QB365 Question Bank Software Study Materials

Strings and String Manipulation Important 2, 3 & 5 Marks Questions With Answers (Book Back and Creative)

12th Standard

Computer Science

Total Marks : 75

2 Marks

10 x 2 = 20

1) What is String?

Answer : (i) String is a data type in python, which is used to handle array of characters.

(ii) String is a sequence of Unicode characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

(iii) **Example :**

Welcome to learning Python"
"Welcome to learning Python"
""Welcome to learning Python""

2) Do you modify a string in Python?

Answer : Strings in python are immutable. That means, once you define a string moditications or deletion is not allowed. However, we can replace the existing string entirely with the new string.

3) How will you delete a string in Python?

Answer : Python will not allow deleting a particular character in a string. Whereas you can remove entire string variable using del command.

Example :

```
> > > strl="How about you"
> > > print (str1)
       How about you
> > > del str1
> > > print (str1)
Traceback (most recent call last):
        File "", line 1, in
               print (strl)
```

NameError : name 'str1' is not defined

4) What will be the output of the following python code?

```
str1 = “School”
print(str1*3)
```

Answer : **Output :** School School School

5) What is slicing?

Answer : (i) Slice is a substring of a main string. A substring can be taken from the original string by using [ ]. operator and index or subscript values.

(ii) Thus, [ ] is also known as slicing operator.Using slice operator, you have to slice one or more substrings from a main string.

General format of slice operation :

Str [start : end]

6) What is the use of formatting operator?

Answer : The formatting operator % is used to construct strings, replacing parts of the strings with the data stored in variables.

7) Match the following.

	Escape sequence	Description

i)	\a	(1)	Backspace
ii)	\b	(2)	Horizontal tab
iii)	\n	(3)	Bell
iv)	\t	(4)	Line feed

Answer : 3, 1,4, 2

8) What is meant by positive subscript?

Answer : The positive subscript 0 is assigned to the first character and n-1 to the last character, where n is the number of characters in the string.

9) What is concatenation string?

Answer : Joining of two or more strings is called as concatenation. The plus (+) operator is used to concatenate strings in Python.

```
> > > "Welcome" + "Students"
'WelcomeStudents'
```

10) What is striding when slicing string?

Answer : When the slicing operation, we can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string. The default value of stride is 1.

3 Marks 10 x 3 = 30

11) Write a Python program to display the given pattern

```
C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
C O M
C O
C
```

Answer : Str 1 = "COMPUTER"

```
index = 0
for i in str 1:
print (str 1[:index -1])
index =1
```

12) Write a short about the followings with suitable example:

(a) capitalize( )

(b) swapcase( )

Answer :

Syntax	Description	Example
(a) capitalize()	Used to capitalize the first character of the string	> > > city="chennai" > > > print(city, capitalize( )) Chennai
(b) swapcase( )	It will change case of every character to its opposite case vice-versa.	> > > strl="tAmilL NaDu" > > > print(strl.swapcase( )) TaMIl nAdU

13) What will be the output of the given python program?

```
str1 = "welcome"
str2 = "to school"
str3 = str1[:2] + str2[len(str2)-2:]
print (str3)
```

Answer : **Output:**

```
Weool
```

14) What is the use of format( )? Give an example.

Answer : (i) The format( ) function used with strings is very versatile and powerful function used for formatting strings.

(ii) The curly braces {} are used as placeholders or replacement fields which get replaced along with format( ) function.

Example:

```
num1 = int (input("Number 1:"))
num2 = int (input("Number 2:"))
print ("The, sum of {} and {} is {}". format (num1, num2, (num1 + num2)))
```

Output:

```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88.
```

15) Write a note about count( ) function in python.

Answer :

Syntax	Description	Example
count (str,beg,end)	Returns the number of substrings Occurs within the given range. Remember that substring may be a single character. Range (beg and end) arguments are optional. If it is not given, python searched in whole string. Search is case sensitive.	> > > str1="Raja Raja Chozhan" > > > print(str1.count("Raja")) 2 > > > print(str1.count("r")) 0 > > > print(str1.count("R")) 2 > > > print(str1.count("a")) 5 > > > print(str1.count('a',0,5)) 2 > > > print(str1.count('a',11)) 1

16) Write the usage of the following format string characters.

(i) % c

(ii) % d (or) % i

(iii) % s

Answer : (i) character

(ii) Signed decimal integer

(iii) String

17) Write the formats string characters for the following

(i) Exponential notation

(ii) Floating point numbers

(iii) Short numbers in exponential notation

Answer : (i) % e or % E

(ii) % f

(iii) % g or % G

18) How will you accessing characters in a string?

Answer : Once we define a string, Python allocates an index value for its each character. These index values are otherwise called as subscripts which are used to access and manipulate the strings. The subscript can be positive or negative integer numbers.

The positive subscript 0 is assigned to the first character and n-1 to the last character, where n is the number of characters in the string. The negative index assigned from the last character to the first character in reverse order begins with -1.

Example:

String	S	C	H	O	O	L
Positive subscript	0	1	2	3	4	5
Built in functions	-6	-5	-4	-3	-2	-1

19) Write a program to delete a particular character in strings.

Answer : Similar as modification, python will not allow deleting a particular character in a string. Whereas you can remove entire string variable using del command.

Codelines to delete a particular character in a string:

```
> > > str1="How are you"
> > > del str1 [2]
Traceback (most recent call last):
File"< pyshell #7)",line1,in< module >
del str1 [2]
```

Output:

TypeError:'Str' object doesn't support item deletion.

20) Write a program to display the following pattern.

```
*
* *
* * *
* * * *
* * * * * *
```

Answer : Str1 = '*'

```
i = 1
While i< =5:
print(str1*i)
i+=1
```

Output:

```
*
* *
* * *
* * * *
* * * * *
```

5 Marks 5 x 5 = 25

21) Explain about string operators in python with suitable example.

Answer : **String Operators:**

Python provides the following operators for string operations. These operators are useful to manipulate string.

(i) Concatenation (+):

Joining of two or more strings is called as Concatenation. The plus (+) operator is used to concatenate strings in python.

Example:

```
> > > "welcome" + "Python"
'welcome python'
```

(ii) Append (+ =):

Adding more strings at the end of an existing string is known as append. The operator += is used to append a new string with an existing string.

Example:

```
> > > str1 = "welcome to"
> > > str1 + = "Learn Python"
> > > print (str1)
Welcome to Learn Python
```

(iii) Repeating (*):

The multiplication operator (*) is used to display a string in multiple number of times.

Example:

```
> > >  str1= "welcome "
> > > print (str1 *4)
Welcome Welcome Welcome Welcome
```

(iv) String slicing:

Slice is a substring of a main string.

A substring can be taken from the original string by using [ ] operator and index or subscript values. Thus [ ] is also known as slicing operator. Using slice operator, we have to slice one or more substrings from a main string.

General format of slice operation:

str[start:end]

Where start is the beginning index and end is the last index value of a character in the string. Python takes the end value less than one from the actual index specified.

Example: I

Slice a single character from a string

```
> > > str 1 = "THIRUKKURAL"
> > > print (str 1 [0])
      T
```

(v) Stride when slicing string:

22) When the slicing operator, we can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string.

23) The default value of stride is 1

24) Python takes the last value as n-1

25) We can also use negative value stride, to print data in reverse order.

Example :

```
> > > str1 = "Welcome to learn python"
> > > print(str1[::-2])
```

Output :

```
nhy re teolW
```

26) Write a python program to print the following pattern

```
*
**
***
****
*****
```

Answer :

```
strl=' * '
i=1
while i< = 5:
print (strl*i)
i+=1
```

27) Explain how will you create and access characters in a string with an example?

Answer : String is a data type in python, which is used to handle array of characters. String is a sequence of unicode characters that may be a combination of letters, numbers or special symbols enclosed within single, double or even triple quotes.

Example:

'Welcome to learning python'

"Welcome to learning python"

""Welcome to learning python""

In Python, strings are immutable, it means once we define a string, it cannot be changed during execution.

Creating Strings:

A string in Python can be created using single or double or even triple quotes. String in single quotes cannot hold any other single quoted character in it, because the compiler will not recognize where to start and end the string. To overcome this problem, we have to use double quotes. Strings which contains double quotes should be define within triple quotes. Defining strings within triple quotes also allows creation of multiline strings.

Once we define a string, Python allocate an index value for its each character. These index values are otherwise called as subscripts which are used to access and manipulate the strings. The subscript can be positive or negative integer numbers. The positive subscript 0 is assigned to the first character and n-1 to the last character, where n is the number of characters in the string. The negative index assigned from the last character to the first character in reverse order begins with - 1.

Example:

String	S	C	H	O	O	L
Positive subscript	0	1	2	3	4	5
Negative subscript	-6	-5	-4	-3	-2	-1

Program to access each character with its positive subscript of a given string:

```
Str1 = input("Enter a string:")
index = 0,
for i in str1
print("Subscript[",index;"]",i)
index+=1
```

Output:

```
Enter a string: Welcome
Subscript [0]: W
Subscript [1]: e
Subscript [2]: I
Subscript [3]: c
Subscript [4]: o
Subscript [5j: m
Subscript [6]: e
```

Program to access each character with its negative subscript of a giving string:

```
str1 = input("Enter a string")
index = -1
while index > =-(len(str1)):
print("Subscript[",index,"]:" + str1 [index])
index + = -1
```

Output:

```
Enter a string: Welcome
Subscript [-1]: e
Subscript [-2]: m
Subscript [-3]: o
Subscript 1-4): c
Subscript [-5]:l
Subscript [-5]: e
Subscript [-71:W
```

28) Explain in detail about built-in function.

Answer :

Syntax	**Description**	**Example**
1. len(str)	Returns the length of the string	> > > A = "Corporation" > > > print (len(A)) 11
2. capitalize()	Used to capitalize the first character of the string.	> > > city = "Chennai" > > > print(city.capitalize( )) Chennai
3. center(width, fillchar)	Returns a string with the original string centered to a total of width columns and filled with fillchar in columns that do not have characters	> > > str1 = "Welcome" > > > print(str1.center (15, *')) *** Welcome***
4. find(sub[,start [,end])	The function is used to search the first occurrence of the sub-string in the given string. it returns the index at which the substring starts. It returns -1 if the substring does not occur in the string.	> > > strl= 'mammals' > > > strl.find ('ma') 0 On omitting the start parameters, the function starts the search from the beginning. > > > str1.find('ma'. 2) 3 > > > str1.find('ma'. 2,4) -1 Displays - 1, because the substring could not be found between the index 2 and 4-1. > > > str1. find ('ma',2,5) 3
5. isalnum()	Returns True if the string contains only letters and digits. It returns false. If the string contains any special character like, @, #, *, etc	> > > str1='Save Earth' > > > str1.isalnum( ) false The function returns false as space is an alphanumeric character. > > > 'Save/Earth' isalnum( ) True.
6. isalpha()	Returns True if the string contains only letters. Otherwise return false	> > > 'Click 123'.isalpha( ) False. > > > 'Python'.isalpha( ) True
7. isdigit()	Returns True if the string contains only numbers. Otherwise, it returns False.	> > > str1='Save Earth' > > > print(str1.isdigit( )) False
	Returns the exact copy of the string	> > > str1= "SAVE

8. lower()	with all the letters in lowercase.	EARTH', > > > print(str1.lower( )) save earth
9. is lower()	Returns True if the string is in lowercase	> > > str1='Welcome', > > print (strl.islower( )) True
10. isupper()	Returns True if the string is in uppercase	> > > str1='welcome' > > > print(str1.isupper( )) False
11. upper()	Returns the exact copy of the string with all letters in uppercase	> > > strl='Welcome' > > > print (str.upper( )) WELCOME
12. title()	Returns a string in title case	> > > str1='education department' > > > print(str1.title ( )) Education Department
13. swapcase()	It will change case of every character to its opposite case vice-versa.	> > > str1= "tAmilNaDu" > > > print(str1.swapcase( )) TaMIL nAdu
14. count(str,beg,end)	Returns the number of substrings occurs within the given range. Remember that substring may be as single character. Range arguments are optional. If it is not given, python searched in whole string. Search is case-sensitive.	> > > str1= "Raja Raja chozhan" > > > print(str1.com('Raja')) 2 > > > print(str1.count('r')) 0 > > > print (str1.count('R')) 2 > > > print (str1.count('a')) 5 > > > print (str1.count('a',0,5)) 2 > > > print(str1.count('a',11)) 1
15. ord (char)	Returns the ASCII code of character.	> > > ch = 'A' > > > print(ord (ch)) 65 > > > print (ord ('B')) 66
16. chr(ASII)	Returns the character represented by a ASCII	> > > ch =97 > > > print(chr(ch)) a > > > print(chr(87)) w

29) Write a program to remove vowels from the given string.

Answer :

```
def rem_ vowels (s):
temp_str ="
for i in s:
if i in "aAeEiOoOuU":
pass
else:
temp_str + =i
print("The string without vowels:",temp_str)
str1 = input ("Enter a string")
rem_vowels(str1)
```

Output:

```
Enter a string: Mathematical functions of Computer Science
The string without vowels: Mthmtcl fndtns fcmptr Scnc
```