

* Difference between Artificial Intelligence, machine learning, Deep learning

Artificial Intelligence	Machine learning	Deep learning
-------------------------	------------------	---------------

- 1) AI stands for ML stands for DL stands for
Artificial Intelligence machine learning Deep learning
- 2) process which enables The study that machines to mimic Uses statistical The study that make use of human behaviour methods enabling Neural network through particular machine to impr- to imitate ove with experience functionality.
- 3) AI is the broader ML is the sub- DL is the family consisting set of AI subset of of ML&DL as Components.
- 4) The aim is basically increase chances The Aim is to IT attains of success & not Accuracy. increase accuracy the highest not caring much rank in terms about the success of Accuracy ratio. when trained large amount data.
- 5) Examples AI Applicat Examples ^{ML} Applic Examples DL
Google's AI powered virtual personal Image analysis predictions Apps Assistant! Siri, & caption like Uber & Htt. Alexa, Email spam generation.

Q. 1. what is PEP - 8.

Ans :-

PEP Python Enhancement proposal (PEP)
It can be defined as a document that helps us to provide the guidelines on How to write python code.

It is basically set of rules that specify How to format python code for maximum readability.

Q. 2. what are the list & tuple what is key difference

Ans:-

List & tuple both Sequence data type that can store A collection of Objects in python. The objects stored in Both Sequences can have different data types.

Lists are represented with Square bracket Ex. ['Sara', 6, 0.99]

while tuple represented Paranthesis ('Aunsh')

The key Diff b/w the two is that while list are immutable tuples other hand In mutable objects.. This means that list can be modified, appended or sliced on the go but Tuples remain Constant & cannot be modified in Any manner

(documentation)

Q.3:- What is doc string in Python ?

Ans:-

doc string is multiline string. Uses the to document a specific code segment.

The doc string should describe what the method function does.

Q.4:- What is slicing in a Python ?

Ans:-

Slicing is a python feature that enables Accessing parts of sequences like string, tuple, list.

You can also use modified or delete the items of mutable sequences. Such as list.

Slices can also be applied numpy's pandas can series & dataframes

Ex. :- String = 'AI' [+0]

O/P:- A

We can use slice function () as well for doing. Some way.

H.W. Write a program reverse to string

H.W. Diffn slicing & indexing.

Q.5

Write a program reverse String

Ans:-

1) Slice Notation :-

Input :-

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

Print (a [::-1])

Output :-

Enter string : Sam

Enter string : Sambhaii

ijahbmäs.

2)

For loop :-

Input :-

```
s = input ("Enter string:")
```

l = len (s) # length store

```
for i in range (l-1, -1, -1):
```

```
    print (s[i], end = "")
```

Output :-

Enter string : Sambhaii Sonwane
enawnos ijahbmäs.

3)

reverse string :-

I/P :-

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

```
print (list (reversed (a)))
```

O/P :-

Enter string : Python

```
[‘n’, ‘o’, ‘h’, ‘t’, ‘y’, ‘p’]
```

- * Set value, Sill Value & floor value. Next lecture maybe.
- * Bitwise & shift operators prepare Kora ka
- * left shift, Right shift, Refactoring.
- * god!

Q.1. What is diff b/w python arrays & list ?

Ans:-

- Arrays & list in python have the same way of storing data.
- But arrays can hold only single data type elements whereas lists can hold Any data type.

Ex:- import array as array

my_array = array.array("i", [1, 2, 3, 4])

my_list = [1, 'abc', 1-20]

Q.2. How to Add values to python array ?

Ans:-

- Elements can be added : to append(), extent & The insert(i,x)

H.W. Write a program in Python to check sequence Palindrome.

1 Diffn betn for loop & while loop.

Ans:-

	Basis of Comparison	for loop	while loop
1) Used	for loop is Used when the no. of iteration is known.	while loop is used when no. of iteration is unknown.	
2) key word	Uses for keyword	Uses while keyword	
3) Absence of Condition	The loop runs infinite times in absence of condition.	Reentrants the compile time error in the absence of condition.	
4) Nature of Initialization	Once done it can not be repeated.	In the while loop it can be repeated at every iteration.	
5) functions	To iterate, the range function is used.	There is no such function in the while loop.	
6) Initialization	Done at the begin of the loop based on condition.	In the while loop it is possible to do these anywhere in loop body.	
7) Generator support	Python for loop can iterate over generator.	While loop cannot be directly iterated on generator.	

Basis of
comparison

for loop

while loop.

8) Speed

faster

Relative slower

2. Diff betn break & continue Statement Pass.

Ans:-

- Break statement: In python terminates the current loop & resumes execution at the next statement.

- Use is when same external condition is triggered requiring a 'hasty' exit from the loop.

Ex. for letter in 'python'
 if letter == 'h':
 break
 print("current letter:", letter)

O/p:-

Current letter: p

Current letter: y

Current letter: t

Current letter: h

Current letter: o

Current letter: n

- Pass Continue Statement:-

The continue statement in python returns the control to the beginning of to the while loop.

We can use Continue Statement to skip current iteration Continue next iteration.

Ex:- for letter in 'python':
 if letter == 'h':
 continue.
 print ("current letter", letter)

O/p:-

Current letter : P
Current letter : Y
Current letter : T
Current letter : O
Current letter : N

* Pass Statement:-

The pass statement in python is used in when statement is required synthetically required but you do not want any command & execute.

The pass statement is null operation. Nothing happen to execute.

Ex:- for letters in 'python':
 if letters == 'h':

 pass

 print("This is pass block")
 print("current letter", letters)

O/p:- Current letter : P | current letter in
Current letter : Y

Current letter : T

This is pass block

Current letter : h :

Current letter : o

H/W what is the difference between module, package, & library.

Ans	Module	Package	Library
1)	The module is a simple Python file that contains collections of functions & global variables with having a .py extension file.	The package is a simple directory having collection of modules.	A Library is an Umbrella term referring to a reusable chunk of code.
2)	A module is a single file (or files) that are imported under <code>__init__.py</code> file one import & by which the are used.	This directory contains python modules, a interpreter it is a package	A package is a collection of modules, a library is a collection of packages.
3)	It is an executable file & to organize all the modules we have the concept called Package in python.	The package is simply a name - Space.	A python library contains a collection of related modules & packages.
E.g:-	<code>import my_module</code> <code>import json</code> .	E.g:- <code>from my_package import abc</code> <code>import a</code>	E.g:- <code>import my_library</code> <code>Import numpy</code>

Tuesday

Date 22-11-2023
Page 1

- Q. Write a program to remove n^{th} index character from non empty string.

Ans:-

The Approach to Use the idea of extraction of sequence of character within range of index values.

Here we extract All the character of string in two parts, 1st Until the n^{th} index & Other begin with $n+1^{\text{th}}$ index. we the approach is two parts.

Program:-

```
# declaring a string variable.  
str = "python is fun"
```

```
# index to remove character at  
n = 8
```

```
# extracts 0 to  $n-1^{\text{th}}$  index  
first_part = str[0:n]
```

```
# extracts characters from  $n+1^{\text{th}}$  index until  
it. Second_part = str[n+1:]
```

```
print("modified string after removing ", n, "th  
character")
```

```
# Combining both the Parts together.
```

```
print(first_part + second_part)
```

Q. Explain the string formating in python?
(search on geeksforGeeks)

Ans:- String formating

- There are four different ways to perform String formating in python:

- ~~String~~ formating with % Operator
- ~~String~~ formating with format() string method.
- ~~String~~ formating with String literals, called f-string
- ~~String~~ formating with string Temple class.

- String formating is the process of infusing Things in the String dynamically & presenting The string.

* Method I: formating string Using % Operator:-

It is the oldest method of string formating.
Here we Use the modulo % operator.
The modulo % also known as the string formating operator.

Wednesday

List comprehensions. Imp.

CLASSMATE
Date 23-May-2023
Page

* Sort descending order

Q.1 what do you mean List Comprehensions In Python or Explain.

Ans:-

Python list comprehensions consist of bracket containing the expression which is executed for each element along with the "for loop" to iterate over each element in the Python list.

This compact way list object for in with like list equal dictionary, range, tuple

It is compact way of creating list object from any iterable object like list, tuple, dictionary, e.g. range, etc. based on some condition.

e.g. $s = [x^2 \text{ for } x \text{ in range}(1, 11)]$
print(s)

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Q.2 How Can you Count Occurrence Each Value.

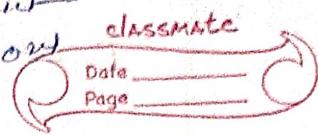
Ans:-

One option is iterable iterate over a list & add count to a dictionary but the easy option is to import the Counter task from collection library Pass the list to it.

```
from collections import Counter  
li = ["a", "b", "c", "c", "a"]  
Counter()
```

Reusable
tuple.
proto type.

In Deep copy use it for project management
Int, float, String, Tuple, Jira - search out
Market basket.



Q.3. What is difference betn shallow copy & deep copy in python.

Ans:- COPY Intbuilf function.

- Deep Copy :-

In Deep Copy any changes made to a copy of object do not reflect in the original object.

Ex:-

```
I = [[1,2], [3,4]]
```

```
import copy
```

```
CP = copy.deepcopy(I)
```

```
CP
```

```
[[1,2], [3,4]]
```

```
CP[0][0] = 45
```

```
CP
```

```
[[45, 2], [3, 4]]
```

```
I
```

```
[[1,2], [3,4]]
```

-

Shallow Object :-

a reference of object an object that stores the reference of the original elements.

IT creates the new collection object then occupying it with reference to the child objects founds in the original.

* intersection (Q.1)

what is diff' bet'n local & global.

Q.1. How would use * args ** in a given function
-5 in a given function.

Ans:-

- They make python function more flexible by accepting a variable numbers of arguments.
- * args pass a variable number of non-keyword argument in a list
- ** keyword pass a variable numbers of keyword argument in a dictionary

Q.2. what do you mean by lambda function & explain it with example?

Ans:-

- lambda function are anonymous function means that the lambda function is without a name.
- The lambda keyword is used to define an anonymous function in Python.

Syntax :- lambda arguments list : expression

E.g:-

$$S = \lambda x, y : x * y$$

$S(4,5)$

$\rightarrow 20$

Q.3. what are local & global variable in Python?

Ans:-

global variables are those which are not defined inside any function (defined outside the functions) & have a global scope where as local variable are those which are defined inside a function & its scope is limited to that function only. local variables can't be accessed anywhere outside the function.

Q.4. what is map & reduce & filter() function in Python?

Ans:-

map() :-

for every element present in the given sequence apply some functionality & generate new element with the required modification.

This requirement we should go for map() function.

Syntax:-

`map(function, sequence)`

* re. sub()

- re. Subn()

- re. escape()

- re. search()

1) re. sub():-

- The sub in the function stands of for substrin -g, a certain regular expression pattern is searched in the given string (3rd parameter).
- finding the substring pattern is replaced by repl (2nd parameter), Count checks & maintains the number of times this occurs.

* Syntax:-

re. sub(Pattern, repl, String, count=0, flags=0)

2) re. subn():-

- Subn() is similar to sub() in all ways except in its way of providing output.
- It returns a tuple with a count of the total of replacement & the new string rather than just the string.

* Syntax:-

re. Subn(Pattern, repl, String, count=0, flags=0)

3) `re.escape()`:

- Returns String with all non-alphanumeric backslashed, this is useful if you want to match an arbitrary literal string that may have regular expression metacharacters.

Syntax:-

`re.escape(String)`4) `re.Search()`:

- This method either returns None.
- `re.match` object contains information about the matching part of the string.
- This method stops after the 1st match, So this is best suited for testing a regular expression more than extracting data.

Thursday

Pandas Very Import

classmate

Date

Page

1-12-22

Regular Expression.

Q.1. What are regular Expression?

Ans:-

regular Expression a way describing pattern text, They are commonly used Search & replace operation As well as in Text Processing & Data mining Application.

Q.2. How many Types of character can be Used To Create patterns for regular expression.

Ans:-

There are types can be Use pattern for regular expression.

1) Literal characters :-

Are Simply Any characters that you want to match directly

2) meta characters! :-

Are the characters that have Special meaning in regular expression & are Use to create more complex patterns.

3) Escape characters!:-

Are Use to insert literal characters into pattern when meta character would otherwise be interpreted by interpreter

Q.3. Can you explain what greed matching in the context regular expression.

Ans:-

Greed matching when a regular expression matches as much text as possible.

scrolling file.

CLASSMATE

Date _____

Page _____

- This can be opposite to non greed matching which matches as little text as possible e.g:-

If you have a text "a" "a" a a a" you are using the regular expression "a+" Then greed match ("a,a,a,a") while non the greed matching match will be "a"

Q.4. Can you list out some common function & methods provided by module.

Ans:-

- 1) re.match(); - attempts to match a given regular expression pattern to a string
- 2) re.search(); - Searches for given regular expression pattern within a string
- 3) re.findall(); - finds all occurrence of given regular expression pattern within a string
- 4) re.split(); - Splits a string along occurrences of a given regular expression pattern
- 5) re.sub(); - Substitutes all occurrences given regular expression pattern within a string with replacement string.

jashm. PySpark language learning medium blogs. classmate
rows = Entert, records towardsdatascience

Q5. How do you match multiple line Using regular expression?

Ans:

- We can use the re module to match multiple lines by using "re.MULTILINE" flag.

Q6. What is diffn betn the replace() & sub()

Ans:

- The replace() method replaces all occurrences of a given string, while the sub method only replaces the 1st occurrence.

Q7. iloc method () In Pandas.

Pandas Reposr Separate create. CSV file include-

(Q.1) loc vs iloc in Pandas?

Ans:-

- Locat & iloc Used in slicing Data-frame
- the pandas Data-frame
- locat Use to get rows & columns with particular labels
- iloc gets rows & / or columns integer location.

e.g. loc :-

```
rows = df.loc[['Avery Bradley', 'R.  
Hunter']]  
print(type(rows))
```

e.g. iloc :-

```
row2 = data.iloc[4,8]
```

(Q.2) mention the different types of Data Structure in Pandas - Two Data Structure.

Ans:-

- Series is Data structure
- Panel - 3 dimensional Data structure It includes items, major axis, minor axis
- Data frame - 2 dimensional Data structure

(Q.3) Explain the indexing in Python

Ans:-

Re-indexing to Conform to Data frame
New index with optional filling logic
Placing NAN / NA in location having
No value in the index IT changes.

The Row labels & Column labels of Dataframe

Q.1 What is Recursive function in python?

Ans:-

- It's funkt that's refer to itself
- Two funn it can call each other this is mutual recursion.

e.g:-

```
def recursive_factorial(n):  
    return n * recursive_factorial(n-1)
```

```
recursive_factorial(3)
```

O/P:- Recursion error :-

maximum recursion depth exceeded.

- The Above Program never stop IT run end
- If you execute the function with n equals to recursive_factorial(n) will be called with n=3 Then, n=2, Then n=1, n=0, Then n=-1 etc.
- This lead infinite hence we are at getting above error.
- The Algorithm never stops is problem
- * The Solution Therefore to seepilyastop condition. which will always depend on our Problem.
- we know that the different factors (3, 2, 1) are -ve not equal to zero. That is 3! equals to $3 \times 2 \times 1$.

- The factor n ever must never be less than or equal to zero.
 - e.g. -
- ```

def recursive_factorial(n):
 if n <= 1:
 return 1
 else:
 return n * recursive_factorial(n-1)

```

### ① How to program recursive function

- 1- break down the problem into 1 or more subproblem on the same type.  
The sub problem solved by recursive call
- 2- The sub problem should Smaller insight  
the original problem.
- 3- finally The decomposition must lead to An elementary IT is not decompose into sub problem (This is the Stop condition)