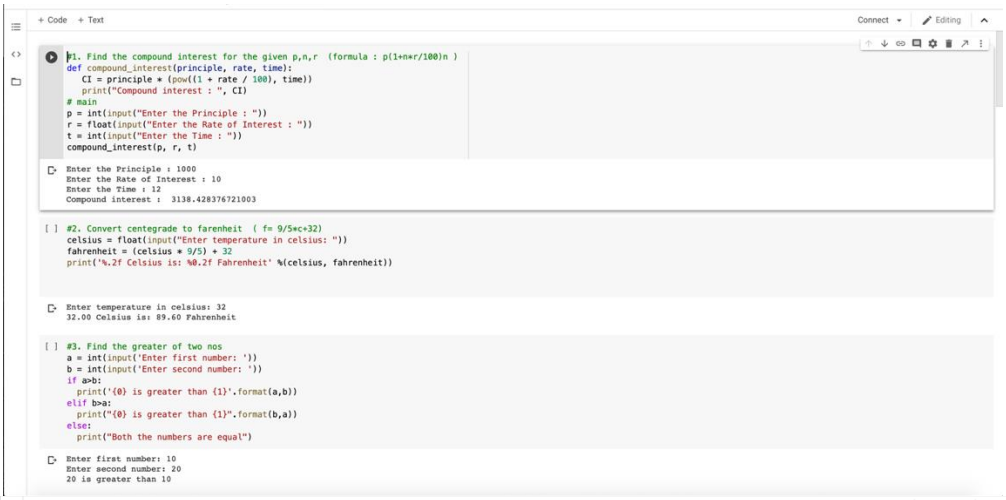
DAILY ASSESSMENT FORMAT

Date:	21/07/2020	Name:	K B KUSHI
Course:	Python	USN:	4AL17EC107
Topic:	• Pythonic	Semester & Section:	6 & B
Github Repository:	https://github.com/alvas-education-foundation/KUSHI-COURSES.git		

SESSION DETAILS

Session images



```
#1. Find the compound interest for the given p,n,r  (formula : p(1+n*r/100)n )
def compound_interest(principle, rate, time):
    CI = principle * (pow((1 + rate / 100), time))
    print("Compound interest : ", CI)
# main
p = int(input("Enter the Principle : "))
r = float(input("Enter the Rate of Interest : "))
t = int(input("Enter the Time : "))
compound_interest(p, r, t)
```

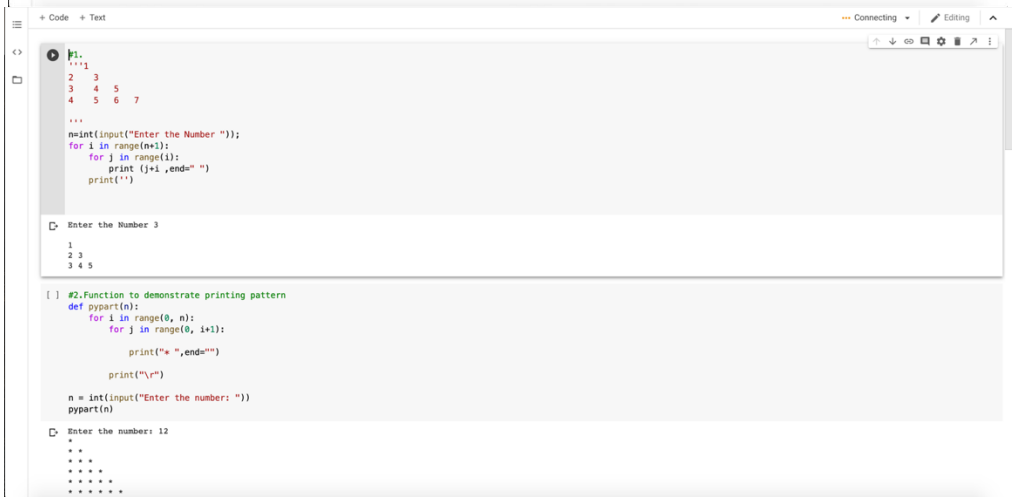
```
Enter the Principle : 1000
Enter the Rate of Interest : 10
Enter the Time : 12
Compound interest :  3138.428376721003
```

```
#2. Convert centegrade to farenheit  ( f= 9/5*c+32)
celsius = float(input("Enter temperature in celsius: "))
fahrenheit = (celsius * 9/5) + 32
print('%.2f Celsius is: %0.2f Fahrenheit' %(celsius, fahrenheit))
```

```
Enter temperature in celsius: 32
32.00 Celsius is: 89.60 Fahrenheit
```

```
#3. Find the greater of two nos
a = int(input('Enter first number: '))
b = int(input('Enter second number: '))
if a>b:
  print('{0} is greater than {1}'.format(a,b))
elif b>a:
  print("{0} is greater than {1}".format(b,a))
else:
  print("Both the numbers are equal")
```

```
Enter first number: 10
Enter second number: 20
20 is greater than 10
```



```
#1.
'''1
2   3
3   4   5
4   5   6   7

'''
n=int(input("Enter the Number "));
for i in range(n+1):
    for j in range(i):
        print (j+i ,end=" ")
    print('')
```

```
Enter the Number 3

1
2 3
3 4 5
```

```
#2.Function to demonstrate printing pattern
def pypart(n):
    for i in range(0, n):
        for j in range(0, i+1):

            print("* ",end="")

        print("\r")

n = int(input("Enter the number: "))
pypart(n)
```

```
Enter the number: 12
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

Report:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting

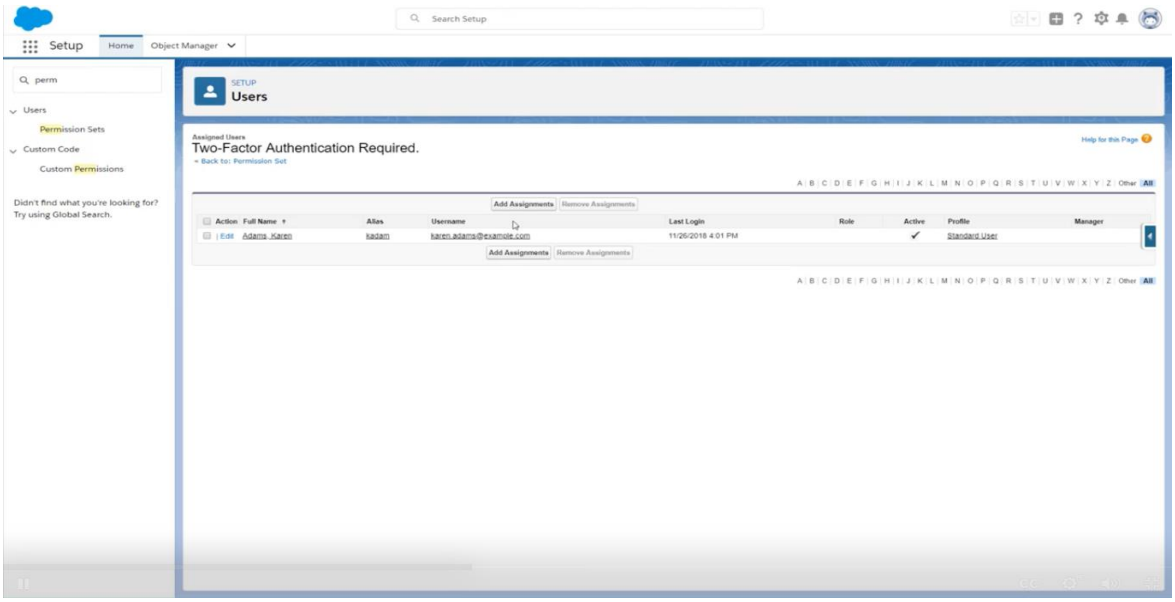
or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. **Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.**

DAILY ASSESSMENT FORMAT

Date:	21/07/2020	Name:	K B KUSHI
Course:	Salesforce	USN:	4AL17EC107
Topic:	• Build-your-career-with-salesforce-skills	Semester & Section:	6 & B
Github Repository:	https://github.com/alvas-education-foundation/KUSHI-COURSES.git		

SESSION DETAILS

Session images



Report:

Secure Your Users' Identity

- **Learning Objectives**

  After completing this module, you'll be able to:

  Describe ways to identify your users in addition to a username and password.

  Set up two-factor authentication.

  Use the Salesforce Authenticator app to verify identity.

  Get login information about users who log in to your org.

- **Secure Identity with Two-Factor Authentication and Salesforce Authenticator:**

As an admin, you probably walk a fine line between making sure that your Salesforce org is secure and that your users can log in quickly and easily.

- **The most effective way to protect your org and its data is to require that users provide more than just their username and password. Security experts call this two-factor authentication, or 2FA for short.**

What Is Two-Factor Authentication?

- **Sounds like a mathematical equation, right? Whether math thrills you or fills you with dread, just know that 2FA has nothing to do with high school algebra. But it has everything to do with making sure that your users are who they say they are.**

What are the two factors?

- **Something users know, like their password**
- **Something users have, such as a mobile device with an authenticator app installed**
- **That second factor of authentication provides an extra layer of security for your org.**

As an admin, you can require it every time your users log in. Or you can require it only in some circumstances, such as when users log in from an unrecognized device or try to access a high-risk application.

- **After users successfully verify their identity with both authentication factors, they can access Salesforce and start working.**

How Two-Factor Authentication Works

- **You might not have known what it's called, but you've probably already used two-factor authentication.**
- **Every time you get cash from the ATM, you use something you have (your bank card) plus something you know (your PIN). And maybe you already have an authenticator app on your phone. For instance, you enter a verification code that you get from the app when you log in to some of your online accounts.**
- **This unique code is sometimes called a time-based one-time password (or TOTP for short) because it expires after a set amount of time.**
- **Several vendors, including Salesforce and Google, provide apps that generate these time-sensitive codes.**

Set Up Two-Factor Authentication for Every Login

- **Now that you know the basics of two-factor authentication, let's see how easy it is to set up.**

Suppose you're a Salesforce admin for Jedeye Technologies, a company *not* located in a galaxy far, far away.

- **Your chief security officer has handed you a mission: Make all employees supply more than their username and password every time they try to access the company's Salesforce org.**

Step 1: Set the session security level for two-factor authentication

Step 2: Create a user

Step 3: Create a permission set for two-factor authentication

Step 4: Assign the permission set to Sia's users.