ValIf

August 13, 2023

1 Att göra val

1.1 Flöde

Ofta hamnar man i en situation där man måste välja bland olika fall vad som ska göras. Tex. om en variabels värde är större än 0 så ska programmet göra på ett sätt men om variabelns värde är mindre än 0 så ska programmet göra något annat. Detta hanteras med så kallade 'if-satser'(om-så-satser).

I figuren nedan illustreras ett vanligt linjärt flöde där satserna följer på varandra uppifrån och ner. Inga alternativ. Python läser rad för rad uppifrån och ner (radnummer) och utför kommandona. Python börjar med att tilldela a värdet 4, sedan beräkna 2 gånger a och tilldela c detta värde osv.

I figuren nedan illustreras en punkt i programmet där vi ska utföra olika satser beroende på värdet på en variabel. Överst i figuren visas ett flödesschema med de logiska alternativen. Nedanför visas motsvarande kod i python.

Se nedersta figuren i bilden ovan. I flödet har jag skrivit in en rad som ska göras då det är sant att `d < 0` och det är satsen `x = x + 1`. Om det inte är sant att `d < 0` så görs det som står i `else` delen dvs `x = x - 1`.

Så om `d < 0` görs bara `x = x + 1`. Även om `x = x - 1` kommer efter `x = x + 1` så utförs den *inte* om `d < 0` är sant; python hoppar.

När python läst antingen `x = x + 1` eller `x = x + 1` (den har bara läst en av dem) så läser den satsen `d = d*2`; den satsen har inget indrag.

Observera att satserna `x = x + 1` och `x = x - 1` har ett indrag (indent). Det är inte av estetiska skäl utan det *måste* vara så. Det är en del av språkets syntax (grammatiska regler, absoluta och måste följas). Detta kan ibland ställa till problem om man kopierar och klistrar in kod; indragen går ibland förlorade. `d = d*2` är utanför if-satsen eftersom den inte har något indrag.

Observera tecknet kolon dvs : Det *måste* vara där.

Man kan också skriva in flera satser; ett kodblock. Python tolkar alla indragna satser som ett *block* eller *svit* av kod; båda uttrycken används i svenskan. Se nästa cell.

[1]:
```
x = 4  # Ändra och prova
if x > 0:
    x = x + 1  # Dessa 3 rader är ett block/en svit
    d = 7
    print("x var större än 0")
else:
```

```
    x = x - 1  # Dessa 3 rader är ett block/en svit
    d = 2*x
    print(d)
print(x) # utanför if-satsen
```

```
x var större än 0
5
```

Det finns också en enklare konstruktion.

[2]:
```
x = int(input("Ange ett heltal"))

if x > 0:
    print(x)
print("Denna skrivs alltid ut")
```

```
3
Denna skrivs alltid ut
```

Anger du tex. 3 så skrivs 3 ut. Anger du tex. -5 så skrivs inte x ut. Däremot skrivs den sista (inte indragen) printsatsen *alltid* ut.

Indragen i python måste vara minst ett mellanslag. Varje rad i ett block av kod måste ha samma storlek på indraget. Men olika block kan ha olika indrag. Dock är seden den att man i regel använder 4 mellanslag och ALDRIG har olika indrag i en kod; alltid heltalsmultiplar av 4.

Stödprogram för att skriva kod har funktioner som hjälper dig i ditt skrivande med att hålla reda på indrag och olika block av kod.

På python.org kan du läsa i PEP 8 om stilregler för kod. PEP står för "Python Enhancement Proposals". När man skriver kommentarer på en rad med kod (inline) så ska det vara 2 mellanslag innan # och sedan ett mellanslag innan texten börjar. Här nedan följer några utdrag från PEP 8. https://www.python.org/dev/peps/pep-0008/

Comments should be complete sentences. The first word should be capitalized, unless it is an identifier that begins with a lower case letter (never alter the case of identifiers!).

An inline comment is a comment on the same line as a statement. Inline comments should be separated by at least two spaces from the statement. They should start with a # and a single space.

1.2 Uppgifter

Uppgift 1

Skriv ett program som beräknar tionde-roten ur ett tal som användaren anger. Om talet inte är större än 0 ska programmet meddela att det inte går och avsluta.

Lösningsförslag

Uppgift 2

Skriv ett program som beräknar $|x - y|$. Användaren anger x och y. Använd inte absolutbeloppsfunktion utan beräkna skillnaden så tecknet alltid är positivt. Testa vilket tal som är störst.

Lösningsförslag

Uppgift 3

Skriv ett program som beräknar $x//y$ där $y \leq x$. Användaren anger x och y. Med en if-sats kontrolleras vilket tal som är störst. Be inte användaren ange största först eller liknande.

Lösningsförslag

1.3 Att göra flera val

Ibland är valmöjligheterna flera och då behövs mer komplicerade strukturer. Vi har tittat på

```
if x > 0:
    x = x + 1
else:  # x=0 och x<0
    x = x - 1
```

och den enklare

```
if x > 0:
    x = x + 1
```

Det finns en konstruktion för upprepade val. Ett `else` skrivs ihop med ett nytt `if` till `elif`. I exemplet nedan har vi 3 fall, inte 2 denna gång:

```
if x > 7:
  print("x är större än 7")
elif x == 7:
  print("x är lika med 7")
else:
  print("x är mindre än 7")
```

Vi använder koden och ber användaren ange ett heltal (och omvandlar strängen i input till ett heltal).

[3]:
```
x = int(input("Ange ett heltal"))

if x > 7:
  print("x är större än 7")
elif x == 7:
  print("x är lika med 7")
else:
  print("x är mindre än 7")
print("klart")
```

```
x är mindre än 7
klart
```

Python tillåter ett obegränsat antal elif men bara ett else.

Exempel på uppdelning i 4 olika fall. Vi gör olika kommandon beroende på om x = 1, eller x = 2, eller x = 3, eller övriga fall:

[4]:
```
x = int(input("Ange ett heltal"))

if x == 1:
    x = x + 1
elif x == 2:
    x = x + 3
elif x == 3:
    x = x - 2
else:
    x = 0

print(x)
```

```
0
```

Se koden nedan. Python besöker bara ett av alternativen sedan lämnar den. Säg att vi matar in 1. Den testar inte fler om ett av fallen inträffat. Om vi först testar om x>0 och den är sann så utförs tillhörande indraget kodblock. Visserligen är också testet x=1 sann, men python testar inte den.

'elif' står för else...if, vilket innebär att utfallet från ett tidigare if måste vara att gå till else(elif) och sedan testas ett nytt if(elif). Så är det första if uppfylld så hamnar man inte på else och kan då inte testa if som finns i denna elif.

Se koden nedan. Om x anges som 1 så utförs första print-satsen. Faktum är att om x>0 så utförs endast första print-satsen. 0 och alla negativa tal gör att sista print-satsen skrivs. Andra printsatsen kommer aldrig att användas. Testa tex. med 1, 5, 0, -5.

[5]:
```
x = int(input("Ange ett heltal"))

if x > 0:
    print("x större än 0")
elif x == 1:
    print("x är lika med 1")
else:
    print("trillade igenom")
```

```
x större än 0
```

Vi kan ha if-satser inuti if-satser. Nedanstående skriver ut att x är större än 0 och om x är 1 så skriver den det också. Om x inte är 1 men större än noll skrivs bara att x är större än 0. Om x är 0 eller negativt så skrivs 'trillade igenom' ut. Testa med 1, 2, 0, -5.

Lägg märke till att skillnader i indrag visar att 'else' tillhör den yttre if-satsen (inget indrag); indrag måste vara av rätt längd.

[6]:
```
x = int(input("Ange ett heltal"))

if x > 0:
    print("x större än 0")
    if x == 1:
```

```
        print("x är 1")
else:
    print("trillade igenom")
```

```
x större än 0
```

I nedanstående har vi 2 else med olika indrag, det markerar vilket if det tillhör.

[1]:
```
x = int(input("Ange ett heltal"))

if x > 0:
    print("x större än 0")
    if x == 1:
        print("x är 1")
    else:
        print("x är inte 1")
else:
    print("trillade igenom")

print("Denna rad är utanför if-satserna, utförs alltid")
```

```
trillade igenom
Denna rad är utanför if-satserna, utförs alltid
```

Utan indrag fungerar det inte.

[2]:
```
x = int(input("Ange ett heltal"))

if x > 0:
print("x större än 0")
if x == 1:
print("x är 1")
else:
print("x är inte 1")
else:
print("trillade igenom")

print("Denna rad är utanför if-satserna, utförs alltid")
```

```
  Cell In[2], line 4
    print("x större än 0")
    ^
IndentationError: expected an indented block after 'if' statement on line 3
```

1.4 Uppgifter

Uppgift 4

Skriv ett program som ber användaren ange sin längd i hela centimetrar. Om längden är större än 172 cm så ska programmet skriva ut “Du är längre än jag är”. Om längden är mindre så ska det skriva ut “Du är kortare än jag är”. Och om längden är 172 cm så ska det skriva ut “Vi är lika långa”.

Lösningsförslag

Uppgift 5

Skriv ett program som undersöker om ett tal är positivt eller negativt eller noll. Om talet är positivt ska det kontrollera om det är jämnt delbart med 5 eller inte; skriv ut slutsatsen. Om talet är negativt ska det kontrollera om talet är jämnt delbart med 2; skriv ut slutsatsen. Om det är 0 ska inget göras mer än att skriva ut att det är talet 0.

Lösningsförslag

Uppgift 6

Skriv ett program som räknar antalet siffror i ett heltal som användaren anger; talet kan vara positivt eller negativt. Om talet är negativt måste minustecknet naturligtvis räknas bort. Antalet element i en sträng ges av len() där strängen anges mellan parenteserna tex. len(“hej”) är 3. Om strängen betecknas med variabeln x skriver man len(x); skriver du len(“x”) får du 1 som svar.

Lösningsförslag