Final Exam

This is a preview of the published version of the quiz

Started: Apr 27 at 2:08pm

Quiz Instructions

Question 1 — 1 pts

Please select from the following, statement(s) that are **false** for the `assert` statements. **Select all the false statements!**

- [ ] assert does not require a condition
- [ ] An AssertionError is raised, if assert condition evaluates to False
- [ ] If assert condition evaluates to False, then the code keeps running
- [ ] Assert statements enable us to convert semantic errors to runtime errors

Question 2 — 1 pts

Please select from the following, statement(s) that can be **executed more than once** during the invocation of a function.

- [ ] return
- [ ] continue
- [ ] while
- [ ] break
- [ ] for

Question 3

1 pts

If a function defined in Python doesn't have an explicit "return" statement, it automatically returns the last variable present in the body of the function.

- ◯ True
- ◯ False

Question 4

1 pts

Please match the type of error to its corresponding code snippet.

Syntax Error	[Choose]
Runtime Error	[Choose]
Semantic Error	[Choose]

Question 5

1 pts

Which of the following **can be used** as a key for a Python dictionary?

- ☐ strings
- ☐ negative integers
- ☐ custom type created using recordclass

- [ ] lists
- [ ] tuples
- [ ] custom type created using namedtuples

Question 6

1 pts

Please select from the following, variable name(s) that are **valid. Select all valid options!**

- [ ] my-variable
- [ ] _my_var
- [ ] 007Secret
- [ ] which exam
- [ ] this_is_theFinalExam
- [ ] goodbye2020

Question 7

1 pts

Assume this code is being used to scrape this html page: http://states.com:300/list_of_us_states.html (part of the code is hidden by ????):

```
import requests
from bs4 import BeautifulSoup
try:
    r = requests.get("http://states.com:300/list_of_us_states.html")
    r.raise_for_status() # raises an HTTPError if page is missing
    doc = BeautifulSoup(????, "html.parser")
except requests.exceptions.HTTPError as e:
    print("WARNING! Could not fetch page")
```

If list_of_us_states.html is deleted from that website so that the server returns a 404 status, what will be the output of the above code?

- ◯ prints “WARNING! Could not fetch page”
- ◯ HTTPError
- ◯ Syntax Error
- ◯ Does nothing

Question 8

1 pts

Assume the content for the website http://states.com:300/list_of_us_states.html is equivalent to the following HTML:

```
<table>
    <tr>
        <td><a href="Alabama.html" title="Alabama">Alabama</a></td>
        <td><i>Montgomery</i></td>
    </tr>
    <tr>
        <td><a href="Alaska.html" title="Alaska">Alaska</a></td>
        <td><b>Juneau</b></td>
    </tr>
  <tr>
        <td><a href="Wisconsin.html" title="Wisconsin">Wisconsin</a></td>
        <td><i>Madison</i></td>
  </tr>
  <tr>
        <td><a title="Wyoming">Wyoming</a></td>
        <td><b>Cheyenne</b></td>
  </tr>
</table>
```

Assume this code is being used to scrape that page (part of the code is hidden by ????):

```
import requests
from bs4 import BeautifulSoup
try:
    r = requests.get("http://states.com:300/list_of_us_states.html")
    r.raise_for_status() # raises an HTTPError if page is missing
    doc = BeautifulSoup(????, "html.parser")
except requests.exceptions.HTTPError as e:
    print("WARNING! Could not fetch page")
```

How many clickable links are there in the above HTML? Be careful!

- ◯ 4

- ◯ 3
- ◯ 1
- ◯ 2

Question 9

1 pts

Assume the content for the website http://states.com:300/list_of_us_states.html is equivalent to the following HTML:

```
<table>
    <tr>
        <td><a href="Alabama.html" title="Alabama">Alabama</a></td>
        <td><i>Montgomery</i></td>
    </tr>
    <tr>
        <td><a href="Alaska.html" title="Alaska">Alaska</a></td>
        <td><b>Juneau</b></td>
    </tr>
  <tr>
        <td><a href="Wisconsin.html" title="Wisconsin">Wisconsin</a></td>
        <td><i>Madison</i></td>
  </tr>
  <tr>
        <td><a title="Wyoming">Wyoming</a></td>
        <td><b>Cheyenne</b></td>
  </tr>
</table>
```

Assume this code is being used to scrape that page (part of the code is hidden by ????):

```
import requests
from bs4 import BeautifulSoup
try:
    r = requests.get("http://states.com:300/list_of_us_states.html")
    r.raise_for_status() # raises an HTTPError if page is missing
    doc = BeautifulSoup(????, "html.parser")
except requests.exceptions.HTTPError as e:
    print("WARNING! Could not fetch page")
```

Match the following expressions to the correct output

???? [Choose]

doc.find_all("td")[-1].get_text()	[Choose]
next(doc.find_all("tr")[0].children)	[Choose]
doc.find_all("a")[-1].attrs	[Choose]

Question 10

1 pts

In URL http://states.com:300/list_of_us_states.html, what is states.com?

- ◯ a status code
- ◯ a port number
- ◯ a domain name
- ◯ a resource

Question 11

1 pts

The following **vote_df** dataframe represents the votes for a county. Note: different states can contain same county name!

	state	county	Republicans Votes	Democrats Votes
2	Kansas	Allen	3552	2189
3	Kentucky	Anderson	6885	3462
0	Kansas	Anderson	2362	1175
1	Kentucky	Allen	5258	2024

The following **pop_df** dataframe represents the total population for each county.

	state	county	total populations

0	Kansas	Anderson	8001
1	Kentucky	Allen	19377
2	Kansas	Allen	13387
3	Kentucky	Anderson	21271

Which of the following statements will return **True**? **Select all that apply.** Hint: Pay close attention to indexes of the dataframes!

- [ ] (vote_df['Democrats Votes']/vote_df['Republicans Votes'] > 0.5)[3]
- [ ] vote_df.iloc[0,0] == vote_df.iloc[2,0]
- [ ] vote_df['Republicans Votes'][0] == 3552
- [ ] vote_df.loc[:3,'county'][3] == 'Allen'
- [ ] vote_df['county'][2] == pop_df['county'].iloc[2]

Question 12

1 pts

DataFrame **grade_df** is shown below:

	HW1	HW2	Final Project
Sheldon	94	88	90.0
Penny	78	75	NaN
Leonard	89	70	80.0
Howard	65	46	67.0

Which of the following are **true**? **Select all that apply!**

- [ ] After you run the below code, everyone's grade for HW2 in the original dataframe, grade_df, will be increased by 5.

```
grade_df['HW2'] + 5
```

- [ ] If 1<= x <= 4, grade_df.iloc[[x]] and grade_df.iloc[x] will ouput the same thing with the same type.
- [ ] When you are calculating the average of the final project with .mean() method, all the NaN values will be skipped in the average calculation.
- [ ] grade_df.loc[['Sheldon','Leonard']] will output the grades of Sheldon, Penny, and Leonard.

Question 13

1 pts

Which of the following Python data structures can be used to create a pandas DataFrame? **Select all options that could work.**

- [ ] dictionary of Series
- [ ] dictionary of lists
- [ ] list of dictionaries
- [ ] None of the other options
- [ ] dictionary of dictionaries
- [ ] list of lists

Question 14

1 pts

The following **vote_df** dataframe represents the votes for a county. Note: different states can contain same county name!

	state	county	Republicans Votes	Democrats Votes
2	Kansas	Allen	3552	2189
3	Kentucky	Anderson	6885	3462
0	Kansas	Anderson	2362	1175

1	Kentucky	Allen	5258	2024

The following **pop_df** dataframe represents the total population for each county.

	state	county	total populations
0	Kansas	Anderson	8001
1	Kentucky	Allen	19377
2	Kansas	Allen	13387
3	Kentucky	Anderson	21271

Which of the following are true about the above pandas DataFrames? **Select all options that are true!**

- [ ] pop_df.iloc[0]["total populations"] enables us to extract total population of Anderson county in the state of Kansas.
- [ ] vote_df['Democrats Votes'] < 2500 will output a dataframe with rows where Democrats votes are less than 2500.
- [ ] If 1 <= x <= 4, pop_df.iloc[:x] will output a dataframe with x rows.
- [ ] The ratio of Republicans votes to total population for each different county, can be computed with vote_df['Republicans Votes']/pop_df['total populations'].

Question 15

1 pts

Assume that the below "books" table is inside a SQL database.

Title	Author	Date	Pages	Country
Anna Karenina	Leo Tolstoy	1877	864	Russia
The Great Gatsby	F. Scott Fitzgerald	1925	192	United States
War and Peace	Leo Tolstoy	1869	1225	Russia
Lolita	Vladimir Nabokov	1955	336	France

The Adventures of Huckleberry Finn	Mark Twain	1884	366	United States

Which SQL query answers the following question: **which authors belong to the "United States"?**

- ◯ SELECT Author, Country FROM books WHERE Country != "France";
- ◯ SELECT Author FROM books ORDER BY Date;
- ◯ SELECT * FROM books WHERE Country != "United States";
- ◯ SELECT Author FROM books GROUP BY Country HAVING Country = "United States";
- ◯ SELECT Author FROM books WHERE Country = "United States";

Question 16

1 pts

Assume that the below "books" table is inside a SQL database.

Title	Author	Date	Pages	Country
Anna Karenina	Leo Tolstoy	1877	864	Russia
The Great Gatsby	F. Scott Fitzgerald	1925	192	United States
War and Peace	Leo Tolstoy	1869	1225	Russia
Lolita	Vladimir Nabokov	1955	336	France
The Adventures of Huckleberry Finn	Mark Twain	1884	366	United States

Which SQL query answers the following question: **which authors have published more than 1 book that has at least 800 pages?**

Note: The query should display one row for each author, along with the number of books that have at least 800 pages.

- ◯ SELECT Title FROM books;

- ◯ SELECT Author, COUNT() as c FROM books WHERE Pages>800 GROUP BY Pages HAVING c>1;
- ◯ SELECT Author, COUNT(Country) as c FROM books WHERE Pages>800 GROUP BY Author HAVING c>1;
- ◯ SELECT Author, COUNT(Country) as c FROM books WHERE c>1 GROUP BY Author HAVING Pages>800;
- ◯ SELECT Author, COUNT(Pages) FROM books WHERE Pages>800 GROUP BY Pages;

Question 17

1 pts

Assume that the below "books" table is inside a SQL database.

Title	Author	Date	Pages	Country
Anna Karenina	Leo Tolstoy	1877	864	Russia
The Great Gatsby	F. Scott Fitzgerald	1925	192	United States
War and Peace	Leo Tolstoy	1869	1225	Russia
Lolita	Vladimir Nabokov	1955	336	France
The Adventures of Huckleberry Finn	Mark Twain	1884	366	United States

Choose the query that returns: **the book title and corresponding author name sorted by publication date with the newest book first.**

- ◯ SELECT Title, Author FROM books ORDER BY Date ASC;
- ◯ SELECT Title, Author FROM books ORDER BY Date;
- ◯ SELECT * FROM books ORDER BY Date DESC;
- ◯ SELECT Title, Author FROM books ORDER BY Date DESC;
- ◯ SELECT Title, Author, SORT(Date) FROM books;

Question 18

1 pts

Assume that the below "books" table is inside a SQL database.

Title	Author	Date	Pages	Country
Anna Karenina	Leo Tolstoy	1877	864	Russia
The Great Gatsby	F. Scott Fitzgerald	1925	192	United States
War and Peace	Leo Tolstoy	1869	1225	Russia
Lolita	Vladimir Nabokov	1955	336	France
The Adventures of Huckleberry Finn	Mark Twain	1884	366	United States

Which SQL query answers the following question: **how many books are available per country?**

- ◯ SELECT Country, SUM(books) FROM books GROUP BY Country;
- ◯ SELECT Country, COUNT(Country) FROM books GROUP BY Date;
- ◯ SELECT Country, COUNT(*) FROM books;
- ◯ SELECT Country, COUNT(*) FROM books GROUP BY Country;
- ◯ SELECT Country, MIN(Pages) FROM books GROUP BY Country;

Question 19

1 pts

Assume that the below "books" table is inside a SQL database.

Title	Author	Date	Pages	Country
Anna Karenina	Leo Tolstoy	1877	864	Russia
The Great Gatsby	F. Scott Fitzgerald	1925	192	United States

War and Peace	Leo Tolstoy	1869	1225	Russia
Lolita	Vladimir Nabokov	1955	336	France
The Adventures of Huckleberry Finn	Mark Twain	1884	366	United States

What is the result of the following SQL query?

```
SELECT Author FROM books
WHERE Country != "Russia" AND Pages < 350
ORDER BY Date
LIMIT 1;
```

○

	Author
0	F. Scott Fitzgerald

○

	Author
0	F. Scott Fitzgerald
1	Vladimir Nabokov

○

	Author
0	Mark Twain

○

	Author
0	Leo Tolstoy

○

	Author
0	Vladimir Nabokov

Question 20

1 pts

Consider the following lines of code. Each line in the code is represented by a line number (for example, #1) to the right.

```
# Line 1 def predict(glucose = 120, bmi = 50, age = 30):
# Line 2    if glucose <= 127.5:
# Line 3        if age <= 28.5:
# Line 4            if bmi <= 45.4:
# Line 5                return False
# Line 6            elif bmi <= 55.3:
# Line 7                return True
# Line 8            else:
# Line 9                return False
# Line 10       else:
# Line 11            return True
```

Select the option that best describes the correct order of execution for the following case:

```
predict (glucose = 120, bmi = 60, age = 25)
```

- ◯ 1, 2, 3, 4, 6, 7
- ◯ 1, 2, 3, 4, 6, 8, 9
- ◯ 1, 2, 3, 4, 8, 10, 11
- ◯ 1, 2, 3, 4, 5, 6, 7
- ◯ 1, 2, 10, 11

Question 21 — 1 pts

Code snippet

```
def predict(glucose = 120, bmi = 50, age = 30):
    #this function has some lines of code here!
```

Which of the following function calls have the correct syntax for the function call? **Select all that apply!**

- ☐ answer = predict(50, 30, glucose = 120)
- ☐ answer = predict(110, glucose = 120, 50, 30)

- [ ] answer = predict(70, 30)
- [ ] answer = predict(glucose = 120)
- [ ] answer = predict(glucose = 120, 50, 30)

Question 22

1 pts

Which of the following statements are **True**? **Select all that apply.**

- [ ] sorted(["a", "C", "b", "D"]) will return ['C', 'D', 'a', 'b']
- [ ] You need to iterate over the dictionary to get the value corresponding to a search key.
- [ ] When you open a file with mode 'r', if the file exists, the existing file is overwritten with the new file.
- [ ] When you create a folder that already exists with os.mkdir(...), the code will produce FileExistsError!

Question 23

1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
{"title": "A",  "year": "disaster year",  "style": "long",  "genres": ["g1", "g
2"]},
{"title": "B",  "year": 19,  "style": "short",  "genres": ["g2", "g3"]},
{"title": "C",  "year": 20,  "style": "short",  "genres": ["g1", "g3"]},
{"title": "D",  "year": 20,  "style": "long",  "genres": ["g1", "g2", "g3"]},
{"title": "E",  "year": 20,  "style": "long",  "genres": ["g2"]}
]
```

What will be printed? Be careful!

```
buckets = {}
bucket = []
for movie in movies:
    year = movie["year"]
    if not year in buckets:
        buckets[year] = bucket
```

```
    for genre in movie["genres"]:
        buckets[year].append(genre)
print(len(buckets[19]))
```

- ◯ 2
- ◯ SyntaxError
- ◯ 4
- ◯ 10
- ◯ KeyError

Question 24

1 pts

Which of the following statements are true after we successfully run the following code? **Select all the options that apply!**

```
import os
os.mkdir ("final_exam")

problems = {"easy": "Question 1",
            "medium": "Question 2",
            "hard": "Question 3"
}

for difficulty in problems:
    folder_path = os.path.join ("final_exam", difficulty)
    os.mkdir (folder_path)
    f = open(os.path.join (folder_path, difficulty+".txt") , "w")
    f.write(problems[difficulty])
    f.close()

    if difficulty == "hard":
        f = open("hard.txt", "w")
        f.write ("This question was challenging\n")
        f.close()
```

- ☐ os.path.isfile(os.path.join("final_exam", "hard.txt")) will return True.
- ☐ os.path.exists("easy.txt") will return True.
- ☐ os.path.isdir("medium") will return True.
- ☐ os.path.isfile("hard.txt") will return True.

- [ ] len(os.listdir("final_exam")) will return 3.

Question 25

1 pts

Consider the following code snippet and then answer the question below it.

```
campus_id = {
    1510121: "Yifan",
    2010333: "Pikachu",
    2194790: "Jay",
    4328132: "Sehun",
    4187523: "Oliver"
}
```

What is the output for:

```
print(type(campus_id), type(campus_id.get("Jay")), type(campus_id.get("1510121",
 0)), type(campus_id.get(2194790)))
```

- ◯ <class 'dict'> <class 'int'> <class 'str'> <class 'NoneType'>
- ◯ <class 'dict'> <class 'int'> <class 'int'> <class 'str'>
- ◯ <class 'dict'> <class 'NoneType'> <class 'str'> <class 'str'>
- ◯ <class 'dict'> <class 'NoneType'> <class 'int'> <class 'str'>
- ◯ <class 'dict'> <class 'NoneType'> <class 'NoneType'> <class 'str'>

Question 26

1 pts

Which lines are executed in the following code? Use the marked line number (comment) to answer this question. Be careful!

Hint: Note the difference between "which lines are executed" and "which lines are successfully executed". The question is asking "which lines are executed".

```
#line 1 def f(a, b):
#line 2     return a+b

#line 3 def g(a):
#line 4     return len(a)

#line 5 def h():
#line 6     try:
#line 7         c = 3
#line 8         d = 4
#line 9         return g(a) * f(c,d)
#line 10     except:
#line 11        return "An error occurred!"

#line 12 try:
#line 13    c = 6
#line 14    d = -100
#line 15    print(f(c,d) + h())
#line 16 except:
#line 17    print("An error occurred!")
```

- ◯ All lines except line 17.
- ◯ All lines except line 11.
- ◯ All lines except line 4, 16, and 17.
- ◯ All lines except line 4.
- ◯ All lines except line 9, 16, and 17.

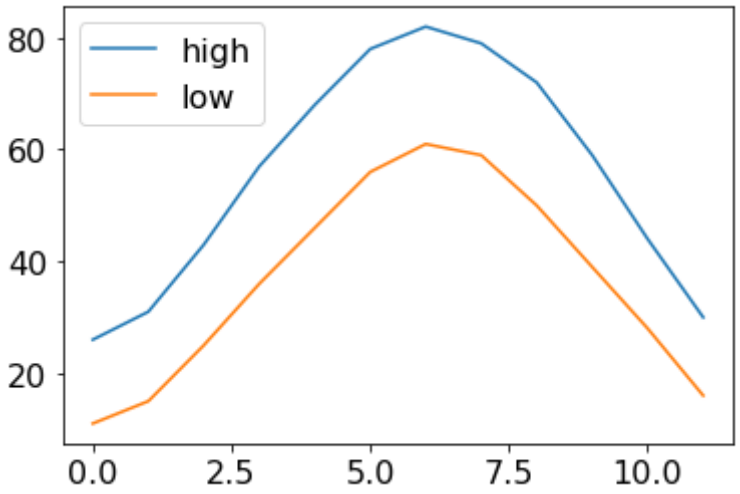
Question 27

1 pts

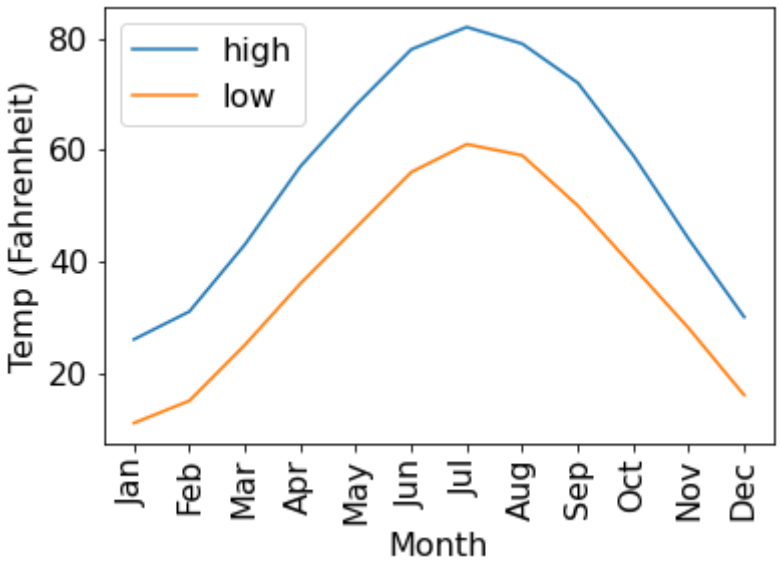
```
df = pd.DataFrame({
"high":  [26, 31, 43, 57, 68, 78, 82, 79, 72, 59, 44, 30],
"low":  [11, 15, 25, 36, 46, 56, 61, 59, 50, 39, 28, 16]
})

ax = df.plot.line()
```

The above code generates the following plot:



In order to improve the plot, we executed a few statements and generated the below plot.



Select all functions that must be called to generate the second plot.

- [ ] ax.set_ylim
- [ ] ax.set_yticks
- [ ] ax.set_ylabel
- [ ] ax.set_xticks
- [ ] ax.set_yticklabels
- [ ] ax.set_xlim
- [ ] ax.set_title

- [ ] ax.set_xticklabels
- [ ] ax.set_xlabel

Question 28

1 pts

Which of the following options creates a bar plot where the x-axis bars are ordered as "green", "blue", and "red" for the following code snippet:

```
s = Series(["red", "red", "red",
            "green", "blue", "blue"])

vc = s.value_counts()
```

- ◯ vc.plot.bar()
- ◯ vc.sort_values().plot.bar()
- ◯ s.set_index("color").plot.bar()
- ◯ vc.sort_index().plot.bar()

Question 29

1 pts

Which of the following enables us to increase font size for plots?

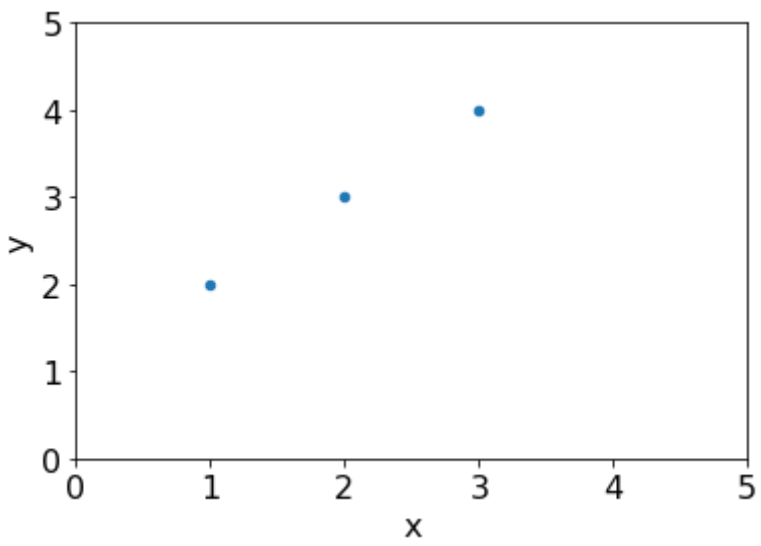
- ◯ `ax["font.size"] = 16`
- ◯ `s.plot.PLOT_FN(figsize = (WIDTH, HEIGHT))`
- ◯ `matplotlib.font.size = 16`
- ◯ `matplotlib.rcParams["font.size"] = 16`

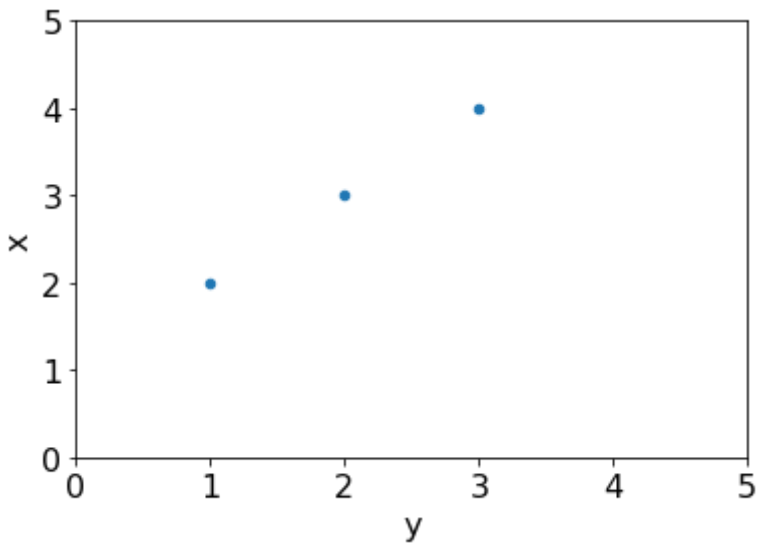
Question 30

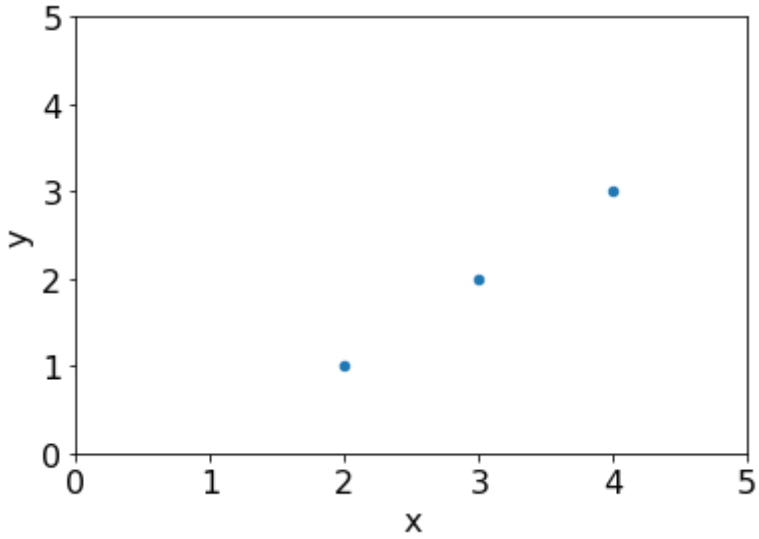
1 pts

Which plot corresponds to the following code snippet?

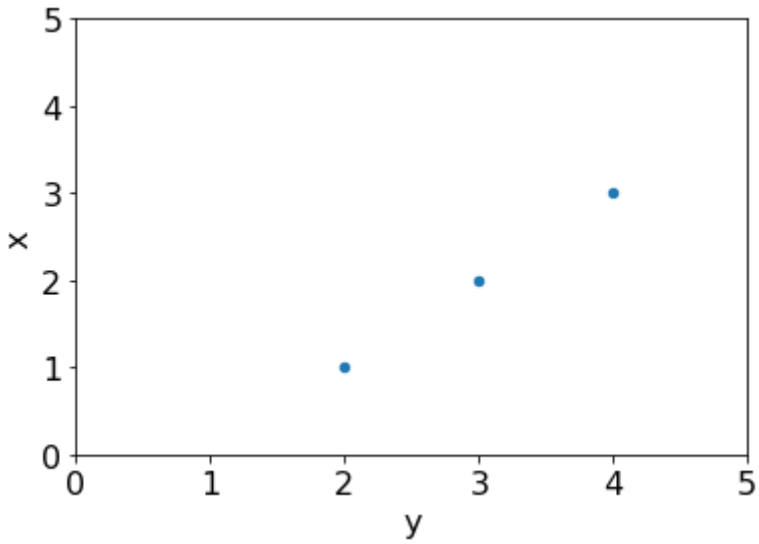
```
data = []
for i in range(3):
    data.append({"y": i+1, "x": i+2})
DataFrame(data).plot.scatter(x='y', y='x')
```







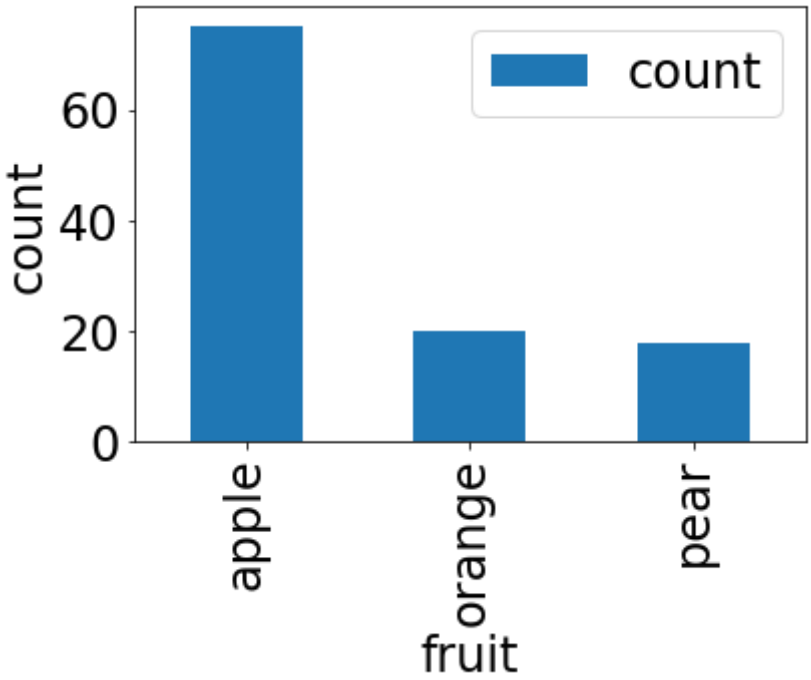
○



Question 31

1 pts

Which code snippet produce the following plot?



Assume the df DataFrame was produced as follows:

```
df = pd.DataFrame({
    "fruit": ["apple", "apple", "pear", "pear", "orange"],
    "color": ["red", "green", "red", "green", "orange"],
    "count": [50, 25, 8, 10, 20]
})
```

- ```
  df.set_index("fruit")["count"].sort_index()[: 3].plot.bar()
  ax.set_xlabel("fruit")
  ax.set_ylabel("count")
  ```

- ```
  s_fruit_count= Series()
  s_fruit_count["apple"] = df[df["count"] == 75]["fruit"]
  s_fruit_count["orange"] = df[df["count"] == 20]["fruit"]
  s_fruit_count["pear"] = df[df["count"] == 18]["fruit"]
  ax = s_fruit_count.plot.bar()
  ax.set_xlabel("fruit")
  ax.set_ylabel("count")
  ```

- ```
  s_fruit_count = Series()
  s_fruit_count["apple"]=df[df["fruit"] == "apple"]["count"].sum()
  s_fruit_count["orange"]=df[df["fruit"]=="orange"]["count"].sum()
  s_fruit_count["pear"]=df[df["fruit"]=="pear"]["count"].sum()
  ax =s_fruit_count.plot.bar()
  ax.set_xlabel("fruit")
  ax.set_ylabel("count")
  ```

- ```
  ax = df.set_index("fruit").sum().plot.bar()
  ax.set_xlabel("fruit")
  ax.set_ylabel("count")
  ```

Question 32

1 pts

The following code attempts to draw a tic-tac-toe board.

```
#sample output
X| |
-+-+-
 | |
-+-+-
 | |
```

```
def draw(x=0, y=0, move="X"):
    i = 1
    while(i < 6):
        if i%2 == 0:
            print("-+-+-", end="")
        else:
            j = 0
            while j < 5:
                if j % 2 != 0:
                    print("|", end="")
                elif i == 2*x + 1 and j == 2*y:
                    print(move, end="")
                else:
                    print(" ", end="")
                j += 1
        print ()
        i += 1
```

What will the following function call produce?

```
draw(-1,-1,'O')
```

- ◯ An empty tic-tac-toe board.
- ◯ A tic-tac-toe board with an 'O' in the lower right corner.
- ◯ A tic-tac-toe board with an 'O' in the upper left corner.
- ◯ TypeError
- ◯ A tic-tac-toe board with an 'X' in the upper left corner.

Question 33

1 pts

The following code attempts to draw a tic-tac-toe board.

```
#sample output
X| |
-+-+-
 | |
-+-+-
 | |
```

```
def draw(x=0, y=0, move="X"):
    i = 1
    while(i < 6): # Highlighted for this question
        if i%2 == 0:
            print("-+-+-", end="")
        else:
            j = 0
            while j < 5:
                if j % 2 != 0:
                    print("|", end="")
                elif i == 2*x + 1 and j == 2*y:
                    print(move, end="")
                else:
                    print(" ", end="")
                j += 1
        print ()
        i += 1
```

How many times does the outer while loop line of code (highlighted in above code) execute for the below function call?

```
draw(2, 2, "X")
```

- ◯ 3
- ◯ 5
- ◯ 30
- ◯ 6

Question 34

1 pts

The following code attempts to draw a tic-tac-toe board.

```
#sample output
X| |
-+-+-
 | |
-+-+-
 | |
```

```
def draw(x=0, y=0, move="X"):
    i = 1
    while(i < 6):
        if i%2 == 0:
            print("-+-+-", end="")
        else:
            j = 0
            while j < 5:
                if j % 2 != 0:
                    print("|", end="")
                elif i == 2*x + 1 and j == 2*y:
                    print(move, end="")
                else:
                    print(" ", end="")
                j += 1
        print ()
        i += 1
```

Which of the following function calls will draw the board with an X in the top row center cell?

- ◯ draw(0,1,X)
- ◯ draw(1,2,"X")
- ◯ draw(1,0,'X')
- ◯ draw(0,1)

Question 35

1 pts

The following code attempts to draw a tic-tac-toe board.

```
#sample output
X| |
-+-+-
 | |
-+-+-
 | |
```

```
def draw(x=0, y=0, move="X"):
    i = 1
    while(i < 6):
```

```
        if i%2 == 0:
            print("-+-+-", end="")
        else:
            j = 0
            while j < 5:
                if j % 2 != 0:
                    print("|", end="")
                elif i == 2*x + 1 and j == 2*y:
                    print(move, end="")
                else:
                    print(" ", end="")
                j += 1
        print ()
        i += 1
```

What will the following code produce?

```
x=int("1")
y="1"
draw(x,y)
```

- ◯ A tic tac toe board with an X in the center.
- ◯ A tic tac toe board with an X in the upper left cell.
- ◯ TypeError
- ◯ An empty tic-tac-toe board.

Not saved Submit Quiz