

DD1312

Programmeringsteknik

Föreläsning 2

Frågor

Övningssal

Se lista över "Assistenter och salar" på webben

Idag

- Villkor
- Funktioner

Villkor

- Ett villkor har värdet `True` eller `False`.
- Datatypen kallas *boolean*.
- En *boolesk* variabel kan sättas direkt till ett villkorsvärde.
- Exempel: `spara = True`

jämföra=compare
villkor=condition

operatorer i villkor

<	mindre än
>	större än
<=	mindre än/lik med
>=	större än/lik med
==	lik med
!=	skilt från

Operatorer i villkor

Operator	Om vi satt dag=20 blir	
==	dag==20	True
!=	dag!=20	False
>	dag>5	True
<	dag<5	False
>=	dag>=5	True
<=	dag<=20	True

Villkor med strängar

Strängar kan också jämföras!

Exempel:

```
"allsång" < "allting"
```

Jämföra strängar

- Strängar kan jämföras med avseende på likhet:

```
lösenord = input("Ge lösenordet: ")  
if lösenord == "qwerty":  
    print("Du är inloggad.")
```

- Och även med alfabetisk ordning:

```
if "elefant" < "elmätare":  
    print("elefant kommer före\  
        elmätare i ordlistan")
```


operatorer i villkor

- `and` Kombinerar två villkor. Blir True om bägge villkoren är True.
- `or` Kombinerar två villkor. Blir True om den ena eller bägge villkoren är True.
- `not` Verkar på ett villkor. Gör True till False och False till True.

kombinera villkor

Villkor kan kombineras med operatorerna and, or, not

True and True	True
True and False	False
True or True	True
True or False	True
not True	False
not False	True

Givet ålder=21, namn="Mireille";
vilket/vilka av följande villkor blir False?

- A. (ålder == 21) and (namn == "Mireille")
- B. (ålder < 22) or (namn == "Algot")
- C. (ålder == 5) or (namn > "Mireille")
- D. (ålder >= 21) and (namn > "Algot")
- E. Inget av villkoren

funktioner (kap 5)

- Inbyggda funktioner
- Anropa funktioner
- Definiera egna funktioner
- Parameter & returvärde

Funktioner



Gör programmet mer
överskådligt och lättläst.

Enklare att
testa delarna
var för sig.



Man slipper skriva
om samma satser
på flera ställen.

Kan återanvändas i
nästa program



inbyggda funktioner, exempel

Funktion	Indata	Utdata
type <code>typ=type(variabel)</code>	variabel	typen
print <code>print("Oj!")</code>	sträng	None
int <code>antal=int(nr)</code>	nr	heltal

inbyggda funktioner, alla

<http://docs.python.org/3/library/functions.html>

uppgift

Vad gör funktionen **round**?

Vad är indata?

Vad är utdata?

anropa funktioner

Indata skickas in via *parameter* till funktionen

Funktionen *returnerar* utdata

```
utdata = funktion(indata)
```



skriva en egen funktion

Funktionshuvud - funktionens namn och ev parametrar

Kommentar som berättar vad funktionen gör

Satser

Return-sats för att returnera ett eller flera värden

funktionshuvud

```
def f(x, y):
```

beskrivande kommentar

```
    """Beräknar x+y"""
```

satser

```
    summa = x + y
```

retur

```
    return summa
```

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

Funktionen *anropas* så här:

```
vinst = ränta(saldo)
```

parameter pengar

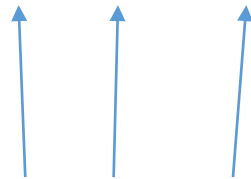
Hitta felet!

```
def f(x):  
    y = 2*x+4  
    return y
```

```
x = int(input("Ge x: "))  
f(x)  
print(f)
```

flera parametrar

```
def f(a, b, c) :
```



Anrop

```
f(x, y, z)
```

flera returvärden

```
return m, n
```

Anrop `m, n = f(x)`

uppgift

Du vill skriva en funktion som avgör om en låneansökan ska beviljas.

1. Vad är indata (parametrar)?
2. Vad är utdata (returvärden)?


```
def egetArbete(hp, veckor, schemalagt):  
    """ Räkna ut antal timmar  
        eget arbete per vecka """  
  
    totaltimmar = hp/1.5*40  
    egnaTimmar = totaltimmar - schemalagt  
    perVecka = egnaTimmar/veckor  
    return perVecka
```

```
timmar = egetArbete(8, 13, 73)  
print(timmar)
```

```
def egetArbete(hp, veckor, schemalagt):  
    """ Räkna ut antal timmar  
        eget arbete per vecka """  
  
    totaltimmar = hp/1.5*40  
    egnaTimmar = totaltimmar - schemalagt  
    perVecka = egnaTimmar/veckor  
    return perVecka  
  
def main():  
    timmar = egetArbete(8, 13, 73)  
    print(timmar)  
  
main() #Huvudprogram
```

Läxa

- Till nästa labb:
- *Skriv och testa fyra egna funktioner*
- Mer info på övningen!