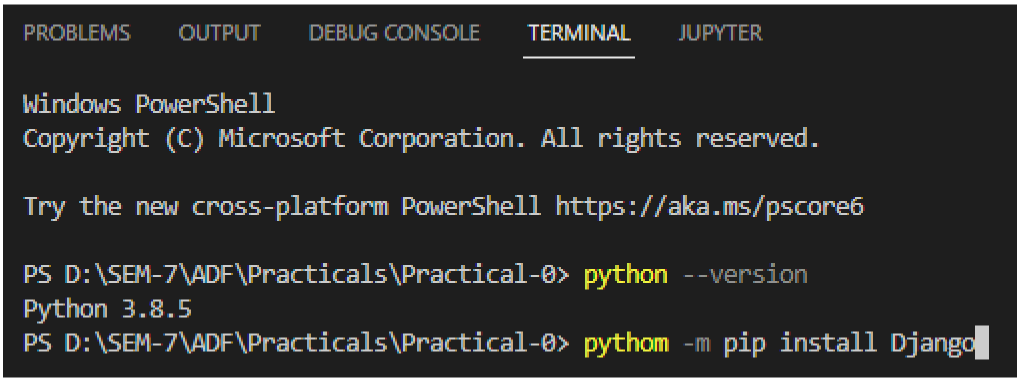
Practical-0

Configuring and Installation of Django

1. Install Python

2. Open prompt (Ensure you have internet connection working)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\SEM-7\ADF\Practicals\Practical-0> python --version
Python 3.8.5
PS D:\SEM-7\ADF\Practicals\Practical-0> pythom -m pip install Django
```

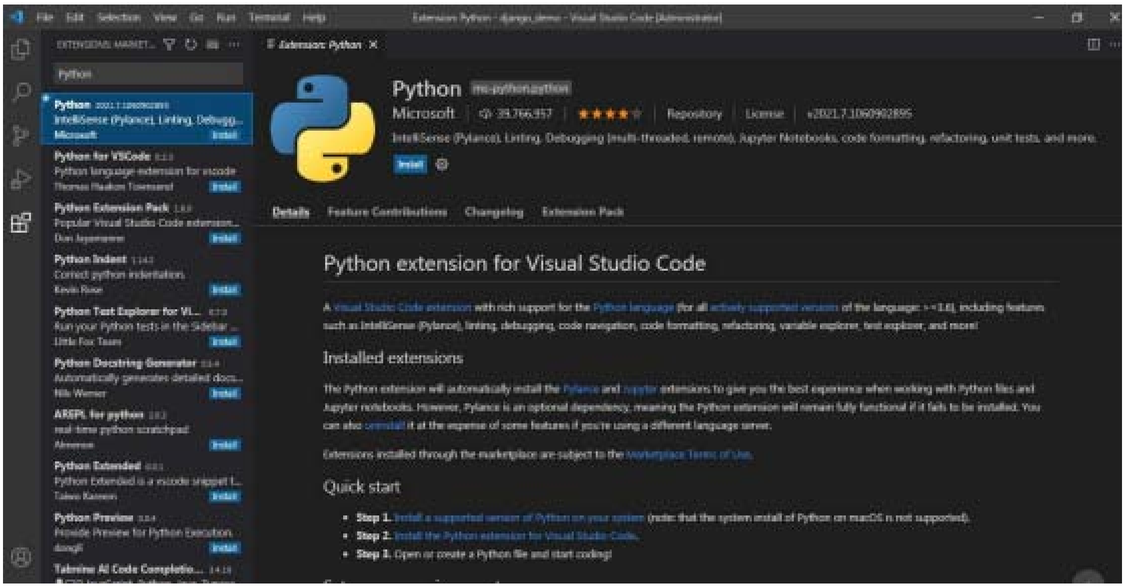
3. Installing Editor (pycharm or visual studio code can be used. It is recommended to use visual studio code i.e vscode)

Go to the link: https://code.visualstudio.com/ and install visual studio code in your system.

After successful installation of vscode, open it and go to the project folder

Install extentions named:

1) Python



IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and more.

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.6), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Installed extensions

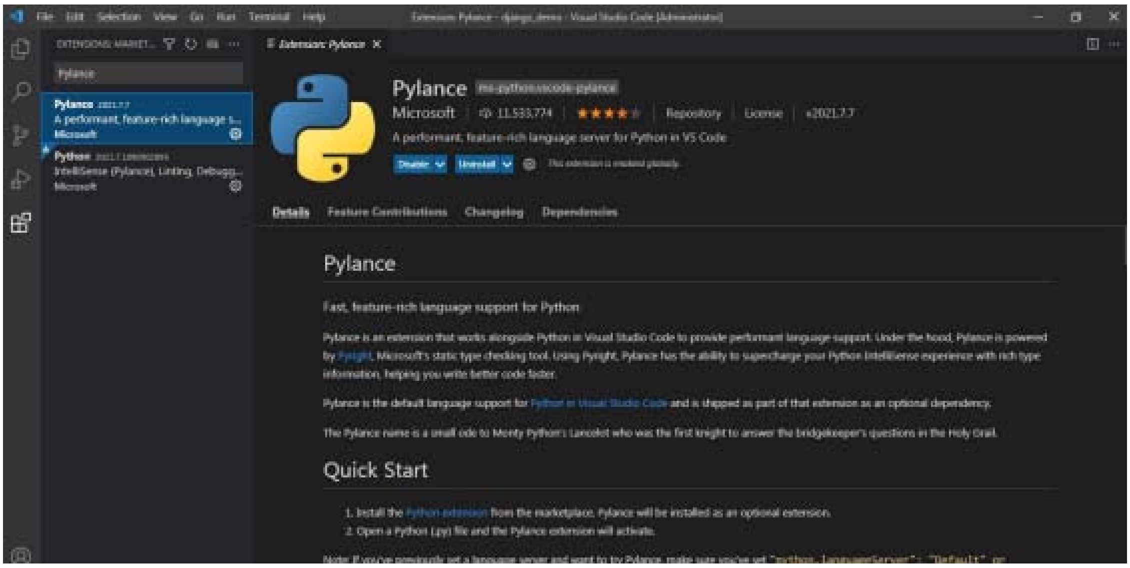
The Python extension will automatically install the Pylance and Jupyter extensions to give you the best experience when working with Python files and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also uninstall it at the expense of some features if you're using a different language server.

Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

Quick start

- Step 1. Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- Step 2. Install the Python extension for Visual Studio Code.
- Step 3. Open or create a Python file and start coding!

2) Pylance:



File Edit Selection View Go Run Terminal Help
Extension: Pylance - django_demo - Visual Studio Code [Administrator]
EXTENSIONS: MARKET...
Pylance
Pylance 2021.7.7
A performant, feature-rich language s...
Microsoft
Python 2021.7.1060902895
IntelliSense (Pylance), Linting, Debugg...
Microsoft
Extension: Pylance
Pylance ms-python.vscode-pylance
Microsoft | 11,533,774 | ★★★★☆ | Repository | License | v2021.7.7
Disable
Uninstall
This extension is enabled globally.
Details Feature Contributions Changelog Dependencies

A performant, feature-rich language server for Python in VS Code

Pylance

Fast, feature-rich language support for Python

Pylance is an extension that works alongside Python in Visual Studio Code to provide performant language support. Under the hood, Pylance is powered by Pyright, Microsoft's static type checking tool. Using Pyright, Pylance has the ability to supercharge your Python IntelliSense experience with rich type information, helping you write better code faster.

Pylance is the default language support for Python in Visual Studio Code and is shipped as part of that extension as an optional dependency.

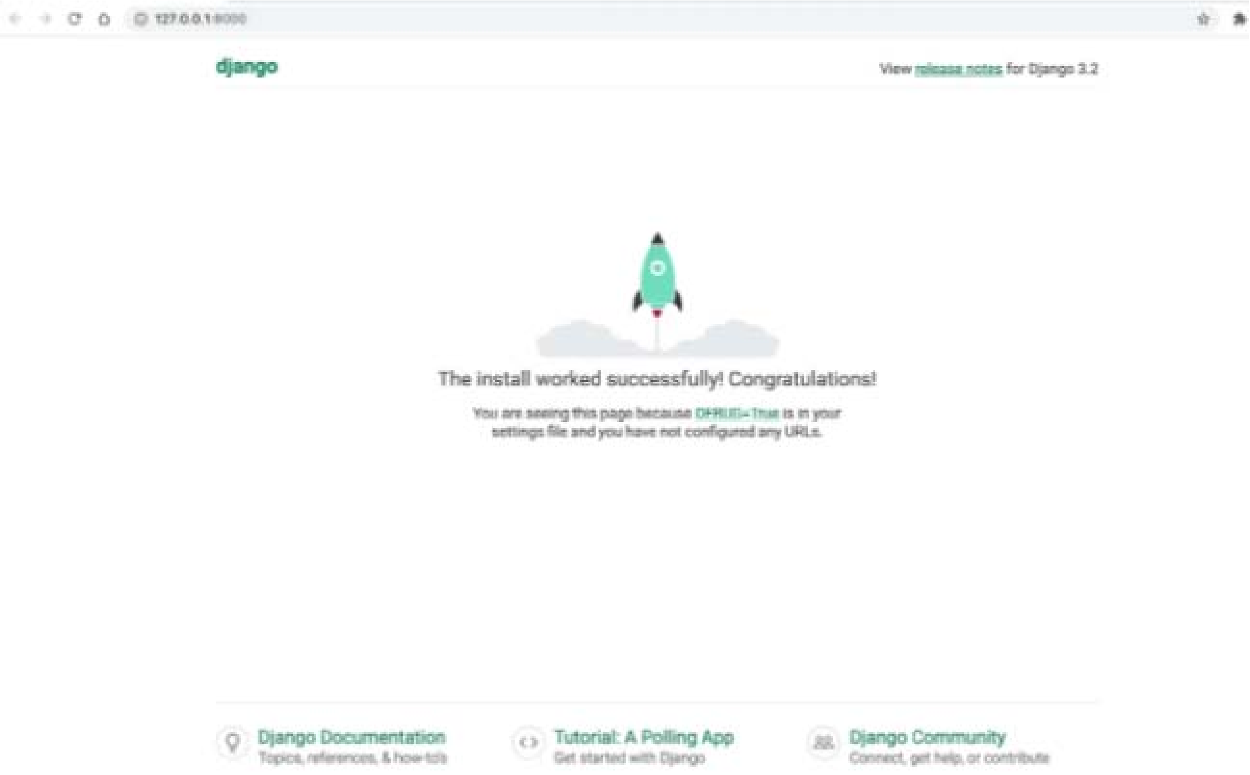
The Pylance name is a small ode to Monty Python's Lancelot who was the first knight to answer the bridgekeeper's questions in the Holy Grail.

Quick Start

1. Install the Python extension from the marketplace. Pylance will be installed as an optional extension.
2. Open a Python (.py) file and the Pylance extension will activate.

Note: If you've previously set a language server and want to try Pylance, make sure you've set "python.languageServer": "Default" or

In terminal run python manage.py runserver from created project. That will run your server.



The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Basic python programming:

RollNo:21BCE528

Aim:a.Developapythonprogramandmakethesimplecalculatorwhichusingandconditional loop.

b. WriteafunctionareaTrianglethattakesthelengthsofthreesidesofthetriangleasinput parametersandreturnstheareaofthetriangleasaoutput.Also,assertthatsumofthelength ofanytwosidesisgreaterthanthethirdside.Writeamainfunctionthatacceptsascommand lineargumentsandcomputerstheareaoftriangleusingthefunctionareaTriangle.

c. WriteafunctionthattakestwonumbersasinputparametersandreturnsTrueorFalse dependingonwhethertheyareco-primes.Twonumbersaresaidtobeco-primeiftheydonot have any common divisor other than one.

d. Writeafunctionthattakesastringasaparameterandreturnsastringwitheverysuccessive repetitive character replaced with a star(_). For Example, ‘balloon’ is returned as ‘balo_*n’.

e. Writeafunctionthattakesanumberasninputparameterandreturnsthecorrespondtextin words,forexample,oninput452,thefunctionshouldreturn‘FourFiveTwo’.Useadictionaryfor mappingtodigitstotheirstringrepresentation.

f. Writearecursivefunctionthattakesnumbernasaninputparameterandprintn-digitstrictly increasing number.

UserInputsrelatedPrograms

a. ThebellshapedGaussianfunction,

f(x)=1/(s*√(2π)) exp 〖[(-1)/2 〖((x-m)/s)〗 ^2]〗

isoneofthemostwidelyusedfunctionsinscienceandtechnology.Theparametersmands>0areprescribedrealnumbers.Makeaprogramforevaluatingthisfunctionfordifferentvalues ofs,xandm.Askusertoinputthevalues.

b. Acardriver,drivingatvelocityv0,suddenlyputsonthebrake.Whatbrakingdistancedis neededtostopthecar?Onecanderive,usingNewton’ssecondlawofmotionora corresponding energy equation, that:

d=1/2(v_0^2)/µg

Makeaprogramforcomputingdaboveequation,whentheinitialcarvelocityv0andthefriction coefficientµaregivenonthecommandline.Runtheprogramfortwocases:v0=120andv0= 50 km/h, both with µ = 0.3 (µ is dimensionless).

(Note:convertthevelocityinm/s)

PythonClassesrelated

a. DefineaclassBankthatkeepstrackofbankcustomers.Theclassshouldcontainthe following data member:

Datamembername

Details

name

Nameofcustomer

accountNum

Account Number

type

AccountType

amount

Amountdepositedinthebankaccount

interest

Interestearnedbythecustomer

Theclassshouldsupportthefollowingmethods:

(a) **init**forinitializingthedatamembers.

(b) depositfordepositingmoneyinthemembers.

(c) withdrawalforwithdrawingmoneyfromtheaccount

(d) findInterestthatdeterminestheinterestonthebasisofamountintheaccount

Programs:

```
num1=int(input("EnterNumber1:"))
num2=int(input("EnterNumber2:"))  op
= input("Enter operator")
if op=='+':
 print(num1+num2)
elif op=='-':
 print(num1-num2)
elif op=='*':
 print(num1*num2)
elif op=='/':
 print(num1/num2)
else:
 print("Please Enter correct Operator")
```

```
     Please Enter correct Operator
```

```
#AreaofTraingle
def Area(a,b,c):
 if(a+b>canda+c>bandb+c>a): s =
    (a+b+c)/2
    area = round((s*(s-a)*(s-b)*(s-c))**0.5,5)

    print("Area of Traingle:"+ str(area))
 else:
    print("Cannot from Triangle")

a = int(input("Enter 1st side:"))
b = int(input("Enter 2st side:"))
c=int(input("Enter3stside:"))
Area(a,b,c)
```

```
     Area of Traingle:6.0
```

```
#coprimes
n1=int(input("Enter1stnumber:"))
n2=int(input("Enter2ndnumber:")) def
divisor(num):
 list=[]
 foriinrange(2,int(num-1)): if
    num%i==0:
        list.append(i)
 return list
l1=divisor(n1)
l2=divisor(n2)
print(l1)
print(l2)
same=False
for i in l1:
    for j in l2:
        if i==j:
            same=True
if(same == False):
    print("Givennumbersarecoprime")
else:
    print("Given numbers are not coprime")
```

```
     [2,5]
     []
     Given numbers are coprime
```

```
s=input("EnterString")
list =[]
s1 = ""
for i in s:
    ifinotinlist: s1+=i
    elifiinlist: s1+='*'
    list.append(i)
print(s1)
```

```
     bal*o*n
```

```
defnumber_to_words(number):
    digits = {
 '0': 'zero',
 '1': 'one',
 '2': 'two',
 '3': 'three',
 '4': 'four',
 '5': 'five',
 '6': 'six',
 '7': 'seven',
 '8': 'eight',
 '9': 'nine'
 }

    word = ''
    fordigitinstr(number): if
        digit in digits:
            word+=digits[digit]+''
    return word.strip()
number = 718
word=number_to_words(number)
print(word)
```

```
     seven one eight
```

```
speed=int(input("enterspeedinkm/h:")) speed1 =
round(speed * 5/18,4)
friction = 0.3
g = 9.8
distance=round((1/2)*((speed1**2)/(friction*g)),3) print(distance)
```

```
     1181.027
```

```
s = int(input("Enter s:"))
x = int(input("Enter x:"))
m=int(input("Enterm:")) if
s>0:
 f=round(((1/(s*((2*3.14)**0.5)))**((-1/2)*((x-m)/s)**2)),3)
print(f)
```

```
     1.119
```

```
def distance(v0, mu):
    v0_mps=v0*1000/3600 g =
    9.81
    d=0.5*v0_mps**2/(mu*g) return d
v0_1 = 120
mu_1 = 0.3
d_1 = distance(v0_1, mu_1)
print(f"Brakingdistance(d)forv0={v0_1}km/handmu={mu_1}is{d_1:.2f}meters.") v0_2 = 50
```

```
mu_2 = 0.3
d_2 = distance(v0_2, mu_2)
print(f"Braking distance (d) for v0 = {v0_2} km/h and mu = {mu_2} is{d_2:.2f}
meters.")
```

```
     Brakingdistance(d)forv0=120km/handmu=0.3is188.77meters. Braking
     distance (d) for v0 = 50 km/h and mu = 0.3 is32.77 meters.
```

```
class Bank:
    definit(self,name,accountNum,accountType,amount):
        self.name = name
        self.accountNum =
        accountNum
        self.accountType=acco
        untType self.amount =
        amount self.interest
        = 0
    defdeposit(self,amount):
        self.amount += amount
        print(f"Deposited{amount}intheaccount.Currentbalance:{self.amount}
    ") def withdrawal(self, amount):
        ifself.amount>=
            amount:
            self.amount
            -= amount
            print(f"Withdrawn{amount}fromtheaccount.Currentbalance:{self.amount}
        ") else:
            print("Insufficientbalance.Withdrawalnotallo
    wed.") def findInterest(self):
        if self.amount >= 500000:
            self.interest=self.amoun
            t*0.08
        elif 300000 <= self.amount <
            500000:
            self.interest=self.amount
            *0.07
        elif 100000 <= self.amount <
            300000:
            self.interest=self.amount
            *0.05
        else:
            self.interest = self.amount * 0.03
        print(f"Interestearned:{self.interest:.2f}"
        )
```

```
customer1 = Bank("abc", "1111111111", "Savings", 400000)
customer2=Bank("xyz","2222222222","Current",800000)
customer1.deposit(50000)
customer1.findInterest()
customer2.withdrawal(100000)
customer2.findInterest()
```

```
Deposited50000intheaccount.Currentbalance:450000
Interest earned: 31500.00
Withdrawn100000fromtheaccount.Currentbalance:700000
Interest earned: 56000.00
```