

DD1310/DD1314/DA3009 Programmeringsteknik

Föreläsning 1

Kursinfo
Diagnostiskt prov
Demo
print
variabler
reserverade ord
input
kommentarer
beräkningar
datatyper
if-satser

Vilka läser kursen?

- Samhällsbyggnad *Labbar på V (Windows)*
- Civilingenjör och Lärare *Labbar på CSC (Ubuntu)*
- Datorlingvistik (SU)
- Kursnämndsrepresentanter?

Lärandemål...

Efter godkänd kurs ska du kunna

- Pythons syntax
- god programmeringsstil
- hitta och rätta fel
- ändra färdiga program
- hämta data från fil
- lagra data på fil

...fler lärandemål

- villkor och slingor
- funktioner
- problem -> program
- datastrukturer (t ex listor, klasser)
- enkla grafiska gränssnitt
- granska andras program

VARFÖR?

för att du ska kunna:

- använda programmering för att lösa problem,
- tillämpa problemlösningsmetodiken även inom andra områden än programmering,
- diskutera programutveckling med experter,
- bedöma kommersiella program.

Undervisning

- Entimmesföreläsningar med läsanvisningar. Kom förberedd för bästa resultat!
- Labbtimmar - handledning
- Övningstimmar - redovisning

VAD SKA DU GÖRA?

Laborationer (LAB1; 1,5hp)
• Fem labbar: program & instuderingsuppgifter
Prov (LAB2; 1,5 hp)
• Skriftligt prov på grunderna
P-uppgift (LAB3; 3p)
• Personlig uppgift
Algoritmlabb (endast Datorlingvistik)
Webblabbar (endast CLGYM)

REKLAM FÖR PYTHON

Lätt att lära sig
Kraftfullt
Objektorienterat
Kan kombineras med andra applikationer
Flyttbart
Används av många
Gratis



ENKLASTE PROGRAMMET

```
print("Hej")
```

En *sats* som skriver ut Hej på skärmen.
Kommandot heter print (PRINT eller Print fungerar inte)
Texten "Hej" kallas för ett *uttryck*
sats=statement
kommando=command
uttryck=expression

VARIABLER

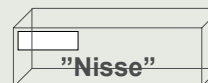
Variabler används för att lagra data i programmet.

En variabel skapas en tilldelning.

Variabelnamn får bara innehålla bokstäver, siffror och understreck, men får inte börja med en siffra.

tilldelning=assignment

```
# Variabeldemonstration  
namn = "Olle"  
print("Grattis" +  
      namn)
```



RESERVERADE ORD

Följande ord är reserverade i Python:

```
and del for is raise  
assert elif from lambda try  
break else global not while  
class except if or return  
continue exec import pass  
def finally in print
```

Reserverade ord har betydelse i språket och får inte användas som variabelnamn.

UPPGIFT: Vilka av följande är OK som variabelnamn?

anka
and
må
moment22
pass
passfoto
akademiskaHus
17tåget
el-ledning
jord_prover

INLÄSNING (PYTHON 3)

- Funktionen input() används vid inläsning av strängar:

```
namn=input("Vad heter du?")  
print("Nämen", namn, "då!")
```

- Vill vi ha tal kan vi konvertera med t ex int(input())

```
storlek=int(input("Ge skostorlek: "))  
print("Ta "+str(storlek+1)+" i skridskor")  
print("så får du plats med sockor också!")
```

Inläsning=user input

Kommentarer

- Alla rader som börjar med # blir kommentarer, som datorn inte bryr sig om.

```
# Programmet som ger  
# komplimanger  
# Skrivet av Linda Kann 120117  
print("Hej, ")  
print("Vilken fin klänning,")  
print("och vad gott du luktar!")
```

BMI-PROGRAM

```
# Läser in längd och vikt,  
# skriver ut BMI (Body Mass Index)  
print("Välkommen till BMI-beräknaren!")  
langd=input("Hur lång är du (i meter)? ")  
vikt=input("Hur mycket väger du (i kg)? ")  
bmi=vikt/(langd*langd)  
print("Din bmi är: ", bmi)
```

UPPGIFT: I vilken ordning ska satserna stå?

- a. `sidorPerDag = sidor/dagar`
- b. `sidor = 63`
- c. `print("Antal sidor per dag:", sidorPerDag)`
- d. `dagar = 3`

ALGORITM

1. Läser in indata
2. Gör beräkningar
3. Skriver ut resultatet



Heltalsberäkningar

Operator:	Beskrivning:	Exempel:	Resultat:
*	multiplikation	3*4	12
/	division	53//10	5
%	modulo	53%10	3
+	addition	10+12	22
-	subtraktion	5-8	-3

heltal=integer

**UPPGIFT: HUR KAN MAN
ANVÄNDA % FÖR ATT TA REDA PÅ
OM ETT TAL ÄR JÄMNT ELLER
UDDA?**

Flyttalsberäkningar

Operator	Beskrivning	Exempel	Resultat
*	multiplikation	2.0*1.5	3.0
/	division	10.0/8.0	1.25
%	modulo	4.25%4.0	0.25
+	addition	0.3+0.4	0.7
-	subtraktion	1.0-0.1	0.9

flyttal=floating-point number

Strängar

En *sträng* är en följd av tecken.

Strängar *konkateneras* med +

tex blir "kus" + "lig" strängen "kuslig"

Strängar upprepas med *

tex blir "nä"*3 strängen "nännänä"

Sträng är en *datatyp*. Andra datatyper är *heltal* och *flyttal*.

sträng=string

Konkatenera (slå ihop)=concatenate

Typkonvertering

Funktion	Beskrivning	Exempel:	Blir
float(x)	Konverterar till flyttal	float("3.14")	3.14
int(x)	Konverterar till heltal	int("17")	17
str(x)	Konverterar till en sträng	str(39)	"39"

parameter=argument

Styrstrukturer

I ett program utförs satserna i ordning uppifrån och ner:

```
print("So long")
print("and thanks")
print("for all the fish!")
```

Hur gör man för att hoppa över en sats eller för att upprepa en sats flera gånger?

```
talet = 17
print("Gissa talet - fina priser!")
gissning=int(input("Din gissning:"))
if gissning == talet:
    print("Hurra, du vann!")
    print("Tyvärr är priserna slut.")
else:
    print("Fel svar.")
print("\n--- Välkommen åter ---")
```

if-satsen

```
if villkor:
    block1
elif:
    block2
else:
    block3
```

if-satsen används för val mellan två alternativ,
elif och *else* kan användas vid behov.

Raderna efter kolon bildar ett *block*: en eller flera
satser som är *indenterade* (tabbar i början av raden).