

Övning 2

```
0 #-*-decoding:latin-1 -*-
```

Sammanfattning

Idag fick vi svar på frågorna; Hur gör man för att hoppa över en sats och hur gör man för att upprepa en sats flera gånger? För att hoppa över en sats användes **if**, **elif** eller **else** och för att upprepa en sats används **for** eller **while** slinga/loop. Vi gick igenom logiska operatorer **and (&)**, **or (!)** och ****not** samt samt jämförelseoperatorer **==**, **!=**, **<**, **<=**, **>**, **>=**. Vi lekte med dessa genom att bl.a. skriva ett **rektangel**, ett **palindrom** och ett **ränteprogram**.

Organisationer som använder python

Från <http://wiki.python.org/moin/OrganizationsUsingPython>

- **Yahoo Maps** Yahoo acquired Four11, whose address and mapping lookup services were implemented in Python. Yahoo Maps still uses Python today, as can be seen by examining its URLs.
- **Google** Many components of the Google spider and search engine are written in Python (mentioned on Slashdot).
- **Battlefield 2** Battlefield 2 uses python to implement core elements of the gameplay such as score keeping and team balancing. Check out the BF2 Tech Wiki for sample BF2 Python scripts and information about API.
- **Civilization 4** The new addition to the legendary strategy game series has all its inner logic, including AI, implemented in Python. The interface is exposed, so the modding community can easily take advantage of it.
- **NASA** Johnson Space Center uses Python in its Integrated Planning System as the standard scripting language. Efforts are underway to develop a modular collection of tools for assisting shuttle pre-mission planning and to replace older tools written in Perl and shell dialects. Python will also be installed in the new Mission Control Center to perform auxiliary processing integrated with a user interface shell. Ongoing developments include an automated grammar based system whereby C++ libraries may be interfaced directly to Python via compiler techniques. This technology can be extended to other languages in the future.
- **SMHI (Swedish Meteorological and Hydrological Institute)** SMHI is the home of the Swedish civilian weather, hydrological and oceanographic services. Its Python-based remote sensing software for automatic product generation, using NOAA and Meteosat data, provides information to bench forecasters, objective analysis schemes, and commercial interests such as the media. This system receives raw data from several weather satellites, and produces images for many purposes, including the satellite images used for the presentation of the daily weather on Swedish TV 4.
- **Nokia** Nokia is using Python to provide high level programming environment for their S60 (Symbian) and Maemo (Linux) platforms
- **IBM** IBM East Fishkill is using Python to create the business practice logic for factory tool control applications. These are used to control material entry, exit and data collection at the Micrus semiconductor plant in East Fishkill. Code that had been previously written in an internal, proprietary scripting language has been ported to Python.
- **CIA** The site of the USA Central Intelligence Agency (CIA) is powered by Python/Zope/Plone.

Innehåll

- **Datotypen bool**
 - **True** / 1
 - **False** / 0

- **Jämförelseoperatorer**

- == lika med
- != skillt från
- < mindre än
- <= mindre eller lika med
- > större än
- >= större eller lika med

- **Logiska operatorer**

- **and (&)** och
- **or (|)** eller
- **not** icke

- **Villkorssatsen if**

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

- **Slingor: while och for**

- **while(villkor):**, **break**, En while-slinga upprepar ett antal satser så länge som ett villkor är uppfyllt. break bryter slingan
- **for i in [1,2,5]:** En for-slinga upprepar ett antal satser för varje element i en följd.

Logiska operatorer

- **not** True = False
- **not** False = True
- True **and** True = True
- True **and** False = False
- False **and** True = False
- False **and** False = False
- True **or** True = True
- True **or** False = True
- False **or** True = True
- False **or** False = False

```
121
122 # Övning 2
123 # Programexempel för if, for och while
124 # Mikael Lindahl 110906
```

INMATNING, Bra exempel på hur man kan göra menyval med ****while****.

1. Skriv in ett tal mellan 0-4.
2. Säg till om det inte var mellan 0-4.
3. Säg om det var ett
4. Säg om det inte var två,
5. Säg om det var två.

Variabler: svar, tal

```
136 svar=input("Skriv ett tal mellan 0-4:")
```

Skriv ett tal mellan 0-4:

```

137 tal=int(svar)
138 while not (0<=tal<=4):      # slinga, observera att tal måste vara int/float
139
140     print("Felaktigt tal")
141     tal=input("Skriv ett tal mellan 0-4:")

```

IF-sats, vilkorsats

```

143
144 # Jämföra heltal
145 if tal==1:
146     print("Du valde ett!")
147 elif (tal!=2): # Observera att eftersom vi använder elif istället
148
149                 # för if kommer den inte komma hit om tal==1!
150     print("Du valde inte två!")
151 else:
152     print("Du valde två!")

```

Du valde två!

Jämföra strängar med avseende på likhet

```

154 prognos='moln'
155 if prognos=='moln':
156     print('Det blir molning')

```

Det blir molning

Men även alfabetisk ordning

```

159 ord='apa'
160 if ord < 'b':
161     print(ord + ' börjar på a')

```

apa börjar på a

Skriva ut en rektangel. Exempel på **for-loop**.

1. Ge antal rader, n,
2. Ge antal kolumner, m.
3. Skriv ut en rektangel av stjärnor med n rader och m kolumner.

Variabler: kol, rad

```

177 def rektangel(rad, kol):
178     for i in range(rad):
179         s = ''
180         for j in range(kol):
181             s = s + '*'      # Adderar strängar
182
183         print(s)
184
185 svar = input('Hur många rader? ')

```

Hur många rader?

```

179 rad = int(svar)
180 print(rad, ' rader')

```

(3, ' rader')

```

181 svar = input('Hur många kolumner? ')

```

Hur många kolumner?

```
182 kol = int(svar)
183 print(kol, ' kolumner')

(4, ' kolumner')
```

```
184 rektangel(rad, kol)
```

```
****
****
****
```

Samma fast med globala variabler i rektangel

```
194 def rektangel_global():
195     global rad, kol
196     for i in range(rad):
197         s = ''
198         for j in range(kol):
199             s = s + '*'      # Adderar strängar
200
201         print(s)
202
203 svar = input('Hur många rader? ')
```

Hur många rader?

```
196 rad = int(svar)
197 print(rad, ' rader')

(4, ' rader')
```

```
198 svar = input('Hur många kolumner? ')
```

Hur många kolumner?

```
199 kol = int(svar)
200 print(kol, ' kolumner')

(5, ' kolumner')
```

```
201 rektangel_global()
```

```
*****
*****
*****
*****
```

Kontrollerar palindrom. Avänder **for-loop** och **while-slinga**. Exemple: anna, otto, ebbe, bob, hannah

1. Fråga efter palindrom
2. Säg om det är ett palindrom
3. Fråga igen
4. Avsluta med retur

Variabler: ord, s (sträng), p (boolean)

```
214 def palindrom(s):
215     langd = len(s)
216     # s.lower() # Hanterar stor bokstav
217     p = True
218     for i in range(int(langd/2)):
219         if s[i] != s[langd-i-1]:
220             p = False
221             break
222     return p
223
```

```

224
225
226 ord = 'Vadsomhelst'
227 while ord != '':
228     ord = raw_input('Ange ett ord, avluta med endast retur: ')
229     if ord == '':
230         break
231     if palindrom(ord):
232         print( ord + ' är ett palindrom' )
233     else:
234         print( ord + ' är ej ett palindrom' )

```

Ange ett ord, avluta med endast retur: olasalo är ett palindrom

Ange ett ord, avluta med endast retur: fisk är ej ett palindrom

Ange ett ord, avluta med endast retur:

Sparprogram. Anropas fel.

1. Funktion som returnerar hur mycket pengar har jag efter x år och med y ränta och z kr från start.

Ränta: 1 %, 1000 kr från början

1år: $(1000+1000*0.01) = 1000*(1+0.01)=1000*1.01$ **2år:** $1000*1.01+1000*1.01*0.01 = 1000*0.01*(1+0.01) = 1000*1.01^2$ **xår:** $1000*1.01^x$

Variabler: amount, time, interest

```

248
249 import math
250 def savings(amount, time, interest):
251     #time in years
252     #interest in percent
253     if time < 0:
254         print( "error" )
255     else:
256         return amount*math.pow(1+interest/100.0,time)
257
258 print( savings(1000, 20,5) )

```

2653.29770514

```

257 #2653.29770514

```

Hitta felet

```

282 '''
283 print( "hello )
284
285
286 i = 0
287 while i < 10
288     i = i + 1
289
290 i = 0
291 while i < 10 :
292     i = i + 1
293     print i
294
295 "a" = 5
296
297 if True :
298     print( 1 )
299 else if True :

```

```
300     print( 2 )
301     '''
```