

Python del 1

Daniel Bosk¹

Avdelningen för informations- och kommunikationssystem (IKS),
Mittuniversitetet, Sundsvall.

python.tex 1317 2013-09-23 13:12:56Z danbos

¹Detta verk är tillgängliggjort under licensen Creative Commons Erkännande-DelaLika 2.5 Sverige (CC BY-SA 2.5 SE). För att se en sammanfattning och kopia av licenstexten besök URL <http://creativecommons.org/licenses/by-sa/2.5/se/>.

Översikt

- 1 Introduktion
 - Vad är programmering?
 - Ett enkelt exempel
- 2 Variabler
 - Datatyper
 - Aritmetiska operatorer
 - Identifierare
 - Variabler
 - Typkonvertering
- 3 Funktioner
 - Översikt
- 4 Ett programexempel
 - Beräkning av energi hos en boll
 - Tips och kommentarer

Översikt

- 1 Introduktion
 - Vad är programmering?
 - Ett enkelt exempel
- 2 Variabler
 - Datatyper
 - Aritmetiska operatorer
 - Identifierare
 - Variabler
 - Typkonvertering
- 3 Funktioner
 - Översikt
- 4 Ett programexempel
 - Beräkning av energi hos en boll
 - Tips och kommentarer

Vad är programmering?

Ett enkelt exempel

Översikt

- 1 Introduktion
 - Vad är programmering?
 - Ett enkelt exempel
- 2 Variabler
 - Datatyper
 - Aritmetiska operatorer
 - Identifierare
 - Variabler
 - Typkonvertering
- 3 Funktioner
 - Översikt
- 4 Ett programexempel
 - Beräkning av energi hos en boll
 - Tips och kommentarer

Datatyper

Man kan säga att nästan allt i Python är av någon datatyp. Dessa datatyper håller Python själv reda på, men man måste även som programmerare hålla koll på dessa för att programmet ska fungera som man vill.

De datatyper som Python definierar är bl.a.

int Heltal (Integers),

float Flyttal (Floating point numbers),

str Strängar (Strings).

Aritmetiska operatorer

På dessa datatyper finns operatorer definierade, t.ex. + är en operator definierad för heltal. Andra operatorer som finns är
+ - * / // %.

Aritmetiska operatorer

Några exempel (som körs i Pythons tolk från terminalen):

```
1 \ $ python
2 Python 2.2.3 (#1, Jan 5 2005, 16:36:30)
3 [GCC 3.4.2] on sunos5
4 Type "help", "copyright", "credits" or
   "license" for more information.
5 >>> 5+5
6 10
7 >>> 3*2
8 6
9 >>> 5//3
10 1
11 >>> 5\%3
12 2
13 >>> "hej"+"svejs"
14 'hejsvejs'
15 >>> "hej"*3
16 'hejhejhej'
17 >>>
```

Aritmetiska operatorer

Alla operatorer är dock inte definierade för alla datatyper, en mycket kort och ej fullständig sammanfattning är:

Heltal kan använda operatorerna + - * / // %,

Flyttal kan använda operatorerna + - * /, heltalsdivision och modulo går inte att beräkna för flyttal,

Strängar har bara operatorerna + *.

Identifierare

- Vilka regler gäller då för namnet på konstanterna?
- De får bestå av bokstäver, siffror och även understreck (_).
- Men de får dock inte börja med siffror.
- De får heller inte vara något av följande reserverade ord: and assert break class continue def del elif else except exec finally for from global if import in is lambda not or pass print raise return try while
- Viktigt att tänka på är att man skiljer på gemener och versaler, d.v.s. PI är inte samma sak som Pi.

Variabler

- En variabel är en identifierare.
- Den representerar ett minnesutrymme i vilket man kan lagra data.
- Namnet uppfyller kravet för en identifierare, men får inte vara enbart versaler (för det tolkar Python som en konstant).

Variabler

```
1 >>> x=5
2 >>> y=3
3 >>> z=x*y*3
4 >>> print x, y, z
5 5 3 45
6 >>> print x
7 5
8 >>> print y
9 3
10 >>> print z
11 45
12 >>> x=x+1
13 >>> print x
14 6
15 >>> print z
16 45
17 >>>
```

Variabler

- Notera att värdet på z inte ändras när vi ändrar värdet på x .
- Detta för att det är värdet 45 som lagras i z och inte relationen $x*y*3$.

Typkonvertering

- Ibland kan man vilja konvertera vissa typer till andra.
- Det kan vara att man läst in en sträng från tangentbordet och vill konvertera den till ett tag (om användaren matade in ett tal).

Typkonvertering

```
1 >>> x="3.14"
2 >>> pi=float(x)
3 >>> print pi
4 3.14
5 >>> print x+2
6 Traceback (most recent call last):
7   File "<stdin>", line 1, in ?
8     TypeError: cannot concatenate 'str' and 'int'
       objects
9 >>> print pi+2
10 5.14
11 >>>
```


Typkonvertering

Ett annat bra exempel är procentberäkningar,

```
1 >>> 99/100*100
2 0
3 >>> float(99)/100*100
4 99.0
5 >>>
```

där man får fel svar om man inte explicit typkonverterar.

Översikt

- 1 Introduktion
 - Vad är programmering?
 - Ett enkelt exempel
- 2 Variabler
 - Datatyper
 - Aritmetiska operatorer
 - Identifierare
 - Variabler
 - Typkonvertering
- 3 Funktioner
 - **Översikt**
- 4 Ett programexempel
 - Beräkning av energi hos en boll
 - Tips och kommentarer

Översikt

- Funktioner utgör en viktig byggsten i programmeringen.
- Liksom inom matematiken kan den användas för att dela upp större problem i mindre och bidra till en bättre ordning.
- De används dessutom på samma sätt.
- $f(x) = 2 \cdot g(x) + 3$ och $g(x) = x^2$ inom matematiken skulle kunna skrivas i Python som `def g(x): return x*x`
`def f(x): return 2*g(x)+3`

Översikt

- När vi talar om funktioner består de av flera delar, funktionen består av ett funktionshuvud och en funktionskropp.
- Funktionshuvudet är den första raden i funktionsdefinitionen.
- Den består av funktionsnamn och formella parametrar. D.v.s.

funktionshuvud

def functionname(argument1, argument2, ...):

formella parametrar

- Resten av funktionen är funktionskroppen, d.v.s. all kod som hör funktionen till.

Översikt

- 1 Introduktion
 - Vad är programmering?
 - Ett enkelt exempel
- 2 Variabler
 - Datatyper
 - Aritmetiska operatorer
 - Identifierare
 - Variabler
 - Typkonvertering
- 3 Funktioner
 - Översikt
- 4 Ett programexempel
 - Beräkning av energi hos en boll
 - Tips och kommentarer

Beräkning av energi hos en boll I

```
#encoding: utf8

import math

def kinetic_energy(m,v):
    return 0.5*m*math.pow(v,2)

def potential_energy(m,h):
    G = 9.82
    return m*G*h

def skriv(e1, e2):
    print "Bollens rörelseenergi = ", e1,
        "Joule" # alt + str(e1)
    print "Bollens potentiella energi", e2,
        "Joule"
```

Beräkning av energi hos en boll II

```
svar = raw_input('Ange bollens massa: ')
m = float(svar)

svar = raw_input('Ange bollens höjd: ')
h = float(svar)

svar = raw_input('Ange bollens hastighet: ')
v = float(svar)

e1 = kinetic_energy(m,v);
e2 = potential_energy(m,h);

skriv(e1,e2)
```

Beräkning av energi hos en boll

En körning av detta program (från en terminal) kan se ut så här:

```
1  $ python 01.py
2  Ange bollens massa: 2
3  Ange bollens höjd: 3
4  Ange bollens hastighet: 4
5  Bollens rörelseenergi = 16.0 Joule
6  Bollens potentiella energi 58.92 Joule
7  $
```


Tips och kommentarer

Några punkter att tänka på:

- Blanda aldrig språk, antingen skriver ni alla variabelnamn och funktioner på engelska eller på svenska – aldrig båda!
- Använd förklarande namn till alla variabler och funktioner!
- Skriv kommentarer!
- Dela upp programmen i mindre delar – funktioner! Det blir lättare att följa och det blir snyggare kod, men framför allt mycket enklare att programmera!

