

Nama : Bagus Ahfad Fadhilah

NIM : 20210801103

Teknik Informatika

UAS Bahasa Pemrograman

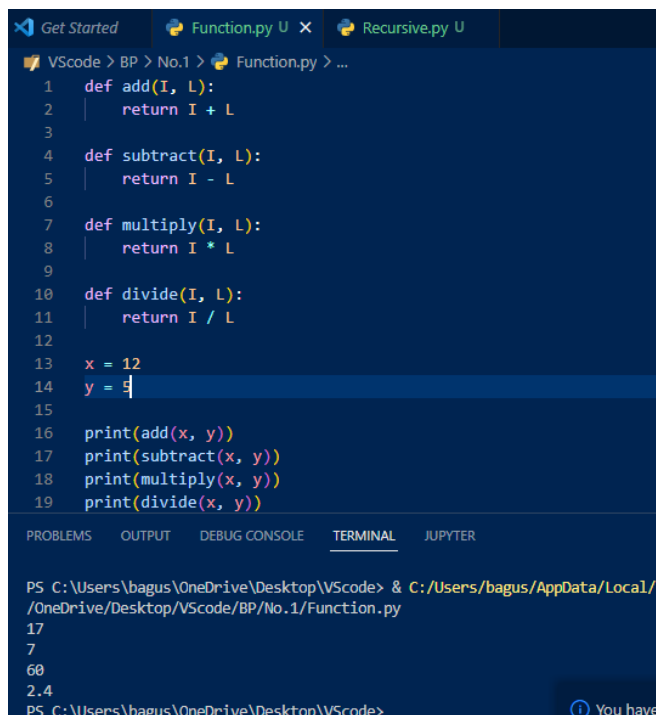
No.1

Function adalah bagian dari kode yang dapat digunakan kembali. Fungsi didefinisikan dengan menggunakan kata kunci "def", diikuti oleh nama fungsi dan tanda kurung. Fungsi dapat menerima argumen dan mengembalikan nilai.

Fungsi dapat digunakan untuk memisahkan kode yang terkait dengan tugas tertentu dari kode utama program Anda, sehingga membuat kode lebih mudah dibaca dan di maintain. Fungsi juga dapat digunakan untuk mengekstrak kode yang digunakan secara berkala dalam program Anda dan menyimpan kode tersebut dalam satu tempat untuk digunakan kembali. Hal ini akan membuat program Anda lebih efisien dan mudah diperbaiki.

Fungsi dapat digunakan untuk mengekstrak kode yang digunakan secara berkala dalam program Anda dan menyimpan kode tersebut dalam satu tempat untuk digunakan kembali. Hal ini akan membuat program Anda lebih efisien dan mudah diperbaiki.

Selain itu, Python juga memiliki built-in function seperti print, input, len, dll yang dapat digunakan untuk melakukan tugas tertentu tanpa harus mendefinisikan fungsi sendiri. Secara keseluruhan, fungsi adalah komponen penting dari pemrograman yang membantu dalam menyederhanakan kode dan meningkatkan efisiensi.



```
Get Started | Function.py U x | Recursive.py U
VSCode > BP > No.1 > Function.py > ...
1  def add(I, L):
2      return I + L
3
4  def subtract(I, L):
5      return I - L
6
7  def multiply(I, L):
8      return I * L
9
10 def divide(I, L):
11     return I / L
12
13 x = 12
14 y = 4
15
16 print(add(x, y))
17 print(subtract(x, y))
18 print(multiply(x, y))
19 print(divide(x, y))

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | JUPYTER
PS C:\Users\bagus\OneDrive\Desktop\VScode> & C:/Users/bagus/AppData/Local/
/OneDrive/Desktop/VScode/BP/No.1/Function.py
17
7
60
2.4
PS C:\Users\bagus\OneDrive\Desktop\VScode> You have
```

Recursive adalah konsep di mana sebuah fungsi mengacu pada dirinya sendiri dalam pemanggilannya. Fungsi rekursif akan terus memanggil dirinya sendiri sampai kondisi akhir dicapai, yang biasanya dikenal sebagai kondisi basis. Kondisi basis adalah kondisi yang menyebabkan rekursi untuk berhenti dan mengembalikan nilai yang sesuai.

Contohnya, dalam kasus mencari faktorial, kondisi basis adalah $n = 0$, di mana faktorial dari 0 adalah 1. Jika n tidak sama dengan 0, maka faktorial dari n adalah n dikalikan dengan faktorial dari $n-1$. Fungsi akan terus memanggil dirinya sendiri dengan argumen yang berkurang sampai $n = 0$, di mana kondisi basis tercapai dan nilai 1 dikembalikan.

Pendekatan rekursif sering digunakan dalam pemrograman untuk menyelesaikan masalah yang memiliki struktur atau pola yang sama. Contohnya dalam kasus traversal pada tree, backtracking dalam algoritma pencarian, dll. Namun perlu diingat bahwa pendekatan rekursif dapat menyebabkan overhead yang cukup besar jika digunakan secara tidak benar seperti dalam kasus pemanggilan rekursi yang tidak berhenti atau pemanggilan rekursi yang berlebihan.

Secara keseluruhan, pendekatan rekursif adalah cara yang efektif untuk menyelesaikan masalah tertentu dan dapat digunakan untuk menyederhanakan kode, namun perlu digunakan dengan hati-hati untuk menghindari overhead yang tidak perlu.

```
VSCode > BP > No.1 > Recursive.py > Node > __init__
1  class Node:
2      def __init__(self, data):
3          self.left = None
4          self.right = None
5          self.data = data
6  def count_leaf_nodes(node):
7      if node is None:
8          return 0
9      if node.left is None and node.right is None:
10         return 1
11     else:
12         return count_leaf_nodes(node.left) + count_leaf_nodes(node.right)
13 root = Node(1)
14 root.left = Node(2)
15 root.right = Node(3)
16 root.left.left = Node(4)
17
18 print(count_leaf_nodes(root))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

PS C:\Users\bagus\OneDrive\Desktop\VScode> & C:/Users/bagus/AppData/L
/OneDrive/Desktop/VScode/BP/No.1/Recursive.py
2
PS C:\Users\bagus\OneDrive\Desktop\VScode>
```