



وظيفة برمجة الشبكات

إعداد الطالبة: سالي عز الدين سلطان



2023/2024

TISHREEN UNIVERSITY

Question 1: Python Basics?

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 }

الشرح: استخدمت التابع dict() لبناء القاموس d تم تمرير التابع zip() كبارمتر داخل التابع وذلك لمقابلة قيم القائمتين حسب الفهارس، ومن ثم عن طريق الباني dict() حولنا النتيجة لقاموس.

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
L2 = [80, 443, 21, 53]

d = dict(zip(L1, L2))

print(d)
```

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
PS C:\Users\ASUS\Documents\Python Scripts>
```

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

```
def factorial(n):
    if n < 0:
        raise ValueError("n must be non-negative")

    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

number = int(input("Enter a non-negative number: "))

try:
    result = factorial(number)
    print("The factorial of {} is {}".format(number, result))
except ValueError as e:
    print("Error: {}".format(e))
```

```
Enter a non-negative number: 7
The factorial of 7 is 5040
PS C:\Users\ASUS\Documents\Python Scripts>
```

تم بناء التابع factorial() لحساب عاملي العدد المدخل ثم باستخدام مفهوم العودية تم حساب العاملي وباستخدام بنية try- except تم معالجة الاستثناء الحاصل في حال إدخال عدد سالب.

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.

قمت بتعريف القائمة L وباستخدام الحلقة for قمنا بالمرور على عناصر القائمة واختبار بواسطة الميثود startswith() الكلمات التي تبدأ ب B.

```
L = ['Network' , 'Bio' , 'Programming' , 'Physics' , 'Music']  
  
for item in L:  
    if item.startswith('B'):  
        print(item)
```

Bio
PS C:\Users\ASUS\Documents\Python Scripts>

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}

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

{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
PS C:\Users\ASUS\Documents\Python Scripts>


باستخدام طريقة بناء القواميس Dictionary comprehension تم إنشاء القاموس d وعناصره (المفاتيح، القيم) تم الحصول عليها بواسطة استخدام الحلقة for وتابع الـ range.


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.

```
def binary_to_decimal(binary):
    try:
        if not all(char in ('0', '1') for char in binary):
            raise ValueError("Invalid binary string. Only '0' and '1' characters allowed.")
        decimal_value = int(binary, 2)
        return decimal_value
    except ValueError as err:
        print("Error: {}".format(err))
        return None

binary_number = input("Enter a binary number: ")
decimal_equivalent = binary_to_decimal(binary_number)

if decimal_equivalent is not None:
    print("The decimal equivalent of {} is {}".format(binary_number, decimal_equivalent))
```



```
Enter a binary number: 010111101
The decimal equivalent of 010111101 is 189
PS C:\Users\ASUS\Documents\Python Scripts> 
```

تم أخذ العدد الثنائي من المستخدم على شكل سلسلة محارفية، ثم أجرينا اختبار إن كانت عناصر هذه السلسلة هي أرقام من النظام الثنائي (0,1) أم لا، إن لم تكن سنظهر رسالة خطأ. باستخدام الباني `int()` وبتمرير العدد المدخل ثم أساس النظام الثنائي 2 قمنا بتحويل العدد الثنائي لـ عشري.

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.

```
import csv

try:
    with open('my_qes.csv') as file:
        reader = csv.reader(file)
        questions = []
        answers = []
        for row in reader:
            questions.append(row[0])
            answers.append(row[1])
except FileNotFoundError:
    print("The file 'my_qes.csv' was not found.")
    exit()
except IndexError:
    print("There was an error reading the 'my_qes.csv'. Please check the file format.")
    exit()

name = input("Write Your name: ")

count = 0

for i in range(len(questions)):
    answer = input(questions[i] + " ")
    if answer.lower() == answers[i].lower():
        print("correct")
        count += 1
    else:
        print("Incorrect answer !")
```

```
what is your name saly
correct
How old are you good
Incorrect answer !
1 + 1 2
correct
2 + 2 4
correct
3 +3 6
correct
4 + 4 8
correct
5+5 10
correct
6+6 12
correct
7+7 14
correct
YOU got : 8
```

```
my_qes.csv > data
1  what is your name,saly
2  How old are you,23
3  1 + 1,2
4  2 + 2,4
5  3 + 3,6
6  4 + 4,8
7  5 + 5,10
8  6 + 6,12
9  7 + 7,14
10 8 + 8,16
11 9 + 9,18
12 10 + 10,20
```

تم وضع الأسئلة في ملف

my_qes.csv حيث أن العنصر ذو الفهرس 0 في السطر يمثل السؤال والعنصر ذو الفهرس 1 يمثل الجواب. يتم تخزين اسم المستخدم الذي يجيب على الأسئلة مع نتيجته في الملف my_ans.csv. يتم التعامل من خلال الـ csv module في بايثون في برنامجي مع هذه البيانات وتخزينها في قائمة الأسئلة بعد جلبها. ونستخدم قائمة أخرى هي قائمة الأجوبة لتخزين النتيجة.

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.

```
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
        print("Deposited ${:.2f}. New balance: ${:.2f}".format(amount, self.balance))

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient funds. Current balance: ${:.2f}".format(self.balance))
        else:
            self.balance -= amount
            print("Withdrew ${:.2f}. New balance: ${:.2f}".format(amount, self.balance))

    def get_balance(self):
        return self.balance

    def __str__(self):
        return "Account Holder: {}\nAccount Number: {}\nBalance: ${:.2f}".format(
            self.account_holder, self.account_number, 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):
        interest = self.balance * self.interest_rate
        self.balance += interest
        print("Applied interest: ${:.2f}. New balance: ${:.2f}".format(interest, self.balance))

    def __str__(self):
        return "{}, Interest Rate: {:.2%}".format(
            super().__str__(), self.interest_rate
        )

my_account = BankAccount("7833438", "Saly_1")
my_account.deposit(1000.00)
my_account.withdraw(500.00)
print("Current balance: ${:.2f}".format(my_account.get_balance()))

savings_account = SavingsAccount("4520314", "Saly_2", 0.05)
savings_account.apply_interest()
print(savings_account)

```

```

Deposited $1000.00. New balance: $1000.00
Withdrew $500.00. New balance: $500.00
Current balance: $500.00
Applied interest: $0.00. New balance: $0.00
Account Holder: Saly_2nAccount Number: 4520314nBalance: $0.00, Interest Rate: 5.00%
PS C:\Users\ASUS\Desktop\اسم > 

```

يتم تعريف الكلاس BankAccount الذي يحتوي على متغيرات لرقم الحساب، اسم صاحب الحساب، والرصيد. يحتوي الكلاس أيضًا على دوال لإيداع المبلغ، سحب المبلغ، الحصول على الرصيد الحالي، وعرض معلومات الحساب.

ثم يتم تعريف الكلاس SavingsAccount الذي يرث من BankAccount ويضيف متغير لمعدل الفائدة. يحتوي هذا الكلاس على دالة لتطبيق الفائدة على الرصيد.

تم إنشاء حساب my_account من نوع BankAccount وتم تنفيذ عمليات إيداع وسحب وطباعة الرصيد الحالي.

ثم تم إنشاء حساب savings_account من نوع SavingsAccount وتم تنفيذ دالة تطبيق الفائدة وطباعة معلومات الحساب.

بكل تأكيد! الكود الموجود يعرف نظاماً بسيطاً لإدارة حسابات بنكية في لغة البرمجة بايثون. إليك شرحاً مبسطاً لمكونات الكود:

1. كلاس BankAccount

- هذا الكلاس يمثل حساب بنكي ويحتوي على ثلاث خصائص: رقم الحساب (account_number)، اسم صاحب الحساب (account_holder)، والرصيد (balance) الذي يبدأ من الصفر.
- لديه دالة deposit تسمح لك بإيداع مبلغ في الحساب وتضيف هذا المبلغ إلى الرصيد الحالي، ثم تطبع رسالة تخبرك بالمبلغ المودع والرصيد الجديد.
- ودالة withdraw تسمح لك بسحب مبلغ من الحساب إذا كان الرصيد كافياً، وإذا لم يكن كافياً تطبع رسالة تخبرك بأن الأموال غير كافية.
- ودالة get_balance تعطيك الرصيد الحالي للحساب.
- ودالة __str__ تعطيك معلومات الحساب عندما تريد طباعة الكائن.

2. كلاس SavingsAccount

- هذا الكلاس يمثل نوع خاص من الحسابات البنكية وهو حساب التوفير، وهو يرث من كلاس BankAccount.
- يحتوي على خاصية إضافية وهي نسبة الفائدة (interest_rate).

- لديه دالة `apply_interest` تحسب الفائدة بناءً على الرصيد الحالي ونسبة الفائدة وتضيف الفائدة إلى الرصيد.

- ودالة `__str__` معدلة لتظهر معلومات الحساب بالإضافة إلى نسبة الفائدة.

3. إنشاء واستخدام الحسابات:

- يتم إنشاء كائن من كلاس `BankAccount` باسم `my_account` ويتم إيداع وسحب مبالغ منه.

- ثم يتم إنشاء كائن من كلاس `SavingsAccount` باسم `savings_account` ويتم تطبيق الفائدة عليه.

- في النهاية، يتم طباعة معلومات كل حساب والرصيد الحالي.