

Kapitel 5: Ti\_system-modulen

Övning 1: Att utbyta data

I den här lektionen kommer vi att bekanta oss med ti\_system-modulen med vilken data kan utbytas mellan Python andra TI-Nspire-applikationer.

**Mål:**

- Exportera värden till Nspire
- Anropa funktioner i Nspire.

Ti\_system-modulen gör det möjligt att utbyta data mellan Python och andra applikationer. Du kan utbyta variabler, listor och funktionsvärden.

Öppna ett nytt dokument och välj en miniräknarsida (alternativ 1).  
Definiera en funktion, till exempel  $f(x) = x^2$ .

(Observera att när du definierar en funktion i räknarläget på TI-Nspire så måste du använda := tecknet.)

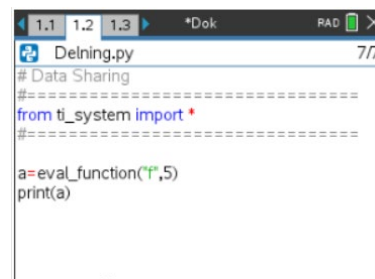


Lägg till en ny sida i dokumentet (CTRL + Doc (+Page)) och välj en Python-sida av typen Datadelning. Med detta val importeras modulen ti-system automatiskt. (Denna modul finns också i menyn under Meny>Fler moduler>TI system och sedan alternativ 1.)

Menyn ti\_system innehåller funktionen **eval\_function()**, under meny>Fler moduler.

Denna funktion har två parametrar. Det första är namnet på den funtion i Nspire du vill anropa (i vårt fall "f") och den andra är det värde du vill skicka över till funktionen.

Kontrollera att programmet returnerar 25.

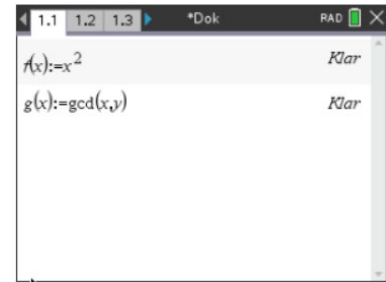


Vi kan använda detta för funktioner som inte finns i Python men som finns i TI-Nspire. Vi ska titta på två sådana funktioner. Den första är att beräkna den största gemensamma delaren till två tal, gcd(a,b).

Den andra är att kontrollera om ett tal är ett primtal.

Vi börjar med att gå tillbaka till räknarsidan. Vi kan göra det med ctrl och uppåt-pil och sedan välja sidan.

Väl där så definierar vi funktionen  $g(x)=\text{gcd}(x,y)$



```
f(x):=x^2      Klar
g(x):=gcd(x,y)  Klar
```

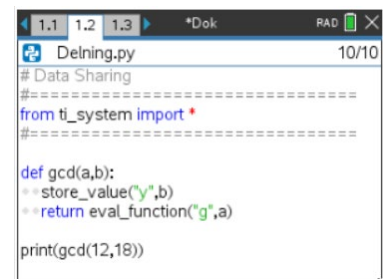
Gå tillbaka till Python-sidan och definiera funktionen  $\text{gcd}(a,b)$

I exemplet till höger börjar funktionen med `store_value("y", b)`.

Detta beror på att `eval_function()` i Python endast kan hantera funktioner med en parameter. Andra värden som också måste också vara kända kan anges med `store_value()`. I detta fall ger vi `y` ett värde innan vi anropar funktionen med hjälp av `eval_function()`.

Funktionen `store_value()` finns i `Ti_system`-menyn.

Kontrollera att du kan använda Python för att beräkna gcd för två på detta sätt.

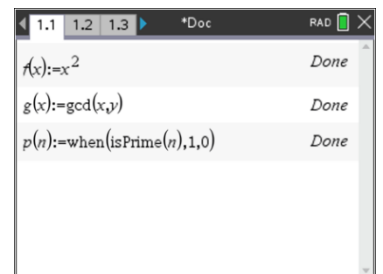


```
Delning.py 10/10
# Data Sharing
#=====
from ti_system import *
#=====
def gcd(a,b):
    +=store_value("y",b)
    +=return eval_function("g",a)
print(gcd(12,18))
```

I följande exempel använder vi funktionen `isprime()` i TI-Nspire.

Denna funktion returnerar **true** om in-värdet är ett primtal och **falskt** om det inte är primtal.

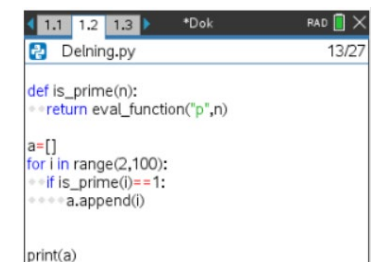
Funktionen `eval_function()` i Python kan inte hantera värden av denna typ (Boolean), men den fungerar bra om beräkningsresultatet är ett tal. För att få detta att fungera så definierar vi på räknarsidan en funktion  $p(n)$  sådan att resultatet är 1 för ett primtal och 0 om det inte är det.



```
f(x):=x^2      Done
g(x):=gcd(x,y)  Done
p(n):=when(isPrime(n),1,0)  Done
```

I Python ser funktionen ut som den till höger.

Vi har också lagt till några rader som skriver ut en lista med alla primtal under 100.



```
Delning.py 13/27
def is_prime(n):
    +=return eval_function("p",n)
a=[]
for i in range(2,100):
    +=if is_prime(i)==1:
    +=+=a.append(i)
print(a)
```