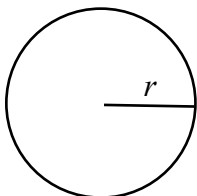


Matematisk förberedelse

Cirkel-kvadrat problemet

Vi betecknar cirkelns radie med r och kvadratens sida med a :



Cirkelns omkrets: $O_{cirkel} = 2 \pi r$

Kvadratens omkrets: $O_{kvadrat} = 4 a$

Båda figurernas omkrets ska vara lika stor, vilket innebär:

$$O_{kvadrat} = O_{cirkel}$$

$$4 a = 2 \pi r$$

$$a = \frac{2 \pi r}{4}$$

Samband mellan r och a :

$$a = \frac{\pi r}{2}$$

Sambandet ovan gäller om och endast om cirkeln och kvadraten har samma omkrets. Vi tar över den till den digitala lösningen, se blåmarkerad kod nedan:

Digital lösning

Circle_Square_1

Öppna Python, mata in följande och tryck efter varje rad på Enter:

```
>>> from math import pi
>>> r = 4
>>> a = pi * r / 2
>>> A_circle = pi * r**2
>>> A_square = a**2
>>> print('Cirkelns area är', A_circle)
Cirkelns area är 50.26548245743669
>>> print('Kvadratens area är', A_square)
Kvadratens area är 39.47841760435743
>>>
```

Lösningen visar:

Cirkelns area är större än kvadratens.

I denna version av lösningen har vi besvarat frågan om cirkeln eller kvadraten får större area när de har samma omkrets endast för specialfallet cirkelns radie $r = 4$.

Men kommer resultatet vara det samma om man väljer andra värden för radien än $r = 4$? Denna version kan inte besvara frågan därför att värdet på r är hårdkodat i programmet. Denna version måste vidareutvecklas så att man kan testa programmet för vilken radie som helst. Detta kräver att vi *generaliserar* koden, t.ex, genom att läsa in värdet på radien. I nästa lektion ska vi lära oss att läsa in data till Python och kombinera detta med att skriva och spara koden i en fil.

Kodförklaring

Circle_Square_1

Första raden

`from math import pi` importerar konstanten π från modulen `math` till vår session så att vi kan komma åt den sedan. Konstanten `pi` är lagrad i modulen `math`. En modul är en samling fördefinierade koder i Python.

Raderna 6 & 8

`print()` skriver ut text och en variabel. Kommat skiljer åt texten från variabeln. Texten omges i koden av `' '` som är apostrofer. Variablerna `A_circle` och `A_square` däremot som är beräknade och tilldelade värden innan, står med sitt namn utan apostrofer, skilda med komma.

if-satsen

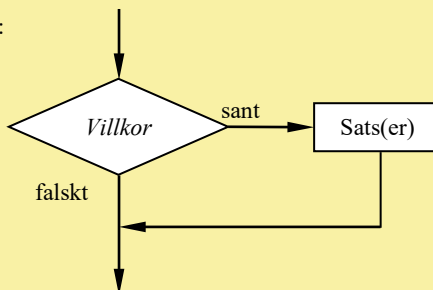
Övningar 1301
1305-1307

`if villkor :`
Sats(er)

Om villkoret är sant utförs en eller flera satser. Är villkoret falskt, görs ingenting. Ett *val* mellan ett alternativ eller ingenting (*enkel selektion*).

Indragning(ar) avgör satsernas räckvidd.

Flödesschemat visar `if`-satsens logik:



Frågor

1. Kör koden `Circle_Square_1` (sid 18) utan den första raden `from math import pi`. Tolka Pythons felmeddelande.
2. Varför definieras i koden `Circle_Square_1` kvadratens sida `a` till uttrycket `pi * r / 2` medan cirkelns radie `r` sätts godtyckligt till 4?
3. Vilket resultat får man om man kör `Circle_Square_1` med andra värden på `r` än 4? Testa med några i Python. Tror du att resultatet är oberoende av `r`-värdena?
4. Varför blir det fel om man i `print`-satsen utelämnar kommat i parentesen?
5. Sätt i `print`-satsen `A_circle` och `A_square` inom apostrofer och kör. Tolka resultatet.
6. Varför står i `print`-satsen apostrofer kring den första, men inte kring den andra *parametern* (delar i `print()`-satsens parentes åtskilda med komma)?

Övningar

1301

1. Ersätt i koden `Circle_Square_1` (sid 18) den första raden `from math import pi` med:

```
import math
```

2. Ersätt den tredje raden med:

```
a = math.pi * r / 2
```

3. Ersätt de två `print`-satserna var med följande `if`-sats:

```
if A_circle > A_square :  
    print('CirkeIn är störst.')
```

```
if A_circle < A_square :  
    print('Kvadraten är störst.')
```

Gör andra ändringar som behövs. Testa den nya varianten av `Circle_Square_1` med olika värden på `r`.

1302

Lös ut `r` ur sambandet mellan `r` och `a` (sid 17). Modifiera koden `Circle_Square_1` genom att ge först `a` ett värde. Använd sedan det nyvunna sambandet för att definiera `r`. Jämför resultaten för olika värden på `a` och med tidigare.

1303

Modifiera koden `Circle_Square_1` så att den besvarar frågan: Hur många hela procent är cirkelns area större än kvadratsens? Kör koden för olika värden på `r`. Påverkar detta procentsatsen?

1304

Lös **Cirkel-kvadrat** problemet (sid 17) generellt: Ställ upp ett algebraiskt uttryck för förhållandet mellan figurernas areor, t.ex.:

$$A_{\text{cirkel}} / A_{\text{kvadrat}}$$

- a) Ange kvoten *exakt* genom att både under uträkningen och i resultatet bibehålla π som bokstav och använda bråk istället för decimaltal.
- b) Vilken generell slutsats kan man dra av resultatet? Är slutsatsen beroende av figurernas storlek?
- c) Ange svaret i hela procent. Jämför resultatet med Pythons svar i övn 1303.

Hittills har vi använt Python både som smart och programmerbar kalkylator (sid 9 & 13). I båda fall har vi kört Interactive mode vars nackdel dock är att koden är borta för gott så snart man lämnat Python. Man har inte längre kvar koden varken för att testa den igen eller för att utveckla den vidare med nya idéer som man kommer senare på. I längre program blir det här arbetssättet ohållbart. Att varje gång behöva mata in den gamla koden för att lägga till en liten förbättring, är slöseri med tiden, dessutom jätte tråkigt. I sådana fall vill man ha möjligheten att spara koden i en fil för att återanvända och vidareutveckla den.

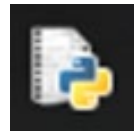
En miljö där koden kan både skrivas, sparas i en fil och köras är Pythons **IDLE** som står för *Integrated Development and Learning Environment* och automatiskt följer med när man installerar Python enligt instruktionerna i appendix **A** (sid 101).

Öppna Pythons IDLE genom att klicka på ikonen →

Den ser lite annorlunda ut än pythonikonen på sid 9

och ligger ofta bredvid den. Följande vitt fönster öppnas

som är IDLEs *kommandofönster*, även kallat Shell:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 3 Col: 4

Gå till menyraden längst upp och klicka på menyn File och sedan på undermenyn New File: **File → New File**

Ett annat vitt, tomt fönster öppnas, IDLEs *editfönster*:

```
Untitled
File Edit Format Run Options Window Help
```

Ln: 1 Col: 0

Mata in koden på nästa sida (utan radnumren) i *editfönstret* ovan:

```
1 # Circle_Square_2.py
2 # Läser in cirkelns radie, beräknar kvadratens sida och
3 # båda figureernas area samt jämför dem med varandra
4
5 import math # Importerar modulen math
6 r = int(input('\n\tMata in cirkelns radie:\t')) # Inläsning
7 a = math.pi * r / 2 # Kvadratens sida, sid 17
8
9 A_circle = math.pi * r**2 # Cirkelns area
10 A_square = a**2 # Kvadratens area
11
12 if A_circle > A_square :
13     print('\n\tCirkelns area är större än kvadratens.\n')
14 else :
15     print('\n\tKvadratens area är större än cirkelns.\n')
```

OBS! Endast av estetiska skäl har vi avbildat editfönstrets kod i en helt annan grafisk form än i det vita, lite tråkiga **IDLE**-editfönstret som visas på förra sidan. Dessutom tillämpas radnumrering för att kunna hänvisa till kod som ska förklaras, se **Kodförklaring** på nästa sida.

Spara koden i en fil

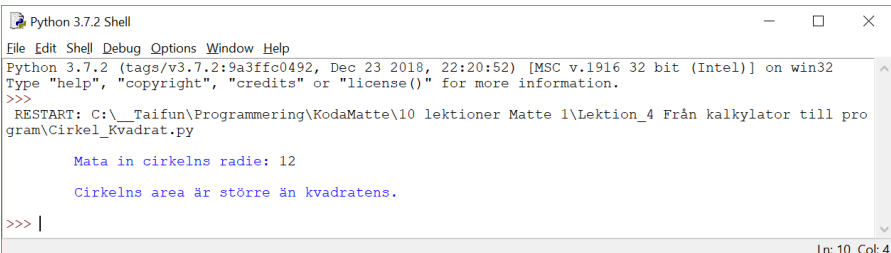
Välj i **editfönstrets** menyrad **File** → **Save As...**, välj lämplig plats på din dator, ange filnamnet **Circle_Square_2.py** och klicka på **Spara**.

Exekvera programmet

Välj i **editfönstrets** menyrad:

Run → **Run Module**.

Du borde nu se följande utskrift i **kommandofönstret** efter du matat in ett värde för **r** :



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\__Taifun\Programmering\KodaMatte\10 lektioner Matte 1\Lektion_4 Från kalkylator till program\Cirkel_Kvadrat.py
Mata in cirkelns radie: 12
Cirkelns area är större än kvadratens.
>>> |
```

Ln: 10 Col: 4

Här har vi återvänt och visar igen IDLE:s *kommandofönster* i originalskick (Shell) som vi började med när vi startade IDLE (sid 21).

Circle_Square_2

Kodförklaring

Raderna 1-3, 5, 7, 9-10

Tecknet # inleder i Python en *radkommentar*. Kommentarer utförs inte av Python utan ska endast förklara koden.

Rad 6

Koden `input(...)` läser in användarens inmatning. Koden `int(...)` omvandlar det inlästa till ett heltal som tilldelas variabeln `r`, cirkelns radie.

Raderna 6, 13, 15

Koden `\n` (newline) genererar radbyte i utskriften. `\t` (tabulator) skapar horisontellt avstånd.

På Mac-datorer kan man få tecknet backslash (\) genom tangentkombinationen `<alt>-<shift>-</>`, dvs / utan att släppa tangenterna `<alt>` och `<shift>`.

Tecknen ' ' är apostrofer. I pythonkod måste text alltid omges av apostrofer. Själva tecknet ' är endast kod och kommer inte att visas i utskriften.

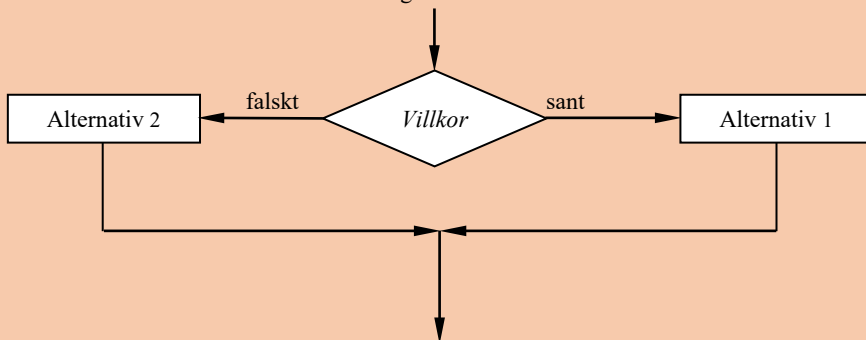
Raderna 12-15

```
if villkor :  
    Alternativ 1  
else :  
    Alternativ 2
```

Ett s.k. *tvåvägsval* mellan två alternativ som utesluter varandra. Beroende på om *villkoret* är sant eller falskt utförs ett av alternativen.

Indragningarna avgör alternativens räckvidd. (raderna 13, 15).

Flödesschemat visar *if-else*-satsens logik:



Frågor

1. Vad är skillnaden mellan att köra Python i Interactive mode och att använda **IDLE**?
2. Vad är skillnaden mellan **IDLE**:s kommando- och editfönster? I vilket skrivs koden?
3. Går det i **IDLE** att skriva pythonkod och exekvera den utan att spara den i en fil? Om ja, gör det. Om nej, förklara när det inte går.
4. Kan man skriva om i programmet **Circle_Square_2** (sid 22) satsen i rad **6** som läser in cirkelns radie, till två separata satser? Om ja, vilka satser blir det då?
5. Varför blir det fel om man i rad **6** av **Circle_Square_2** tar bort koden `int(...)`?
6. Är det möjligt att i raderna **12-15** av programmet **Circle_Square_2** ersätta **if-else**-satsen med två **if**-satser? Om ja, gör det.

Övningar

1401

Öppna Pythons **IDLE** eller din favorit utvecklingsmiljö, mata in koden till programmet **Circle_Square_2** (sid 22), spara och exekvera den. Kör flera gånger med olika värden på cirkelns radie.

Har du ingen favorit pythonmiljö, ladda ner och installera Python enligt appendix **A** (sid 101). Då följer **IDLE** med. Det går på 5 min. Skriv, spara och exekvera **Circle_Square_2** enligt instruktionerna på sid 21.

1402

Skriv ett pythonprogram i en fil så att du får följande utskrift:

```
Välkommen till
Koda matte med Python!
Programmering i matematik.
```

Spara filen och exekvera programmet.

1403

Det duala **Cirkel-kvadrat**-problemet:

En cirkel och en kvadrat har *samma area*. Vilken av dem har den *minsta omkretsen*?

- a) Ställ upp ett samband mellan cirkelns radie r och kvadratens sida a . Använd sambandet i ett pythonprogram liknande **Circle_Square_2** (sid 22). Läs in ett exempel, beräkna figurernas omkrets och jämför dem med varandra. Kör programmet för andra exempel. Besvara frågan ovan.
- b) Lös problemet generellt: Ställ upp ett algebraiskt uttryck för förhållandet mellan figurernas omkretsar:

$$O_{\text{cirkel}} / O_{\text{kvadrat}}$$

Svara *exakt*, se övn **1304**

- c) Beräkna den procentuella skillnaden mellan figurernas omkretsar.