**Cheat Sheet**

*(python)*

**1, chapter**

**Which of the following is a python program extension?**

.py  
.c  
.p  
.json

**Have you ever learned any programming language? (True for yes. False for no)**

True  
False

**Cloud computing and fog computing have the same concept.**

True  
False

**IoT stands for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Input Output Technology.

Internet for Teaching.

Internet of Things.  
All of the above.

**2, chapter**

**In the second part of the course, which python environment will be used?**

Jupyter

VScode

IDLE

As I like

**3, chapter**

**What is the output?**

print(2 \*\* 2 \*\* 3)

256

**Usually, Interpreter is faster than the complier.**

True

False

**This code has no error.**

x= input('Enter a number')

y= x+1

print (y)

True

False

**Each function may have 1) an effect 2) a result.**

True  
False

**Functions in Python can come from (Select more than one answer if needed)**

modules  
Built-in  
import from another languages like c++  
own functions

**What do you call a file containing a program written in a high-level programming language?**

A code file

A source file

A machine file

A target file

**print () function is a built-in function type**

True

False

**What is the output?**

var = 2

var = 3

print(var)

Error

var

2

3

**Computers have a native language; just like us. Computers' native language is called Machine Learning.**

True

False

**what is the output?**

print ('2' + '2')

22

**Select the true statements  (Select two)**

Python is free, open-source, and multiplatform

python is a good choice for creating and executing tests for applications  
Python2 is compatible with Python3  
Python is faster compared to c++

**Python is a scripting language.**

True

False

**4, chapter**

list\_1 = ["A", "B", "C"]

list\_2 = list\_1

list\_3 = list\_2

del list\_1[0]

del list\_2[:]

print(list\_3)

Error

[]

’A’

„B”, „C”

lst = []

del lst

print(lst)

Error  
True  
False  
[]

for i in range(0, 6, 3): print(i)

0 and 3  
3 and 0  
0 and 0  
Error

**What is the output?**

for i in range (-1,1):

     print ('%')

%%%%  
%  
%%  
out of range

x = 4

y = 1

a = x & y

b = x | y

c = ~x

d = x ^ 5

e = x >> 2

f = x << 2

print(a, b, c, d, e, f)

0 5 -5 1 1 16  
Error  
0 -5 5 1 1 16

0 0 -9 1 1 16

**what is the output?**

m\_l = [3,1,-1]

m\_l[-1] = m\_l[-2]

print (m\_l)

[3, -1, 1]  
Error  
[3,1,1]  
[3,-1,-1]

What is the output?

def add\_numbers(a, b=2, c):

     print(a + b + c)

add\_numbers(a=1, c=3)

4  
Syntax Error  
6  
abc

x = 1

y = 0

z = ((x == y) and (x == y)) or not(x == y)

print(not(z))

0  
False  
True  
1

What is the output?

hi()

def hi():

     print("hi!")

hi!  
nothing  
Error  
None

**What is the output?**

**a = 1**

**def fun():**

**a = 2**

**print(a)**

**a = 3**

**fun()**

**print(a)**

1 3  
2 2

1 2  
2 3

**5, chapter**

**A PWG-lead repository, collecting open-source Python code, is called:**

PyPI  
PyRep  
PWGR  
PyCR

**the name pip comes from:**

pip install packages

python internal packages

package in package

all the above

**How to get information about a package in python**

pip --version will tell you that.

pip3 --version will tell you that.

pip show package will tell you that.

All the above.

**What is the expected output of the following code?**

for ch in "abc":

     print(chr(ord(ch) + 1), end='')

bcd

97 98 100

Error

abc

**Which one of the following is true?**

Packages can contain modules.

Modules can contain packages.

Modules can contain modules.

All the above.

**Python is completely internationalized - we can use UNICODE characters inside our code, read them from input and send to output.**

True, because Python 3 is I18N.  
All the above  
True, because Python 3 is UCS-4  
True, because Python 3 is Ascii.

**write a line to import pi from math as PI**

from math import pi as PI

**What is the expected output of the following code?**

the\_list = ['Where', 'are', 'the', 'snows?']

s = '\*'.join(the\_list)

print(s)

Where\*are\*the\*snows

Where are the snows?

Error, it is immutable

Where\*are\*the\*snows?

**You want to invoke the function make\_money() contained in the module named mint. Your code begins with the following line:**

from mint import make\_money

**What is the proper form of the function's invocation?**

mint.make\_money()

make\_money

make\_money()  
All the above

abc

|\_\_

         def

               |\_\_ mymodule.py

**Assuming that D:\Python\Project\Modules has been successfully appended to the sys.path list, write an import directive letting you use all the mymodule entities.**

from abc import \*

import abc.def.mymodule

import mymodule.py

All the above

**6, chapter**

**choose the correct answer.**  
class Python:

   population = 1

   victims = 0

   def \_\_init\_\_(self):

         self.length\_ft = 3

         self.\_\_venomous = False

population and victims are class variables

population and \_\_venomus are class variables

length and \_\_venomous are class variable

population and victims are instance variables

**If we assume that pythons, vipers, and cobras are subclasses of the same superclass, how would you call it?**

Snake or reptile

All the above

People

Cars

**What is the output?**

class Snake:

     pass

class Python(Snake):

    pass

print(Python.\_\_name\_\_, 'is a', Snake.\_\_name\_\_)

print(Python.\_\_bases\_\_[0].\_\_name\_\_, 'can be', Python.\_\_name\_\_)

Error

Python is a Snake Snake can be Snake

Python is a Snake Snake can be Python

Python is a Python Snake can be Python

**Can you name one of your classes just "class"?**

No, class is a function.

No, class is a keyword.

I can, but there is no need for that.

Yes, I can and why not?

**Is there something missing  in the following code?**

class Snakes

     def \_\_init\_\_():

                self.sound = 'Sssssss'

Calling the super class.

Nothing is missing.  
self.\_\_sound = 'Sssssss'  
The \_\_init\_\_() constructor lacks the obligatory parameter (we should name it self to stay compliant with the standards).

**Write only one line.**

Assuming that there is a class named Snakes, write the very first line of the Python class declaration, expressing the fact that the new class is actually a subclass of Snake.

class Python(Snakes):

**The ArithmeticError is not a built-in exception.**

True

False

**What is the name of the most general of all Python exceptions?**

AssertionError  
BaseException  
Except  
MemoryError

**In the procedural approach, the data can use the functions.**

True  
False

**7, chapter**

**How do you encode an open() function’s mode argument value if you're going to create a new text file to only fill it with an article?**

"wb"  
"rt" or "r"  
"rt"

"wt" or "w"

**Fix the code and fill the blank by the fixed line.  
Example wrong solution: def \_\_iter\_\_(self.example):**

class Vowels:

     def \_\_init\_\_(self):

            self.vow = "aeiouy "

            self.pos = 0

     def \_\_iter\_\_(self):

          return self

    def \_\_next\_\_(self):

       if self.pos == len(vow):

          raise StopIteration

     self.pos += 1

    return self.vow[self.pos - 1]

vowels = Vowels()

for v in vowels:

     print(v, end=' ')

if self.pos == len(self.vow):

**What is the output of the following snippet?**

import calendar

print(calendar.weekheader(1))

M T W T F S S

**What is the expected output of the following code?**

import math

try:

     print(math.sqrt(-9))

except ValueError:

    print("inf")

else:

     print("fine")

finally:

     print("the end")

fine the end

inf the end

fine

inf

**What is the problem with this program ? If there is !**

from datetime import timedelta  
from datetime import date  
from dateTime import datetime

delta = timedelta(weeks=2, days=2, hours=2)  
print(delta)

delta2 = delta \* 2  
print(delta2)

d = date(2019, 10, 4) + delta2  
print(d)

dt = datetime(2019, 10, 4, 14, 53) + delta2  
print(dt)

No problem found, it is working

we should write from datetime import \*  
dateTime no such module  
we should use try except form

What is the output of the following snippet? (Assume the file is not exist)

import errno

try:

    stream = open("file", "rb")

    print("exists")

         stream.close()

except IOError as error:

     if error.errno == errno.ENOENT:

          print("absent")

    else:

          print("unknown")

exists  
errno.ENOENT → No such file or directory  
unknown  
absent

**You're going to process a bitmap stored in a file named image.png, and you want to read its contents as a whole into a *bytearray* variable named image. Add a line to the following code to achieve this goal.**

try:

   stream = open("image.png", "rb")

   # Insert a line here.

      stream.close()

except IOError:

     print("failed")

else:

     print("success")

image = bytearray(stream.read())

Write a **lambda** function, setting the least significant bit of its integer argument, and apply it to the map() function to produce the string 1 3 3 5 on the console.

any\_list = [1, 2, 3, 4]

even\_list = # Complete the line here.

print(even\_list)

Hint: the " opertation does the following :

even | 1 = even + 1

odd | 1 = odd

list(map(lambda n: n | 1, any\_list))

**x = lambda a,b : a \*\* b**

**print (x(2, 10))**

1024  
Error  
2222222222222

What is the output of the following snippet?

from datetime import datetime

dt1 = datetime(2020, 9, 29, 14, 41, 0)

dt2 = datetime(2020, 9, 28, 14, 41, 0)

print(dt1 - dt2)

1 day, 0:00:00

**What is the expected output of the following code?**

import math

try:

     print(math.sqrt(9))

except ValueError:

    print("inf")

else:

    print("fine")

3.0 fine  
fine  
3.0  
Error

**What is the output of the following snippet?**

from datetime import time

t = time(14, 39)

print(t.strftime("%H:%M:%S"))

Error

strftime is not defined

14:53

14:53:00

**What is the output**

foo = [i + i for i in range(5)]

print (foo)

0 2 4 6 8  
[1 , 3, ,5 7, 9]

[0. 2. 4 .6 .8]

**8, chapter**

**Given the following nested list, use indexing to grab the word "Hi Big Data learner".**

lst = ['a','b',[4,10,'Hi Big Data learner'],['c',[1,66,['this']],2,111],'e',7]

lst[2][2]

**Big data can be defined:**

Data that is so vast, fast, or complex that  
it becomes impossible to store, process,  
and analyze using traditional data storage  
and analytics applications

True  
False

**All data can be used as is.**

True  
False

**Which of the following is unstructured data?**

.db  
.csv  
.xls  
text

**Which of the following is considered traditional data storage.**

excel  
mysql  
sql  
python

**Which of the following ARE type of data?**

Public Data

Private Data  
Close Data  
Open Data

**Data in Motion**

backup data

requires real-time process  
Data stored for analyzing  
Data moves from place to another

**# Modify this code**   
floor\_types = ['Parking', 'Shops', 'Food Court', 'Offices']  
floor\_numbers = # Fill in the blank # put your code here   
zipped = list(zip(floor\_types ,floor\_numbers ))  
print(zipped)

# To get this output

[('Parking', 1), ('Shops', 2), ('Food Court', 3), ('Offices', 4)]

range(1,5)

[1, 2, 3, 4]

**Data at Rest is**

data not used

data can be analyze

data can be stored somewhere

Data cannot be updated

**Today, data is growing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

exponentially

linearly

rapidly

randomly

**9, chapter -missing-?**

**10, chapter**

**An example of categorial variable is**

gender  
number of visits  
0 or 1  
weight

**An example of numerical variable is weight or blood pressure.**

True  
False

To change the datatype of a column called **ping** in a data frame called  **df\_compact\_clean**, you can use a lambda function in this way

df\_compact\_clean['Ping (ms)\_float'] = df\_compact\_clean['Ping (ms)'].apply(lambda val: float(val))

True

False

**What is this table for?**

| **FSIQ** | **VIQ** | **PIQ** | **Weight** | **Height** | **MRI\_Count** |
| --- | --- | --- | --- | --- | --- |
| **FSIQ** | 1.000000 | 0.946639 | 0.934125 | -0.051483 | -0.086002 | 0.357641 |
| **VIQ** | 0.946639 | 1.000000 | 0.778135 | -0.076088 | -0.071068 | 0.337478 |
| **PIQ** | 0.934125 | 0.778135 | 1.000000 | 0.002512 | -0.076723 | 0.386817 |
| **Weight** | -0.051483 | -0.076088 | 0.002512 | 1.000000 | 0.699614 | 0.513378 |
| **Height** | -0.086002 | -0.071068 | -0.076723 | 0.699614 | 1.000000 | 0.601712 |
| **MRI\_Count** | 0.357641 | 0.337478 | 0.386817 | 0.513378 | 0.601712 | 1.000000 |

The is the correlation for a dataframe.

this is the output of describe() method in pandas

This is output of head() method in pandas

this is a dataframe file.

| **Upload (Mbit/s)** | **Date** | **Time** | **Ping (ms)** | **Download (Mbit/s)** |
| --- | --- | --- | --- | --- |
| **0** | 14.31 | 2016-11-24 | 13:36:25 | 26.992 | 91.80 |
| **1** | 14.12 | 2016-11-24 | 13:36:55 | 24.532 | 88.19 |
| **2** | 14.11 | 2016-11-24 | 13:37:25 | 20.225 | 59.86 |
| **3** | 14.22 | 2016-11-24 | 13:37:57 | 19.332 | 91.81 |
| **4** | 14.08 | 2016-11-24 | 13:38:27 | 22.494 | 92.05 |

Before saving the DataFrame, it makes sense to reposition Upload as the last column. This can be achieved using the reindex function.

ex: df\_compact\_clean = df\_compact\_clean.reindex(columns = ['Date', 'Time', 'Ping (ms)','Download (Mbit/s)','Upload (Mbit/s)']);

True

False

**Add a new column called rounded to the dataframe and populate it with rounded weights, the name of the dataframe is da it has the following data, note the round is 2**

wieght

0 1.00000  
1 2.00000  
2 3.00000  
3 4.00000

data['rounded'] = data.weight.lambda (2)(2)

data['rounded'] = data.weight.round(2)

data['rounded'] = data.weight.Round(2)

da['rounded'] = da.weight.round(2)

**in the following table, it is by coincidence, that the diagonal is 1 in this**

correlation table

| **FSIQ** | **VIQ** | **PIQ** | **Weight** | **Height** | **MRI\_Count** |
| --- | --- | --- | --- | --- | --- |
| **FSIQ** | 1.000000 | 0.946639 | 0.934125 | -0.051483 | -0.086002 | 0.357641 |
| **VIQ** | 0.946639 | 1.000000 | 0.778135 | -0.076088 | -0.071068 | 0.337478 |
| **PIQ** | 0.934125 | 0.778135 | 1.000000 | 0.002512 | -0.076723 | 0.386817 |
| **Weight** | -0.051483 | -0.076088 | 0.002512 | 1.000000 | 0.699614 | 0.513378 |
| **Height** | -0.086002 | -0.071068 | -0.076723 | 0.699614 | 1.000000 | 0.601712 |
| **MRI\_Count** | 0.357641 | 0.337478 | 0.386817 | 0.513378 | 0.601712 | 1.000000 |

True  
False

**To find the correlation for a data file imported in pandas as BB, you can use a method like:**

brainFrame.corr()

brainFrame.corr(method='pearson')

BB.describe()

BB.corr()

**Which method would you use to view table statistics? for a data frame called Data?**

| **weight** |
| --- |
| **count** | 10000.000000 |
| **mean** | 20.499212 |
| **std** | 0.199874 |
| **min** | 19.752000 |
| **25%** | 20.365000 |
| **50%** | 20.500000 |
| **75%** | 20.635000 |
| **max** | 21.171000 |

Data.head()

Data.dtype

Data.describe()

All can be used

**Heat map is used to visualize the correlation for a dataframe!**

True  
False

**Write a method to find the datatype of each column in a dataframe called y**

y.dtypes

**Write a command to view the five 5 lines of the dataframe imported by pandas as brain.**

brain.head(5)

**11, chapter**

**A researcher does not use a temperature sensor correctly. What kind of error will be?**

Gross Error

Random Error

Systematic Error

Noise Error

**When using the ML to predict something based on the given data, Calculating the noise ratio is important. One needs to know how correct is the prediction is.**

True

False

**To calculate the error,  which module used (according to the labs)**

order = 1  
p = np.poly1d(np.polyfit(x, y ,order))

from \_\_\_\_\_\_\_\_\_  import r2\_score

r2 = r2\_score(y, p(x))  
r2

sklearn metrics

sklearn

scikit

seaborn

**To draw an arrow on a figure, then you can use the method called \_\_\_\_\_\_\_\_\_\_ from the matplotlib module (plt)**

plt.xticks()

plt.plot()

plt.annotate()

plt.legend()

|  | **district** | **sales** | **stores** |
| --- | --- | --- | --- |
| **0** | 1 | 231.0 | 12 |
| **1** | 2 | 156.0 | 13 |
| **2** | 3 | 10.0 | 16 |
| **3** | 4 | 519.0 | 2 |
| **4** | 5 | 437.0 | 6 |

**If the dataframe above called SA, you can reach the Sales column in this way  SA.['sales']**

True

False

|  | **district** | **sales** | **stores** |
| --- | --- | --- | --- |
| **0** | 1 | 231.0 | 12 |
| **1** | 2 | 156.0 | 13 |
| **2** | 3 | 10.0 | 16 |
| **3** | 4 | 519.0 | 2 |
| **4** | 5 | 437.0 | 6 |

**Drop the District column using the drop method. (Donot create a new dataframe, drop it from this dataframe itself) Important: the dataframe names is *df***

df.drop('district',axis=1,inplace=True)

sales = df.drop('district',axis=1)

df.drop('district',axis=1)

sales = df.drop('district',axis=1, inplace=True)

**To fill a NaN values in a column in the age column , you can use the following code**

**Note: the dataframe called *training***

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 915 entries, 0 to 914

Data columns (total 12 columns):

PassengerId 915 non-null int64

Survived 915 non-null int64

Pclass 915 non-null int64

Name 915 non-null object

Gender 915 non-null object

Age 738 non-null float64

SibSp 915 non-null int64

training["Age"].fillna(training["Age"].mean())

training["Age"].fillna(training["Age"].mean(), inplace=True)  
training["Age"].fillna(training["Age"].mean(), axces=1)  
training["Age"].dropna()

**Need a target to work**

Data Mining  
Unsupervised ML  
Supervised ML  
All are correct

**What method/function produces this output  from a dataframe called PP**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 915 entries, 0 to 914

Data columns (total 12 columns):

PassengerId 915 non-null int64

Survived 915 non-null int64

Pclass 915 non-null int64

Name 915 non-null object

Gender 915 non-null object

Age 738 non-null float64

SibSp 915 non-null int64

PP.describe

PP.head()

PP.info()  
info(PP)

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 915 entries, 0 to 914

Data columns (total 12 columns):

PassengerId 915 non-null int64

Survived 915 non-null int64

Pclass 915 non-null int64

Name 915 non-null object

Gender 915 non-null object

Age 738 non-null float64

SibSp 915 non-null int64  
  
According to the above lines, In which column(s) there are NaN values.

Gender  
Pclass  
No NaN values are there  
Age

**Learn from the data itself**

Supervised ML

Unsupervised ML  
AI  
Data Mining

**12, chapter**

**df\_compact.to\_csv('rpi\_data\_processed.csv', index=False)**

**in the given code, it is necessary to make the index false.  Isn't it?**

Yes, so that you do not introduce an additional column.  
No, it should be True.  
We should make it True to not introduce new column.  
It is to make the extension .csv

**This code can be used to find  how many values in dataframe are NaN?**

**df\_compact.isnull().sum().sum()**

**Which of the following can also be used to find the number of NaN in the dataframe?**

df\_compact.info()

df\_compact.head()

df\_compact.describe()

all the choices

**Select the best visualization:**

**drawing the throughput on a cable connecting two routers.**

Line (curves)

Scatter

Pie

Bar

**How many default subplot will be generated from this code**

fig, ax = plt.subplots(1, 2, figsize=(10, 10))

2

3

4

1

**In this way, we can drop the row from a dataframe named dfp**

dfp.drop(45)

True  
False

**Data can be summarized using visualizations to help others understand  
the data. You need to focus on the data when you present.**

True  
False

**Select the best visualization:**

**drawing the percentage of men and women in a class.**

Pie  
Line  
Scatter  
Bar

**Select the best visualization:**

**Average salary  based on a number of months**

Scatter  
Line  
Pie  
Bar

**Deductive reasoning uses facts, propositions, or other statements of  
truth to arrive at a conclusion.**

True  
False