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

Ans:- Python has a set of built-in methods that you can use on strings. Strings are objects in Python which means that **there is a set of built-in functions that you can use to manipulate strings**. You use dot-notation to invoke the functions on a string object such as sentence.  
  
Python String Methods.

Methods Description

|  |  |
| --- | --- |
| capitalize() | Converts the first character to upper case |
| [casefold()](https://www.w3schools.com/python/ref_string_casefold.asp) | Converts string into lower case |
| [center()](https://www.w3schools.com/python/ref_string_center.asp) | Returns a centered string |
| [count()](https://www.w3schools.com/python/ref_string_count.asp) | Returns the number of times a specified value occurs in a string |
| [encode()](https://www.w3schools.com/python/ref_string_encode.asp) | Returns an encoded version of the string |
| [endswith()](https://www.w3schools.com/python/ref_string_endswith.asp) | Returns true if the string ends with the specified value |
| [expandtabs()](https://www.w3schools.com/python/ref_string_expandtabs.asp) | Sets the tab size of the string |
| [find()](https://www.w3schools.com/python/ref_string_find.asp) | Searches the string for a specified value and returns the position of where it was found |
| [format()](https://www.w3schools.com/python/ref_string_format.asp) | Formats specified values in a string |
| format\_map() | Formats specified values in a string |
| [index()](https://www.w3schools.com/python/ref_string_index.asp) | Searches the string for a specified value and returns the position of where it was found |
| [isalnum()](https://www.w3schools.com/python/ref_string_isalnum.asp) | Returns True if all characters in the string are alphanumeric |
| [isalpha()](https://www.w3schools.com/python/ref_string_isalpha.asp) | Returns True if all characters in the string are in the alphabet |
| [isascii()](https://www.w3schools.com/python/ref_string_isascii.asp) | Returns True if all characters in the string are ascii characters |
| [isdecimal()](https://www.w3schools.com/python/ref_string_isdecimal.asp) | Returns True if all characters in the string are decimals |
| [isdigit()](https://www.w3schools.com/python/ref_string_isdigit.asp) | Returns True if all characters in the string are digits |
| [isidentifier()](https://www.w3schools.com/python/ref_string_isidentifier.asp) | Returns True if the string is an identifier |
| [islower()](https://www.w3schools.com/python/ref_string_islower.asp) | Returns True if all characters in the string are lower case |
| [isnumeric()](https://www.w3schools.com/python/ref_string_isnumeric.asp) | Returns True if all characters in the string are numeric |
| [isprintable()](https://www.w3schools.com/python/ref_string_isprintable.asp) | Returns True if all characters in the string are printable |
| [isspace()](https://www.w3schools.com/python/ref_string_isspace.asp) | Returns True if all characters in the string are whitespaces |
| [istitle()](https://www.w3schools.com/python/ref_string_istitle.asp) | Returns True if the string follows the rules of a title |
| [isupper()](https://www.w3schools.com/python/ref_string_isupper.asp) | Returns True if all characters in the string are upper case |
| [join()](https://www.w3schools.com/python/ref_string_join.asp) | Converts the elements of an iterable into a string |
| [ljust()](https://www.w3schools.com/python/ref_string_ljust.asp) | Returns a left justified version of the string |
| [lower()](https://www.w3schools.com/python/ref_string_lower.asp) | Converts a string into lower case |
| [lstrip()](https://www.w3schools.com/python/ref_string_lstrip.asp) | Returns a left trim version of the string |
| [maketrans()](https://www.w3schools.com/python/ref_string_maketrans.asp) | Returns a translation table to be used in translations |
| [partition()](https://www.w3schools.com/python/ref_string_partition.asp) | Returns a tuple where the string is parted into three parts |
| [replace()](https://www.w3schools.com/python/ref_string_replace.asp) | Returns a string where a specified value is replaced with a specified value |
| [rfind()](https://www.w3schools.com/python/ref_string_rfind.asp) | Searches the string for a specified value and returns the last position of where it was found |
| [rindex()](https://www.w3schools.com/python/ref_string_rindex.asp) | Searches the string for a specified value and returns the last position of where it was found |
| [rjust()](https://www.w3schools.com/python/ref_string_rjust.asp) | Returns a right justified version of the string |
| [rpartition()](https://www.w3schools.com/python/ref_string_rpartition.asp) | Returns a tuple where the string is parted into three parts |
| [rsplit()](https://www.w3schools.com/python/ref_string_rsplit.asp) | Splits the string at the specified separator, and returns a list |
| [rstrip()](https://www.w3schools.com/python/ref_string_rstrip.asp) | Returns a right trim version of the string |
| [split()](https://www.w3schools.com/python/ref_string_split.asp) | Splits the string at the specified separator, and returns a list |
| [splitlines()](https://www.w3schools.com/python/ref_string_splitlines.asp) | Splits the string at line breaks and returns a list |
| [startswith()](https://www.w3schools.com/python/ref_string_startswith.asp) | Returns true if the string starts with the specified value |
| [strip()](https://www.w3schools.com/python/ref_string_strip.asp) | Returns a trimmed version of the string |
| [swapcase()](https://www.w3schools.com/python/ref_string_swapcase.asp) | Swaps cases, lower case becomes upper case and vice versa |
| [title()](https://www.w3schools.com/python/ref_string_title.asp) | Converts the first character of each word to upper case |
| [translate()](https://www.w3schools.com/python/ref_string_translate.asp) | Returns a translated string |
| [upper()](https://www.w3schools.com/python/ref_string_upper.asp) | Converts a string into upper case |
| [zfill()](https://www.w3schools.com/python/ref_string_zfill.asp) | Fills the string with a specified number of 0 values at the beginning |

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

ANS:- To create a string, **put the sequence of characters inside either single quotes, double quotes, or triple quotes and then assign it to a variable**. You can look into how variables work in Python in the Python variables tutorial. For example, you can assign a character 'a' to a variable single\_quote\_character .

What are the operations on string in Python?

**Python String Operations**

* Python + operator – String Concatenation operator. The + operator joins all the operand strings and returns a concatenated string.
* Python \* operator – String Replication operator.

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

Ans:-

In order to use non-ASCII characters, Python **requires explicit encoding and decoding of strings** into Unicode. In IBM® SPSS® Modeler, Python scripts are assumed to be encoded in UTF-8, which is a standard Unicode encoding that supports non-ASCII characters.

Using **encode() and decode() method** to remove unicode characters in Python. You can use String's encode() with encoding as ascii and error as ignore to remove unicode characters from String and use decode() method to decode() it back.

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

Ans:- The major difference between these two is that a **text file contains textual information in the form of alphabets, digits and special characters or symbols**. On the other hand, a binary file contains bytes or a compiled version of a text file.

While both binary and text files contain data stored as a series of bits (binary values of **1s and 0s**), the bits in text files represent characters, while the bits in binary files represent custom data. While text files contain only textual data, binary files may contain both textual and custom binary data.

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

Ans:- All text (str) is Unicode by default. Encoded Unicode text is represented as binary data (bytes). The str type can contain any literal Unicode character, such as "Δv / Δt", all of which will be stored as Unicode.

The default encoding in str.encode() and bytes.decode() is UTF-8.

There is one other property that is more nuanced, which is that the default encoding to the built-in open() is platform-dependent and depends on the value of locale.getpreferredencoding():

>>>

>>> # Mac OS X High Sierra

>>> import locale

>>> locale.getpreferredencoding()

'UTF-8'

>>> # Windows Server 2012; other Windows builds may use UTF-16

>>> import locale

>>> locale.getpreferredencoding()

'cp1252'

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

Ans:- write() to write unicode text to a text file. **Call str.** **encode(encoding) with encoding set to "utf8**" to encode str . Call open(file, mode) to open a file with mode set to "wb"

.

**convert file encoding to utf-8 python”**

1. with open(ff\_name, 'rb') as source\_file:
2. with open(target\_file\_name, 'w+b') as dest\_file:
3. contents = source\_file. read()
4. dest\_file. write(contents. decode('utf-16'). encode('utf-8'))

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

Ans:- unicode is **the Information Technology standard** that is used for encoding, representation, and handling of texts in the writing systems whereas ASCII (American Standard Code for Information Interchange) represents text in computers such as symbols, digits, uppercase letters, and lowercase letters.

ASCII == UNICODE? For backward compatibility, the first 128 Unicode characters point to ASCII characters. And since UTF-8 encodes each of those characters using 1-byte. ASCII is essentially just UTF-8, or we can say that **ASCII is a subset of Unicode**.

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

Ans:-

A number can be converted to string using the str() function.I n Python, Strings are **arrays of bytes representing Unicode characters**. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.

Python supports **two types** of strings — Single-line strings and Multi-line strings. Single line strings are enclosed in single or double quotes and terminate in one line. Multi-line strings store multiple lines of text and are enclosed in triple quotes.

**Strings are not mutable** in Python. Strings are a immutable data types which means that its value cannot be updated.