

Kapitel 1: Python med Nspire, en introduktion

Övning 3 : Python-editorn

I den här övningen lär vi oss mer om Pythons syntax och om användningen av variabler

Mål:

- Att använda variabler
- Att använda input()-kommandot.
- Att skapa funktioner

Det är ofta nödvändigt att kunna påverka programmets gång under körningen. Ett av kommandona som kan göra hjälpa oss göra detta är kommandot **input()**. Vi skall skapa ett program som först frågar efter användarens namn och sedan skriver ut namnet på skärmen.

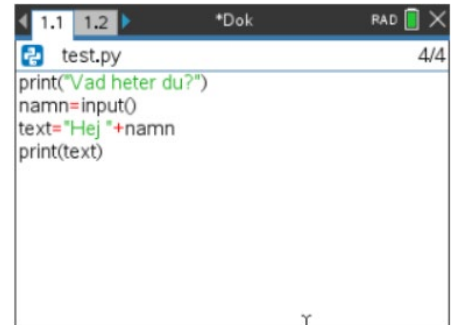
Börja igen med ett nytt Python-program (Nytt dokument, Lägg till Python, Nytt...) och ge programmet något lämpligt namn.

Programmet skall börja att fråga efter användarens namn med hjälp av print-kommandot. Så behöver vi hämta in namnet med kommandot input. Sedan behöver vi spara det angivna namnet någonstans (tilldela det till en variabel). Vi kan till exempel välja att lagra det i en variabel men namnet "namn".

Om vi vill skriva ut "Hej" och så namnet på personen så kan vi göra detta genom att lägga ihop strängar (konkatenera strängar). Vi gör det med ett plustecken. Vi kan lagra resultatet i en ny variabel, "text".


Slutligen så skriver vi ut svaret till konsolen.

Mata in programmet och kör det.



```
test.py 4/4
print("Vad heter du?")
namn=input()
text="Hej "+namn
print(text)
```

Programmet kan göras lite kortare. Frågan vi skrev ut kan flyttas in i input-kommandot och sammanslagningen av texten kan flyttas in i print-kommandot



```
test.py 2/2
namn=input("Vad heter du?")
print("Hej "+namn)
```

Input-kommandot returnerar alltid en sträng, det vill säga den ger alltid ifrån sig en sträng oberoende av vad som matas in. Det kan vara problematiskt om det du vill ha är ett tal. Så om vi vill ha ett tal så måste vi konvertera strängen till ett heltal (**int**) eller decimaltal (**float**).

Vi skall göra ett program som frågar efter ett nummer och sedan skriver ut kvadraten för det numret - så här behöver vi konvertera strängen till ett tal.

Programmet börjar med ett input-kommando och avslutas med ett print-kommando. Programmet visas till höger.



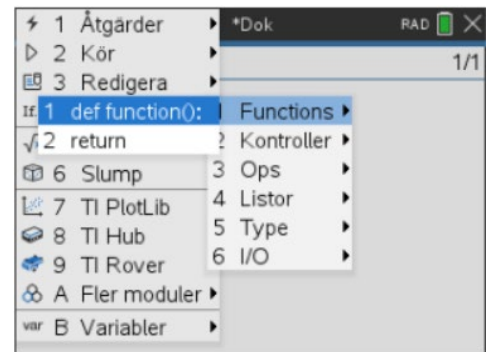
```
*test.py
a=input("Mata in ett ta:")
b=float(a)
c=b**2
print(c)
```

I Python anger du upphöjt till med två asterisk-tecken.

Förutom de inbyggda funktionerna i Python så kan du definiera egna funktioner.

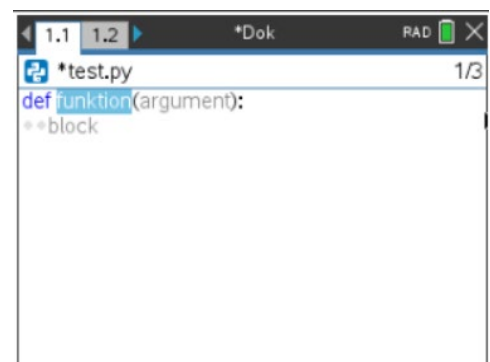
Vi ändrar programmet så att beräkningen av kvadraten kommer att göras i en funktion.

Syntaxen (grammatiken) är viktig, så om du använder menyn minskar du risken för syntax-fel.



Gå till Meny, 4: Inbyggda, 1:Functions, 1: def funktion()).

Detta ger oss en mall för att skapa egendefinierade funktioner.



```
*test.py
def funktion(argument):
+=block
```

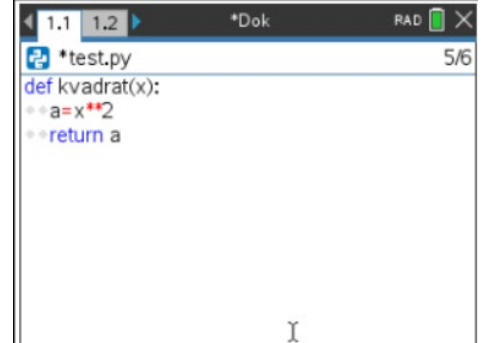


Ersätt ordet funktion med det namn du vill ha på funktionen (till exempel kvadrat).

Ersätt ordet argumentet med det variabelnamn du vill ha på argumentet (till exempel x).

Ersätt ordet block med den kod du vill skall köras då funktionen anropas. (Du kan använda tab för att ta dig från ett inmatningsfält till nästa.)

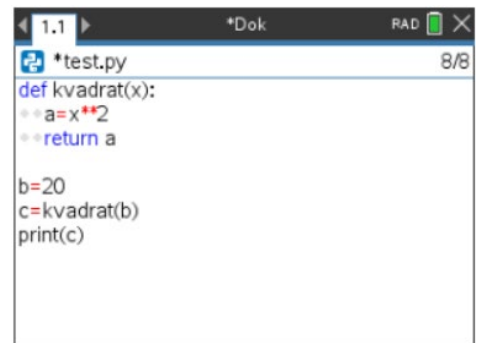
För att tala om för Python vad som skall returneras (svaras) med så använder du kommandot **return** följt av det du vill skall returneras. I exemplet är detta det värde variabeln a har.



```
1.1 1.2 *Dok RAD 5/6
def kvadrat(x):
  a=x**2
  return a
```

Mallen visar hur syntaxen ska se ut:

Som du ser så är alla rader inne i funktionen indragna (indenterade) med samma antal mellanslag. I Python så är all kod på samma nivå indenterade lika mycket. Detta skiljer sig från många andra programmeringsspråk, till exempel det ursprungliga språket i Nspire där motsvarande kod skrivs inom par av ord, Prgm och EndPrgm, eller som i JavaScript där ett motsvarande block av kod skrivs inom { och }.



```
1.1 *Dok RAD 8/8
def kvadrat(x):
  a=x**2
  return a

b=20
c=kvadrat(b)
print(c)
```

Nästa steg är att anropa funktionen. I programmet till höger skrivs kvadraten på 20 ut när programmet körs.