

Skolan för Datavetenskap och kommunikation

DD1319

Programmeringsteknik

Föreläsning 0

Kursansvarig, examinator: Linda Kann

Vad händer på Labb 0?

- Vi använder Ubuntu-datorerna i datorsalarna.
- Om du vill ta med egen laptop går det bra, men vissa delar av labben måste göras på Ubuntu-datorerna.
- Labbarna görs i grupper om två. Om du kan mer än din labbkompis så får du sätta på dig lärarhatten.
- Labben görs på plats på KTH. Om du är sjuk men orkar delta hemifrån så ordnar vi Zoom med en labbkompis på plats.
- I slutet av labben är det redovisning - då tillkallar ni en asse och visar vad ni gjort, och assen rapporterar in i Canvas att ni är godkända på Labb 0.

Kursnämndsrepresentanter sökes

Maila mig på lk@kth.se

Kursinformation

Lärandemål...

Efter godkänd kurs ska du kunna

- konstruera program utan kodupprepningar
- dela upp ett större problem i hanterliga delar
- dela upp ett program
- tillämpa styrstrukturer
- utforma och presentera användarvänliga utdata
- skapa flexibla applikationer
- välja lämpliga identifierarnamn

...fler lärandemål

- ❑ konstruera interaktiva program
- ❑ använda och konstruera sammansatta datatyper och klasser
- ❑ överföra data mellan fil och program
- ❑ granska andras program
- ❑ skapa statiska och dynamiska webbsidor
- ❑ utveckla enkla webbtillämpningar

...för att du ska kunna:

- använda programmering för att lösa problem och i undervisning i gymnasiet
- tillämpa problemlösningsmetodiken även inom andra områden än programmering
- diskutera programutveckling med experter
- bedöma kommersiella program
- självständigt och i grupp lösa problem genom att konstruera program på upp till femhundra rader i ett modernt programspråk
- konstruera en enkel dynamisk webbsida.

Undervisning

- Entimmesföreläsningar. Kom förberedd!
- Datorlabbar (**obligatorisk närvaro**)
- Övningar (**obligatorisk närvaro**)

Labbarna är på plats i datorsalar på KTH.

Vilka moment ingår i kursen?

Laborationer (LAB1; 1,5hp) (obligatorisk närvaro labbar + övningar)

Sju labbar (Labb 0 - Labb 6)

Prov (LAB2; 1,5 hp)

Skriftligt prov på grunderna

P-uppgift (LAB3; 3p)

Större individuellt programmeringsprojekt

Webbprogrammering (LAB4; 2p)

Tre labbar: program & quiz + projekt

Betyg

Betyget på kursen avgörs av moment LAB3: P-uppgiften.

Bonuspoäng för att du gör saker i tid kan höja betyget.

moment		max-bonus
LAB1	<p>1 bonuspoäng för varje lab (av labb 1-6) som redovisas och lämnas in i tid.</p> <p>1 bounspoäng extra för quiz (10 st) som gjorts i tid.</p>	16
LAB2	1 bonuspoäng för att ha klarat provet v. 48 (inklusive komplettering)	1
LAB3	4 bonus för specen, 4 bonus för prototyperna (om godkända i tid)	8
LAB4	<p>1 bonuspoäng för varje labb (av labb W1-W3) som lämnas in i tid</p> <p>1 bounspoäng för varje uppsättning inlämningsuppgifter som lämnas in i tid</p>	6
Totalt		31

22 bonuspoäng eller mer ger ett stegs höjning av betygen E, D, C och B.

Kurslitteratur

Det finns länkar till material (i Möbius) på Canvas

För dig som vill ha en mer omfattande bok rekommenderar vi Tony Gaddis: *Starting Out with Python, Global Edition*.

Vad händer under en typisk vecka?

Från vecka 40 ser schemat ut så här varje vecka:

- ★ Föreläsning 1 tim
- ★ Labb 2 tim (obligatorisk närvaro)
- ★ Föreläsning 1 tim
- ★ Övning 1 tim (obligatorisk närvaro)

(Efter att Labb 6 är klar ändras det)

Att göra-lista i Canvas

Enkät (inte examination)

Fyll i start-enkäten!

Se Startsidan på Canvas

Se till att du är inloggad i Canvas innan du fyller i denna enkät!

Reserverade ord

Följande ord är reserverade i Python:

<code>and</code>	<code>del</code>	<code>from</code>	<code>None</code>	<code>True</code>
<code>as</code>	<code>elif</code>	<code>global</code>	<code>nonlocal</code>	<code>try</code>
<code>assert</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>while</code>
<code>break</code>	<code>except</code>	<code>import</code>	<code>or</code>	<code>with</code>
<code>class</code>	<code>False</code>	<code>in</code>	<code>pass</code>	<code>yield</code>
<code>continue</code>	<code>finally</code>	<code>is</code>	<code>raise</code>	
<code>def</code>	<code>for</code>	<code>lambda</code>	<code>return</code>	

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

Uppgift:

Vilka av följande är OK som variabelnamn?

anka

and

mås

moment22

variabel

pass

pass-foto

akademiskaHus

17tåget

jord_prover

Inläsning

Funktionen `input()` används vid inläsning.

Typen på värdet vi får från `input` är alltid `String`, dvs en textsträng.

Vill vi ha tal kan vi konvertera med **`int(input())`** eller **`float(input())`**

Exempel

```
namn = input("Vad heter du? ")  
print("Välkommen", namn)
```

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

Kommentarer

Alla rader som börjar med # blir kommentarer.
En längre kommentar kan omslutas av tre citattecken.

```
# Detta är en enrads-kommentar
```

```
""" Det här en längre kommentar  
som sträcker sig över  
flera rader. """
```

Uppgift:

I vilken ordning ska satserna stå?

- a. `sidor = boksidor/dagar`
- b. `boksidor = 137`
- c. `print("Läs:", sidor, "sidor om dan.")`
- d. `dagar = 3`

Algoritm

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

Heltalsberäkningar

Operator:	Beskrivning:	Exempel:	Resultat:
/	division	53/10	5.3
//	heltalsdivision	53/10	5
%	modulo	53%10	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
**	upphöjt till	2.5**2	6.25
+	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 +

t ex blir "kus" + "lig" strängen "kuslig"

Strängar upprepas med *

t ex blir "nä"*3 strängen "nänänä"

sträng = *String*

konkatenera (slå ihop) = *concatenate*

Typkonvertering

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

Funktion med returvärde

```
def f(x) :  
    y = 4.18*x-0.92*x**2  
    return y
```

Funktion med returvärde, anrop med x

```
def f(x):  
    y = 4.18*x-0.92*x**2  
    return y
```

```
x = 1  
svar = f(x)  
print(svar)
```

Funktion med returvärde, anrop med x=1

```
def f(x):  
    y = 4.18*x-0.92*x**2  
    return y
```

```
svar = f(x=1)  
print(svar)
```

Funktion med returvärde, anrop med 1

```
def f(x):  
    y = 4.18*x-0.92*x**2  
    return y
```

```
svar = f(1)  
print(svar)
```

exempel på funktion med två parametrar

```
def summa(lunch, middag):  
    """Beräknar kostnad för dagens mat"""  
    kostnad = lunch + middag  
    return kostnad
```


Hur kan vi anropa funktionen?

```
def summa(lunch, middag):  
    """Beräknar kostnad för dagens mat"""  
    kostnad = lunch + middag  
    return kostnad
```