

```
0 # coding: latin
```

Summering

Vi gick igenom bakgrund till python, förberedelser för att börja programmera och köra python. Vidare gick vi igenom **programstruktur**; *identifierare, kommentarer, litteraler, reserverade ord, variabler (lokala och globala)*, och **pythonspråket**; *datatyper, referenser, objekt, konstanter, funktioner (funktionshuvud, funktionaskropp, formella och anropsparametrar, funktion med och utan returvärde), typkonvertering*. Dessa begrepp exemplifierades genom att skriva ett program **energikonvertering.py**. Ytterligare ord som ni ska kunna *inläsning, utskrift och tilldelning*.

Bakgrund python

I december 1989 letade 32 åriga Guido van Rossum efter att göra något under julen. Nått som kunde hålla honom okuperad då hans kontor var stängt. Han bestämde sig för att börja skriva ett programmeringspråk inspirerad av ett tidigare språk ABC, å så var python fött. Namnet python kom sig av att han gillade Monty Python Flying's Circus. Ormen som senare kom att representera python är inget som Rossum hade i huvudet.

Rossum ville att python skulle vara:

- enkelt, intuitivt och kraftfullt
- öppen källkod, så alla kan bidra
- kod ska vara förstålig på vanlig engelska
- korta utvecklingstider

Python har utvecklats från version 1 till 2 till 3 på de 20 år som gått. Rossum har kontrollerat utvecklingen och skämtsamt säger man om honom att han är en "Älskad diktator för livet"

[\[http://en.wikipedia.org/wiki/Guido_van_Rossum\]](http://en.wikipedia.org/wiki/Guido_van_Rossum)

Förberedelse

vilket system? vilken editor? hur körs programmet?

linux, ubuntu

- editors: **emacs**, **gedit** köra: skriv *python3 filens_namn.py*

windows och mac

- editor: **idle** gui-integrerad utvecklingsmiljö, ladda ned från pythons hemsida.
- editor och köra görs i idle

Python 3 används på kursen men kolla gärna in python 2. På ett ytligt plan är mkt lika.

- Python 3an innehåller många små förbättringar, rättat till kända fel och tagit bort föråldrad kod. Dock, det finns mycket kod skrivet 2an som inte ännu går att använda med 3an. Därför för att kunna använda vissa äldre programpaket måste man ibland använda python2.

Pythonprogrammering

Programstruktur

- **identifierare**, ord som användaren skapar i programmet. OBS får inte börja med en siffra.
- **kommentarer**, beskrivande text som hjälper andra och dig själv att förstå programmet.
 - Exempel:
 - # En enradskommentar
 - """Fleradskommentar med fnuttar"""
- **litteral**, värde direkt i programkoden. Kan tex. va heltal, flyttal, strängar och logiska

- **reserverade ord**, ord med fördefinierad betydelse i programmet tex. **def**, **and**, **assert**, **break**, **class**, **continue**, etc.
- **variabler**, identifierare som håller ett visst minnesutrymme. Antas kunna ändra värde i programmet, där av namnet variabel.
 - lokal variabel (variabler initierade i en funktion)
 - global variabel (variabler initierade utanför en funktion)
- **Moduler**, utökar funktionaliteten för ett program.

Pythonspråket

- **datatyper**, värden som lagras i variabler tillhör en fördefinierad datatyp i python. En variables datatyp kan visas genom att skriva `type(variabelnamn)`.
 - heltal: long, int
 - Flyttal: float
 - Boolesk: bool
 - String: str
- **referens**, vid deklaration av en variabel `bil='Saab'` så är `bil` en **referens** till minnespositionen där **objektet** `'Saab'` finns lagrad. Ett objekt kan ha antal metoder, tex för `kurs='prgo'`
 - `kurs.upper()` -> kurs är nu strängen 'PRGO'
 - `kurs.capitalize()` -> kurs är nu strängen 'Prgo'
- **objekt**, en position i datorns minne som har ett **värde** och en **referens** från en **identifierare**. Kan vara en variabel, funktion eller datastruktur.
- **konstanter**, identifierare för ett värde som inte ska ändras i programmet, dvs hållas konstant. Skiljs smidigt från variabler genom att skriva dem versaler Ex: `PI=3.14`
- **Funktioner**
 - funktionshuvud
 - funktionskropp
 - formella parameter (i funktionshuvudet)
 - anropsparameter (vid anropet)
 - funktion med och utan returvärde
 - variabler som initieras i en funktion kallas **lokala** och är bara tillgängliga inne i funktionen.
- **typkonvertering**

Energiprogrammet

```

122
123 # Energiprogram
124 # Beräknar rörelseenergi och potentiell energi
125 # Mikael Lindahl 11-08-30
126 # rörelseenergi=1/2*m*v^2
127 # potentiell energi=E_pot=m*g*h
128 # m = massa
129 # v = hastighet
130 # g = gravitationskonstant
131 #
132
133 import math      # importera en module

```

Funktion med returvärde, med **funktionshuvud**; `def kinetic_energy(m,v):`, **funktionskropp**; `return 0.5*m*math.pow(v,2)`, **formella parametrar**; `m,v`, **litteraler**; `0,5` och `0,2` och **reserverade ord**; `def` och `return` är

```

140 def kinetic_energy(m,v):
141     return 0.5*m*math.pow(v,2)    # v**2 fungerar också

```

G är **lokal konstant**, stora bokstäver för konstanter

```
145 def potential_energy(m,h):
146     G = 9.82                # tilldelning
147
148     return m*G*h
```

Funktion utan returvärde

```
150 def print_result(m,h,v):
151     print "Bollens rörelseenergi (J) = ", kinetic_energy(m,v);      #
152         utskrift
153     print "Bollens potentiella energi (J)", potential_energy(m,h);
```

Anropsparameter 'Ange bollens massa: '

```
153 svar = input('Ange bollens massa (kg): ')    # Inläsning
```

Ange bollens massa (kg):

Typkonvertering, string->float där m är en **global variabel**

```
155 m = float(svar)
156
157 svar = input('Ange bollens höjd (m): ')

```

Ange bollens höjd (m):

```
158 h = float(svar)
159
160 svar = input('Ange bollens hastighet (m/s): ')

```

Ange bollens hastighet (m/s):

```
161 v = float(svar)
162
163 print_result(m,h,v)    # Skriv ut resultat

```

Bollens rörelseenergi (J) = 0.5
 Bollens potentiella energi (J) 9.82

```
164
165 #print "Bollens rörelseenergi = ", kinetic_energy(m,v);
166 #print "Bollens potentiella energi", potential_energy(m,h);
```

Programutveckling schampometoden

1. Skärmen: Rita hur den ser ut efter en lyckad körning. Där ser man all in- och utmatning som gjorts.
2. Algoritmen: Skriv hur man skulle lösa problemet utan dator. I enkla fall är algoritmen en formel.
3. Minnet: Skriv upp variabelnamn och vilka värden dom tildelas i ditt körexempel.
4. Procedurer: Varje deluppgift får en egen procedur eller funktion. Se dom som medarbetare, specialiserade på var sin uppgift. Oppifrån Skriv nu programsatserna oppifrån och ner.

Schampometoden för vårt exempel

1. Skärmen: Lyckad körning:
 - Ange bollens massa: 1
 - Ange bollens höjd: 1
 - Ange bollens hastighet: 1
 - Bollens rörelseenergi = 0.5
 - Bollens potentiella energi = 9.82
2. Algoritmen: Lösa problemet utan dator:

- För en viss massa, höjd och hastighet räkna ut rörelseenergi och potentiell energi utifrån deras formler

3. Minnet: Variabel namn från körexemplet

- m för massan
- h för höjd
- v för hastighet

4. Procedurer:

- Läsa in massa
- Läsa in höjd
- Läsa in hastighet
- Beräkna rörelseenergi
- Beräkna potentiell energi
- Skriv ut resultatet