

# Introduktion till informationsteknologi (1DT051)

## Programmering (3) - workshop

Vad menas med ett objekt?

Vad menas med en klass?

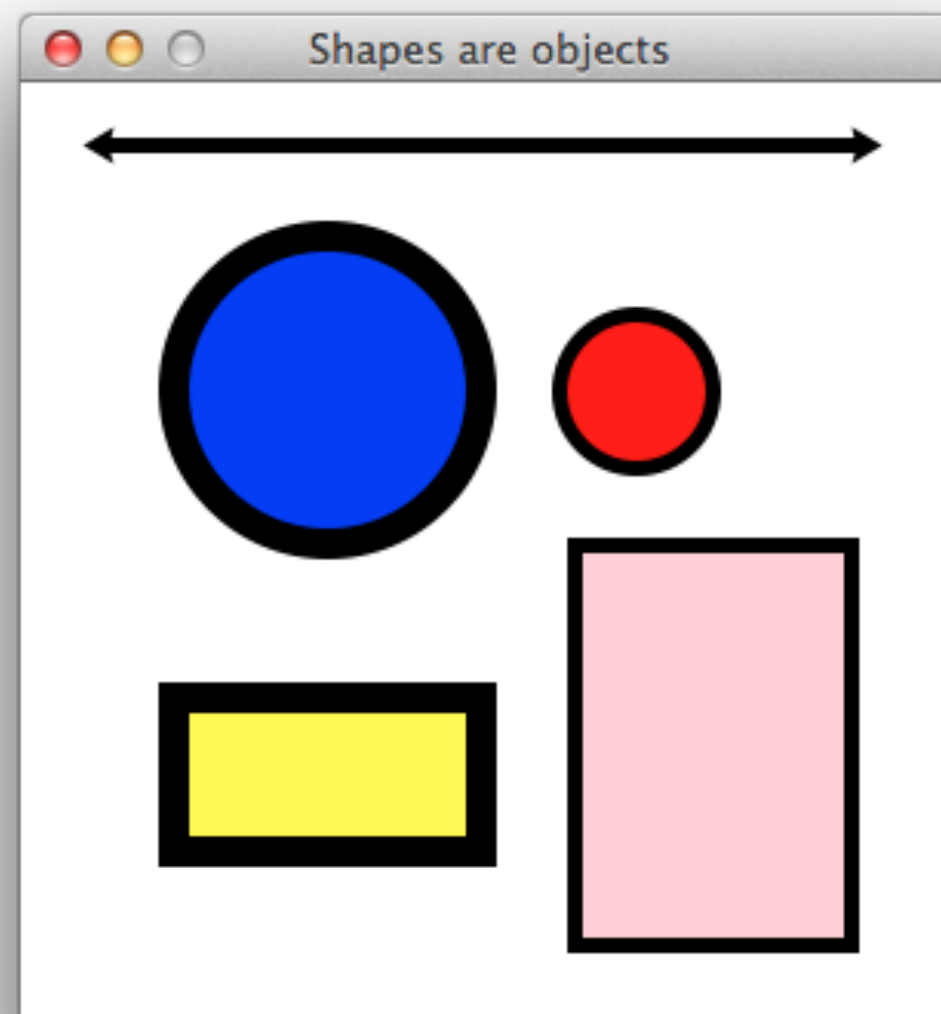
Vad är sambandet mellan objekt och klass?

I Python är listor objekt

september/oktober 2016

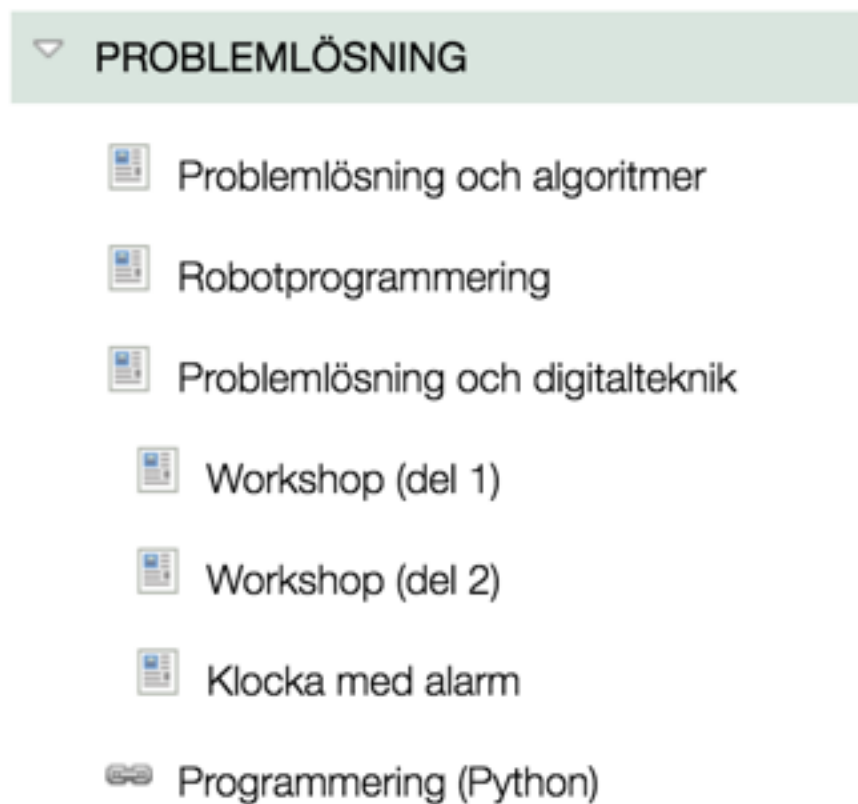
Uppsala universitet

karl.marklund@it.uu.se



# 1 skärm = ett par ur basgruppen

Vilket par du tillhör framgår på kursens sida i **Studentportalen** ➤ **Problemlösning** ➤ **Programmering (Python)** ➤ **Del 3** ➤ **Syfte och översikt** ➤ **Arbete och redovisning i par om två studenter.**



**Introduktion till informationsteknologi (1DT051)**








**Programmering (3) - workshop**

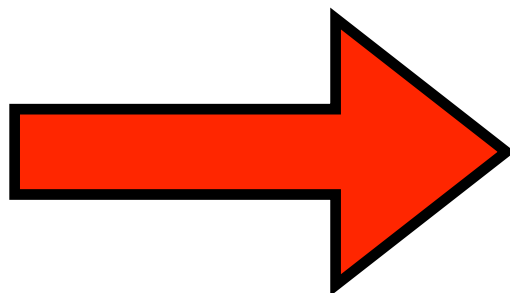
# Förberedelser

(1)

Från kursens sida i Studentportalen, öppna hemsidan om problemlösning och programmering.

## ▼ PROBLEMLÖSNING

-  Problemlösning och algoritmer
-  Robotprogrammering
-  Problemlösning och digitalteknik
-  Workshop (del 1)
-  Workshop (del 2)
-  Klocka med alarm
-  Programmering (Python)



## Introduktion till informationsteknologi

1DT051/2016

## Problemlösning och programmering

## Start

### Syfte och översikt

Syfte

Förkunskaper

Förberedelser

Innehåll

Arbete i par om två studenter

Workshops

# Problemlösning och programmering

¶

På kursen **Introduktion till informationsteknologi (1DT051)** ingår momentet problemlösning och programmering. All information om detta moment hittar du på dessa sidor.

## Syfte

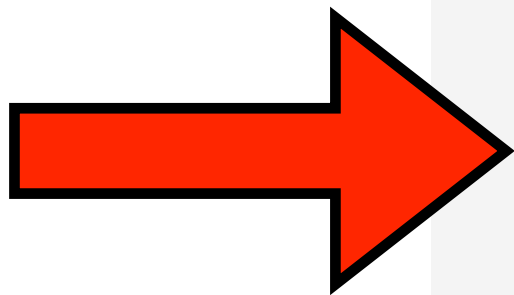
¶

Syftet med detta moment är att se hur metoden **söndra och härska** kan användas för att lösa problem med hjälp av programmering. Du får prova på och lära sig grunderna inom programmering och då särskilt i det **imperativa** programmeringsspråket **Python**.

# Förberedelser

(2)

Navigera till Del 3 ➤ Syfte och översikt



## Del 3

### Syfte och översikt

Syfte

Mål

Metod

Förkunskaper

Arbete och redovisning i par ...

Workshop

Handledning

# Introduktion till informationsteknologi (1DT051)

## Programmering (3) - workshop

Vad menas med ett objekt?

Vad menas med en klass?

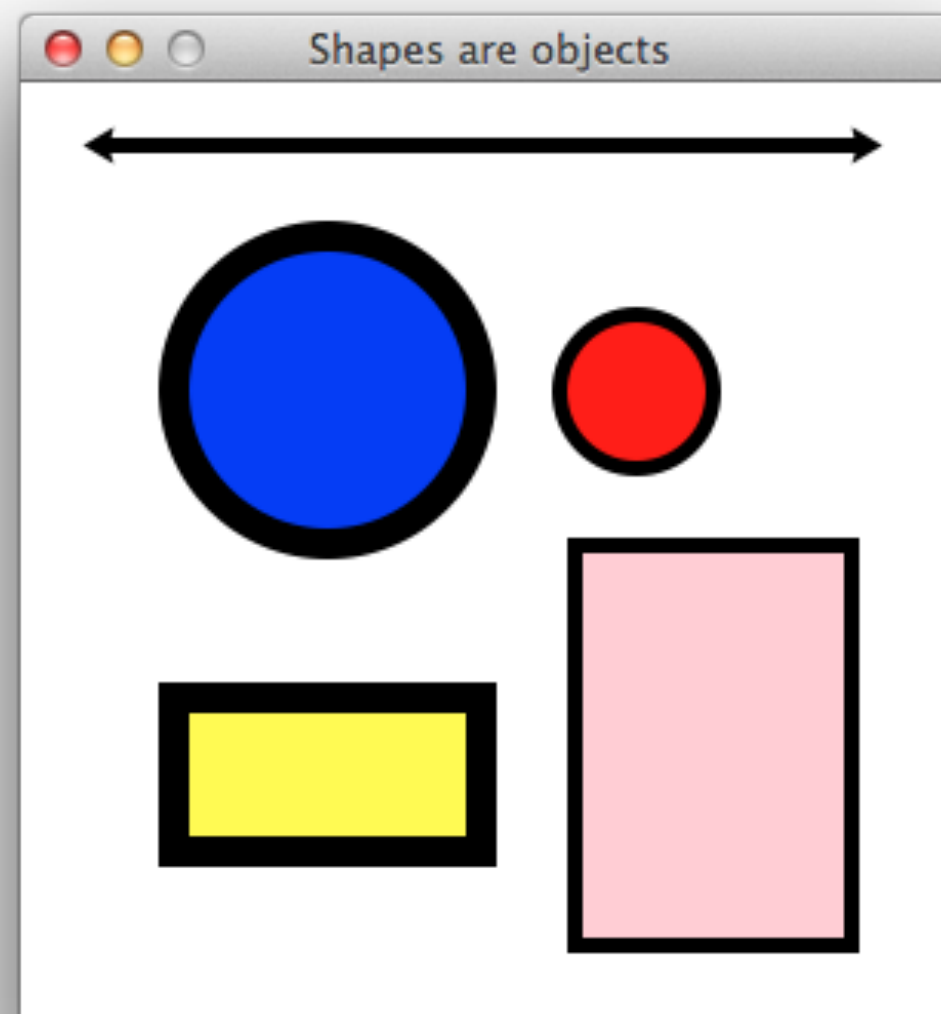
Vad är sambandet mellan objekt och klass?

I Python är listor objekt

september/oktober 2016

Uppsala universitet

karl.marklund@it.uu.se

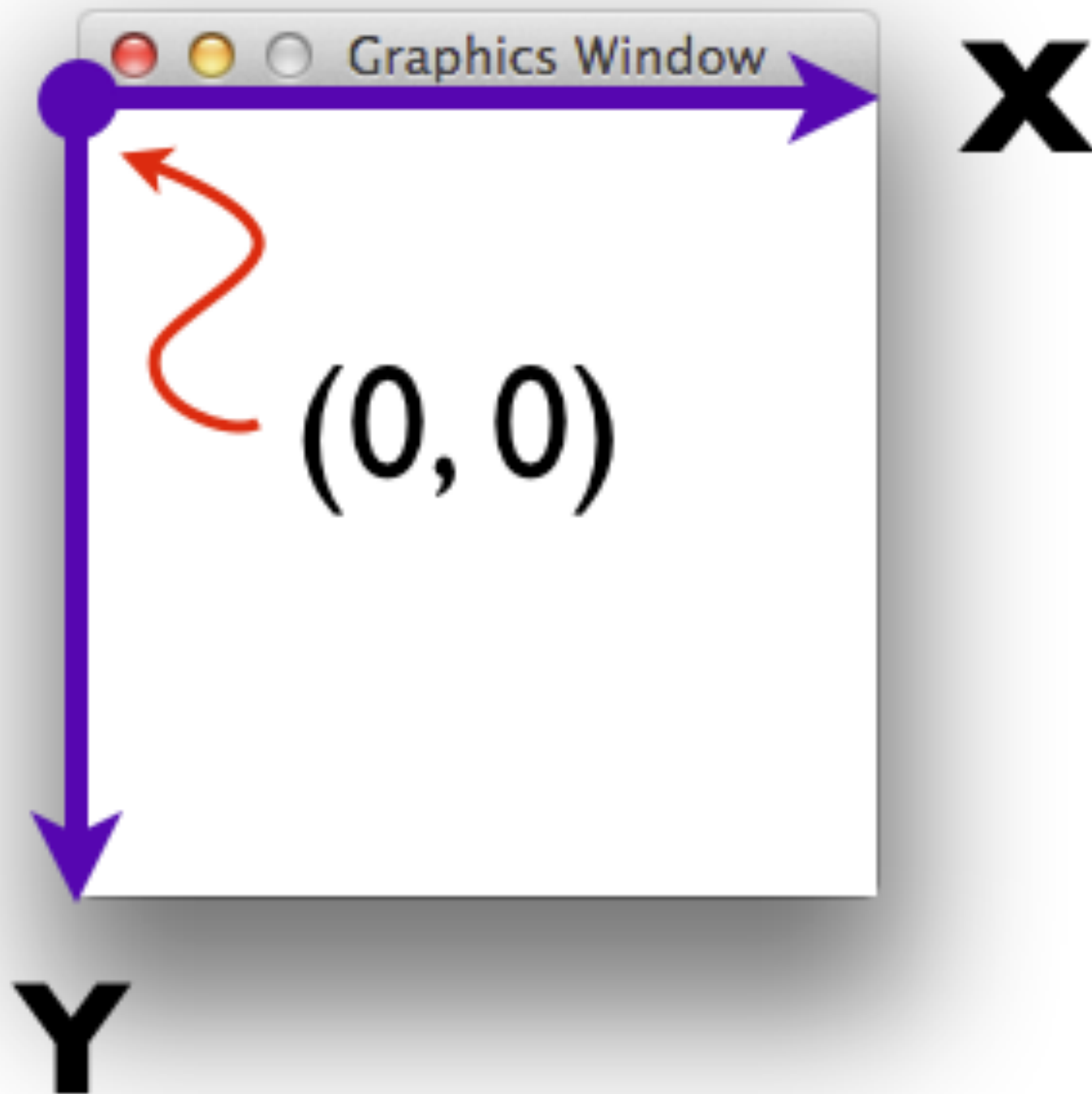


# Objekt och klasser

- Vad menas med ett objekt?
- Vad menas med en klass?
- Vad är sambandet mellan objekt och klass?

# Koordinatsystem

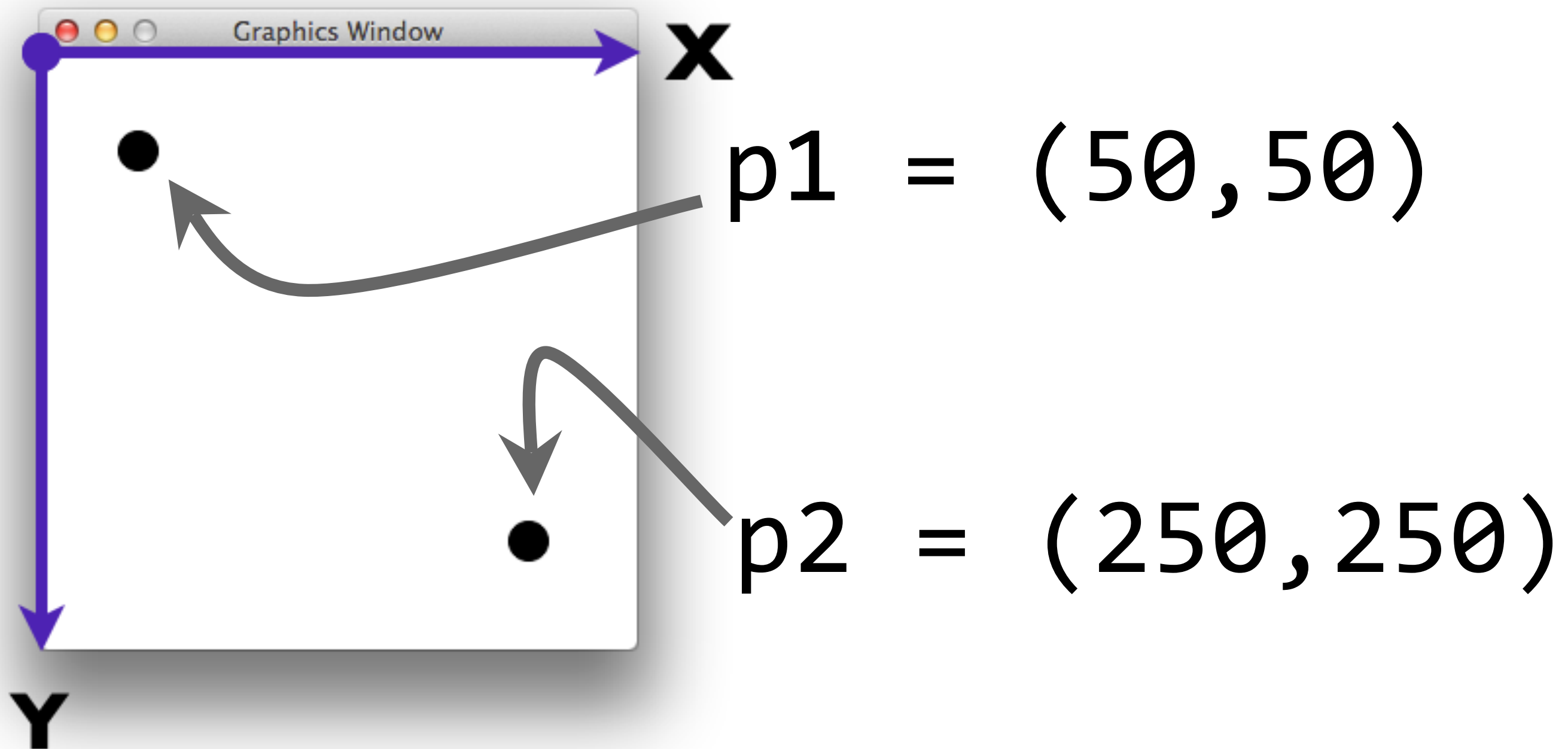
För att studera objekt och klasser kommer vi att använda oss av grafiska figurer som ritas upp i ett fönster.





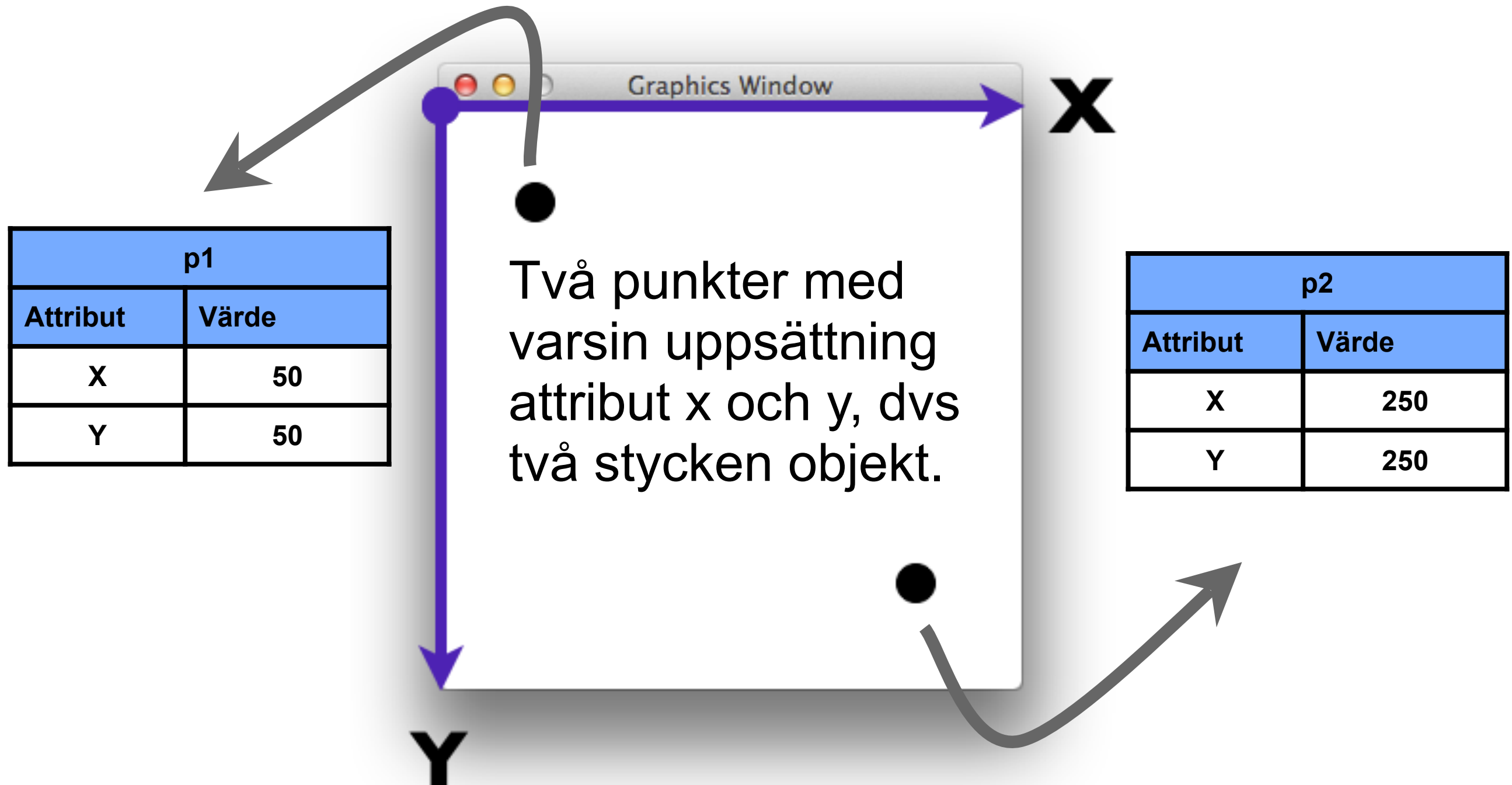
# Punkter i fönstret

En punkt i fönstret beskrivs av sitt position på x- och y-axeln. En punkt  $p$  kan alltså beskrivas av ett par av heltal  $p = (x, y)$ .



# Objekt

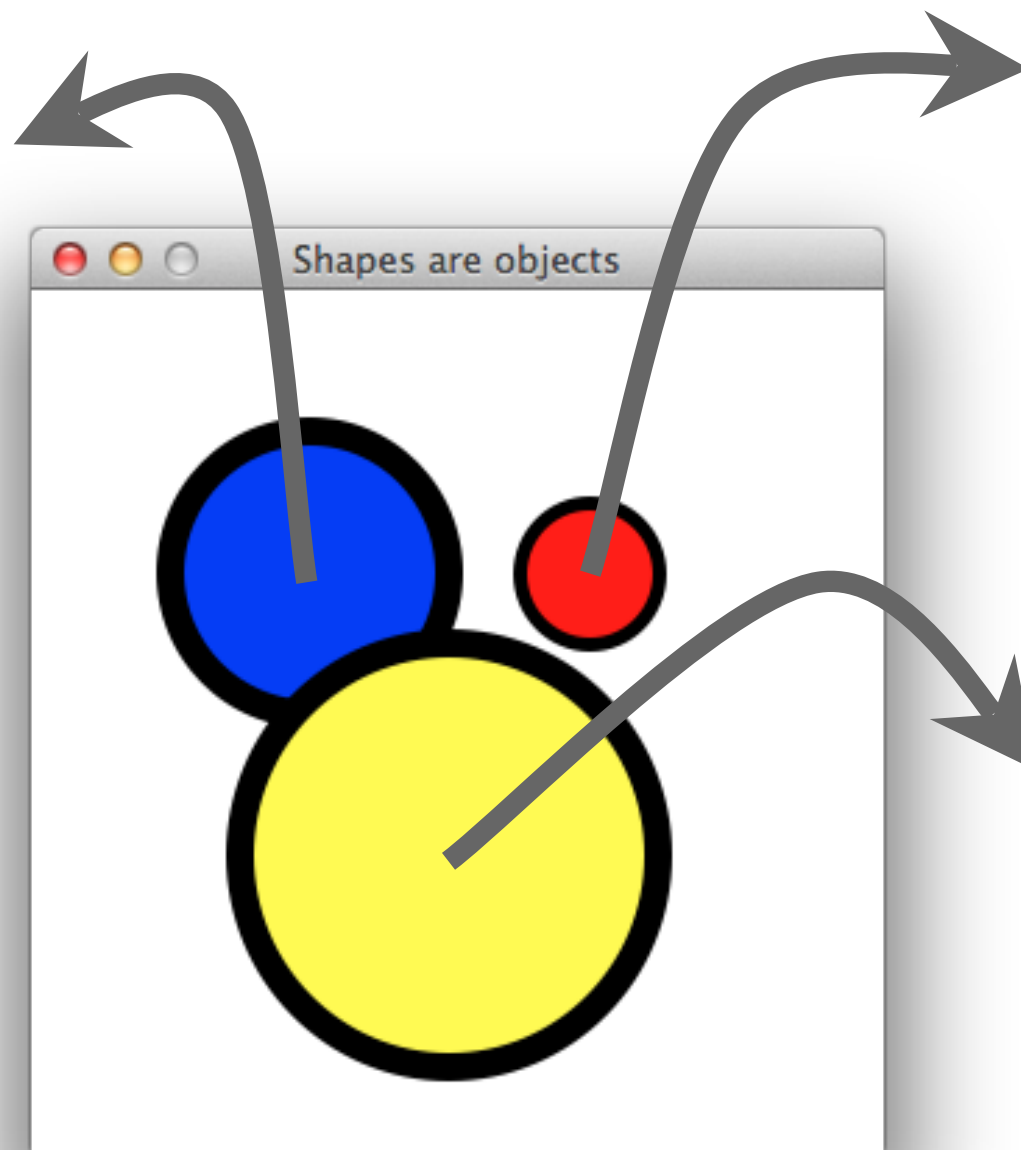
Ett objekt har ett antal **attribut**. En punkt är ett objekt med attribut X och Y.



# Objekt

Ett objekt har ett antal attribut. En cirkel är ett objekt med attribut centrum (X, Y), radie, färg och linjetjocklek.

Attribut		Värde
Centrum	X	100
	Y	100
Radie		50
Färg		Blå
Linjetjocklek		10



Attribut		Värde
Centrum	X	200
	Y	100
Radie		25
Färg		Röd
Linjetjocklek		5

Attribut		Värde
Centrum	X	150
	Y	200
Radie		75
Färg		Gul
Linjetjocklek		10

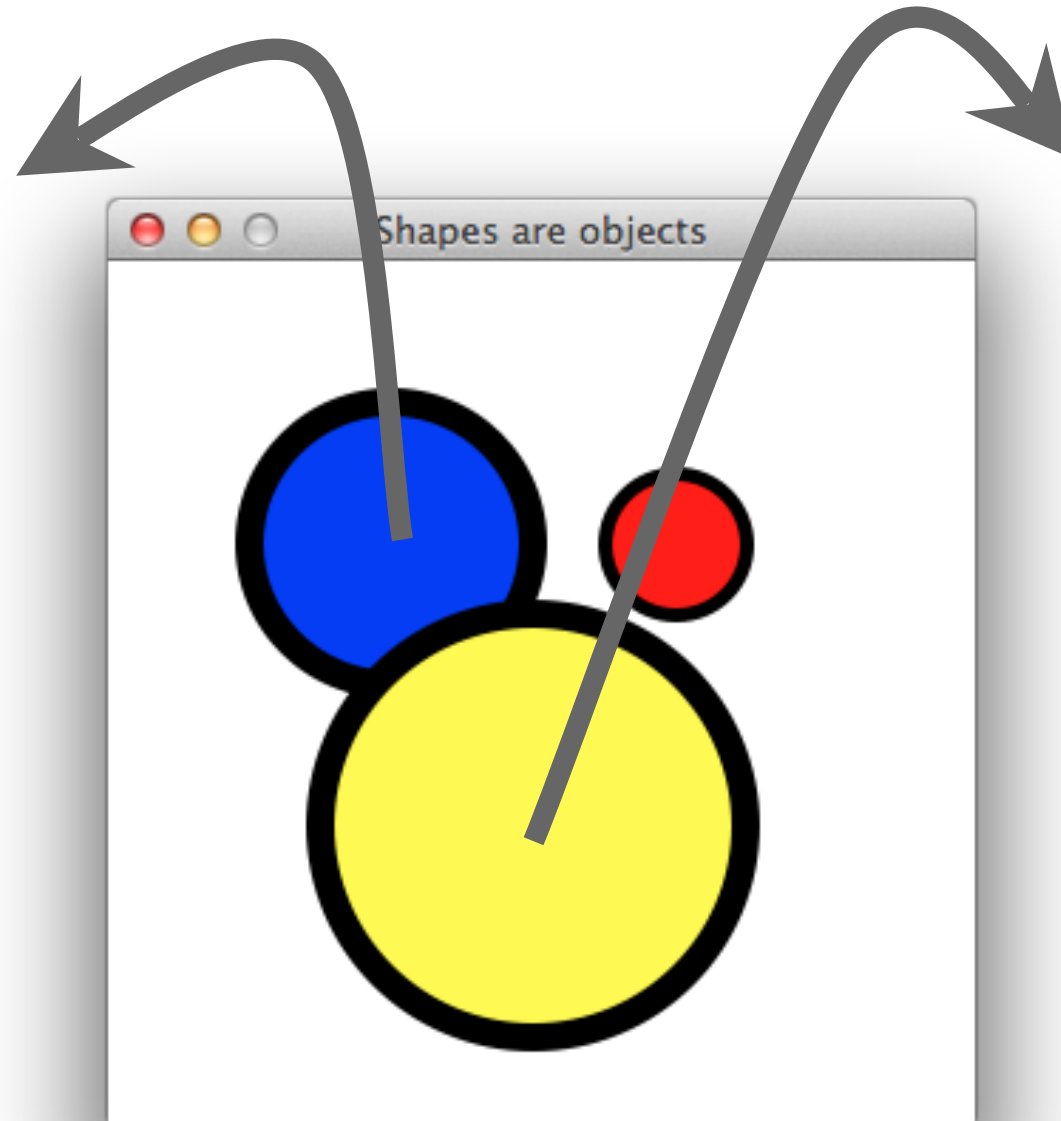
Tre cirklar med varsin uppsättning attribut, dvs tre olika objekt.

# Operationer på objekt

Ett objekt har förutom ett antal attribut även en samling operationer för att ändra på objektets attribut.

## Ett objekt

Attribut		
Centrum	X	100
	Y	100
Radie		50
Färg		Blå
Linjetjocklek		10
Operationer		
Flytta		
Ändra färg		
Ändra linjetjocklek		



Objekten har olika värden på attributen men stöder samma uppsättning operationer.

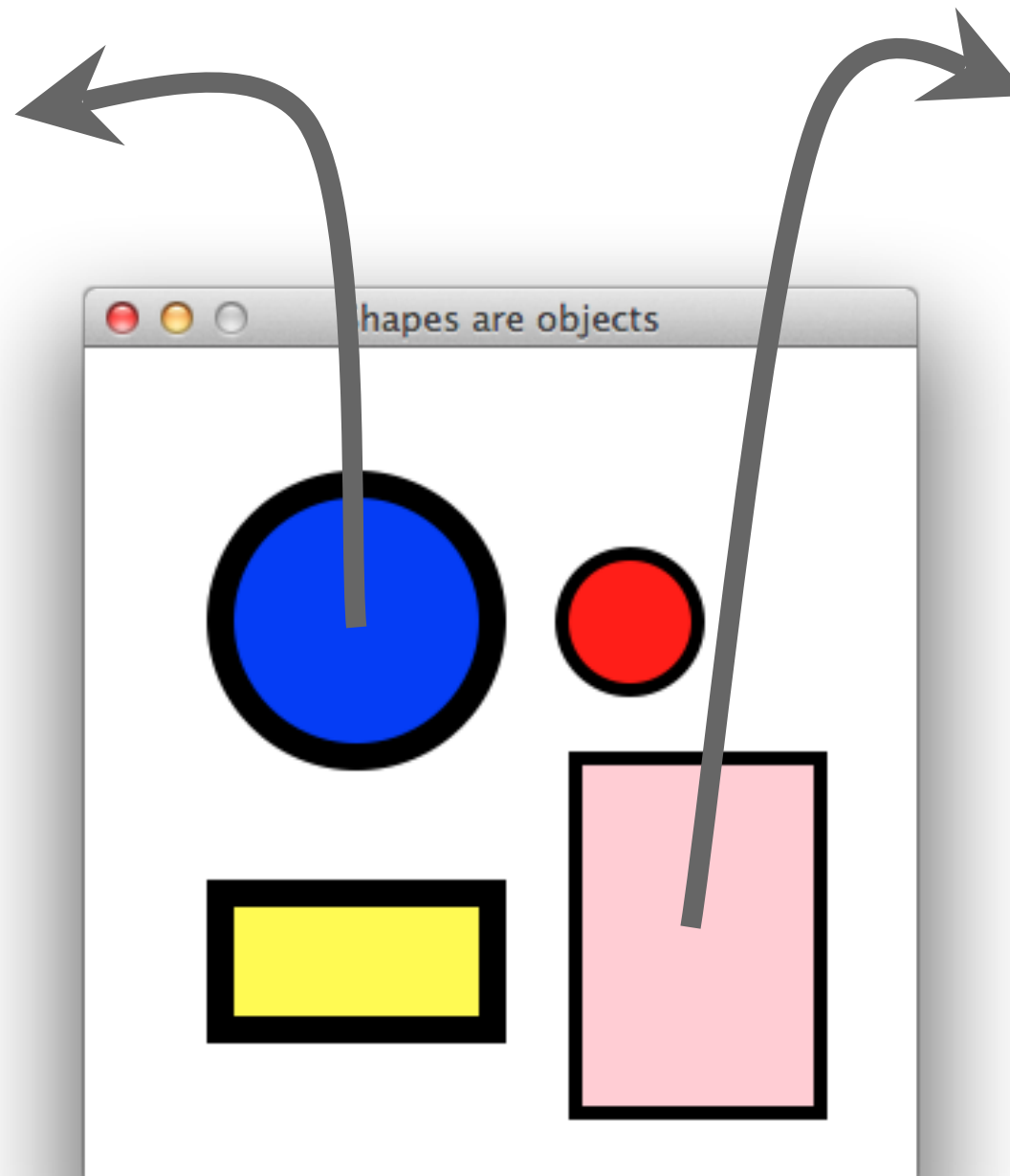
## Ett annat objekt

Attribut		
Centrum	X	150
	Y	200
Radie		75
Färg		Gul
Linjetjocklek		10
Operationer		
Flytta		
Ändra färg		
Ändra linjetjocklek		

# Cirklar och rektanglar är två klasser av objekt

Objekt som delar samma uppsättning attribut och operationer tillhör samma klass. Cirklar utgör därför en klass av objekt och rektanglar en annan klass av objekt.

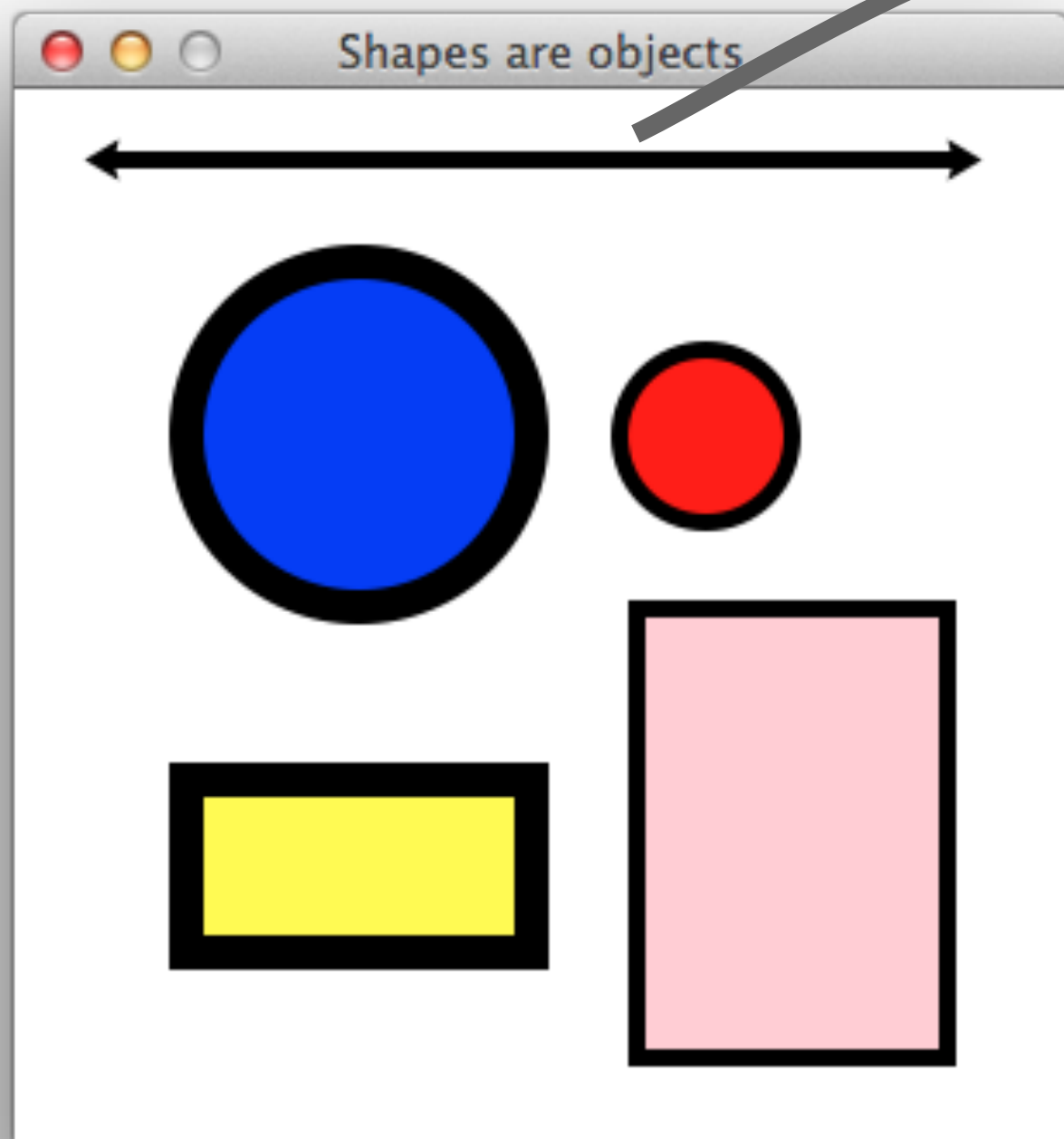
Objekt från klassen Cirklar		
Attribut		
Centrum	X	100
	Y	100
Radie		50
Färg		Blå
Linjetjocklek		10
Operationer		
Flytta		
Ändra färg		
Ändra linjetjocklek		



Objekt från klassen Rektanglar		
Attribut		
Centrum	X	225
	Y	215
Bredd		90
Höjd		130
Färg		Rosa
Linjetjocklek		5
Operationer		
Flytta		
Ändra färg		
Ändra linjetjocklek		

# Linjer är en klass av objekt

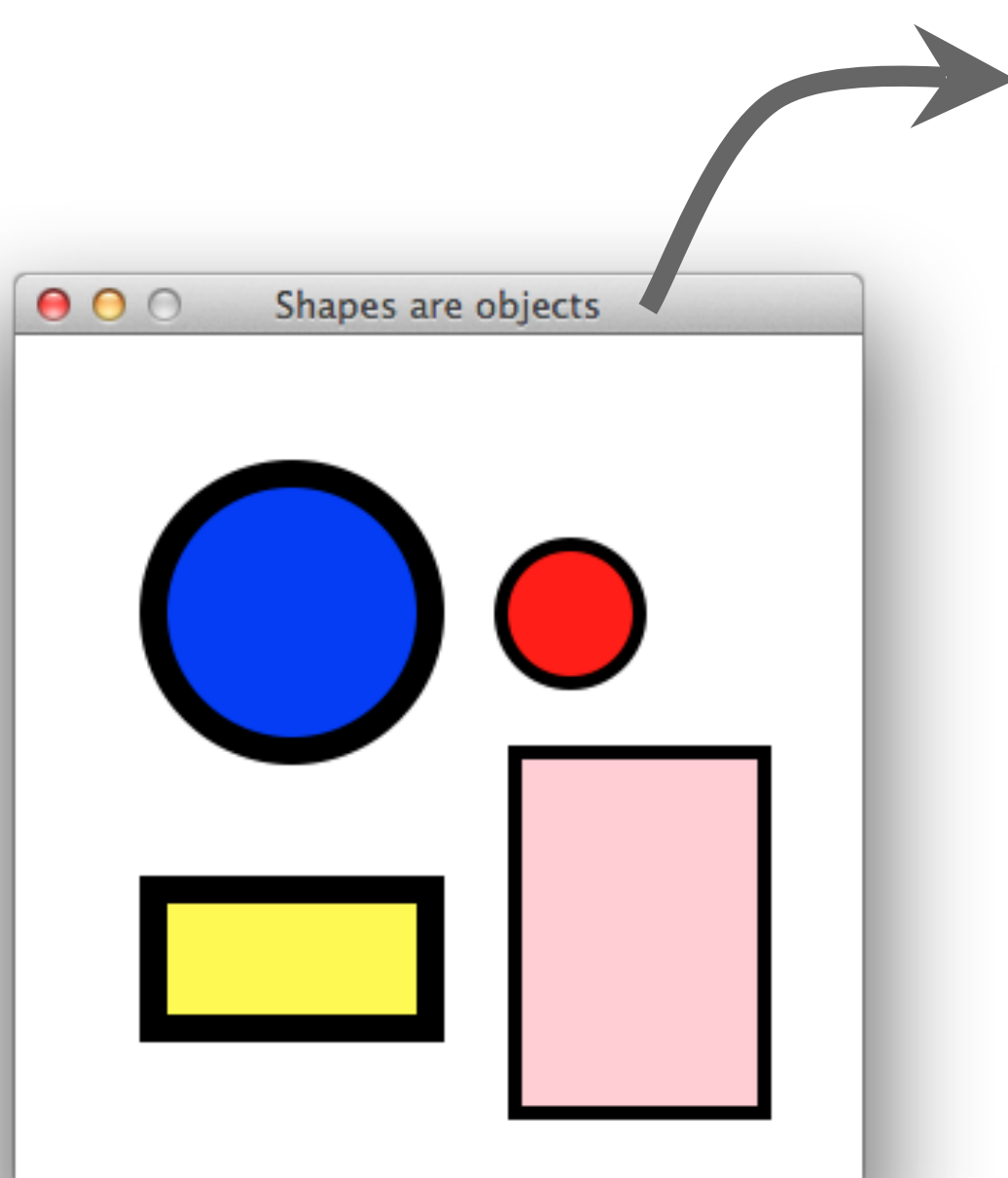
Linjer är objekt med attribut start, slut, linjetjocklek, pilar (inga, vid start, vid slut eller vid båda) och operationer för att ändra linjetjocklek och pilar.



Objekt från klassen Linjer		
Attribut		
Start	X	20
	Y	20
Slut	X	280
	Y	20
Linjetjocklek		5
Pilar		båda
Operationer		
Ändra linjetjocklek		
Ändra pilar		

# Fönster är en klass av objekt

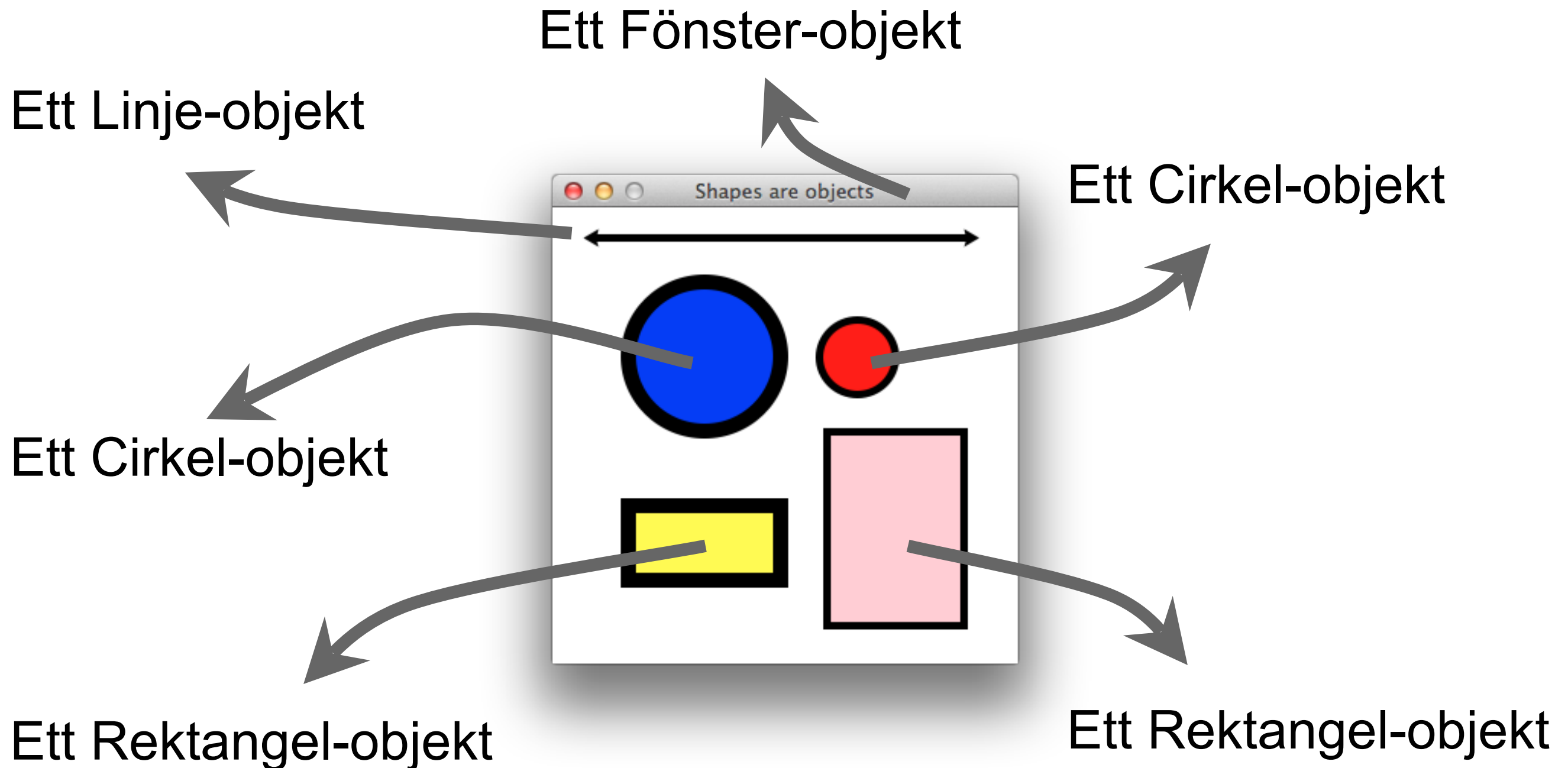
Fönster som vi ritar grafik i är objekt med attribut och operationer, dvs fönster för grafik utgör en klass av objekt.



Objekt från klassen Fönster	
Attribut	
Titel	Shapes are objects
Bredd	300
Höjd	300
Färg	Vit
Operationer	
Stäng fönster	
Vänta på musklick	

# Klasser och objekt

En klass beskriver en samling attribut med tillhörande operationer. Ett objekt är ett konkret exemplar, en **instans**, av det som klassen beskriver. Det går att skapa flera olika objekt av en viss klass, var och ett med olika värden på sina attribut.





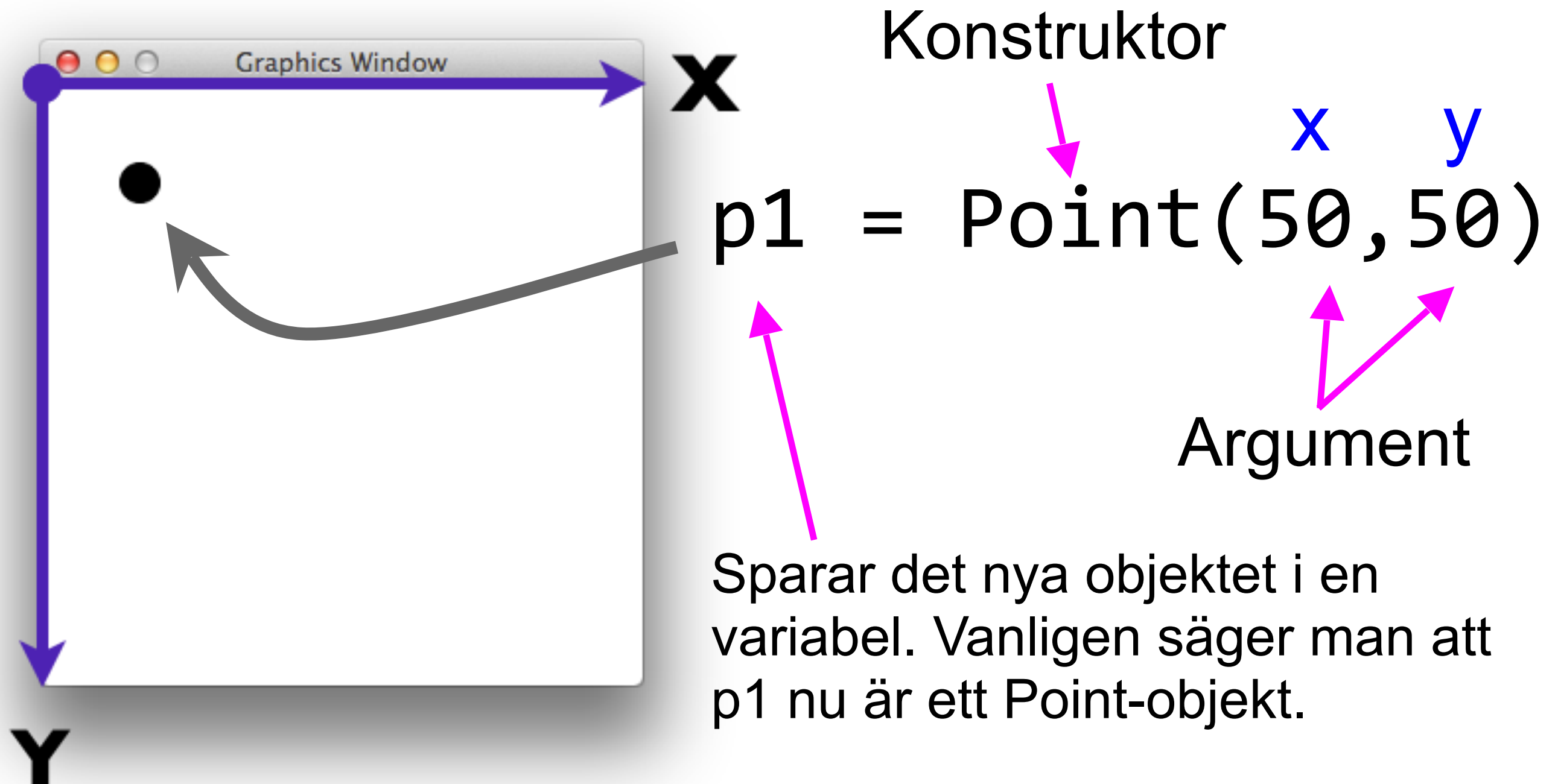
# Objekt och klasser i Python



# Konstruktor

(1)

För att skapa ett objekt av en viss klass anropas klassens konstruktor. Konstruktorn kan ta olika argument som används för att sätta värdet på objektets attribut.



# Konstruktor

(2)

För att skapa ett objekt av en viss klass anropas klassens konstruktor. Konstruktorn kan ta olika argument som används för att sätta värdet på objektets attribut.

Konstruktor

Argument

`c = Circle(Point(100,100), 50)`

Sparar det nya objektet i en variabel. Vanligen säger man att c nu är ett Circle-objekt.

Centrum på en cirkel är i sin tur ett objekt av klassen Point.

Radie på cirkeln.

# Metoder

För att ändra på ett objekts attribut måste vi anropa en av de metoder som finns för klassen objektet tillhör.

```
c = Circle(Point(100,100), 50)
```

```
c • setFill("blue")
```

En punkt •

För att anropa en metod på ett objekt används en punkt mellan objektet och namnet på metoden.

Argument till metoden.

Namn på den metod vi vill anropa.

Objekt från klassen Circle.

# Skapa fönstret *w*

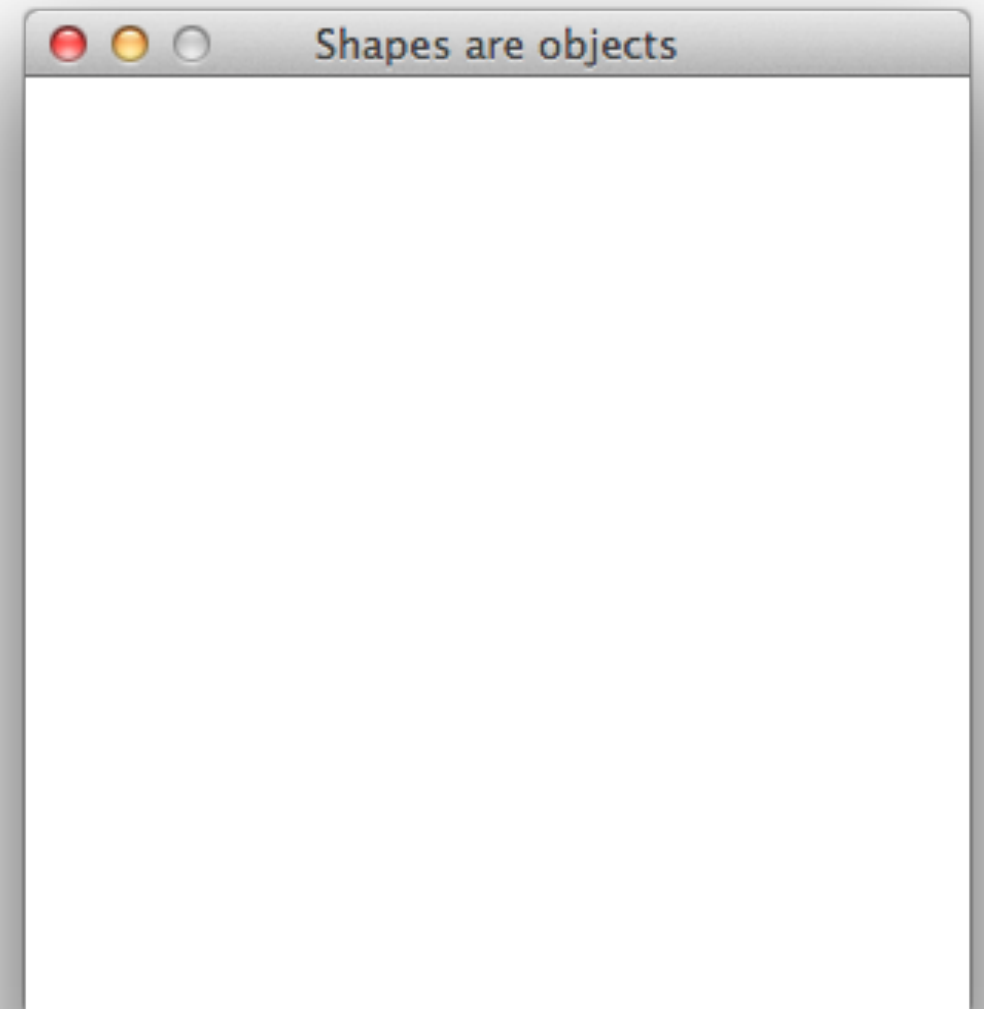
```
from graphics import *
```

```
w = GraphWin("Shapes are objects", 300, 300)
```



Klassen GraphWin	
Attributes	
Title	
Width	
Height	
Background	
Methods	
close()	
getMouse()	
getWidth()	
getHeight()	

Ett objekt från klassen GraphWin	
Attributes	
Title	Shapes are objects
Width	300
Height	300
Background	white
Methods	
close()	
getMouse()	
getWidth()	
getHeight()	

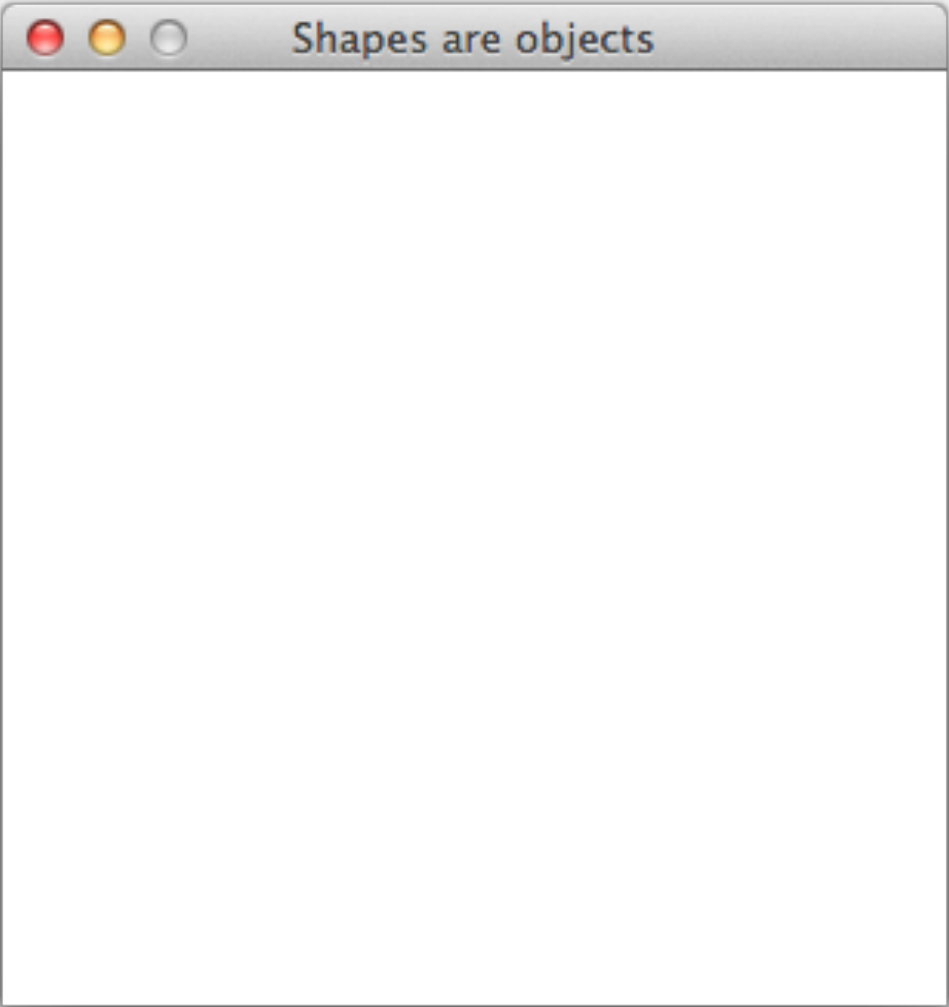


# Skapa cirkeln c

```
c = Circle(Point(100,100), 50)
```

Klassen Circle	
Attributes	
center	X
	Y
radius	
color	
width	
Methods	
draw(window)	
move(dx, dy)	
setFill(color)	
sedWidth(width)	
getCenter()	

Ett objekt från klassen Circle		
Attributes		
center	X	100
	Y	100
radius		50
color		?
width		?
Methods		
draw(window)		
move(dx, dy)		
setFill(color)		
sedWidth(width)		
getCenter()		

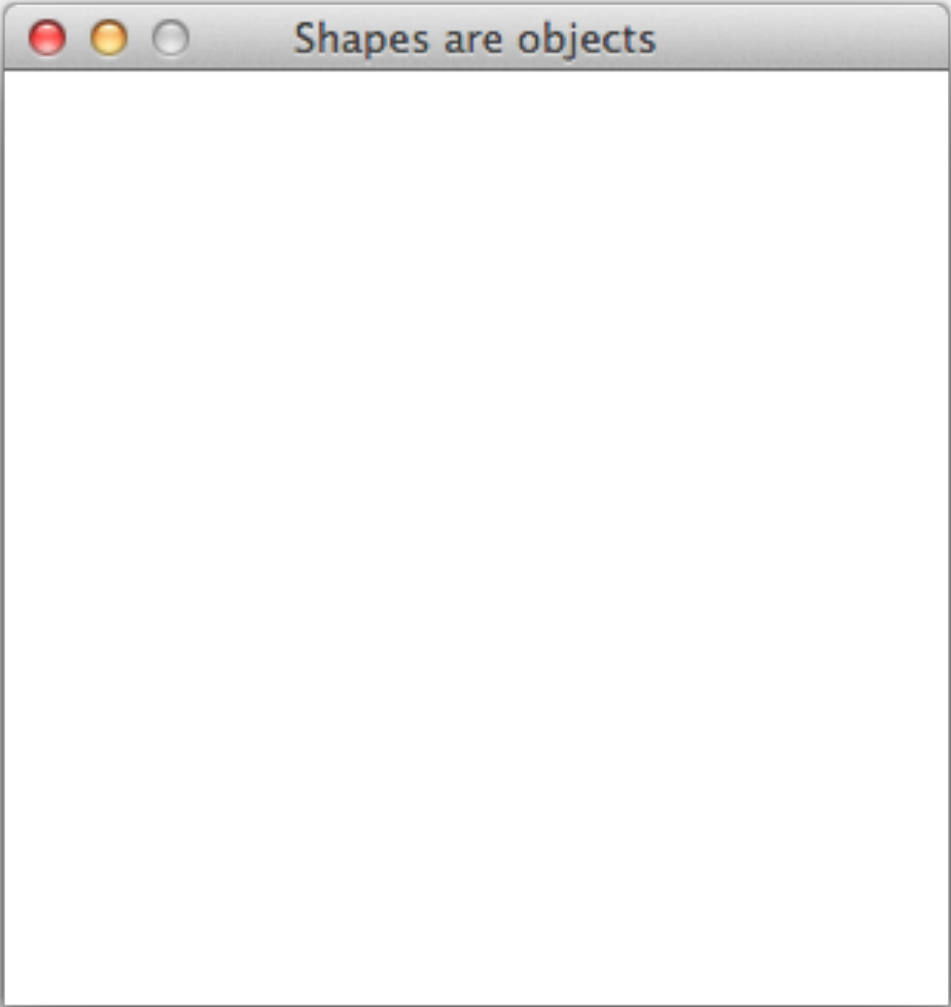


# Sätt attribut för cirkeln c

Klassen Circle	
Attributes	
center	X
	Y
radius	
color	
width	
Methods	
draw(window)	
move(dx, dy)	
setFill(color)	
setWidth(width)	
getCenter()	

Ett objekt från klassen Circle		
Attributes		
center	X	100
	Y	100
radius		50
color		blue
width		10
Methods		
draw(window)		
move(dx, dy)		
setFill(color)		
setWidth(width)		
getCenter()		

```
c.setFill("blue")
c.setWidth(10)
```

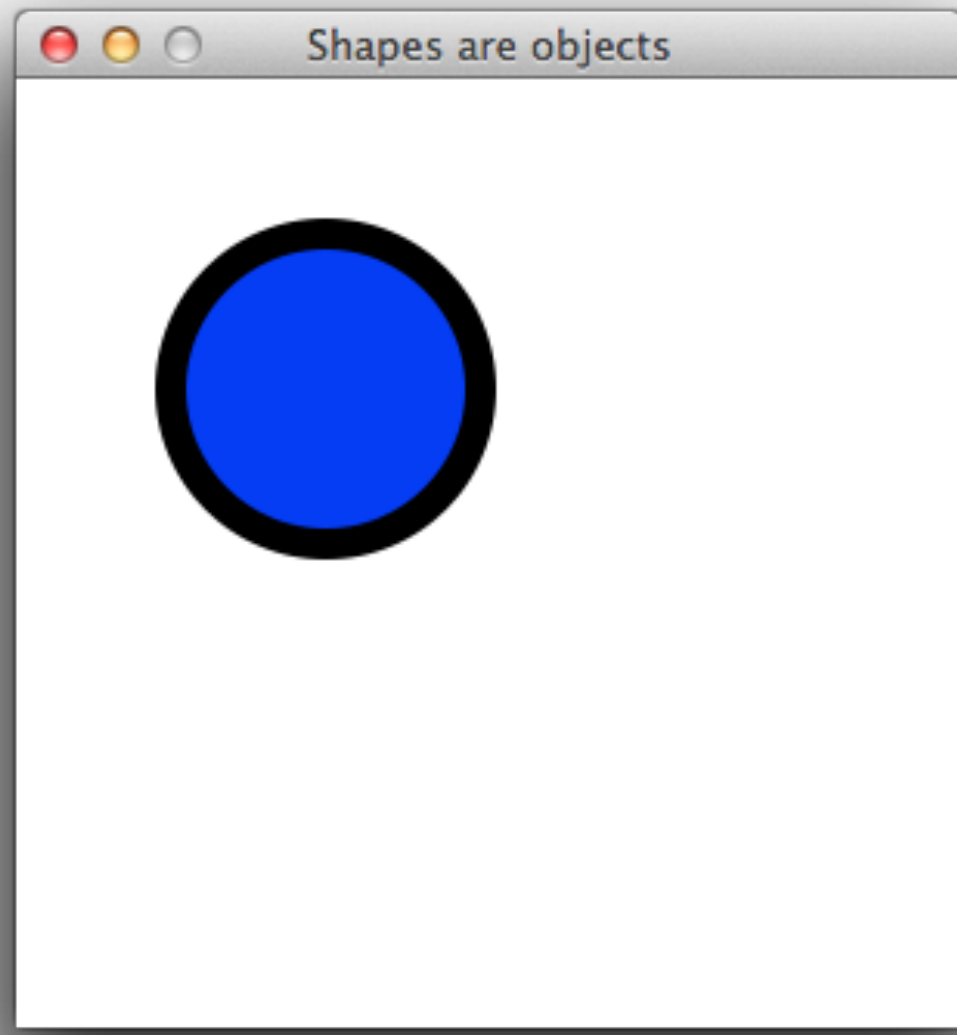


# Rita cikeln **c** i fönstret **w**.

**c**.draw(**w**)

Klassen Circle	
Attributes	
center	X
	Y
radius	
color	
width	
Methods	
draw(window)	
move(dx, dy)	
setFill(color)	
sedWidth(width)	
getCenter()	

Ett objekt från klassen Circle		
Attributes		
center	X	100
	Y	100
radius		50
color		blue
width		10
Methods		
draw(window)		
move(dx, dy)		
setFill(color)		
sedWidth(width)		
getCenter()		



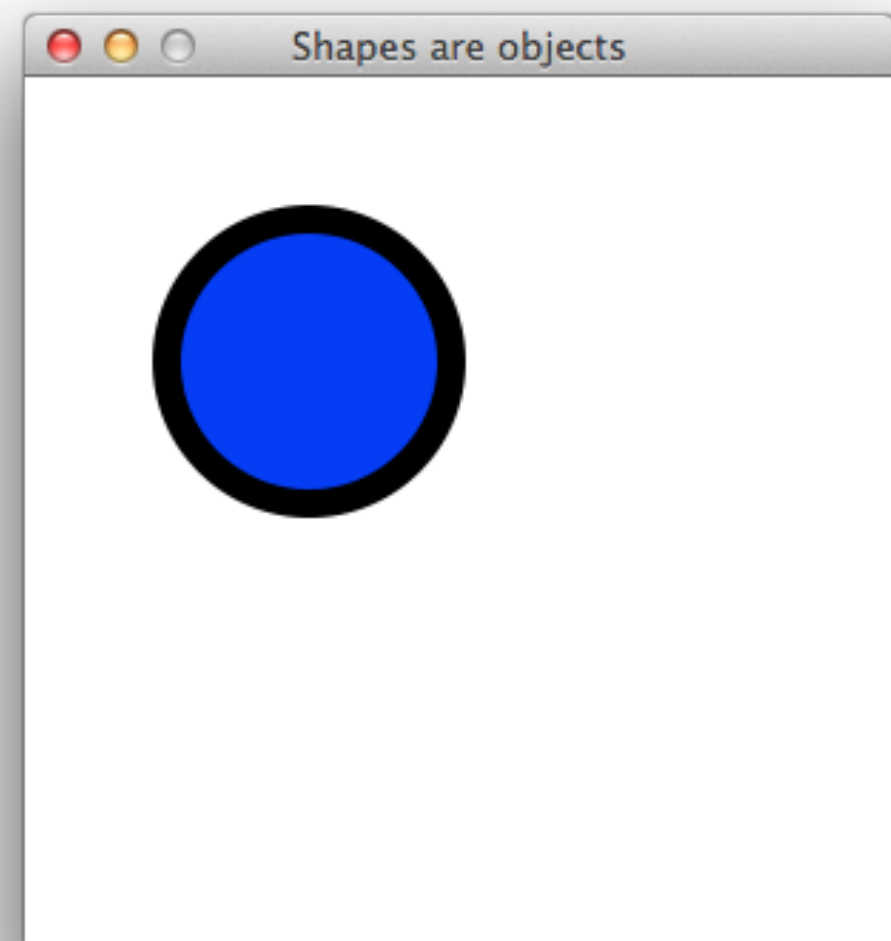


# Flytta på cirkeln c.

Klassen Circle	
Attributes	
center	X
	Y
radius	
color	
width	
Methods	
draw(window)	
move(dx, dy)	
setFill(color)	
setWidth(width)	
getCenter()	

Ett objekt från klassen Circle		
Attributes		
center	X	100
	Y	100
radius		50
color		blue
width		10
Methods		
draw(window)		
move(dx, dy)		
setFill(color)		
setWidth(width)		
getCenter()		

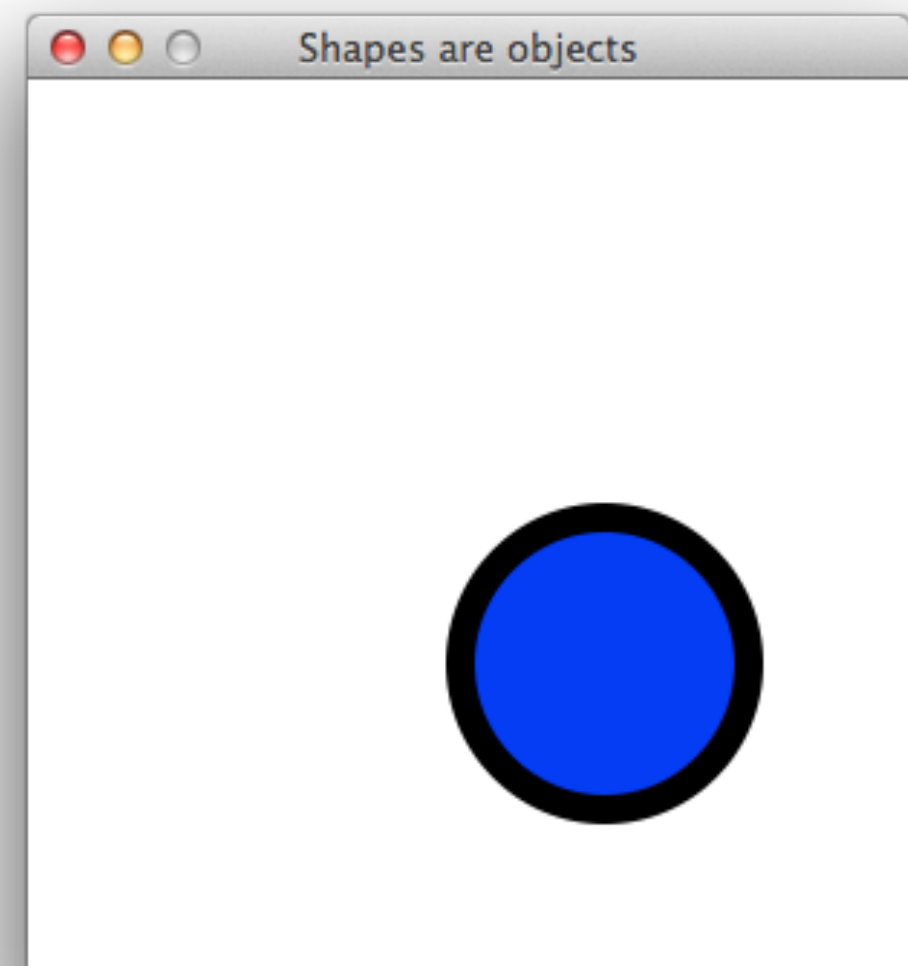
`c.move(100, 100)`



Metoden `move(dx, dy)` ändrar på objektets center-attribut (X, Y)

Klassen Circle	
Attributes	
center	X
	Y
radius	
color	
width	
Methods	
draw(window)	
move(dx, dy)	
setFill(color)	
setWidth(width)	
getCenter()	

Ett objekt från klassen Circle		
Attributes		
center	X	200
	Y	200
radius		50
color		blue
width		10
Methods		
draw(window)		
move(dx, dy)		
setFill(color)		
setWidth(width)		
getCenter()		



Metoden `move(dx, dy)` ändrar på objektets center-attribut (X, Y)

# Syntax

Inom datavetenskapen menas med syntax de **regler** som bestämmer hur **symbolerna** i ett programmeringsspråk måste **kombineras** och struktureras för att vara ett **giltigt program**.

# Syntaxfel (Syntax Error)

Syntaxfel (engelska: Syntax error) är inom datavetenskap ett **felmeddelande** som uppstår när ett **program inte är skrivet enligt syntaxreglerna** för programspråket i fråga.

Dessa fel beror ofta på stav- eller skrivfel och är sällan resultatet av ett logiskt tankefel.

# Pythons syntax

(1)

I Python säger syntaxen att **namn** på **variabler** och **funktioner** endast kan bestå av bokstäverna a...z och A...Z, siffrorna 0...9 och understreck (underscore) men inte får börja med en siffra.

```
>>> def double(n):  
...     return 2*n  
...  
>>>
```

```
>> def 3foo():  
    File "<input>", line 1  
        def 3foo():  
            ^  
SyntaxError: invalid syntax  
>>>
```

# Pythons syntax

(2)

I Python säger syntaxen att ett **decimal** tal:

- ▶ kan börja med minst en siffra (0-9), följas av en punkt (.) för att sedan följas av noll eller flera siffror (0-9)
- ▶ kan börja direkt med en punkt (.), följt av minst en siffra (0-9)

```
>>> .11
0.11
>>> 5.25
5.25
>>> 5.
5.0
>>> .5
0.5
>>> .
File "<input>", line 1
    .
    ^
SyntaxError: invalid syntax
>>> 5B.0
File "<input>", line 1
    5B.0
    ^
SyntaxError: invalid syntax
>>>
```

# Pythons syntax

(3)

I Python säger syntaxen att en **lista** måste börja med **[**, sedan följas av **noll eller flera värden** separerade med **komma** och avslutas med **]**.

```
>>> [1,2]
[1, 2]
>>> [1,2,3)
      File "<input>", line 1
        [1,2,3)
              ^
SyntaxError: invalid syntax
```

```
>>> [1,2,3]
[1, 2, 3]
>>> [1 2 3]
      File "<input>", line 1
        [1 2 3]
              ^
SyntaxError: invalid syntax
```

# Syntaktisk socker

Syntaktiskt socker är **tillägg till ett programspråks syntax** som inte påverkar språkets funktionalitet men **som gör det enklare att använda**.

Uttrycket myntades av den brittiska datorvetaren Peter J. Landin.

De funktioner som det syntaktiska sockret utför kan alltid uttryckas i någon mer grundläggande syntax hos språket.





# I Python är listor objekt

För att skapa en ny lista används konstruktorn **list()**. Vanligen används det **syntaktiska sockret** **[ ]** i stället för **list()**.

Klassen List
Attributes
elements
length
Methods
append()
reverse()
sort()

`ns = []`

Syntaktiskt  
socker för  
konstruktor

Objekt från klassen List	
Attributes	
elements	None
length	0
Methods	
append()	
reverse()	
sort()	

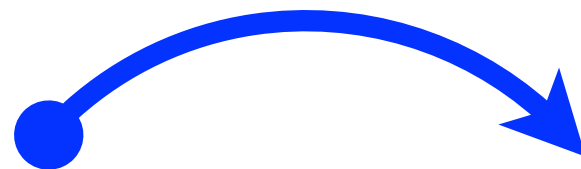
Vi kan även skapa en lista genom att  
anropa konstruktorn `list()`.

`ns = list()`

Konstruktor

# ns.append(10)

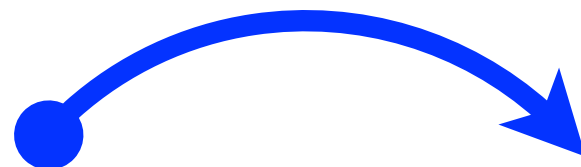
Objekt från klassen List	
Attributes	
elements	None
length	0
Methods	
append()	
reverse()	
sort()	



Objekt från klassen List	
Attributes	
elements	10
length	1
Methods	
append()	
reverse()	
sort()	

# ns.append(77)

Objekt från klassen List	
Attributes	
elements	10
length	1
Methods	
append()	
reverse()	
sort()	



Objekt från klassen List	
Attributes	
elements	10,77
length	2
Methods	
append()	
reverse()	
sort()	

# ns.reverse()

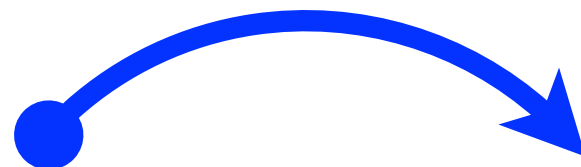
Objekt från klassen List	
Attributes	
elements	10,77
length	2
Methods	
append()	
reverse()	
sort()	



Objekt från klassen List	
Attributes	
elements	77,10
length	2
Methods	
append()	
reverse()	
sort()	

# ns.append(42)

Objekt från klassen List	
Attributes	
elements	77, 10
length	2
Methods	
append()	
reverse()	
sort()	



Objekt från klassen List	
Attributes	
elements	77,10,42
length	3
Methods	
append()	
reverse()	
sort()	

# ns.sort()

Objekt från klassen List	
Attributes	
elements	77,10,42
length	3
Methods	
append()	
reverse()	
sort()	



Objekt från klassen List	
Attributes	
elements	10,42,77
length	3
Methods	
append()	
reverse()	
sort()	