Class 3 Notes

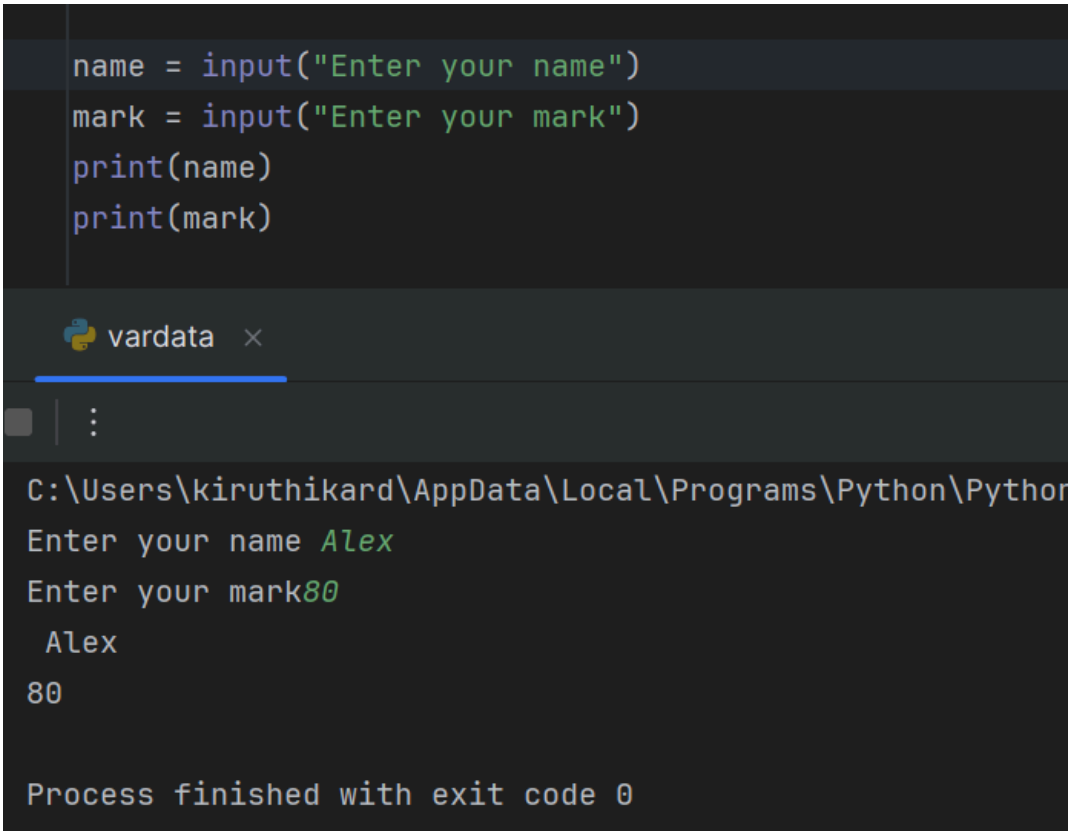
Getting user input

input()

It asks user for a input and print it in the console

Syntax: input(“your string”)

Eg: input(“Enter name”)
input(“Enter mark”)



F-Strings Dynamic variable inside a string

You can insert a variable into a string using f-strings.

Syntax: {variable name}

Eg: name = “Arun”
print(f“There are two students with same name {name}”)

```
name = "Arun"
print(f"There are two students with same name {name}")
```

vardata

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\Python313\python.exe C:\
There are two students with same name Arun

Process finished with exit code 0
```

Arithmetic Operators

```
a = 7
b = 2
print( a + b)
print( a - b)
print( a * b)
print( a / b)
print( a // b)  #only the quotient in division or only the integer part
print( a % b)   # modulo operator. it will give you the remainder part of division
print( a ** b)  # exponential operator
```

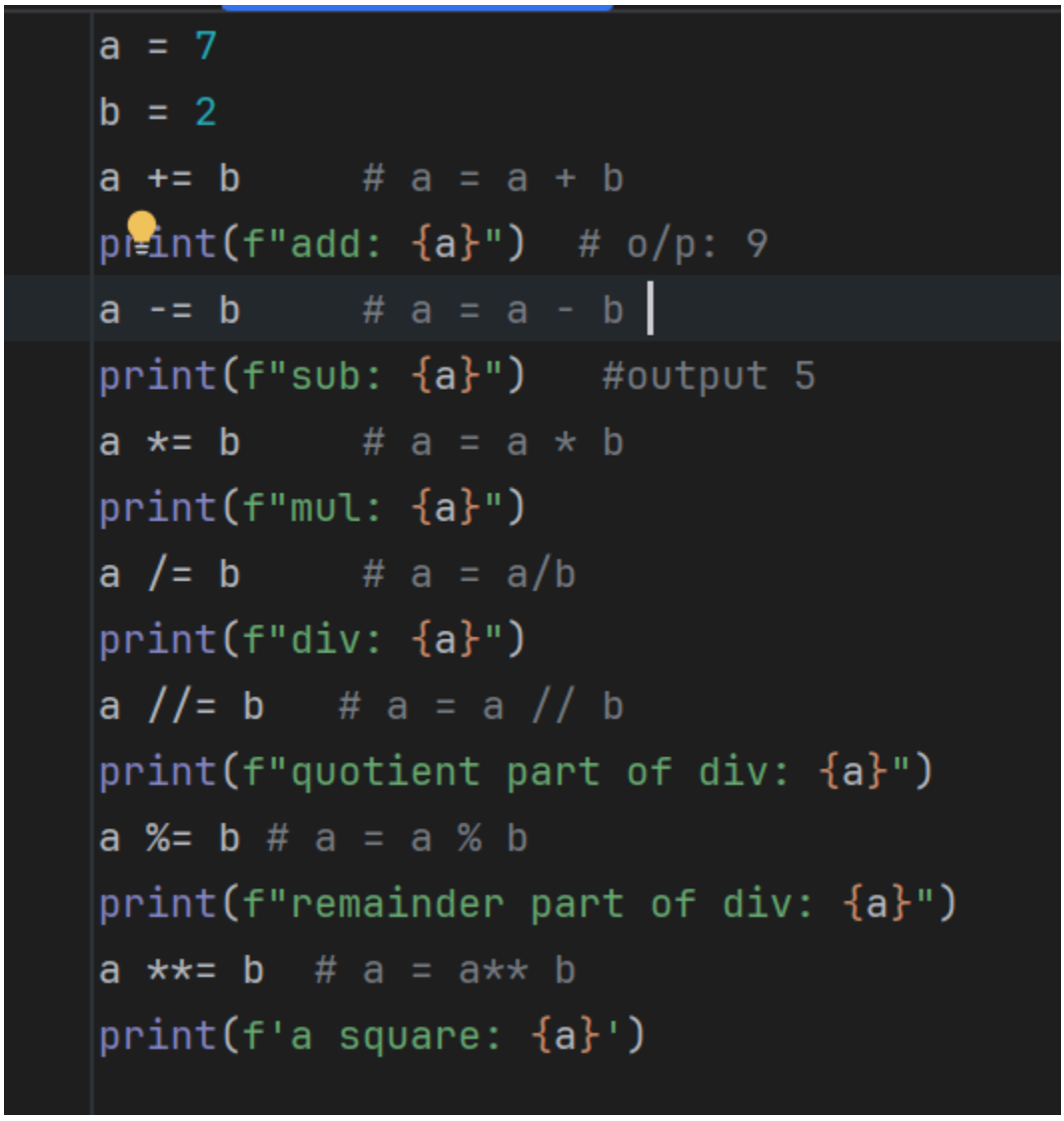
vardata

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\Python313\python.exe C:\Users\kiruthikar
9
5
14
3.5
3
1
49

Process finished with exit code 0
```

Operator & Assignment operator in single line

variable += value
Variable = variable + value



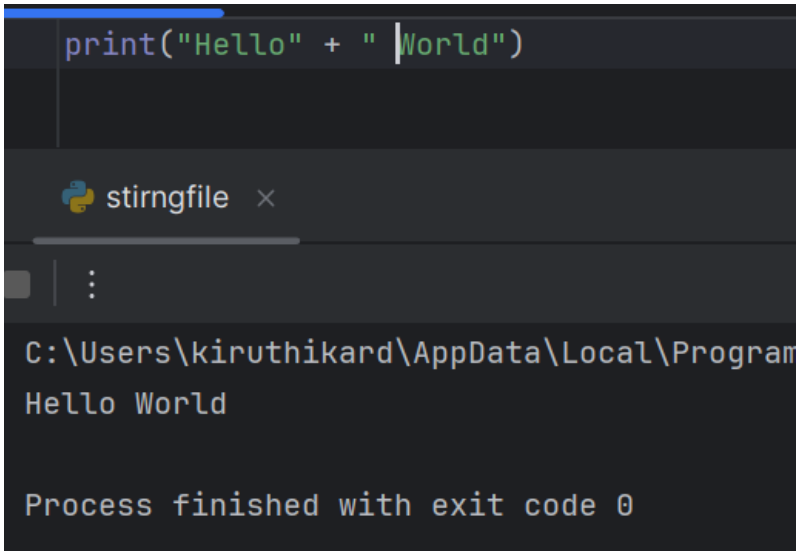
```
a = 7
b = 2
a += b      # a = a + b
print(f"add: {a}")  # o/p: 9
a -= b      # a = a - b
print(f"sub: {a}")   #output 5
a *= b      # a = a * b
print(f"mul: {a}")
a /= b      # a = a/b
print(f"div: {a}")
a //= b   # a = a // b
print(f"quotient part of div: {a}")
a %= b # a = a % b
print(f"remainder part of div: {a}")
a **= b  # a = a** b
print(f'a square: {a}')
```

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\P
add: 9
sub: 7
mul: 14
div: 7.0
quotient part of div: 3.0
remainder part of div: 1.0
a square: 1.0

Process finished with exit code 0
```

String Concatenation

We can add string to string to create a new string. This is called concatenation.



Logical Operator

```
#Logical operators
a = True
b = False
print( a and b)  #logical and
print( a or b)   #logical or
print(not a)     #logical not
```

o/p: false, true, false

Bitwise Operator

```
#Bitwise operator
a = 4  #0100
b = 5  #0101
print( a & b)
print( a | b)
print( a!=b)
```

Output: 4,5,true