

if, elif, else

Programmering 1

16 september 2021

Repetition

if

Om vi ville att något skulle hända enbart vid om ett specifikt villkor använder vi `if`-satser:

```
1 tal = input("Skriv ett tal ")
2 tal = int(tal)
3
4 if tal == 3:
5     print("Bra tal")
```

Repetition

and

Om man ville att flera villkor var uppfyllda kunde man skriva så här:

```
1 namn = input("Ditt namn? ")
2 födelseår = input("När är du född? ")
3 födelseår = int(födelseår)
4
5 if namn == "jesus" and födelseår == 0:
6     print("Säger du det?")
```

Repetition

or

Om man vill minst ett av flera villkor ska vara uppfyllda skriver man:

```
1 år = input("Vilket år är det?")
2 år = int(år)
3
4 if år >= 1769 or år <= 1821:
5     print("Då lever Napoleon.")
```

Boolsk algebra

1 och 0

När vi har ett villkor kan två saker hända:

1. Villkoret är uppfyllt, 1
2. Villkoret är inte uppfyllt, 0

Boolsk algebra

1 och 0

När vi har ett villkor kan två saker hända:

1. Villkoret är uppfyllt, 1
2. Villkoret är inte uppfyllt, 0

Det är likadant i datorn, men Python visar det som **True** och **False**:

```
1 a = input("Tal? ")
2 b = input("Tal? ")
3 a = int(a)
4 b = int(b)
5 c = a < b
6 print(c)
```

Boolsk algebra

Komplicerade villkor

Om du har ett komplicerat villkor:

```
1 if år < 1800 and år%3 == 0 or år > 1000 and år <
   1800:
2     print("?")
```

Vi kan se att för att villkoret ska vara sant så måste **år** vara mindre än 1800, sen kan det antingen vara större än 1000 eller jämnt delbart på 3.

Boolsk algebra

Komplicerade villkor

$A = \text{år} < 1800, B = \text{år} \% 3 == 0, C = \text{år} > 1000$

$A_B_C_A$

Hela uttrycket ska bli antingen 1 eller 0. Vad kan vi sätta på strecken?

Boolsk algebra

Komplicerade villkor

$A = \text{år} < 1800, B = \text{år} \% 3 == 0, C = \text{år} > 1000$

$A_B_C_A$

Hela uttrycket ska bli antingen 1 eller 0. Vad kan vi sätta på strecken?

Vi kan översätta **and** till multiplikation och **or** till addition!

Boolsk algebra

Komplicerade villkor

$$A \cdot B + C \cdot A$$

Det uttrycket kan vi förenkla:

$$A \cdot (B + C)$$

Det uttrycket kan vi översätta till kod:

```
1 if år < 1800 and (år%3 == 0 or år > 1000):
2     print("?")
```

Boolsk algebra

Räkna fram ett villkor

Om vi har ett långt villkor:

```
1 if x == y and x < z or y == z and (x == z or x
   == y) or x < z:
2     print("JA!")
```

Så kan vi räkna ut om talen $x = 3, y = 3, z = 6$ uppfyller kravet:

$$x == y : 1$$

$$x < z : 1$$

$$y == z : 0$$

$$x == z : 0$$

$$1 \text{ and } 1 \text{ or } 0 \text{ and } (0 \text{ or } 1) \text{ or } 1$$

Boolsk algebra

Räkna fram ett villkor

1 and 1 or 0 and (0 or 1) or 1

Vi översätter till addition och multiplikation:

$$1 \cdot 1 + 0 \cdot (0 + 1) + 1$$

$$1 \cdot 1 + 0 \cdot 1 + 1$$

$$1 + 0 + 1$$

Boolsk algebra

Räkna fram ett villkor

1 and 1 or 0 and (0 or 1) or 1

Vi översätter till addition och multiplikation:

$$1 \cdot 1 + 0 \cdot (0 + 1) + 1$$

$$1 \cdot 1 + 0 \cdot 1 + 1$$

$$1 + 0 + 1$$

$$= 1$$

Villkoret är alltså uppfyllt och vi skriver ut JA!

Else

Else



ALWAYS BE
YOURSELF
— — — — —
EVERYONE
ELSE
SUCKS



Else

Annars

Än så länge har vi pratat om vad som händer om ett villkor är uppfyllt.

Men om vi vill göra något särskilt när det inte är uppfyllt, hur gör vi då?

Else

Annars

Än så länge har vi pratat om vad som händer om ett villkor är uppfyllt.

Men om vi vill göra något särskilt när det inte är uppfyllt, hur gör vi då?

```
1 tal = input("Skriv en trea: ")
2 tal = int(tal)
3
4 if tal == 3:
5     print("Bra")
6 if tal != 3:
7     print("Dåligt")
8
9 print("Visst har vi roligt?")
```


Else

Else

Ett snyggare sätt att skriva är genom att använda `else`

```
1 tal = input("Skriv en trea: ")
2 tal = int(tal)
3
4 if tal == 3:
5     print("Bra")
6 else:
7     print("Dåligt")
8
9 print("Visst har vi roligt?")
```

`else` tar hand om ALLA fall när villkoret över inte är uppfyllt.

else

else if

```
1     if tal < 10:  
2         print("Ental")  
3 if tal >= 10 and tal < 100:  
4     print("Tiotal")  
5 if tal >= 100 and tal < 1000:  
6     print("Hundratal")
```

else

else if

```

1     if tal < 10:
2         print("Ental")
3     if tal >= 10 and tal < 100:
4         print("Tiotal")
5     if tal >= 100 and tal < 1000:
6         print("Hundratal")
    
```

```

1     if tal < 10:
2         print("Ental")
3     else:
4         if tal < 100:
5             print("Tiotal")
6         else:
7             if tal < 1000:
8                 print("Hundratal")
    
```

elif

else if

```
1 if tal < 10:  
2     print("Ental")  
3 else:  
4     if tal < 100:  
5         print("Tiototal")  
6     else:  
7         if tal < 1000:  
8             print("Hundratal")
```

elif

else if

```

1 if tal < 10:
2     print("Ental")
3 else:
4     if tal < 100:
5         print("Tiotal")
6     else:
7         if tal < 1000:
8             print("Hundratal")
    
```

```

1 if tal < 10:
2     print("Ental")
3 elif tal < 100:
4     print("Tiotal")
5 elif tal < 1000:
6     print("Hundratal")
    
```

elif

else if

```
1 if tal < 10:
2     print("Ental")
3 elif tal < 100:
4     print("Tiotal")
5 elif tal < 1000:
6     print("Hundratal")
```

- ▶ Här kommer den först att kolla om det första villkoret är uppfyllt.
- ▶ Är det inte uppfyllt kollar det om `tal < 100`
- ▶ Och om det inte är uppfyllt kollar det om `tal < 1000`

Säg att `tal` är 5. Då är alla villkor uppfyllda, men programmet kommer bara att gå in i den första `if`-satsen

Övningar

Blad 1

Utgå från filen `if-else.py` i vklass.

1. Det finns en logisk bugg i raderna 12–15 . Rätta till den.
2. "Tripp trapp trull" sekvensen visar hur det hade sett ut utan `elif`. Ändra den så att den fungerar likadant fast med `elif`.
3. Skriv ett program som kollar om ett ord är skrivet med bara versaler, gemener eller om det är blandat.
4. Skriv ett program som tar emot ett årtal (efter 1753) och returnerar om det var ett skottår eller ej. Tänk på att år jämnt delbara med 100 inte är skottår, men att år delbara med 400 är fortfarande skottår. (1800 och 1900 ej skottår men 2000 och 2400 skottår.)

Övningar

Saxade övningar 1

En operatörför mobiltelefoni erbjuder tre olika abonnemang: *Kontant*, *Normal* och *Plus*. Om man jämför villkoren för abonnemangen visar det sig att *Kontant* är billigast om man ringer högst 33 minuter per månad, *Normal* lönar sig bäst om man ringer mellan 33 och 36 minuter per månad och *Plus* är mest förmånligt om man ringer ännu mer. Skriv ett program som läser in antalet minuter man uppskattar att man kommer att ringa per månad. Programmet ska tala om vilket abonnemang man bör välja.

Övningar

Saxade övningar 2

I en triangel kan man beteckna sidorna med a , b och c . Om man känner till längderna för sidorna a och b samt vinkeln α mellan dessa sidor så kan man räkna ut längden av sidan c med formeln:

$$c = \sqrt{a^2 + b^2 - 2ab \cos(\alpha)}.$$

Skriv ett program som läser in längdenerna på två sidor i en triangel och vinkeln mellan sidorna. Programmet ska avgöra om triangeln är *liksidig* (alla sidor lika), *likbent* (två sidor lika) eller *oliksidig* (inga sidor lika).

Python jobbar i radianer: α grader $\Rightarrow \frac{\pi}{180} \cdot \alpha$ radianer