

Q1. In Python 3.X, what are the names and functions of string object types?

Answer:

Numbers	1234, 3.1415, 3+4j, Decimal, Fraction
Strings	'python', "Jupiter's", b'a\x01c'
Lists	[1, [2, 'three'], 4]
Dictionaries	{'Apple': 'iPhone', 'Google': 'Android'}
Tuples	{1, 'php', 3, 'Y'}
Files	myFile = open('java', 'r')
Sets	set('xyz'), {'x', 'y', 'z'}
Other core types	Booleans, types, None
Program unit types	Functions, modules, classes

Q2. How do the string forms in Python 3.X vary in terms of operations?

Answer: Some of the string operations introduced in python 3 are :-

- 1) Str.upper and str.lower which converts the string variable str to all lower case or uppercase
- 2) Boolean methods has been introduced that will return a boolean result if true for instance, isalnum(), isalpha(), islower(), isupper() etc. Which evaluates if a string contains alphanumeric characters, has only alphabets, is all lower, is all upper, respectively. There are several more such boolean functions introduced.
- 3) A len function has been introduced to determining the length of a string.
- 4) join(), split(), and replace() Methods have been introduced.
 - join() function concatenates two strings but in a way one string passes through another
 - spilt() function splits a string by a given delimiter such as comma or space etc
 - replace() method replaces an original string with a given replacement

Q3. In 3.X, how do you put non-ASCII Unicode characters in a string?

Answer: Using the ascii() method which produces the ASCII representation of an object ,with non-ASCII characters escaped

Example:

Input : `ascii("¥")`

```
Output : "'\\xa5'"
```

Q4. In Python 3.X, what are the key differences between text-mode and binary-mode files?

Answer: A text file contains textual information like texts, alphabets, numbers and special characters and a binary mode file contains the compiled information from a text file, a binary mode file contains bytes

Q5. How can you interpret a Unicode text file containing text encoded in a different encoding than your platform's default?

Answer: To open a text file which is encoded in a different encoding other than system default we can use the `open()` method and mention the encoding method as a parameter. For instance

```
open(blalba, encoding='utf-8')
```

Q6. What is the best way to make a Unicode text file in a particular encoding format?

Answer: We can open a new file with write permission with the desired encoding format, such as :-

```
file = open(fname, 'w', encoding="latin-1")
```

Q7. What qualifies ASCII text as a form of Unicode text?

Answer: The first 128 Unicode code points represent the ASCII characters, which means that any ASCII text is also a UTF-8 text, Hence a string of ASCII text is also valid UTF-8 text.

Q8. How much of an effect does the change in string types in Python 3.X have on your code?

Answer: Python 3 uses the concept of text and binary data instead of Unicode and 8 bit strings. All text is unicode but encoded unicode is represented as binary data in Python 3. The type used to hold text is [str](#), the type used to hold data is [bytes](#). The biggest difference with the 2.x situation is that any attempt to mix text and data in Python 3.0 raises [TypeError](#), whereas if you were to mix Unicode and 8-bit strings in Python 2.x, it would work if the 8-bit string happened to contain only 7-bit (ASCII) bytes, but you would get [UnicodeDecodeError](#) if it contained non-ASCII values