

## PROG 1: Pomocny material k rekurzii

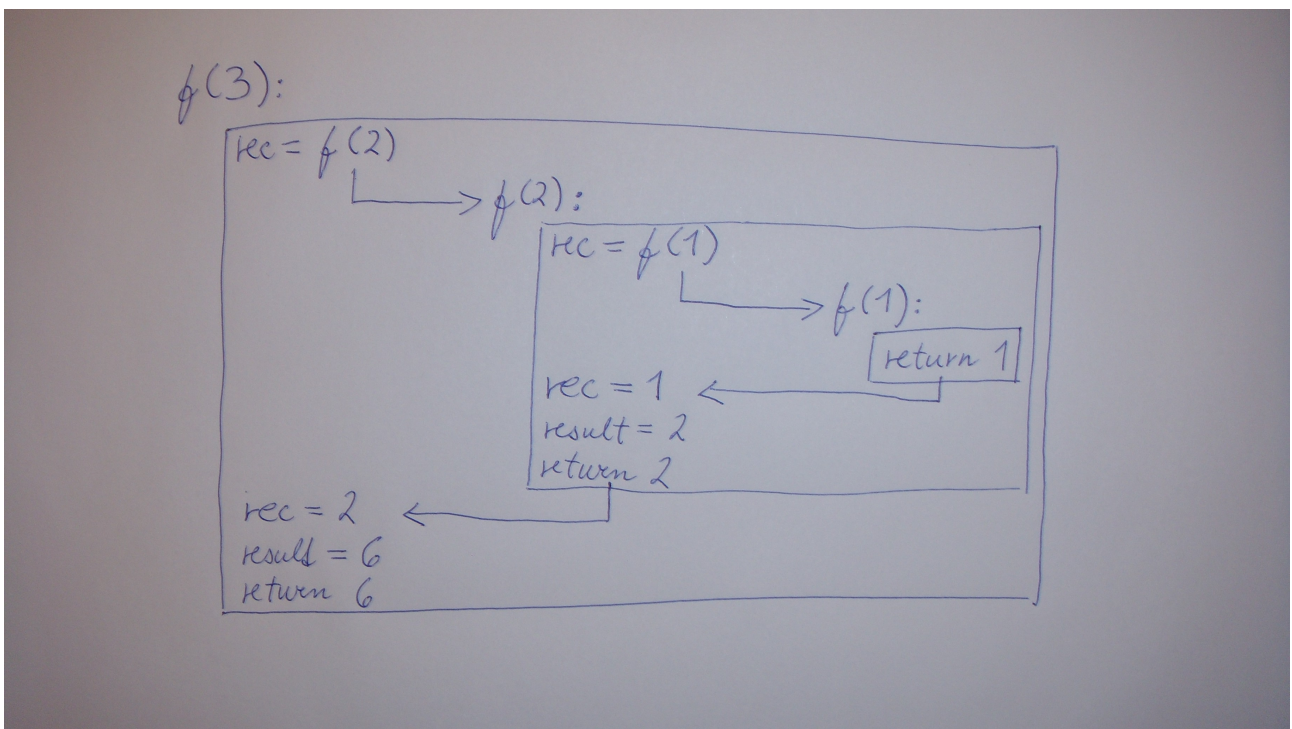
### Vysvetlenie priebehu funkcie faktorial

Uvazujme funkciu faktorial definovanu nasledovne:

```
def factorial(n):  
    if n==1:  
        return 1  
    else:  
        rec=factorial(n-1)  
        result=n*rec  
        return result
```

Ako prebieha vykonavanie funkcie faktorial pre argument n=3 ?

Schema:



Na dalsej strane nasleduje podrobny slovny popis.

Slovný popis:

Začne sa vykonávať funkcia `factorial(3)`.

Najprv sa overí, či platí podmienka `n==1`.

Podmienka neplatí.

Začne sa preto vykonávať kód vo vetve “else”.

V prvom riadku v tomto kóde je priradenie “`rec=factorial(n-1)`”.

Najprv sa ohodnotí prava strana tohto priradenia.

Prava strana tohto priradenia volá funkciu `factorial` s argumentom `n-1`.

Zavola sa teda funkcia `factorial(2)`.

Začne sa vykonávať funkcia `factorial(2)`.

Overí sa, či platí podmienka `n==1`.

Podmienka neplatí.

Začne sa preto vykonávať kód vo vetve “else”.

V prvom riadku v tomto kóde je priradenie “`rec=factorial(n-1)`”.

Najprv sa ohodnotí prava strana tohto priradenia.

Prava strana tohto priradenia volá funkciu `factorial` s argumentom `n-1`.

Zavola sa teda funkcia `factorial(1)`.

Začne sa vykonávať funkcia `factorial(1)`.

Overí sa, či platí podmienka `n==1`.

Podmienka platí.

Začne sa preto vykonávať kód vo vetve “`if n==1`”.

Tento kód obsahuje iba príkaz “`return 1`”.

Funkcia `factorial(1)` preto vráti hodnotu 1 a skončí.

Pokračuje sa vo vykonávaní funkcie `factorial(2)`.

Funkcia `factorial(1)` vrátila hodnotu 1, do premennej `rec` vo funkcii `factorial(2)` sa teda uloží hodnota 1.

Následne sa do premennej `result` uloží hodnota 2.

Následne funkcia `factorial(2)` vráti hodnotu 2 a skončí.

Pokračuje sa vo vykonávaní funkcie `factorial(3)`.

Funkcia `factorial(2)` vrátila hodnotu 2, do premennej `rec` vo funkcii `factorial(3)` sa teda uloží hodnota 2.

Následne sa do premennej `result` uloží hodnota 6.

Následne funkcia `factorial(3)` vráti hodnotu 6 a skončí.