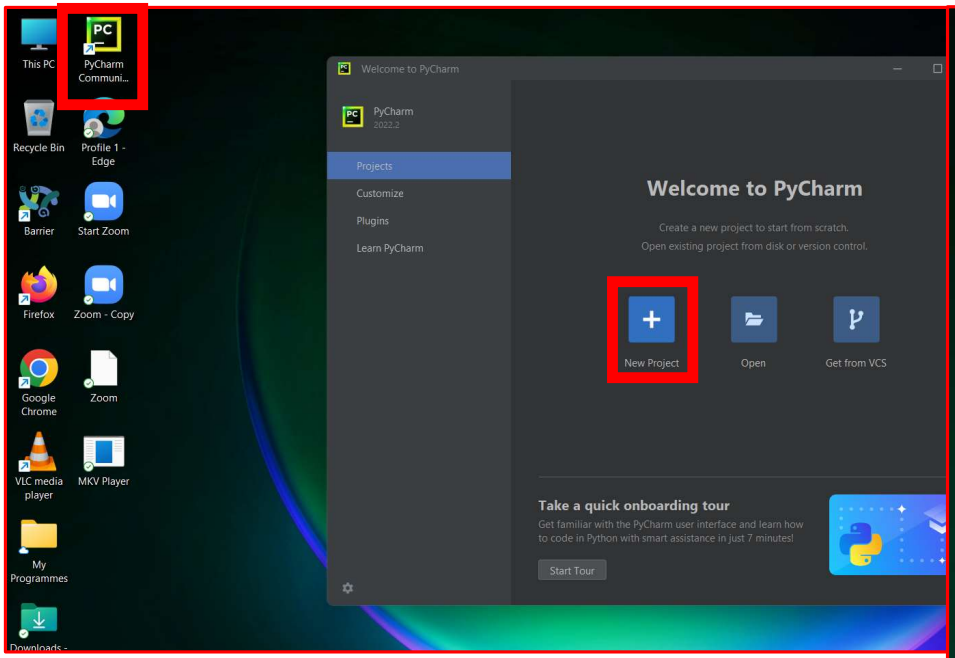
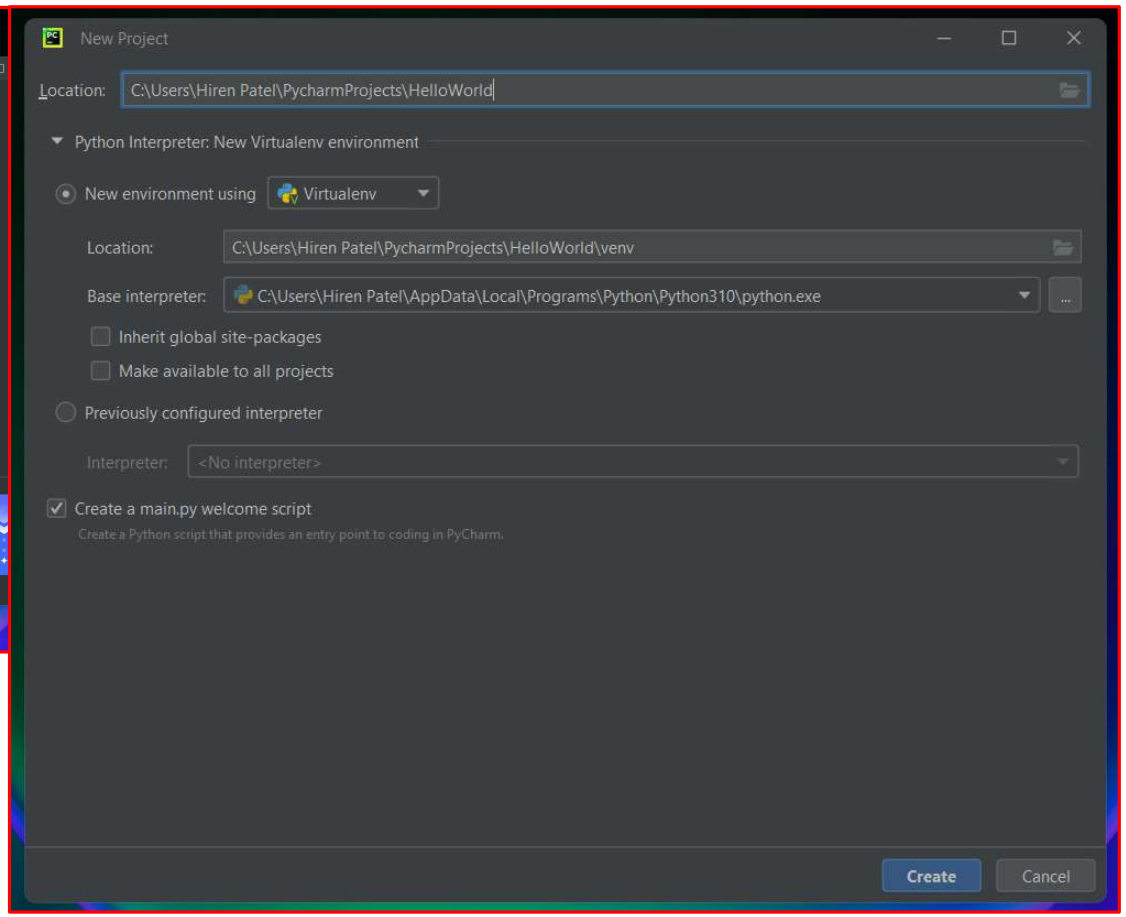
Getting familiar with PyCharm

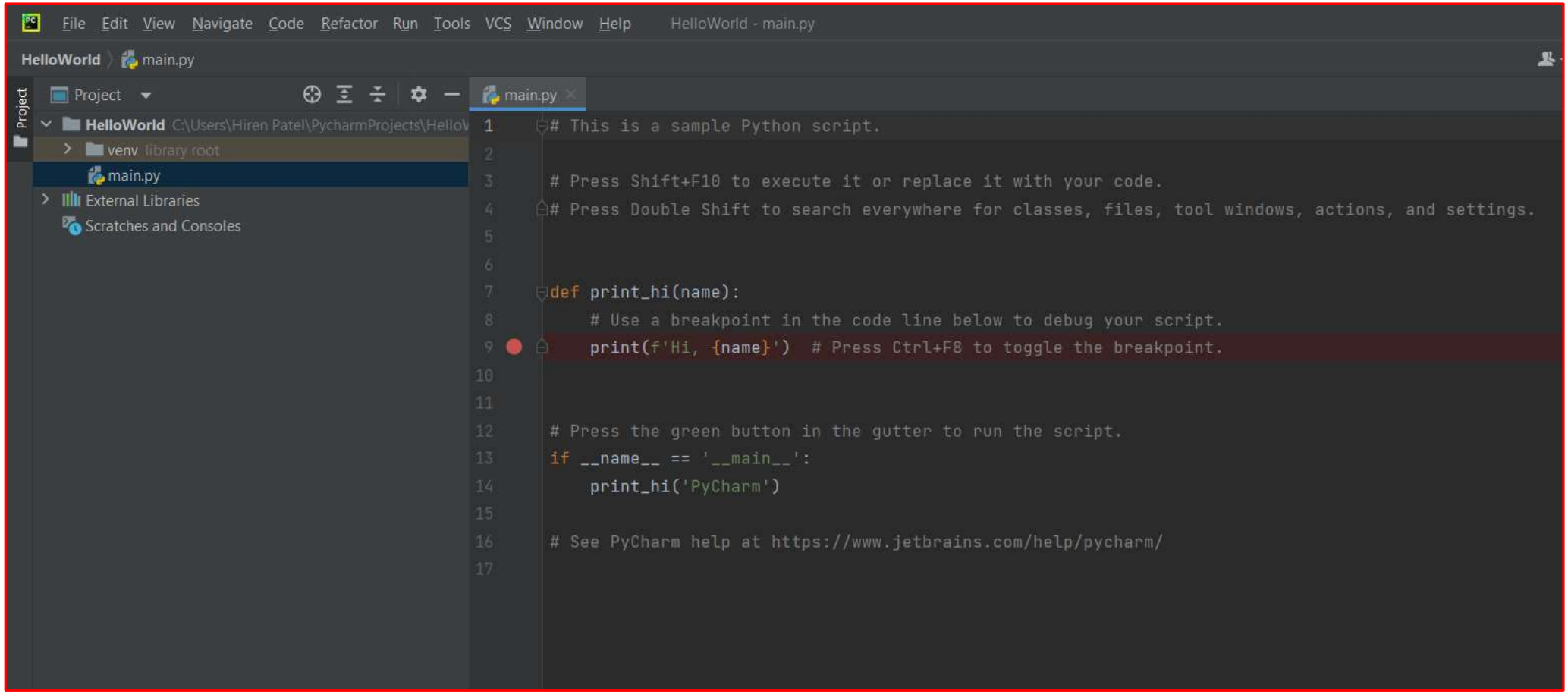
www.hbpatel.in





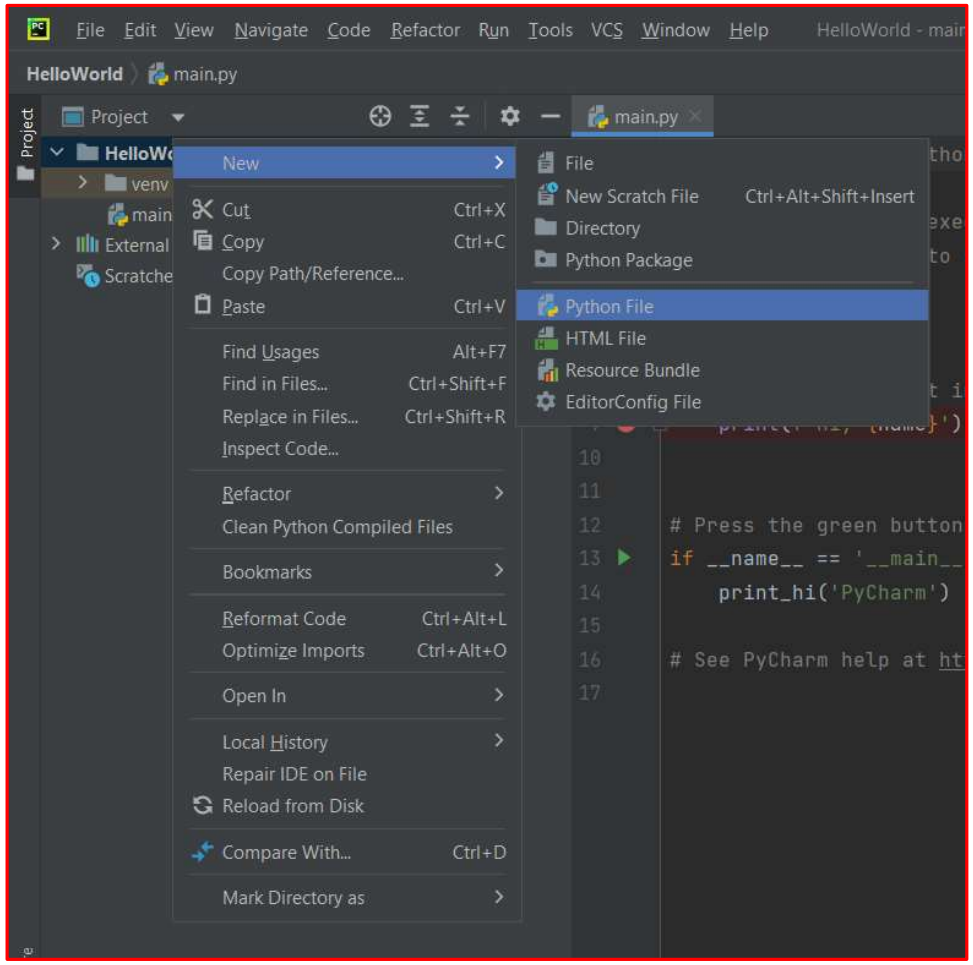
Getting familiar with PyCharm

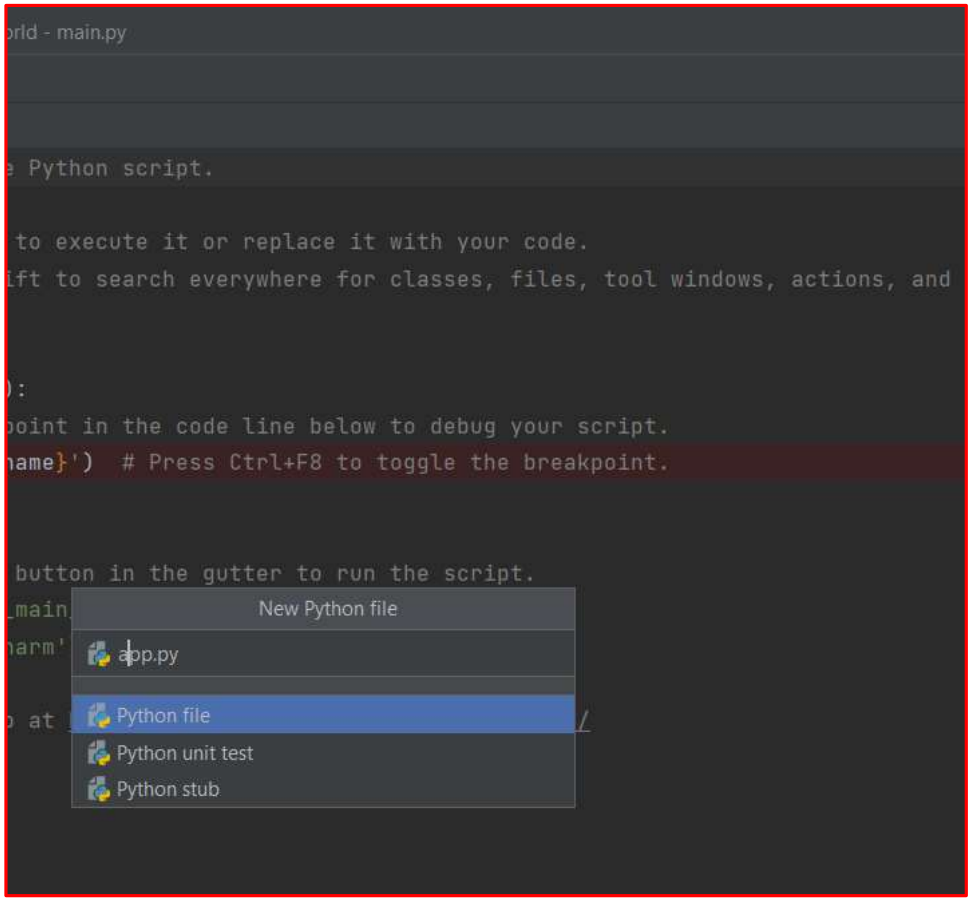
www.hbpatel.in



Getting familiar with PyCharm

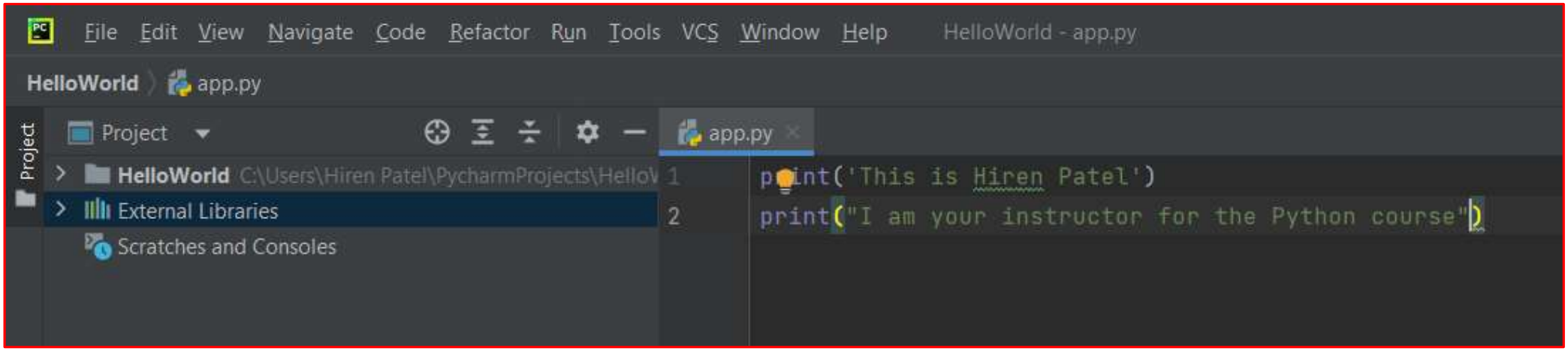
www.hbpatel.in

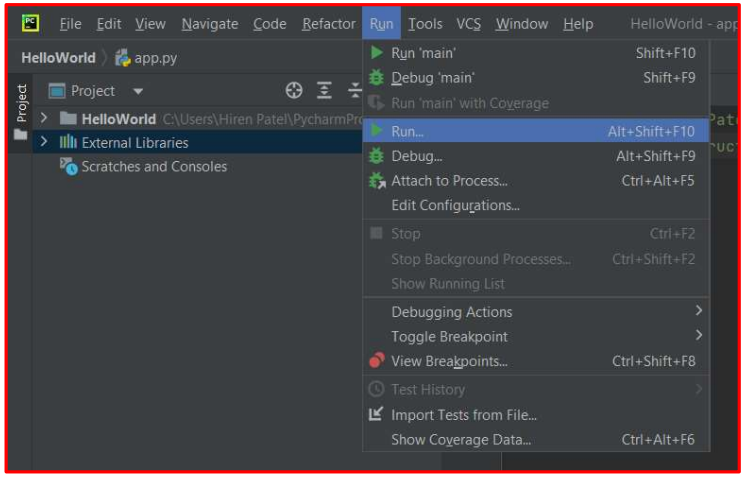


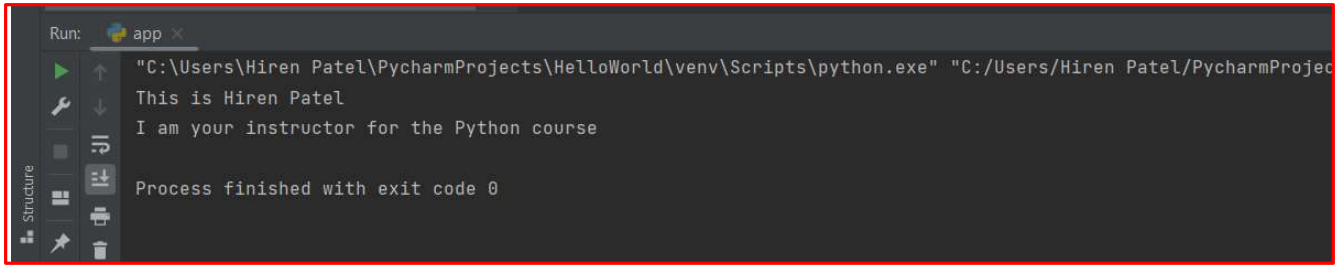


Writing small programs in PyCharm

www.hbpatel.in

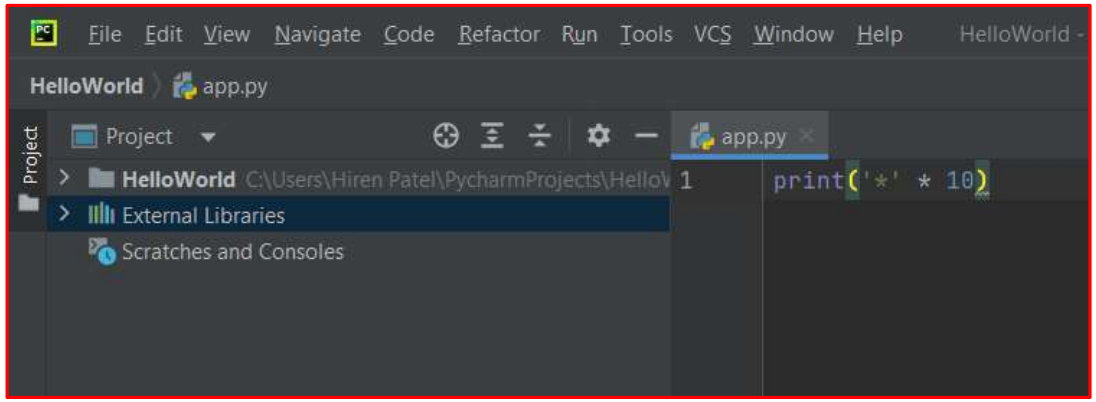


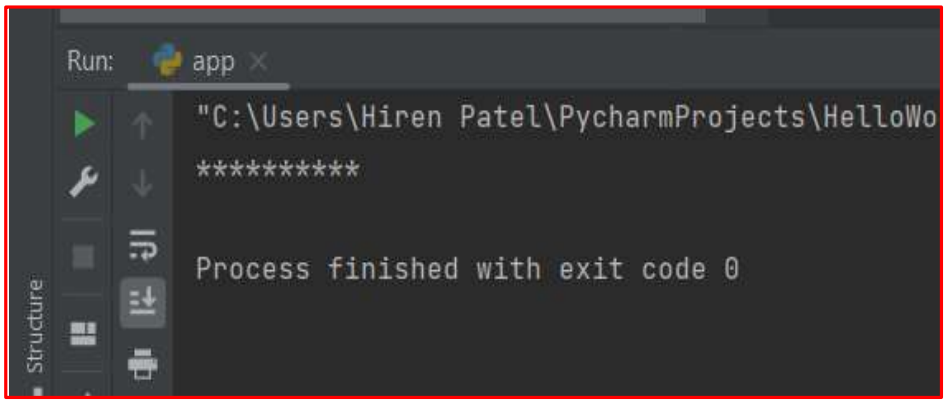


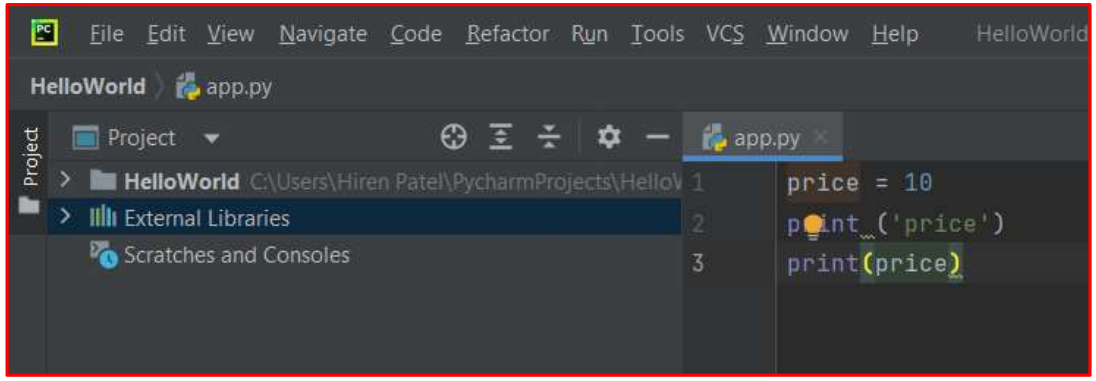


Writing small programs in PyCharm

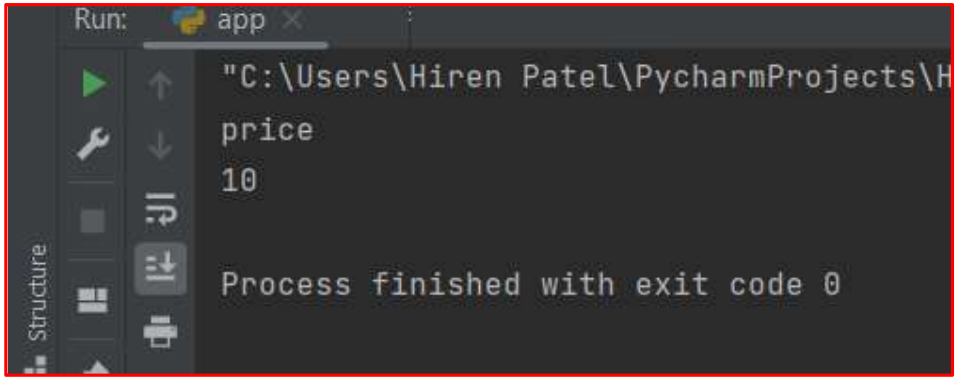
www.hbpatel.in





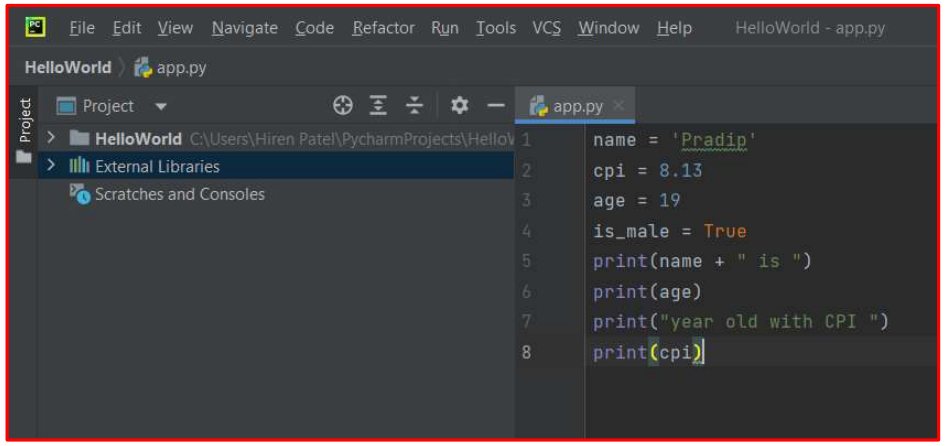


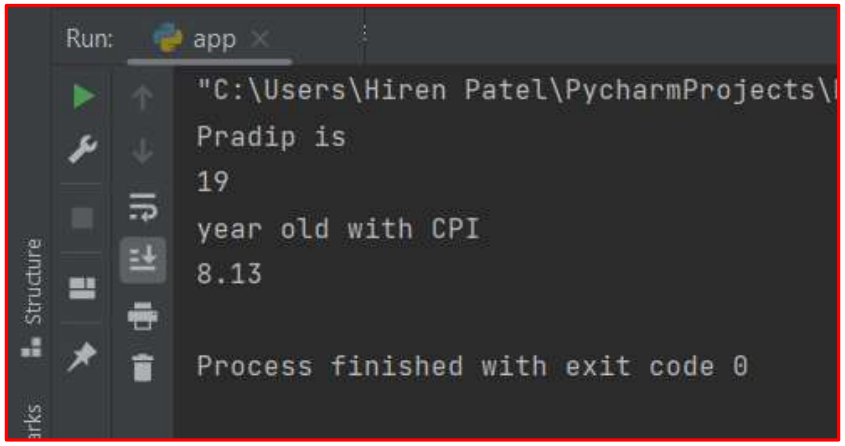
Variables

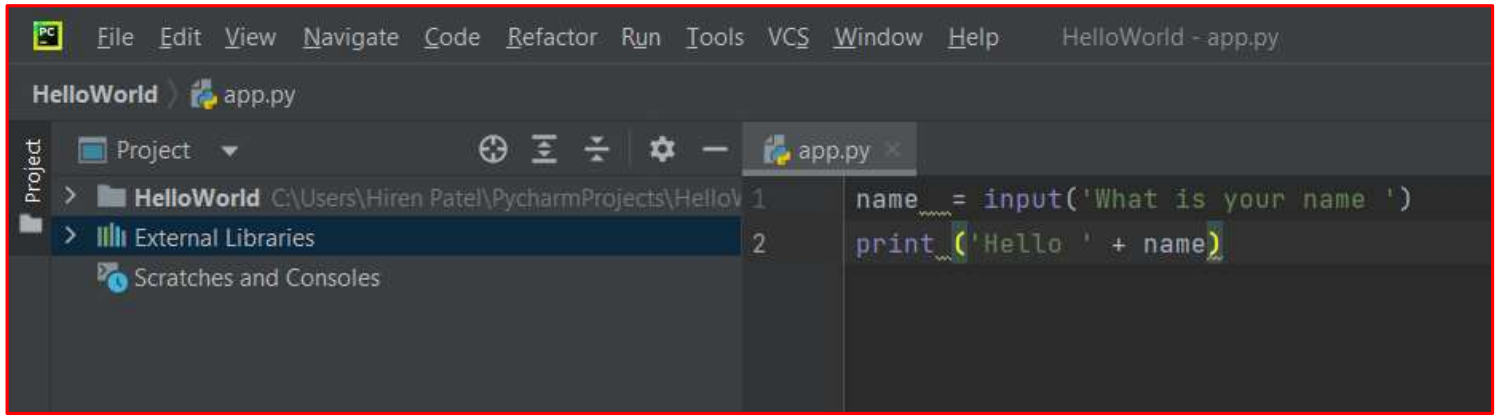


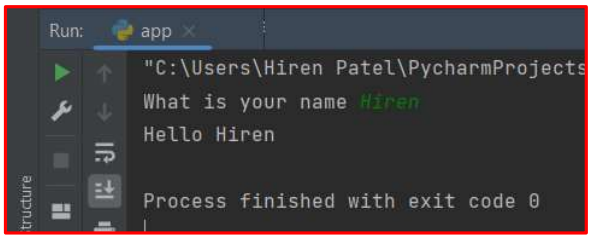
Variables

www.hbpatel.in









Getting input from the keyboard

Input from Keyboard and Type Casting

www.hbpatel.in

```python
name = input('What is your name ')
food = input('What is your favorite food ')
print(name + ' loves ' + food)
```

```
"C:\Users\Hiren Patel\PycharmProjects\Hell
What is your name Hiren
What is your favorite food Gulab Jamun
Hiren loves Gulab Jamun

Process finished with exit code 0
```

Getting input from the keyboard

```python
birth_year = input('Enter your birth year ')
age = 2022 - birth_year
print(age)
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Scripts\python.exe" "C:/Users/Hiren
Enter your birth year 1976
Traceback (most recent call last):
  File "C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py", line 2, in <module>
    age = 2022 - birth_year
TypeError: unsupported operand type(s) for -: 'int' and 'str'

Process finished with exit code 1
```

Type casting / conversion

```python
birth_year = input('Enter your birth year ')
age = 2022 - int(birth_year)
print(age)
```

```
"C:\Users\Hiren Patel\PycharmProjec
Enter your birth year 1976
<class 'str'>
<class 'int'>
46

Process finished with exit code 0
```

Type Casting and Strings

www.hbpatel.in

app.py

```
1  weight_pound = int(input('Enter weight in lbs : '))
2  weight_kg = weight_pound * 0.45
3  print(weight_kg)
```

app

```
"C:\Users\Hiren Patel\PycharmProje
Enter weight in lbs : 100
45.0

Process finished with exit code 0
```

app.py

```
1  subject1 = 'Python Programming'
2  subject2 = "Python Programming"
3  subject3 = "Python's Programming by Hiren Patel"
4  subject4 = '"Python" Programming by Hiren Patel'
5  subject5='''This is a multiline message.
6  Multiple lines are enclosed in three quotes.
7  The three quotes could be single or double.'''
8  print (subject1)
9  print (subject2)
10 print (subject3)
11 print (subject4)
12 print (subject5)
```

app

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWor
Python Programming
Python Programming
Python's Programming by Hiren Patel
"Python" Programming by Hiren Patel
 This is a multiline email.
Multiple lines are enclosed in three quotes.
The three quotes could be single or double.


Process finished with exit code 0
```

Playing with strings

app.py

```
1  first_name = 'Hiren'
2  last_name = 'Patel'
3  msg1 = first_name + ' [' + last_name + '] is an instructor'
4  #formatted string
5  msg2 = f'{first_name} [{last_name}] is an instructor'
6  print(msg1)
7  print(msg2)
```

app

```
"C:\Users\Hiren Patel\PycharmProjec
Hiren [Patel] is an instructor
Hiren [Patel] is an instructor

Process finished with exit code 0
```

app.py

```
1  subject = 'Python programming'
2  print(len(subject))
3  print(subject.upper())
4  print(subject.lower())
5  print(subject.title())
6  print(subject[0].islower())
7  print(subject[0].isupper())
8  print(subject.find('P'))
9  print(subject.find('o'))
10 print(subject.find('p'))
11 print(subject.find('program'))
12 print(subject.replace('programming','coding'))
13 print('program' in subject)
```

app

```
18
PYTHON PROGRAMMING
python programming
Python Programming
False
True
0
4
7
7
Python coding
True
```

PROGRAM

```
s = "hello"
print(s.capitalize())
print(s.upper())
print(s.rjust(7))
print(s.center(7))
print(s.replace('l', '(ell)'))
print('  world '.strip())
```

OUTPUT

```
Hello
HELLO
  hello
 hello
he(ell)(ell)o
world
```

Arithmetic Operators

www.hbpatel.in

app.py

```
1  print (10 + 3)
2  print (10 - 3)
3  print (10 * 3)
4  print(10 / 3)
5  print (10 // 3)
6  print (10 % 3)
7  print (10 ** 3)
```

app

```
"C:\Users\Hiren Patel\
13
7
30
3.3333333333333335
3
1
1000
```

app.py

```
1  x = 10
2  x = x+3
3  print(x)
4  x += 3
5  print(x)
6  x -= 3
7  print(x)
8  x *= 3
9  print(x)
10 x /= 3
11 print(x)
```

app

```
"C:\Users\
13
16
13
39
13.0
```

Arithmetic Operations
(Precedence and In-built functions)

www.hbpatel.in

app.py

```
1  x = 10 + 3 * 2 ** 4
2  print(x)
3  x = (10 + 3) * 2 ** 4
4  print(x)
```

app

```
"C:\Users\Hi
58
208
```

app.py

```
1  x = 3.6
2  print(x)
3  print(round(x))
4  x = -3.6
5  print(abs(x))
```

app

```
"C:\Users\Hi
3.6
4
3.6
```

Arithmetic Operations
(In-built mathematical functions)


```
import math
x = 3.6
print(x)
print(math.ceil(x))
print(math.floor(x))
print(math.factorial(5))
print(math.isfinite(100))
print(math.sqrt(100))
print(math.lcm(50, 40))
print(math.gcd(50, 40))
print(math.exp(10))
print(math.log(100,2))
```

app

```
"C:\Users\Hiren Patel
3.6
4
3
120
True
10.0
200
10
22026.465794806718
6.643856189774725
```

Bitwise Operators

OPERATOR	DESCRIPTION	SYNTAX
&	Bitwise AND	x & y
|	Bitwise OR	x | y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x>>
<<	Bitwise left shift	x<<

PROGRAM

```
a = 10
b = 4

# Print bitwise AND operation
print("a & b =", a & b)

# Print bitwise OR operation
print("a | b =", a | b)

# Print bitwise NOT operation
print("~a =", ~a)

# print bitwise XOR operation
print("a ^ b =", a ^ b)
```

OUTPUT

```
a & b = 0
a | b = 14
~a = -11
a ^ b = 14
==========
```

Explanation

```
a      = 1010 (10)
b      = 0100 (04)
a & b = 0000 (00)
a | b = 1110 (14)
~a     = 0101 (-11, 2's complement)
a ^ b = 1110 (14)
```

Bitwise Operators

OPERATOR	DESCRIPTION	SYNTAX
&	Bitwise AND	x & y
|	Bitwise OR	x | y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x>>
<<	Bitwise left shift	x<<

PROGRAM

```
a = 10


# print bitwise right shift
operator
print("a >> 1 =", a >> 1)


a = 9


# print bitwise left shift
operator
print("a << 1 =", a << 1)
```

OUTPUT

```
a >> 1 = 5
a << 1 = 18
======================
```

Explanation

```
a = 00001010 (10)
a >> 1 = 00000101 (5)


a = 00001001(5)
a << 1 = 00010010 (18)
```