**Python**

**Introduction:**

Python is a high-level programming language.

It has dynamic type system. (type is assigned at runtime instead of compile time)

Emphasis on readability.

Interpreted language. (live coding, live debugging, but slower than complied)

Multi-paradigm (both OOP and SOP)

**History:**

Guido Van Rossum, creator.

First major release in 1991.

Python 2.0 released on October 16, 2000

Python 3.0/ Python 300 / Py3k released in Dec 2008

**Uses:**

Linux scripting and Administration.

Web development

Application scripting

Data Science

**iPython:**

Interactive shell to execute codes.

**PyPI (Python Package Index):**

Codes are published here by programmers so it can be used by others.

Pypi.python.org

**Pip (Package Manager):**

* Install and uninstall packages
* Dependencies of packages are installed automatically
* Python Packages can be managed as Python group.
* Versions are done for compatibility.

**Python 2 vs Python 3**

Python 3 is intentionally not backward compatible.

Python 2 is legacy now and Python is used by default.

* Python command-line is REPL (Read Evaluate Print Loop)
* Standard practice is to use 4 whitespaces for loop.
* PEP 8 gives the conduct of code (standard coding rules)
* Like C/C++ import is used to import std library modules and functions.

**Scalar Types:**

Int – whole number

Integers can be specified in decimal, binary, octo and hexadecimal

Ex: deci – 10, bi – 0b10, octo -0o10, hexa -0x10

Float – Decimal Numbers

15-16significant decimal digits

Scientific notations can be used

Ex: 3e8 , 1.616e-35

None – for null

Bool – true or false

Python does not have a fixed size for datatypes, it depends on the memory of your computer.

**Relational operators same as any other language.**

**Loops:**

While loop

If else

If-elif-else

For loop

Does not support do-while loop like C/C++

**String:**

Sequence of Unicode codepoints.

Immutable.

Single or double quotes.

>>> s = 'parrot'

>>> s[4]

'o'

**Multi Line strings:**

Three quotes can be used to represent multi line or escape character (\n) can be used.

Example: >>> a = """ This is a

... multi line

... string"""

>>> a

' This is a\nmulti line\nstring'

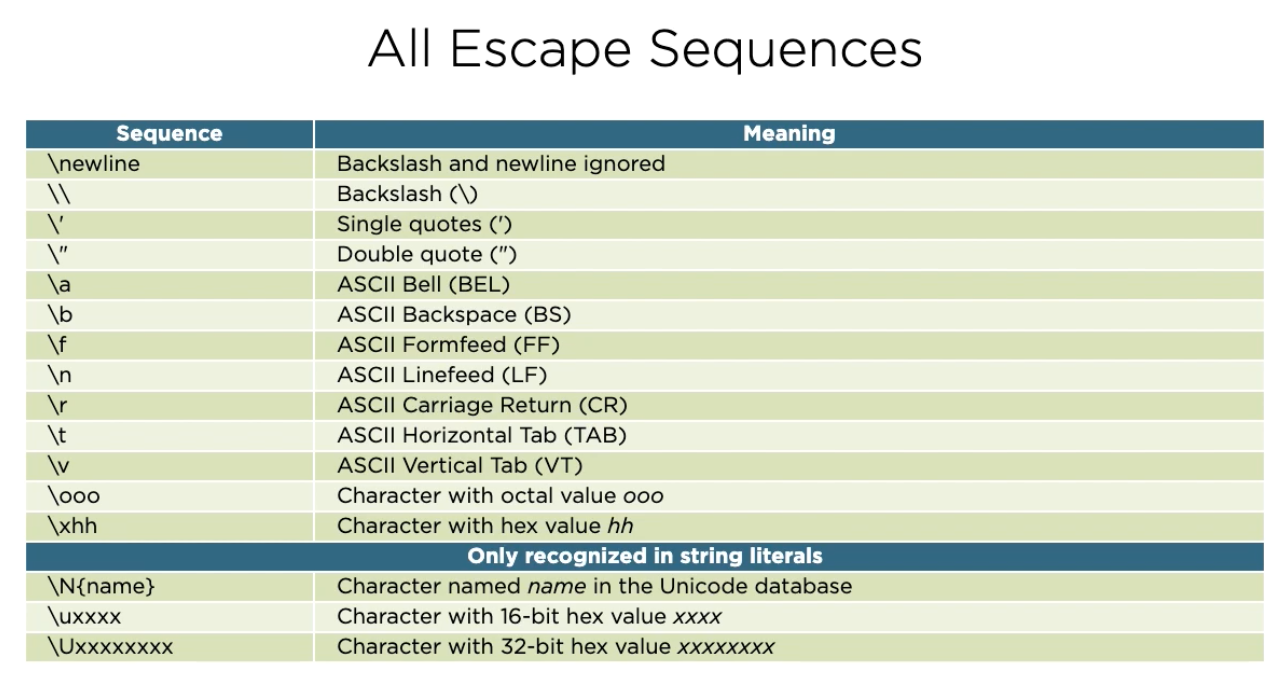
>>> print (a)

This is a

multi line

string

**Escape Sequences:**



**Raw strings:**

Raw strings don’t support any escape sequences.

Syntax: adding lower case r at the beginning of string

Example:

>>> path = r'C:\\\a'

>>> path

'C:\\\\\\a'

>>> print (path)

C:\\\a

**String class also has many methods which can be called using objects(variables).**

Example: >>> a= 'abc'

>>> a.capitalize()

'Abc'

>>> a

'abc'

**Bytes:**

Data type for sequences of bytes.

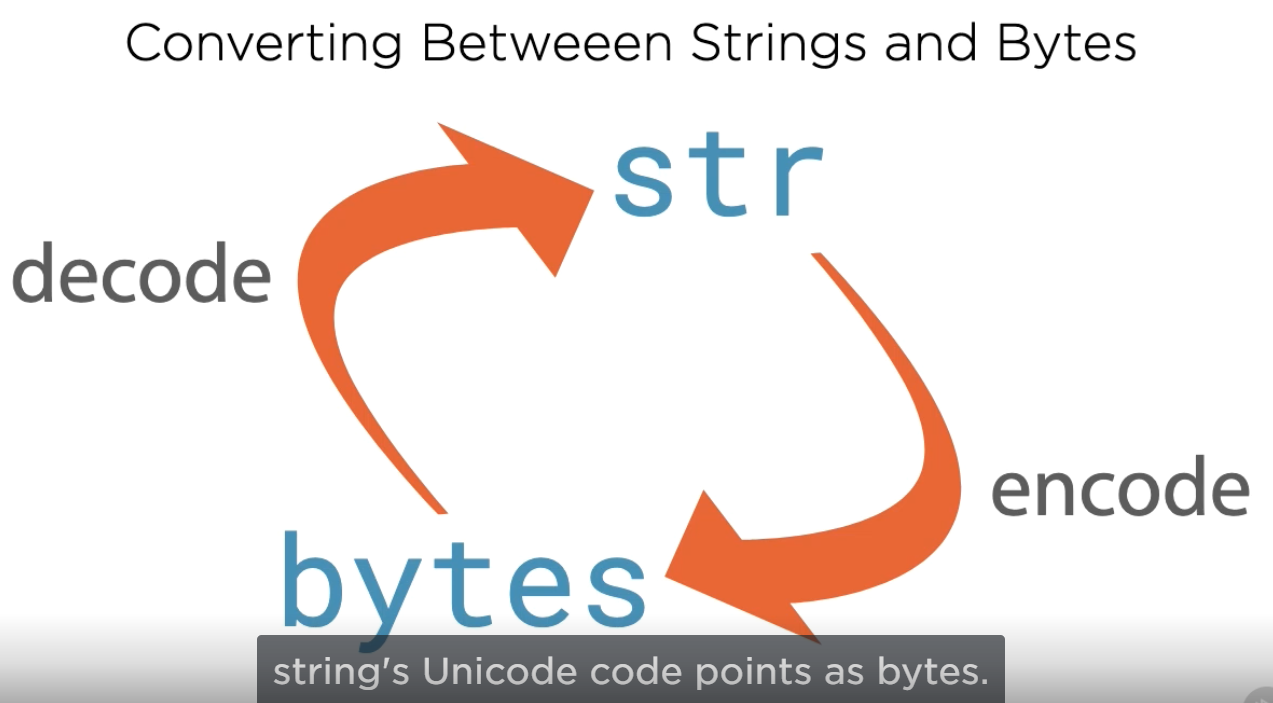
Raw binary data.

Fixed width single byte encoding.

Can be written by prefixing lower case b with literal form with quotes.

Example:

>>> b’data’



>>> nork = 'å fortære'

>>> data = nork.encode('utf8')

>>> data

b'\xc3\xa5 fort\xc3\xa6re'

>>> norwegian = data.decode('utf8')

>>> norwegian

'å fortære'

**Lists:**

Sequence of objects.

Mutable.

List can have data of heterogeneous types.

**Dict:**

Fundamental data structure (widely used)

Maps keys to values

Also known as Map or associative arrays.

**EOF**