

Kapitel 3: Programmera i Python

Övning 2: Rekursiva funktioner

I detta kapitel skall titta på hur man skriver rekursiva funktioner i Python

Mål:

- Definiera rekursiva funktioner
- Rekursiv form kontra sluten form

En rekursiv funktion är en funktion som anropar sig själv.

Ett bra exempel på en funktion som kan definieras rekursivt är fakultet-funktionen som vi tittat på tidigare.

Vi har exempelvis att $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 24$. Detta kan mer allmänt skrivas som $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1$

I rekursiv form blir det att: $n! = n \cdot (n-1)!$ och att $1! = 1$.

Som du ser måste en rekursiv funktion ha någon slags slutvillkor, i detta fall att om $n=1$ så är svaret 1.

Vi ska skall skapa funktionen fakultet(n) där $fakultet(n) = n \cdot fakultet(n-1)$.

Funktionen har formen:

- om $n = 1$ är resultatet 1
- annars är resultatet $n \cdot fakultet(n-1)$

Skapa ett Python-program där du definierar den här funktionen och prova sedan att beräkna några olika resultat.



```
1,2 1,3 1,4 Python2 RAD 5/6
fakultet.py
def fakultet(n):
    if n==1:
        return 1
    else:
        return n*fakultet(n-1)
```

En nackdel med denna funktion är att för stora värden på n är handenhetens minne inte tillräckligt. Ta reda på vad som är det största värdet för n som inte ger ett felmeddelande.

Som vi tidigare sett i kapitel 2, övning 2, så är ett annat tillvägagångsätt att använda en for-loop.

I funktionen skapar vi en variabel som vi sätter till 1 och där det nya resultatet är det gamla multiplicerat med räknaren (loop-variabeln) i en for-loop som går från 1 till n . (Kom ihåg range inte inkluderar sista talet.)

Definiera den här funktionen som fakultet1(n) och kontrollera att du med den funktionen kan beräkna $500!$ Kontrollera gärna om du kan få fram det svaret i det vanliga räknarläget i TI-Nspire.



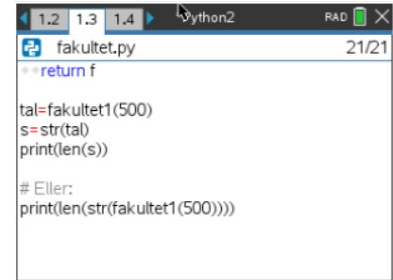
```
1,1 1,2 1,3 Python2 RAD 4/22
fakultet.py
def fakultet(n):
    if n==1:
        return 1
    else:
        return n*fakultet(n-1)

def fakultet1(n):
    f=1
    for i in range(1,n+1):
        f=f*i
    return f
```



Om du vill veta hur många siffror talet 500! har kan du göra detta i Python genom att först konvertera numret till en sträng med **str()** och sedan beräkna längden på den strängen med **len()**.

Kontrollera att 500! har av 1135 siffror.



```
fakultet.py 21/21
--return f

tal=fakultet1(500)
s=str(tal)
print(len(s))

# Eller:
print(len(str(fakultet1(500))))
```