

Strings

Very Short Answer type Questions [1 mark each]

Question 1:

Explain capitalize() method in Python.

Answer:

The method capitalize() returns a copy of the string with only its first character capitalized.

Question 2:

Write the syntax for capitalize() method.

Answer:

Following is the syntax for capitalize() method : str.capitalize()

Question 3:

What value will be returned by center (width, fillchar) method in Python.

Answer:

The method center() returns centered in a string of length width. Padding is done using the specified fillchar.

Question 4:

What are the two parameters of center() method.

Answer:

width — This is the total width of the string,
fillchar — This is the filler character

Question 5:

Describe the count(str, beg=0,end= len(string))

Answer:

The method count() returns the number of occurrences of substring sub in the range [start, end].

Question 6:

Describe the decode (encoding='UTF8', errors='strict')

Answer:

The method decode() decodes the string using the codec registered for encoding.

Question 7:

What do you n .an by encode(encoding= 'UTF- 8,errors='strict')



Answer:

The method `encode()` returns an encoded version of the string. Default encoding is the current default string encoding.

Question 8:

What do you mean by `endswith(suffix, beg=0, end=len(string))`

Answer:

The method `endswith()` returns `True` if the string ends with the specified suffix, otherwise return `False`

Question 9:

Write the syntax for `find()` method

Answer:

Following is the syntax for `find()` method :

```
str.find(str, beg=0 end=len(string))
```

Question 10:

Write the output of the following code.

```
#!/usr/bin/py thon
str1 = "this is string example... ,wow!!!";\
str2 = "exam";
print str1.find(str2);
print str1.find(str2,10);
print str1.find(str2, 40);
```

Answer:

15 15 -1

Question 11:

Write the syntax for `isalnum()` method.

Answer:

Following is the syntax for `isalnum()` method :

```
str.isalnum( )
```

Question 12:

Write the output of the following code.

```
#!/usr/bin/python
str = "this2009"; # No space in this string print str.isalnum( );
str = "this is string example....wow!!!";
print str.isalnum( );
```

Answer:

True False



Question 13:

Write the syntax for isalpha() method.

Answer:

Following is the syntax for isalpha() method :

str.isalpha()

Question 14:

Write the output of the following code.

```
#!/usr/bin/python
```

```
str = "this"; # No space & digit in this string print str.isalpha( );
```

```
str = "this is string example....wow!!!"; print str.isalpha( );
```

Answer:

True False

Question 15:

Describe the isdigit() method

Answer:

The method isdigit() checks whether the string consists of digits only

Question 16:

Why we use islower() method in python?

Answer:

The method islower() checks whether all the case-based characters (letters) of the string are lowercase

Question 17:

Describe the isspace() method

Answer:

The method isspace() checks whether the string consists of whitespace.

Question 18:

Write the output of the following code.

```
#!/usr/bin/python
```

```
str = " ";
```

```
print str.isspace( );
```

```
str = "This is string example....wow!!!";
```

```
print str.isspace( );
```

Answer:

True

False

Question 19:

Write the output of the following code.

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print str.ljust(50, '0');
```

Answer:

```
This is string example
...wow!!!00000000000000000000
```

Question 20:

Write the output of the following code.

```
#!/usr/bin/python
str = "Waltons Technology....wow!!!";
print "str.upper() : "str.upper()
```

Answer:

```
str.upper() : Waltons Technology ....WOW!!!
```

Question 21:

Rectify the error (if any) in the given statements.

```
>>>str = "Hello World"
>>>str[5] = 'p'
```

Answer:

```
Strings are immutable. So convert to 2.
list >>>s = list (str)'p')
>>>s [5]='p'
```

Question 22:

Give the output of the following state-ments :

```
>>>str = 'Honesty is the best policy"
>>>str.replace ('o'.'*')
```

Answer:

```
H*nesty is the best p*licy.
```

Short Answer type Questions [2 mark each]

Question 1:

What do you mean by string in Python ?

Answer:

Strings are amongst the most popular types in Python. We can create them simply by characters in quotes. Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable. For example :



```
var1 = 'Waltons Technology!'
var2 = "Python Programming"
```

Question 2:

What is indexing in context to Python strings ? Why is it also called two-way indexing ?

Answer:

In Python strings, each individual character is ! given a location number, called "index" and this process is called "indexing". Python allocates indices in two directions :

1. in forward direction, the indexes are numbered as 0,1, 2, ..., length-1.
 2. in backward direction, the indexes are numbered as -1, -2, -3,... length.
- This is known as "two-way indexing".

Question 3:

What is a string slice ? How is it useful ?

Answer:

A sub-part or a slice of a string, say s, can be obtained using s[n : m] where n and m are integers. Python returns all the characters at indices n, n+1, n+2,... m-1.

For example,

'Oswaal Books' [1 : 4] will give 'swa'

Question 4:

How you can "update" an existing string ?

Answer:

You can "update" an existing string by (re) assigning a variable to another string.

The new value can be related to its previous value or to a completely different string altogether.

Following is a simple example :

```
# !/usr/bin/python
var1 = 'Hello World!'
print("Updated String:-",var i[:6] + 'Python')
```

Question 5:

Describe Triple Quotes in Python.

Answer:

Python's triple quotes comes to the rescue by allowing strings to span multiple lines, including verbatim NEWLINES, TABs, and any other special characters. The syntax for triple quotes consists of three consecutive single or double quotes.

```
# !/usr/bin/py thon
```

```
para str = """this is a long string that is made up of several lines and non-printable characters
such as
```

```
TAB ( \t ) and they will show up that way when displayed.
```



NEWLINEs within the string, whether explicitly given like this within the brackets [\n], or just a NEWLINE within the variable assignment will also show up.

```
"""
```

```
print para_str;
```

Question 6:

Define raw string with example.

Answer:

Raw strings don't treat the backslash as a special character at all. Every character you put into a raw string stays in the way you wrote it :

```
#!/usr/bin/python
print 'C:\\nowhere'
```

When the above code is executed, it produces the following result :

```
C:\nowhere
```

Now let's make use of raw string. We would put expression in r'expression' as follows :

```
#!/usr/bin/python
print r'C:\\nowhere'
```

When the above code is executed, it produces the following result :

```
C:\\nowhere
```

Question 7:

Explain Unicode String with example.

Answer:

Normal strings in Python are stored internally as 8-bit ASCII, while Unicode strings are stored as 16-bit Unicode. This allows for a more varied set of characters, including special characters from most languages in the world.

Example.....

```
#!/usr/bin/python
print u'Hello, world!'
```

Question 8:

Describe isdecimal() with example.

Answer:

The method isdecimal() checks whether the string consists of only decimal characters. This method is present only on Unicode objects.

Note : To define a string as Unicode, one simply prefixes a 'u' to the opening quotation mark of the assignment.

Below is the example.

Syntax :

Following is the syntax for isdecimal() method :

```
str.isdecimal( )
```

Question 9:

Explain zfill (width) with Syntax and Return Value

Answer:

The method zfill() pads string on the left with zeros to fill width.

Syntax : str.zfill(width)

Parameters: This is final width of the string. This is the width which we would get after filling zeros. Return Value: This method returns padded string

Question 10:

Write the output of the following code

```
# !/usr/bin/python
str = "this is string example....wow!!!";
print str.zfill(40);
print str.zfill(50);
```

Answer:

On compiling and running the above program, this will produce the following result :

OOOOOOOOthis is string example....wow!!! 00000000000000000000this is string example....
wow!!!

Question 11:

Write the output of the following code

```
# !/usr/bin/python
from string import maketrans. # Required to call maketrans. function.
intab = "aeiou" outtab = "12345"
trantab = maketrans(intab, outtab) str = "this is string example....wow!!!"; print
str.trAns.late(trantab, 'xm');
```

Answer:

The given code will produce following result :

th3s 3s str3ng 21pl2....w4w!M

Question 12:

Describe the following method trans.late(table, deletechars="")

Answer:

The method translate() returns a copy of the string in which all characters have been translated using table (constructed with the maketrans() function in the string module), optionally deleting all characters found in the string deletechars.

Question 13:

Give an example of title() in Python

Answer:

The following example shows the usage of title() method

```
# !/usr/bin/python
```

```
str = "this is string example....wow!!!";  
print str.title( );
```

On compile and run the above program, this will produce the following result :
This Is String Example....Wow!!!

Question 14:

Give an example of `swapcase()` in Python

Answer:

The following example shows the usage of `swapcase()` method.

```
# !/usr/bin/py thon  
str = "this is string example....wow!!!";  
print str.swapcase( );  
str = "THIS IS STRING EXAMPLE....WOW!!!";  
print str.swapcase( );
```

This will produce the following result :

```
THIS IS STRING EXAMPLE....WOW!!!  
this is string example....wow!!!
```

Question 15:

Define `strip ([chars])` with its syntax

Answer:

The method `strip()` returns a copy of the string in which all chars have been stripped from the beginning and the end of the string (default whitespace characters).

Syntax: `str.strip([chars]);`

Question 16:

Explain Parameters of `str.startswith(str, beg=0,end=len(string));`

Answer:

`str` — This is the string to be checked.

`beg` — This is the optional parameter to set start index of the matching boundary.

`end` — This is the optional parameter to set end index of the matching boundary.

Question 17:

Explain Parameters of `str.rjust(width[, fillchar])`

Answer:

`width` — This is the string length in total after padding.

`fillchar` — This is the filler character, default is a space.

Question 18:

Write the output of the given Python code # !/usr/bin/python

```
str = "this is really a string example.... wow!!!";  
str = "is";
```




```
print str.rfind(str);
print str.rfind(str, 0,10);
print str.rfind(str, 10, 0);
print str.find(str);
print str.find(str, 0,10);
print str.find(str, 10, 0);
```

Answer:

Above code will produce the following result :

```
5
5
-1
2
2
-1
```

Question 19:

Write the output of the given code `#!/usr/bin/python`

```
str = "this-is-real-string-example....wow!!!";
print "Min character: " + min(str);
str = "this-is-a-string-example....wow!!!";
print "Min character: " + min(str);
```

Answer:

```
Min character: !
Min character: !
```

Question 20:

Write the output of the given code `#!/usr/bin/python`

```
str = "this is really a string example....wow!!!";
print "Max character: " + max(str);
str = "this is a string example....wow!!!";
print "Max character: " + max(str);
```

Answer:

```
Output
Max character: y
Max character: x
```

Question 21:

Describe the function `maketrans()`

Answer:

The method `maketrans()` returns a translation table that maps each character in the intab string into the character at the same position in the outtab string. Then this table is passed to the `translate()` function.

Syntax : `str.maketrans(intab, outtab)];`



Question 22:

Write the output of the following code

```
#!/usr/bin/py thon
str = " this is string example....wow!!! "; print str.lstrip( );
str = "88888888this is string example....wow!!!8888888";
print str.lstrip('8');
```

Answer:

Output

this is string example....wow!!!

this is string example...,wow!!!8888888

Question 23:

Study the given script

```
defmetasearch( ):
import re
p=re.compile('sing+')
searchl=re.search(p,' Some singers sing well')
if searchl:
match=searchl.group( )
index=searchl.start( )
lindex=search 1 ,end( )
print "matched", match, "at index", index ,"ending at", lindex
else:
print "No match found"
metasearch( )
```

What will be the output of the above script if search() from the re module is replaced by match () of the re module. Justify your answer

Answer:

The output would be "N match found"

Justification : re.search() will attempt the pattern throughout the string, until it finds a match. re.match() on the other hand, only attempts the pattern at the very start of the string.

Example :

```
>>>re.match("d", "abcdef") # No match
>>>re.search("d", "abcdef") # Match
```

Question 24:

What will be the output of the script mentioned below? Justify your answer, def find()):

```
import re
p=re.compile(' sing+')
searchl=p.findall('Some singer sing well')
print search
```



Answer:

Output : ['sing', 'sing']

Justification : fmdall() finds all occurrences of the given substring with metacharacter.

Long Answer type Questions [4 mark each]

Question 1:

What is the concept of immutable strings ?

Answer:

Strings are immutable means that the contents of string cannot be changed after it is created.

For example :

```
>>> str = 'Meney'
```

```
>>> str [3] = 'h'
```

Type Error : 'str' object not support item assignment Python does not allow to change a character in a string. So an attempt to replace 'e' in the string by 'h' displays a Type Error.

Question 2:

What do you understand by traversing a string ? Ans. Traversing a string means accessing all the elements of the string one after the other by using the subscript. A string can be traversed using for loop or while loop.

For example :

```
A = 'Python'
```

```
i = 0
```

```
while i < len(A) :
```

```
    print A[i]
```

```
    i = i + 1
```

Output :

P

y

t

h

o

n

Question 3:

Write a program to check whether the string is a palindrome or not.

Answer:

```
def palindrom ( ) :
```

```
    str = input ("Enter the string")
```

```
    l = len (str)
```

```
    P = l - 1
```

```
    inex = 0
```



```

while (index < p) :
if (str[index] == str [p]):
index = index + 1
p = p-1
else :
print "String is not a palindrom" break
else :
print "String is a palindrom"

```

Question 4:

Write a program to count number of 's' in the string 'successor'.

Answer:

```

def letcount ( ) :
word = 'successor'
count = 0
for letter in word :
if letter == 's' :
count = count + 1
print (count)

```

Question 5:

Write a program to determine if the given word is present in the string.

Answer:

```

def wsearch ( ) :
imprt re
word = 'good'
search1 = re.search (word, 'I am a good person')
if search1 :
position = search1.start ( )
print "matched", word, "at position", position
else :
print "No match found"

```

Question 6:

Input a string "Green Revolution". Write a script to print the string in reverse.

Answer:

```

def reverseorder(list 1) :
relist = [ ]
i = len (list 1) -1
while i >= 0 :
relist.append (list [i])
i = i -1
return relist

```

Question 7:

Write a program to print the pyramid ?

Answer:

```
num = eval (raw_input ("Enter an integer from 1 to 5:"))
if num < 6 :
    for i in range (1, num + 1):
        for j in range (num-i, 0,-1):
            print (" ")
        for j in range (i, 0, -1):
            print (j)
        for j in range (2, i+1):
            print (j)
        print (" ")
    else :
        print ("The number entered is greater than 5")
```

Output :

```
1
2 1 2
3 2 1 2 3
4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

Question 8:

Write the syntax of `isdecimal()` and give suitable example

Answer:

The method `isdecimal()` checks whether the string consists of only decimal characters. This method are present only on Unicode objects. Below is the example.

Syntax

```
str.isdecimal()
```

Example

```
# !/usr/bin/python
str = u"this2009";
print str.isdecimal();
str = u"23443434";
print str.isdecimal();
```

This will produce the following result :

False

True

Question 9:

Write the output of the following python code `#!/usr/bin/python`

```
str = "Line1-a b c d e f\nLine2- a b  
c\n\nLine4- a b c d";  
print str.splitlines( );  
print str.splitlines(0);  
print str.splitlines(3);  
print str.splitlines(4);  
print str.splitlines(5);
```

Answer:

Output

```
['Line1-a b c d e f', 'Line2- a b c', '', 'Line4- abed']  
['Line1-a b c d e f', 'Line2- a b c', '', 'Line4- abed']  
['Line1-a b c d e f\\r', 'Line2- a b c\\r', '\\n', 'Line4- a b c d']  
['Line1-a b c d e f\\n', 'Line2- a b c\\r', '\\r', 'Line4- a b c d']  
['Line1-a b c d e f\\r', 'Line2- a b c\\r', '\\n', 'Line4- a bed']
```

Question 10:

Define `split()` with suitable example.

Answer:

The method `split()` returns a list of all the words in the string, using `str` as the separator (splits on all whitespace if left unspecified), optionally limiting the number of splits to `num`.

Syntax

```
str.split(str="", num=string.count(str)).
```

Parameters

`str` — This is any delimiter, by default it is space.

`num` — This is number of lines to be made.

Example

```
#!/usr/bin/python  
str = "Line1-abcdef \nLine2-abc \nLine4-abcd";  
print str.split( ); print str.split(' 1 ');
```

OUTPUT

```
['Line1-abcdef', 'Line2-abc', 'Line4-abcd']  
['Line1-abcdef', '\\nLine2-abc \\nLine4-abcd']
```

Question 11:

Explain `replace(old, new [, max])`

Answer:

The method `replace()` returns a copy of the string in which the occurrences of `old` have been replaced with `new`, optionally restricting the number of replacements to `max`.

Syntax

```
str.replace(old, new[, max])
```

Parameters –

old — This is old substring to be replaced.

new — This is new substring, which would replace old substring.

max — If this optional argument max is given, only the first count occurrences are replaced.

Example

```
# !/usr/bin/python
str = "this is string example....wow!!! this is really string";
print str.replace("is", "was"); print str.replace("is", "was", 3);
```

OUTPUT

```
thwas was string example....wow!!! thwas was really string
thwas was string example....wow!!! thwas is really string
```

Question 12:

Describe `index(str, beg=0, end=len(string))` with example

Answer:

The method `index()` determines if string `str` occurs in string or in a substring of string if starting index `beg` and ending index `end` are given. This method is same as `find()`, but raises an exception if sub is not found.

Syntax

```
str.index(str, beg=0 end=len(string))
```

Example

```
# !/usr/bin/python
str = "this is string example....wow!!!";
str = "exam";
print str.index(str);
print str.index(str, 10);
print str.index(str, 40);
```

OUTPUT

```
15
```

```
15
```

```
Traceback (most recent call last):
```

```
File "test.py", line 8, in
```

```
print str.index(str, 40);
```

```
ValueError: substring not found
```

```
shell returned 1
```

Question 13:

Explain `expandtabs(tabsize=8)` with example

Answer:

The method `expandtabs()` returns a copy of the string in which tab characters ie. `'\t'` have been expanded using spaces, optionally using the given `tabsize` (default 8).

Syntax

Following is the syntax for `expandtabs()` method : `str.expandtabs(tabsize=8)`

Example

```
#!/usr/bin/python
```

```
str = "this is\tstring example....wow!!!";
```

```
print "Original string: " + str;
```

```
print "Default expanded tab: " +
```

```
str.expandtabs( );
```

```
print "Double expanded tab: " + str.
```

```
expandtabs(16);
```

OUTPUT

Original string: this is string example....wow!!! Default expanded tab: this is string example....
wow!!!

Double expanded tab: this is string example.... wow!!!

Question 14:

Write Python script that takes a string with multiple words and then capitalizes the first letter of each word and forms a new string out of it.

Answer:

```
string = raw_input("Enter a string :")
```

```
length = len (string)
```

```
a = 0
```

```
end – length
```

```
string 2 = " # empty string
```

```
while a < length
```

```
if a == 0
```

```
string 2 += string [0].upper()
```

```
a += 1
```

```
elif (string [a] == ' ' and string [a+1] != " ) :
```

```
string 2 += string [a]
```

```
string 2 += string [a+1].upper( )
```

```
a += 2
```

```
else :
```

```
string 2 += string [a]
```

```
a += 1
```

```
print "Original string :", string
```

```
print "Converted string :", string2
```

Question 15:

Write a program that reads a string and display the longest substring of the given string having just the consonants.

Answer:

```
string = raw_input("Enter a string :")
```

```
length = len (string)
```



```

max length = 0
max sub = ''
sub = ''
lensub = 0
for a in range (length) :
if string [a] in aeiou 'or string [a] in 'AEIOU':
if lensub > maxlength :
maxsub = sub
maxlength – lensub
sub = ''
lensub 0
else :
sub += string[a]
lensub = len(sub)
a + = 1
print "Maximum length consonent
substring is :", maxsub,
print "with", maxlength, "characters"

```

Question 16:

Write a program that reads email id of a person in the form of a string and ensures that it belongs to domain @gmail.com.

Answer:

```

email = raw_input ("Enter email ID :")
domain = "@gmail.com"
lendo = len(domain)
lenm = len(email)
sub = email [lenm – lendo:]
if sub == domain :
if lendo != lenm :
print "It is a valid email ID"
else :
print "It is an invalid email ID"
else :
print "It's domain is different"

```

Question 17:

Write a program that reads a string and then prints a string that capitalizes every other letter in the string.

Answer:

```

string = raw_input ("Enter a string:")
length = len (string)
string2 = " "

```

```

for a in range(0, length, 2):
    string 2 + - string [a]
if a < (length-1):
    string 2+ = string [a+1],upper()
print "Original string is", string
print "Converted string is", string2

```

Question 18:

Consider the string str="Global Warming" Write statements in Python to implement the following

- (a) To display the last four characters.
- (b) To display the substring starting from index 4 and ending at index 8.
- (c) To check whether string has alphanu-meric characters or not
- (d) To trim the last four characters from the string.
- (e) To trim the first four characters from the string.
- (f) To display the starting index for the substring „ WaD.
- (g) To change the case of the given string.
- (h) To check if the string is in title case.
- (i) To replace all the occurrences of letter „aD in the string with „*?

Answer:

- (a) print str[-4:]
- (b) print str[4:8]
- (c) str.isalnum()
- (d) str[:-4]
- (g) str.swapcase()
- (h) str.istitle()
- (i) str.replace('a','*')

Output screenshot :

```

>>>
ming
al W
False
Global War
al Warming
7
gLOBAL WARMING
True
Glob*1 W*rning
>>> |

```

