loop through every variable that is there and you can find out its length [`len()`] of the sentence.

Then you have to loop again just to find the length of each word.

b) 1. type the sentence first

1. 2. first find out how many words are there in the sentence

2. 3. then loop through each and every word and find out which word is the longest by using the length function [`len()`].

3. 4. then by this you have to specify which word is the longest and how long is it.

4. 5. After looping through it, it will print the longest word and its length.