**Pseudocode**

Writing pseudocode is a useful way to think about how you will approach a problem before actually beginning to code. Computer scientists in the real world often use it to describe algorithms and how different pieces of code interact with each other.

1. Write pseudocode for a function that takes the current month and day as parameters and calculates & returns the absolute day of the year. For example, January 2 is the second day of the year, April 1 is day 91, and August 6 is day 218. You don’t need to account for leap years (so assume February has 28 days).

Step 1: Store the number of days in each month

Step 2: input month and day

Step 3: sum all days until that month

Step 4: add the number of days to current sum

Step 5: return absolute day of the yeara

1. Write pseudocode for a program that asks the user for their birthday (month, day, and year) and the current date (month, day, and year) and then tells the user how old they are.

**Functions & Returns**

What is the output of the following programs? When you read this code, pay attention to returns!

|  |  |
| --- | --- |
| 1 def secrets(name):  2 print(name + " who has 7 cats")  3  4 def main():  5 name = "Gary"  7 secrets("Ash")  8 print(" is jealous of ")  10 secrets(name)  11  12 main() | Output:  Gary who has 7 cats is jealous of Ash |
| 1 def message(name):  2 s = "Detective " + name  3 return s  4  5 def main():  6 name = "Pikachu"  7 output = message(name)  8 print(name)  9 print("vs.")  10 print(output)  11  12 main() | Output:  Pikachu  vs.  Detective Pikachu |

**Functions**

1. Write a function, dollars\_to\_mexican\_pesos, that takes **no** parameters, prompts the user for an amount in dollars (like 1.50), then **prints out** the equivalent amount in Mexican pesos. Assume the exchange rate to be 1 US dollar = 19.88 Mexican pesos. Call your function.

def dollars\_to\_mexican\_pesos():

dollars = input(“Give an amount of dollars”)

pesos = 19.88 \* dollars

print(pesos)

dollars\_to\_mexican\_pesos()

1. Change your function dollars\_to\_mexican\_pesos, so that instead of prompting the user for the amount in US dollars, the function takes one parameter, amount, whose value will be the number in USD to convert. Call your function.

def dollars\_to\_mexican\_pesos(amount):

pesos = 19.88 \* amount

print(pesos)

dollars\_to\_mexican\_pesos(5)

1. Change your function dollars\_to\_mexican\_pesos, so that instead of **printing** the value of the converted amount, it **returns** the value instead. Call your function. (Be sure to assign the return value to a variable.)

def dollars\_to\_mexican\_pesos(amount):

pesos = 19.88 \* amount

return pesos

1. Change your function so that it takes a conversion\_rate parameter in addition to amount so that it can be used to convert between any two currencies. Why would adding this parameter change this function to convert between any two currencies and not just US dollars and other currencies? Call your function and assign its return value to a variable. Look up some other conversion rates (for example converting the dollar to the krona, yuan, or pound) to call your function.

def convert(amount,conversion\_rate):

currency = conversion\_rate \* amount

return pesos

rupees = convert(60, 73.50) #60 rupees in dollars