Skolan för Datavetenskap och kommunikation

Programmeringsteknik

Föreläsning 2

Funktioner (kap 6)

- Kap 6 i Dawson (och resten av kap 3)
- Funktioner du redan använt
- Anropa funktioner
- Definiera egna funktioner
- Parameter & returvärde
- While-slingan

Funktioner du redan använt

Funktion	Indata (parametrar)	Utdata (returvärde)
`svar=input("Gissa:")`	strängen "Gissa"	den inlästa gissningen
`print(17)`	talet 17	None (inget)

`from math import *`
`x=0.14`
`tal=sin(x)`
	värdet 0.14 i variabeln x	Det beräknade värdet av sin(0.14)

`antal=len("kålrot")`	strängen "kålrot"	6 (antal tecken i strängen)

Anropa funktioner

- Så här ser anrop ut: `utdata = funktion(indata)`
- Indata skickas in via *parametrar* till funktionen
- Utdata returneras via *return-sats* ur funktionen
- Programmet *fortsätter* efter anropet

Indata Utdata

Hur man definierar en funktion

- Funktioner definieras överst i programmet!
- Skriv först `def` *funktionsnamn*`(`*parametrar*`):`
- Sen, indenterat:
  - En kommentarrad som beskriver vad funktionen gör, inom tredubbla citationstecken, tex `"""Beräknar arean"""`.
  - Satserna som funktionen ska utföra.
  - Allra sist `return` *returvärde/returvärden*
  - Anger man inget returvärde blir det `None`

Parameter & returvärde

Funktionen definieras så här:

```
def ränta(pengar):
    """Beräknar och returnerar räntan."""
    if pengar > 100000:
        r = pengar*0.75/100
    else:
        r = pengar*0.40/100
    return r
```

returvärde r

parameter pengar

Funktionen anropas så här:

```
vinst = ränta(saldo)
```

Flera parametrar/returvärden

```
def ränta(pengar, extra):
    """Beräknar och returnerar räntan."""
    if pengar > 100000:
        r = pengar*0.75/100
        rx = extra*0.5
    else:
        r = pengar*0.40/100
        rx = extra*0.4
    return r,rx
```

```
vinst = ränta(saldo, 1000)
```

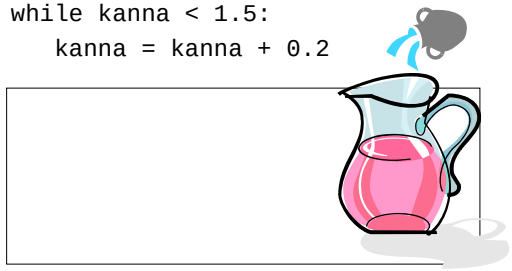
Uppgift: Du vill skriva en funktion som avgör om en låneansökan ska beviljas.
- Vad är indata (parametrar)?
- Vad är utdata (returvärden)?

while-slingan

- En while-slinga upprepar ett antal satser så länge som ett villkor är uppfyllt.
- Så länge som kannan inte rinner över:
  - Fyll på mer vatten!
- Så länge som du inte har somnat:
  - Räkna ett får till!
- Så länge som du inte gissat rätt tal:
  - Gissa en gång till!

Ett exempel

```
kanna = 0
while kanna < 1.5:
    kanna = kanna + 0.2
```



Oändlig slinga

- Om villkoret aldrig uppfylls får man en slinga som upprepas i all oändlighet.
- Kan yttra sig som att programmet "hänger sig" - inget händer
- Eller att massor av text rusar förbi på skärmen (om man har utskrift i slingan).
- Avbryt programmet genom att trycka **Ctrl-C** (Ctrl och C samtidigt).

Läxa

- Till nästa labb ska ni skriva en funktion. För instruktioner, se Övningen i schemat på KTH Social!