

Name: \_\_\_\_\_ حيدر بسام اسماعيل \_\_\_\_\_, Number: 1387, Submitted To GitHub: \_\_\_\_\_

### Question 1:

**A-**If you have two lists, L1=['HTTP','HTTPS','FTP','DNS'] L2=[80,443,21,53], convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53 }

### Answer:

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
```

```
L2 = [80, 443, 21, 53]
```

```
DD = dict(zip(L1, L2))
```

```
print(DD)
```

### Output:

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
```

### Explain :

في السطر الأول :

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
```

تم إنشاء قائمة تحتوي على أسماء بروتوكولات الاتصال مثل HTTP و HTTPS و FTP و DNS.

في السطر الثاني :

```
L2 = [80, 443, 21, 53]
```

تم إنشاء قائمة أخرى تحتوي على أرقام المنافذ المرتبطة بكل بروتوكول

في السطر الثالث :

```
DD = dict(zip(L1, L2))
```

تم دمج الاقائمتين ( L1 و L2 ) معًا باستخدام دالة zip التي تجمع كل عنصر من L1 مع العنصر المقابل له في L2 ثم تم استخدام دالة zip لتحويل النتيجة الى قاموس

وأخيراً في السطر الرابع :

```
print(DD)
```

يتم طباعة القاموس DD الذي يحتوي على اسماء البروتوكولات كمفاتيح وأرقام المنافذ كقيم

**B- Write a Python program that calculates the factorial of a given number entered by user.**

**Answer :**

```
x = int(input("the number: "))

def factorial(a):

    if a == 0:

        return 1

    else:

        return a * factorial(a-1)

fact = factorial(x)

print(f"The factorial of {x} is {fact}")
```

**Output:**

```
| the number: 20
| The factorial of 20 is 2432902008176640000
|>>>
```

**Explain :**

في السطر الأول :

***x = int(input("the number: "))***  
يُطلب من المستخدم إدخال عدد صحيح، ويتم تخزين هذا العدد في المتغير x بعد تحويله إلى نوع البيانات integer.

في السطر الثاني :

***def factorial(a):***  
 ***if a == 0:***  
 ***return 1***  
 ***else:***  
 ***return a \* factorial(a-1)***

تم تعريف دالة باسم factorial تأخذ عددًا صحيحًا كمُدخل. إذا كان العدد المُدخل يساوي صفر، فإن الدالة تعود بقيمة 1. وإذا كان العدد غير صفر، فإن الدالة تقوم بحساب عامل التثنية للعدد بالضرب في عامل التثنية للعدد (a-1)، وهكذا حتى يصل العدد إلى صفر.

في السطر الثالث :

***fact = factorial(x)***

يتم استدعاء الدالة factorial وتمرير قيمة المتغير x إليها، ونتيجة ذلك يتم تخزينها في المتغير fact.

في السطر الرابع :

```
print(f"The factorial of {x} is {fact}")
```

يتم طباعة نص يحتوي على قيمة x التي قام المستخدم بإدخالها، وقيمة fact التي تمثل عامل التثنية لهذا العدد.

**C-** L=['Network' , 'Bio' , 'Programming' , 'Physics' , 'Music']

In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with 'B' letter, then print it on screen.

Tips: using loop, 'len ()' , startswith() methods.

**Answer:**

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
```

```
for z in L:
```

```
    if z.startswith('B'):
```

```
        print(z)
```

**Output:**

```
> Bio
```

**Explain :**

السطر الأول :

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
```

تم إنشاء قائمة L تحتوي على عدة عناصر نصية.

في السطر الثاني والثالث :

```
for z in L:
```

```
    if z.startswith('B'):
```

```
        print(z)
```

يتم استخدام حلقة for للانتقال عبر كل عنصر في القائمة L. ثم يتحقق الشرط if z.startswith('B')، حيث يفحص إذا كان العنصر يبدأ بحرف "B" أو لا. إذا كان الشرط صحيحاً (أي العنصر يبدأ بحرف "B")، فيتم طباعة هذا العنصر.

بناءً على قائمة L التي تم تعريفها، سيتم طباعة العنصر "Bio" لأنه هو الوحيد الذي يبدأ بحرف "B" من بين العناصر الموجودة في القائمة.

**D: Using Dictionary comprehension, Generate this dictionary**

**d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}**

**Answer:**

```
H = {i: i+1 for i in range(11)}
```

```
print(H)
```

**Output:**

```
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
```

**Explain :**

السطر الأول :

***H = {i: i+1 for i in range(11)}***

تم استخدام تعبير قاموس (Dictionary Comprehension) لإنشاء القاموس H. يتم استخدام حلقة for للانتقال عبر الأرقام من 0 إلى 10 (من خلال range(11))، ويتم إنشاء مفتاح لكل رقم i وقيمته تكون i+1.

بمعنى آخر، سيكون القاموس H على النحو التالي:

{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}

السطر الثاني :

***print(H)***

يتم طباعة القاموس H، والذي سيظهر كما ذكرنا أعلاه. يحتوي القاموس على مفاتيح تتبّع سلسلة الأرقام من 0 إلى 10، ولكل مفتاح قيمة تكون بقيمة أكبر بواحد من قيمة المفتاح

---

## **Question 2: Convert from Binary to Decimal**

**Write a Python program that converts a Binary number into its equivalent Decimal number.**

The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen.

**Tips:** solve input errors

**Answer:**

```
def binary_to_decimal(binary_str):  
  
    try:  
  
        for digital in binary_str:  
  
            if digital not in ['0', '1']:  
  
                raise ValueError("Input must be only 1s or 0s.")  
  
        decim = int(binary_str, 2)  
  
        return decim  
  
    except ValueError as y:  
  
        print(y)  
  
        return None  
  
binary_number = input("Enter a binary number: ")  
  
decimal_number = binary_to_decimal(binary_number)  
  
if decimal_number is not None:  
  
    print(f"The equivalent decimal number is: {decimal_number}")
```

**Output:**

```
Enter a binary number: 1010  
The equivalent decimal number is: 10
```

**Explain :**

1. تعريف دالة `binary_to_decimal`:  
تم تعريف دالة باسم `binary_to_decimal` التي تأخذ وسيطة واحدة `binary_str` التي تمثل الرقم الثنائي كنص يتم استخدام (`try`) و (`except`) للتعامل مع الأخطاء المحتملة  
- يتم التحقق من أن كل رقم في النص الثنائي يكون إما '0' أو '1'. إذا كان أي رقم آخر موجود،  
تم رفع خطأ من نوع `ValueError` يُفيد أن الإدخال يجب أن يكون فقط '0' أو '1'.

- يتم تحويل النص الثنائي إلى رقم عشري باستخدام دالة `int(binary_str, 2)`، حيث يتم تحديد أن الرقم هو في النظام الثنائي.
- إذا تم التحويل بنجاح، يتم إرجاع القيمة العشرية.
- 2. قراءة الرقم الثنائي من المستخدم:
  - يُطلب من المستخدم إدخال رقم ثنائي عبر استدعاء `.input("Enter a binary number: ")`.
  - القيمة المدخلة تُخزن في المتغير `binary_number`.
- 3. تحويل الرقم الثنائي إلى عشري:
  - يتم استدعاء دالة `binary_to_decimal(binary_number)` لتحويل الرقم الثنائي إلى عشري.
  - القيمة المُعادَة من الدالة تُخزن في المتغير `decimal_number`.
- 4. طباعة الرقم العشري:
  - إذا كان الرقم العشري غير مساوٍ لـ `None` (أي تم التحويل بنجاح):
  - يُطبع رسالة تحتوي على الرقم العشري المحول باستخدام `.print(f"The equivalent decimal number is: {decimal_number}")`.

#### بإختصار :

هذا الكود يأخذ رقمًا ثنائيًا من المستخدم، يُحوّله إلى عشري، ويُطبع النتيجة. في حال وجود أي خطأ في إدخال الرقم الثنائي، سيتم طباعة رسالة خطأ.

### Question 3: Working with Files” Quiz Program”

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.

#### Answer:

```
import json

import csv

def calculate_score(user_answers, correct_answers):

    score = 0

    for question, answer in user_answers.items():

        if answer == correct_answers.get(question):

            score += 1

    return score

def save_results(user_name, user_answers, score, file_format):

    file_name = f'{user_name}_quiz_results.{file_format}'
```

```

if file_format == 'json':

    with open(file_name, 'w') as json_file:

        json.dump({'User': user_name, 'Score': f'{score}/20', 'User Answers': user_answers}, json_file)

    print('Quiz results saved in JSON format.')

elif file_format == 'csv':

    with open(file_name, 'w', newline='') as csv_file:

        csv_writer = csv.writer(csv_file)

        csv_writer.writerow(['Question', 'User Answer'])

        for question, answer in user_answers.items():

            csv_writer.writerow([question, answer])

        csv_writer.writerow(['Score', f'{score}/20'])

    print('Quiz results saved in CSV format.')

else:

    print('Invalid file format. Results not saved.')

user_name = input('Enter your name: ')

file_name = input('Enter the file name with questions and answers: ')

with open(file_name, 'r') as file:

    data = file.read()

questions_and_answers = json.loads(data)

user_answers = {}

for question in questions_and_answers.keys():

    user_answer = input(f'Your answer for "{question}": ')

    user_answers[question] = user_answer

score = calculate_score(user_answers, questions_and_answers)

print(f'Your score is: {score}/20')

file_format = input('Choose the file format to save the results (JSON or CSV): ')

```

save\_results(user\_name, user\_answers, score, file\_format)

output :

```
Enter your name: haidar ismail
Enter the file name with questions and answers: questions.txt
Your answer for " What does the term black gold refer to? ": petroleum
Your answer for " Who was the first people to use a wristwatch?": octopus
Your answer for " How many types of antibodies are there in the human body? "
Your answer for " In 1865 AD, one of the Presidents of the United States of A
Your answer for " What is the atomic number of the element mercury? ": f
Your answer for " Which science is the oldest science in human history?":
Your answer for " What are the most thirsty animals? ": camels
Your answer for " To which two planets was the Mariner 10 spacecraft sent? "
Your answer for " What sea creature has three hearts? ": f
Your answer for " Who invented the light bulb? ": t
Your answer for " What is the heaviest animal in the world?": f
Your answer for " What is the fastest bird in the world? ": t
Your answer for " Who is the princess who married the Shah of Iran?": t
Your answer for "How many times can female cats give birth per year? ":
Your answer for "Who was the first people to use a wristwatch? ":
Your answer for "What is the name that the ancient Arabs gave to the metal g
Your answer for "Mention the name of the city called the City of the Sun? ":
Your answer for "Who is the king who founded the Kingdom of Saudi Arabia? ":
Your answer for "How many hearts does an octopus have?":
Your score is: 0/20
Choose the file format to save the results (JSON or CSV): json
Quiz results saved in JSON format.
```

Explain :

شرح الكود خطوة بخطوة:

1. يتم استيراد المكتبات **json** و **csv**.
2. تعريف الدالة **calculate\_score(user\_answers, correct\_answers)**:
  - تقوم بحساب درجة المستخدم بناءً على الإجابات التي أدخلها المستخدم والإجابات الصحيحة.
  - تبدأ بتهيئة متغير **score** إلى صفر.
  - تقوم بتفحص كل سؤال وإجابة المستخدم، وإذا كانت الإجابة صحيحة، يتم زيادة قيمة **score** بواحد.
  - تُرجع درجة المستخدم.وذلك باستخدام حلقة **for** للمرور على إجابات المستخدم والإجابات الصحيحة، وزيادة قيمة **score** بواحد لكل إجابة صحيحة.
3. تعريف الدالة **save\_results(user\_name, user\_answers, score, file\_format)**



- تقوم بحفظ نتائج الاختبار في ملف بتنسيق JSON أو CSV.
- تُنشئ اسم الملف باستخدام اسم المستخدم وتنسيق الملف المحدد.
- إذا كان تنسيق الملف هو JSON، يتم حفظ النتائج في ملف JSON.
- إذا كان تنسيق الملف هو CSV، يتم حفظ النتائج في ملف CSV.
- إذا كان التنسيق غير معروف، يُطبع رسالة بأن التنسيق غير صالح.
- 4. يُطلب من المستخدم إدخال اسمه واسم الملف الذي يحتوي على الأسئلة والإجابات.
- 5. يتم فتح الملف وقراءة البيانات منه باستخدام (`json.loads()`)

لتحويل البيانات إلى قاموس `questions_and_answers`.

- 6. يُطلب من المستخدم إدخال إجاباته لكل سؤال.
- 7. يُحسب الكود درجة المستخدم باستخدام دالة `calculate_score()` ويعرضها.
- 8. يُطلب من المستخدم اختيار تنسيق الملف (JSON أو CSV).
- 9. يُستدعى الدالة (`save_results()`) لحفظ نتائج الاختبار في ملف بالتنسيق المحدد.

باختصار، هذا الكود يقوم بإجراء اختبار، حساب درجة المستخدم، وحفظ نتائج الاختبار في ملف بتنسيق JSON أو CSV حسب اختيار المستخدم.

---

#### Question 4: Object-Oriented Programming - Bank Class

Define a class `BankAccount` with the following attributes and methods:

Attributes: `account_number` (string), `account_holder` (string), `balance` (float, initialized to 0.0)

Methods: `deposit(amount)`, `withdraw(amount)`, `get_balance()`

- Create an instance of `BankAccount`, - Perform a deposit of \$1000, - Perform a withdrawal of \$500.
- Print the current balance after each operation.

- Define a subclass SavingsAccount that inherits from BankAccount and adds interest\_rate Attribute and

apply\_interest() method that Applies interest to the balance based on the interest rate.

And Override print() method to print the current balance and rate.

- Create an instance of SavingsAccount , and call apply\_interest() and print() functions.

**Answer:**

```
class BankAccount:
```

```
    def __init__(self, account_number, account_holder):
```

```
        self.account_number = account_number
```

```
        self.account_holder = account_holder
```

```
        self.balance = 0.0
```

```
    def deposit(self, amount):
```

```
        self.balance += amount
```

```
    def withdraw(self, amount):
```

```
        if self.balance >= amount:
```

```
            self.balance -= amount
```

```
        else:
```

```
            print("Insu funds.")
```

```
    def get_balance(self):
```

```
        return self.balance
```

```
class SavingsAccount(BankAccount):
```

```
    def __init__(self, account_number, account_holder, interest_rate):
```

```
        super().__init__(account_number, account_holder)
```

```
        self.interest_rate = interest_rate
```

```
    def apply_interest(self):
```

```
        self.balance += self.balance * self.interest_rate
```

```

def __str__(self):
    return f"Account holder: {self.account_holder}\nAccount Balance: ${self.balance:.2f}\nInterest Rate: {self.interest_rate*100:.2f}%"

bank_account = BankAccount("11223344", "Ismail")

bank_account.deposit(1000)

print(f"Current Balance: ${bank_account.get_balance():.2f}")

bank_account.withdraw(500)

print(f"Current Balance: ${bank_account.get_balance():.2f}")

savings_account = SavingsAccount("44556677", "Haidar ", 0.05)

savings_account.apply_interest()

print(savings_account)

```

**Output:**

```

Current Balance: $1000.00
Current Balance: $500.00
Account holder: Haidar
Account Balance: $0.00
Interest Rate: 5.00%

```

Explain :

هذا الكود يحتوي على تعريفين للفئتين BankAccount و SavingsAccount ،

### 1. الفئة BankAccount:

- يتم تعريف الصنف باستخدام الكلمة المفتاحية class واسم الصنف BankAccount.

- يحتوي الصنف على ثلاثة طرق:

\_\_init\_\_(self, account\_number, account\_holder): يُستخدم لتهيئة البيانات الأساسية لحساب المصرف، مثل رقم الحساب واسم صاحب الحساب.

deposit(self, amount): يزيد الرصيد بالقيمة المحددة.

withdraw(self, amount): يقوم بسحب مبلغ معين من الرصيد إذا كان متاحًا، وإلا يُطبع رسالة بأن الرصيد غير كافٍ

get\_balance(self): - يُرجع قيمة الرصيد الحالية

## 2. الفئة SavingsAccount التي ترث من BankAccount:

- تضيف خاصية interest\_rate لتمثيل معدل الفائدة على الحساب.
- تحتوي على أسلوب apply\_interest(self) لحساب الفائدة وإضافتها إلى الرصيد.
- تُعيد نصًا يحتوي على معلومات الحساب باستخدام أسلوب str(self).

## 3. في الجزء الأخير من الكود:

- يتم إنشاء حساب مصرفي عادي باسم bank\_account وإضافة 1000 دولار كودائع.
- يُطبع الرصيد الحالي للحساب.
- يتم سحب 500 دولار من الحساب وإعادة طباعة الرصيد.
- يتم إنشاء حساب توفير باسم savings\_account بمعدل فائدة 5% وتطبيق الفائدة.
- يُطبع نص يحتوي على معلومات الحساب التوفير.

## باختصار:

الكود يُظهر كيفية استخدام الوراثة في Python لإنشاء فئة مشتقة من فئة أخرى، وكذلك كيفية استخدام أساليب المصفوفات لإجراء عمليات إيداع وسحب وحساب الفائدة على حساب توفير.

---