

FONDAMENTI DI INFORMATICA

Alma Artis

Francesca Pratesi (ISTI, CNR)

Introduzione a Processing – Forme di base,
riempimenti, colori

ESEMPIO: SOMMA DI MATRICI

- Date due matrici A e B, entrambe $M \times N$, è possibile calcolare la loro somma $A+B = C$. C è una matrice $M \times N$ in cui ogni elemento è dato dalla somma tra i corrispondenti elementi nelle matrici A e B.
 - Scrivere una funzione con due parametri A e B, aventi la stessa dimensione. La funzione calcola e restituisce un nuovo array (multidimensionale) che rappresenta la matrice $A+B$.

1	2	3
4	5	6

 +

2	3	4
6	5	4

 =

3	5	7
10	10	10

ESEMPIO: SOMMA DI MATRICI

```
function somma_matrici(A,B){  
    var C = new Array(A.length); // creo un array semplice  
    for(var riga=0;riga<A.length;riga++){ // per ogni riga  
        C[riga]=new Array(A[riga].length); // creo la nuova riga  
        for(var colonna=0;colonna<A[riga].length;colonna++){  
            /* per ogni colonna, ma in un for interno, quindi andrebbe  
            letto «per ogni riga e per ogni colonna», cioè per ogni posizione */  
            C[riga][colonna] = A[riga][colonna] + B[riga][colonna];  
        }  
    }  
    return C;  
}
```



ESERCIZI

- Scrivere una funzione che prende in input una matrice e la popola con numeri letti da tastiera.
- Si scriva una funzione *concatena* che dati due array di interi A e B, calcola e restituisce in output un terzo array C che contiene tutti gli elementi di A seguiti da tutti gli elementi di B. La funzione lascia A e B invariati.
- Scrivere una funzione due array di interi A e B, calcola e restituisce in output un terzo array C che contiene l'unione di A e B, cioè l'insieme (senza duplicati!) di tutti gli elementi che sono in A o in B. La funzione lascia A e B invariati.



ESERCIZI - 2

- Scrivere una funzione con un parametro stringa s. La funzione deve restituire in output la stringa inversa di s, cioè la stringa che si ottiene leggendo s al contrario. Es. dato 'ciao' in input deve essere restituito 'oaic'
- Un palindromo è una sequenza di caratteri che, letta al contrario, rimane identica. Esempi di palindromi: 'ada', 'ahaha', 'radar', 'ossesso', 'oro', 'anna'. Scrivere una funzione che data una stringa, verifica se essa è un palindromo o meno.





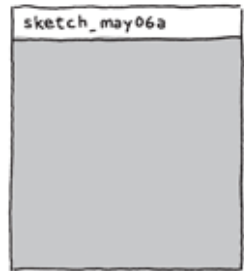
INTRODUCING PROCESSING

PROCESSING.ORG - INSTALLAZIONE

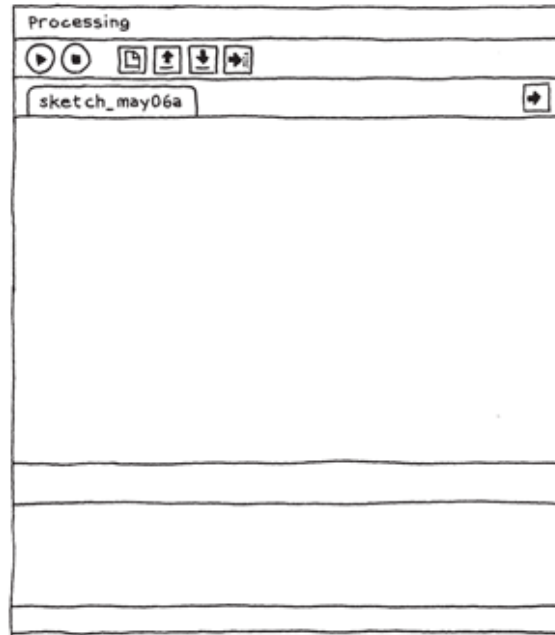
- Scaricare l'IDE da <http://processing.org>
- Estrarre tutti i file
- Eseguire l'applicazione processing.exe



ANATOMY OF CODE EDITOR



Display window



Toolbar

Tabs

Text
editor

Message
area

Console

ANATOMY OF CODE EDITOR



Execute current Sketch



Stop execution of current Sketch



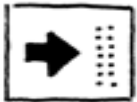
Save current Sketch



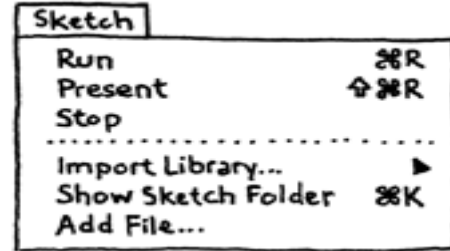
Load an existing Sketch from disk



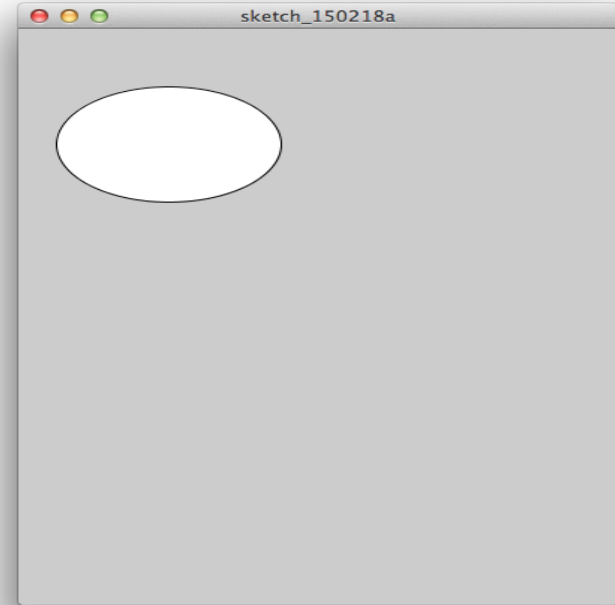
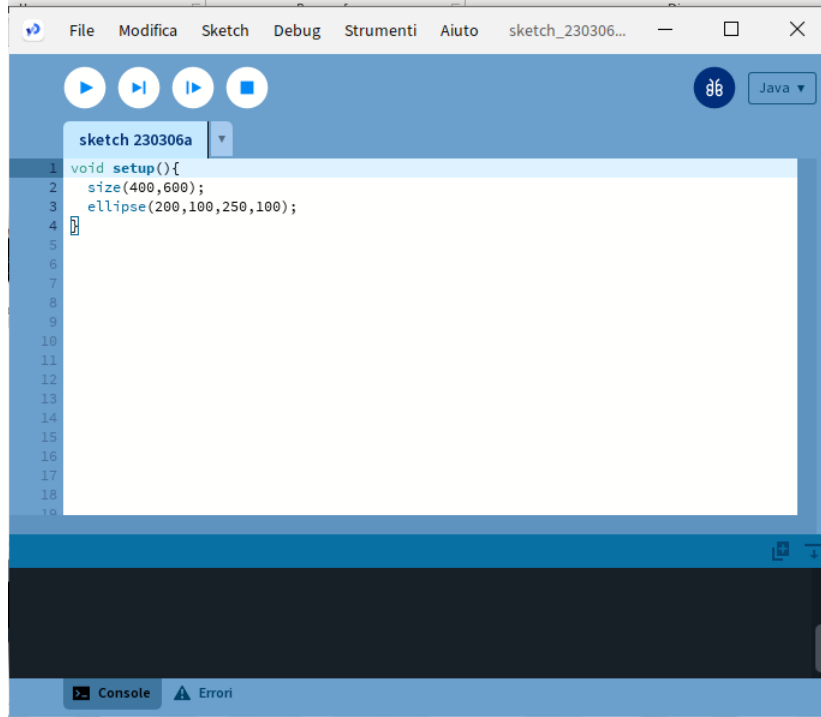
Start a new Sketch



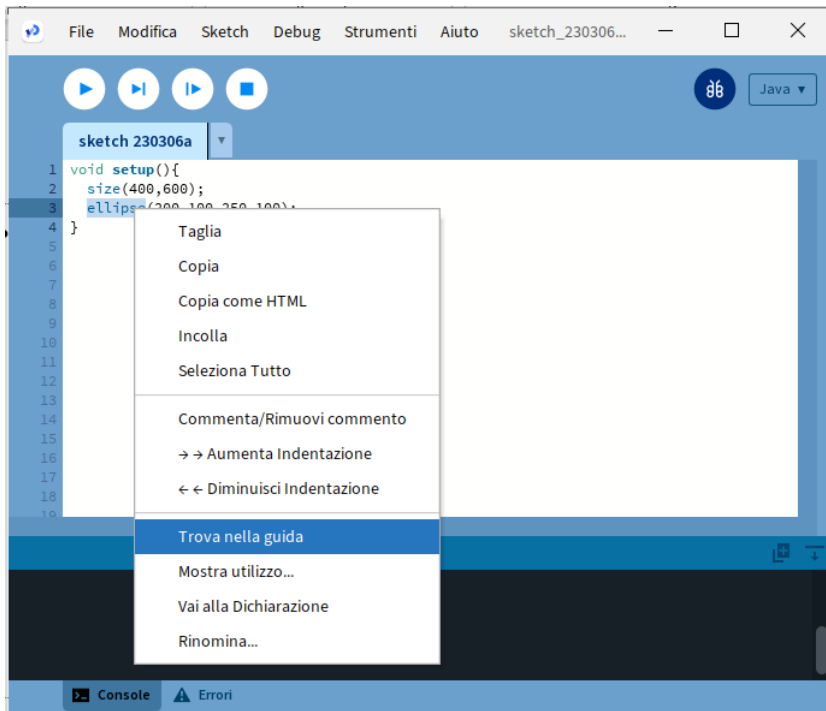
Export current Sketch as a standalone app



IL NOSTRO PRIMO SKETCH



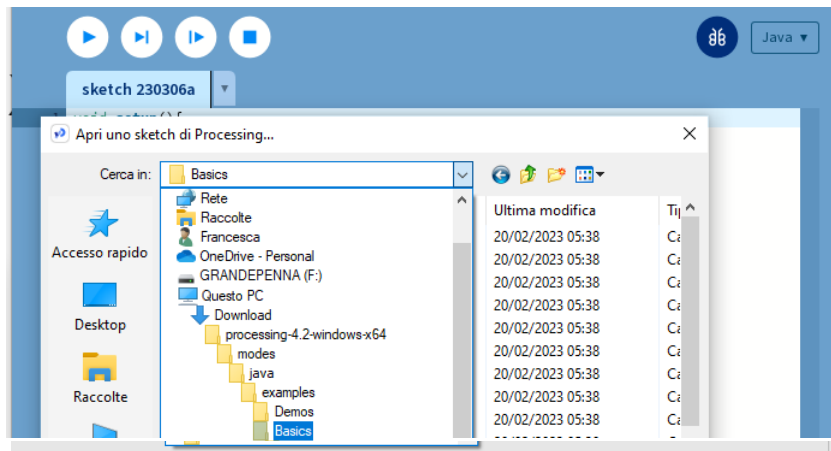
CHIEDERE AIUTO



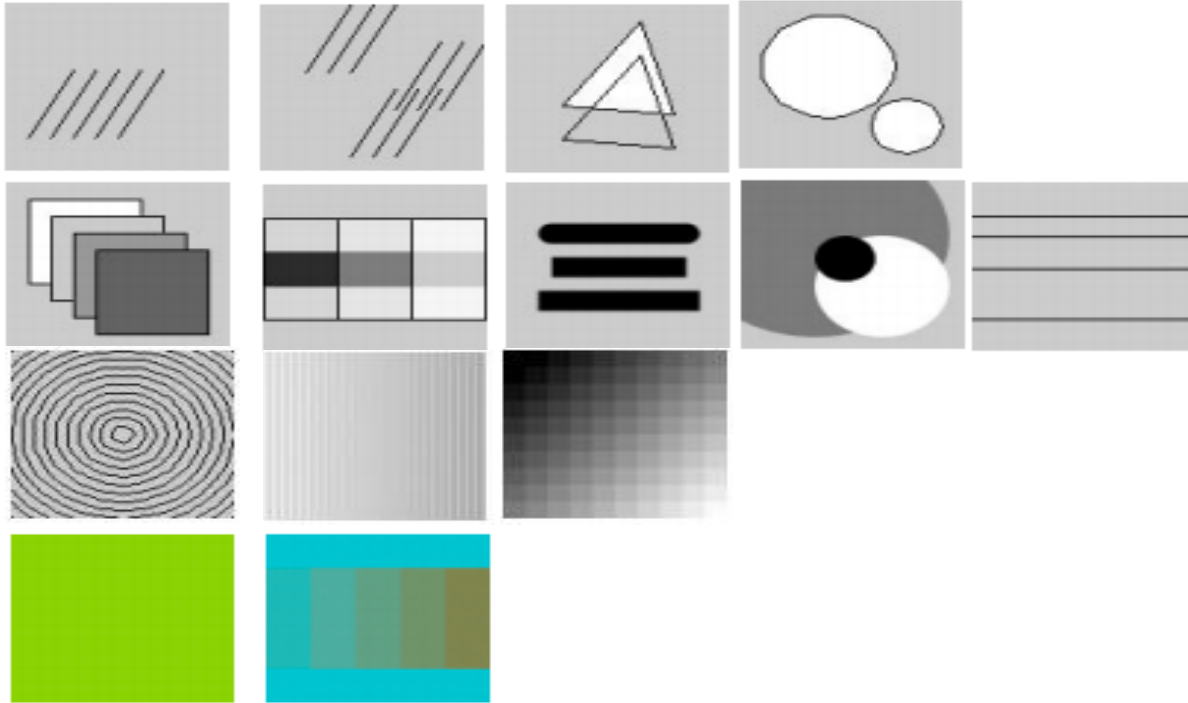
- Ogni parola chiave (keyword) è collegata alla sua documentazione
- Accessibile attraverso il tasto destro del mouse

BROWSE EXAMPLES DIRECTORY

Dal menu “Open...” scegliere la voce
“Examples...”



SOME EXAMPLES



Processing: a programming handbook for visual designers and artists

Casey Reas, Ben Fry
MIT Press, 2007



PIXELS

COORDINATE

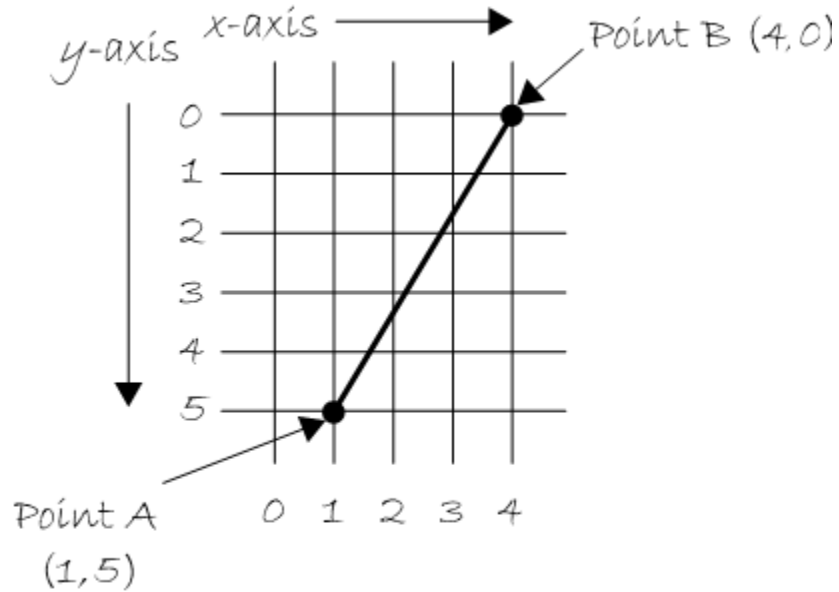


fig. 1.1

`line(0,1,4,5);`

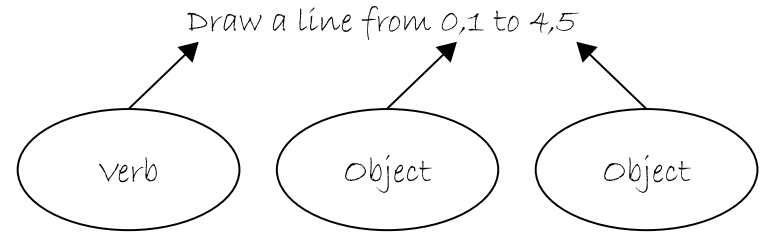
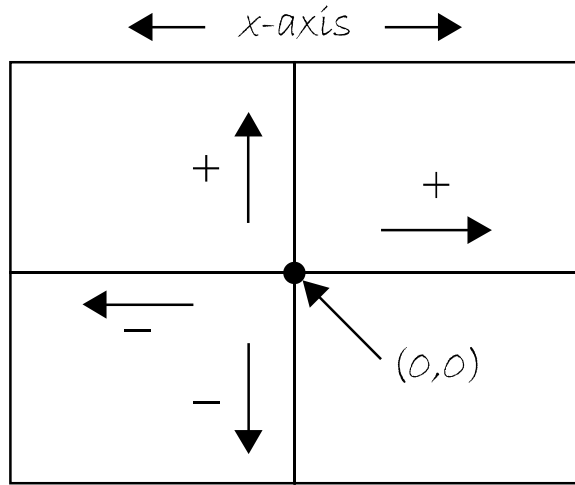
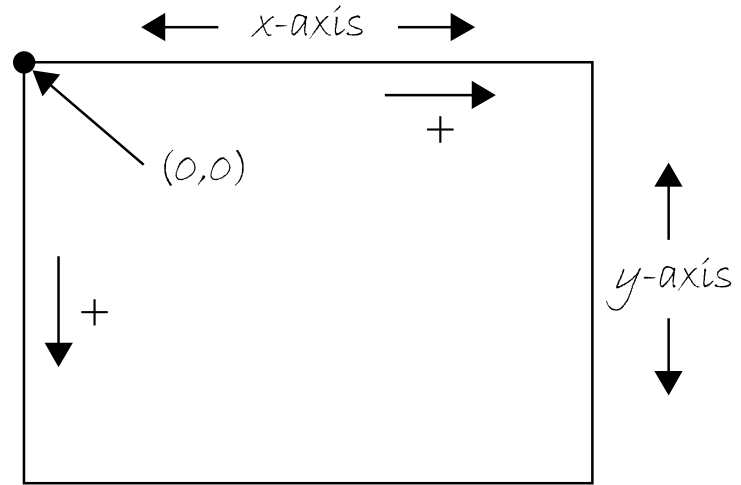


fig. 1.2

SISTEMA DI RIFERIMENTO



Eighth grade



Computer

fig. 1.3

FORME DI BASE



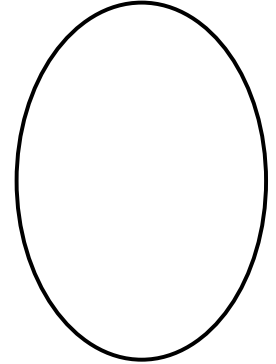
Point
fig. 1.4



Line



Rectangle



Ellipse



FORME DI BASE: POINT

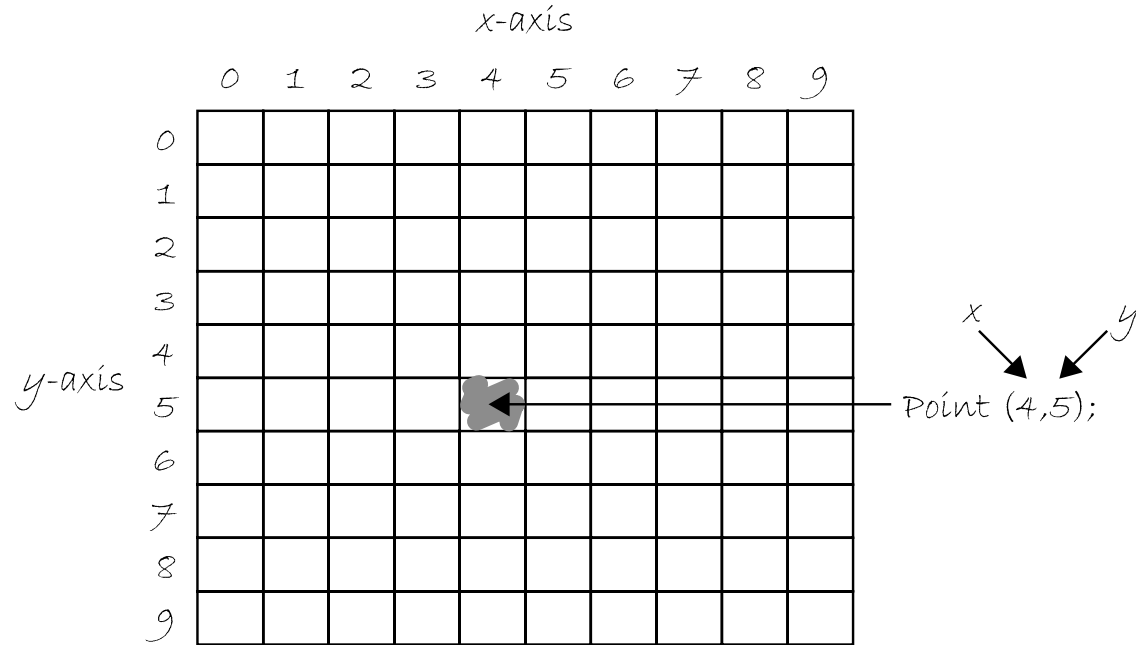


fig. 1.5

FORME DI BASE: LINE

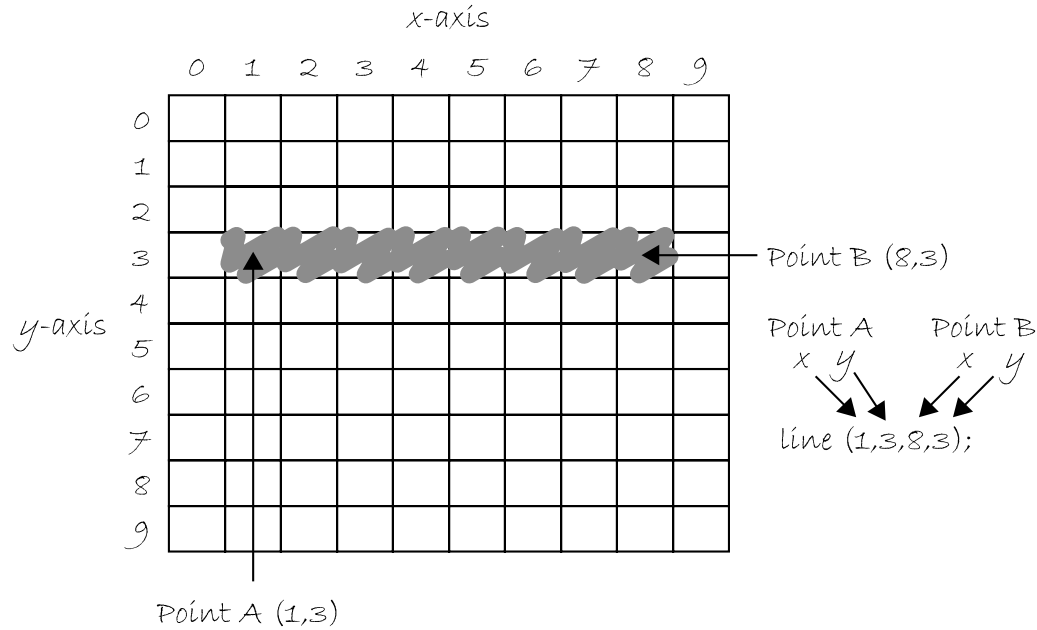


fig. 1.6

FORME DI BASE: RECT

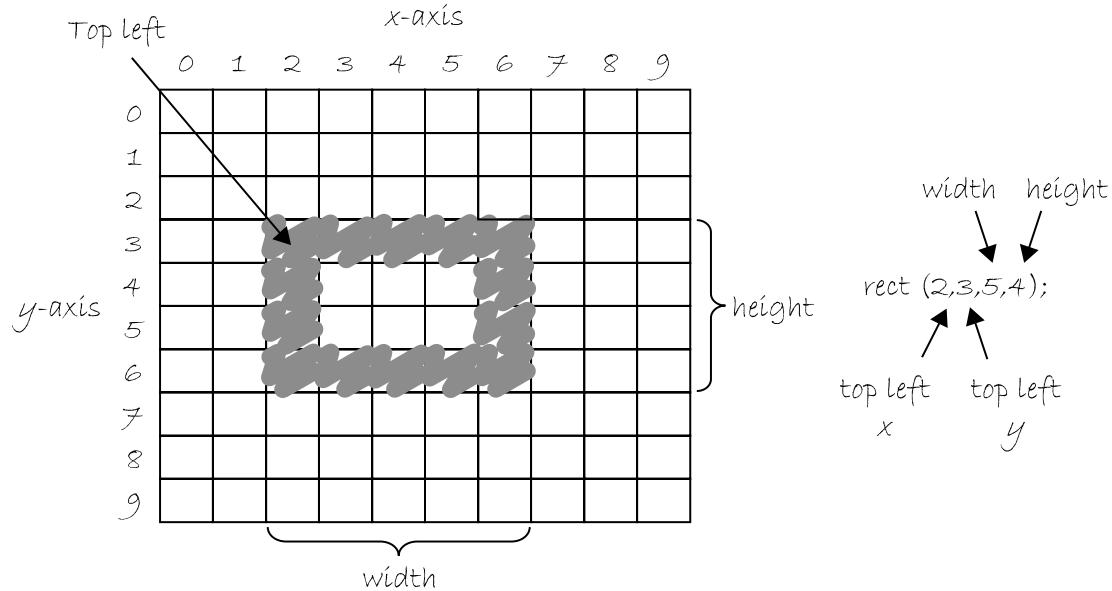


fig. 1.7

FORME DI BASE: RECT (2)

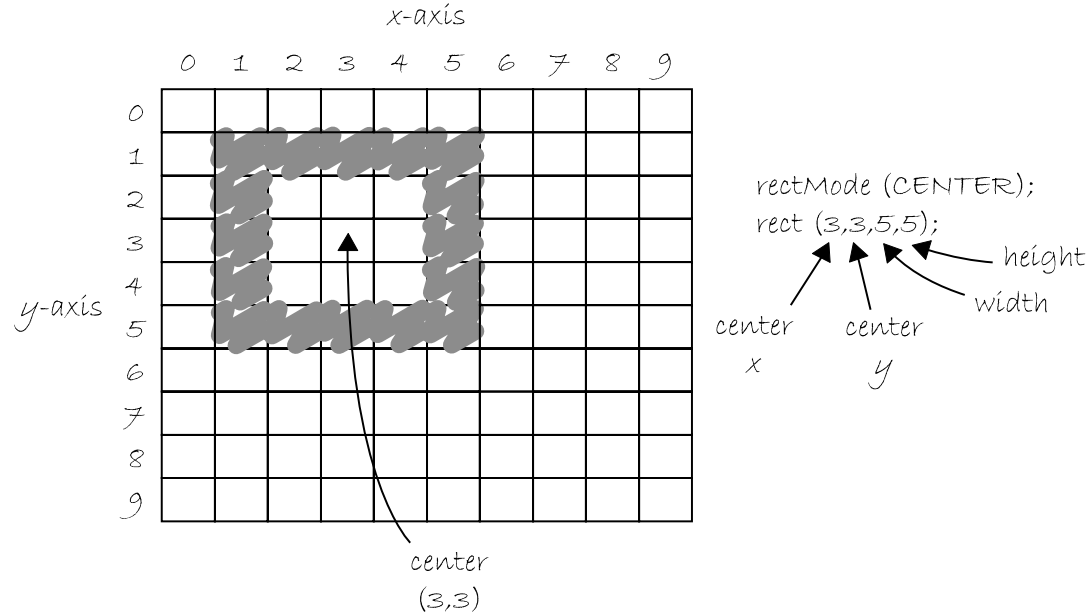


fig. 1.8

FORME DI BASE: RECT (3)

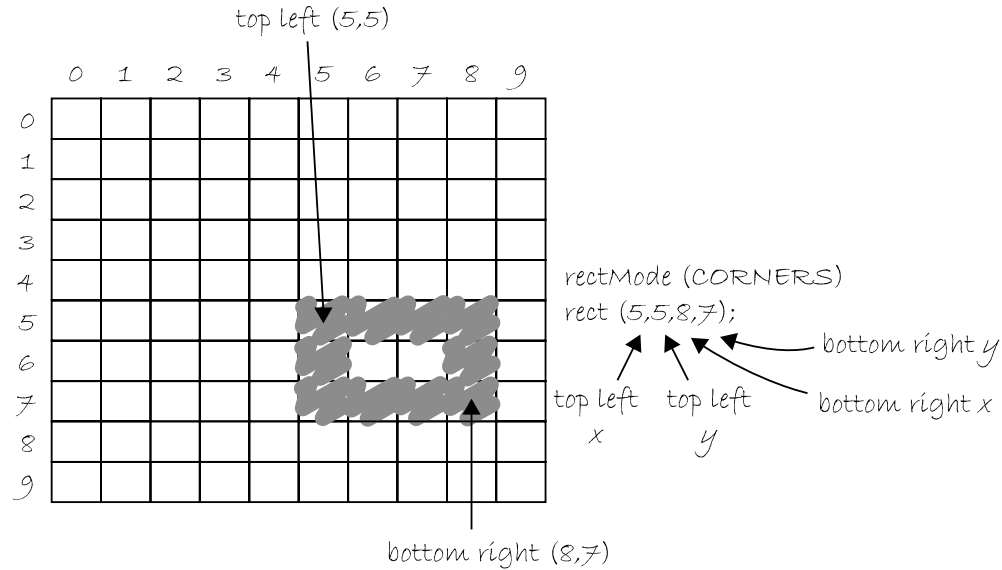
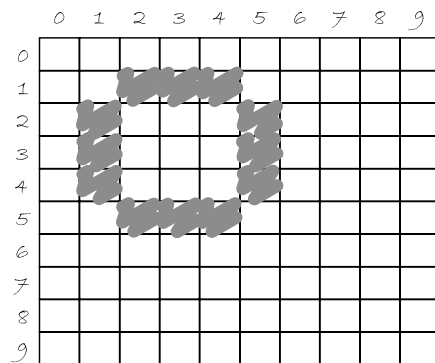
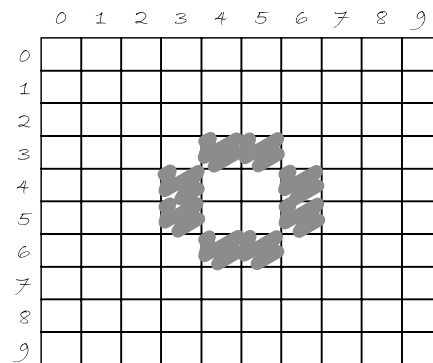


fig. 1.9

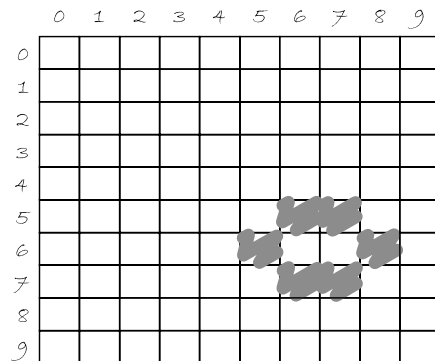
FORME DI BASE: ELLIPSE



`ellipseMode (CENTER);`
`ellipse (3,3,5,5);`



`ellipseMode (CORNER);`
`ellipse (3,3,4,4);`

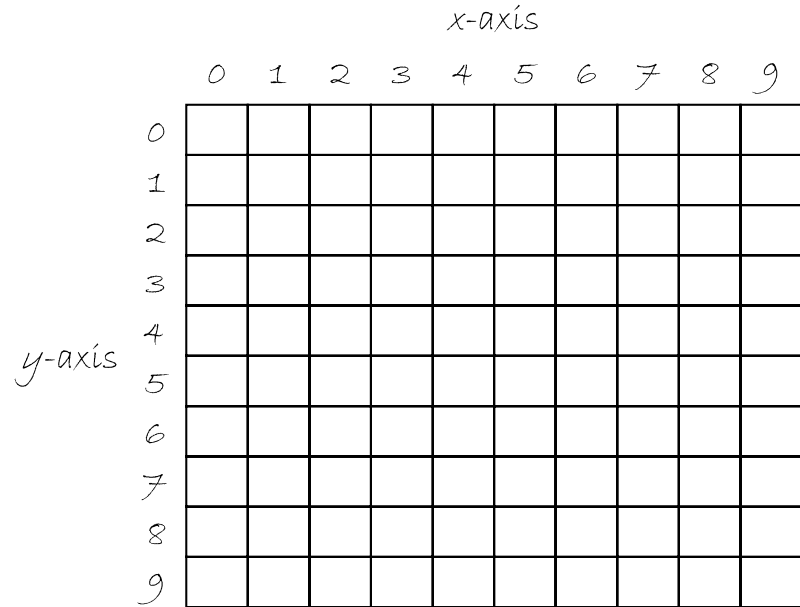


`ellipseMode (CORNERS);`
`ellipse (5,5,8,7);`

fig. 1.10

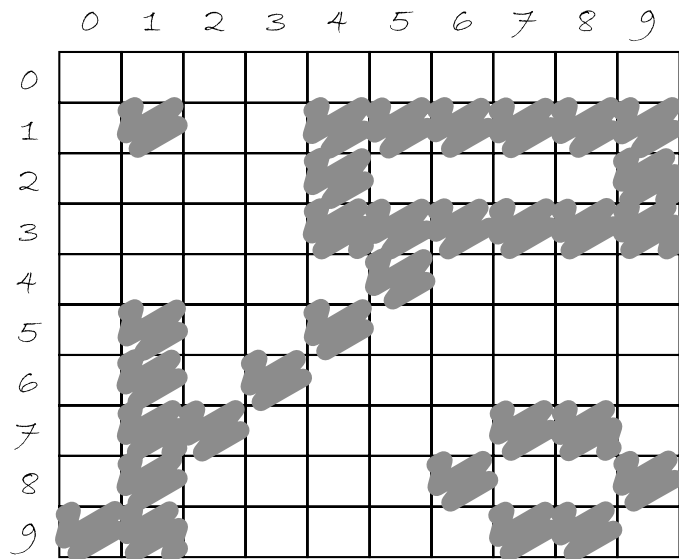
ESERCIZIO

```
line(0,0,9,6);  
point(0,2);  
point(0,4);  
rectMode(CORNER);  
rect(5,0,4,3);  
ellipseMode(CENTER);  
ellipse(3,7,4,4);
```



Disegnare le forme descritte dal codice a sinistra

ESERCIZIO



Note: There is more than one correct answer!

Scrivere un codice per riprodurre il diagramma



COLORI, CONTORNI E RIEMPIMENTI

SCALA DI GRIGIO: LUMINOSITÀ

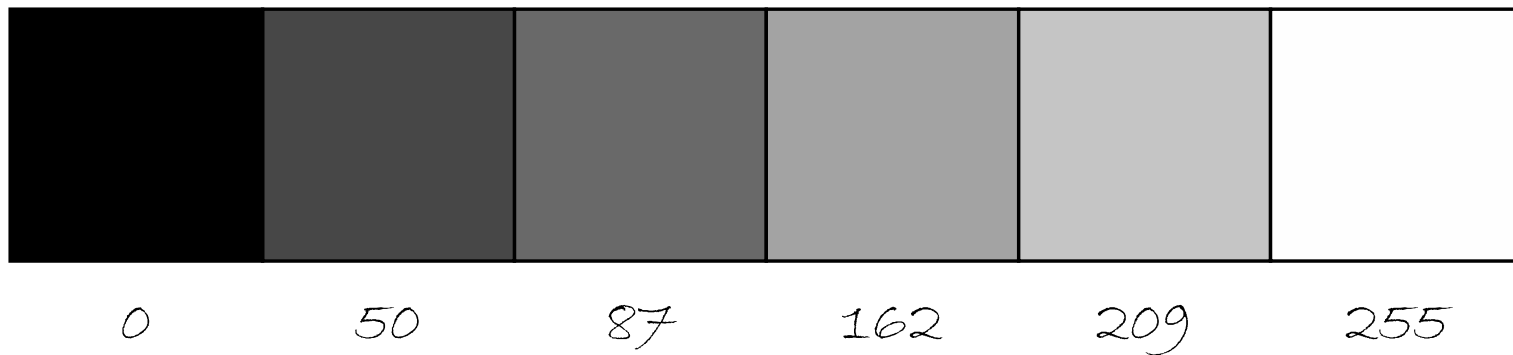


fig. 1.13

I livello di luminosità è rappresentato come un valore compreso tra 0 (nero) e 255 (bianco)

PARTI DI UNA FORMA GEOMETRICA

- Stroke: è la parte che definisce i contorni di una forma
 - Esempio: per un rettangolo sono i 4 lati
- Fill: è la parte interna della forma
 - Esempio: per un rettangolo, l'area racchiusa dai 4 lati



VALORI DI DEFAULT PER stroke E fill

```
rect(50,40,75,100);
```

The background color is gray.

The outline of the rectangle is black

The interior of the rectangle is white

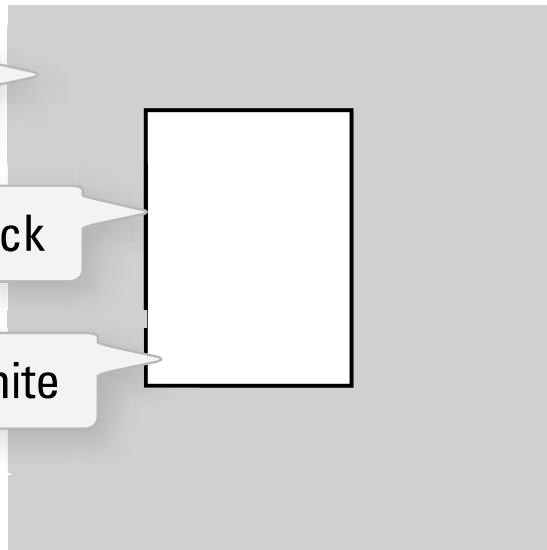


fig. 1.14

strokeEfill

```
background(255) ;  
stroke(0) ;  
fill(150) ;  
rect(50,50,75,100) ;
```

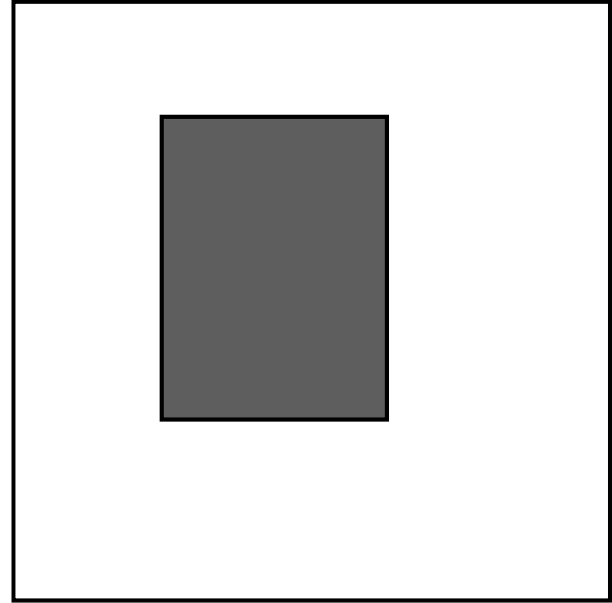


fig. 1.15

stroke Fill

```
background(255);  
stroke(0);  
noFill();  
ellipse(60,60,100,100);
```

If we draw two shapes at one time, *Processing* will always use the most recently specified ***stroke()*** and ***fill()***, reading the code from top to bottom. See Figure 1.17.

noFill() leaves the shape with only an outline

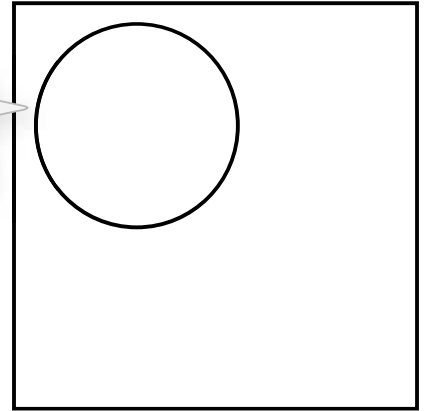


fig. 1.16

strokeFill

```
background(150);  
stroke(0);  
→ line(0,0,100,100);  
stroke(255);  
noFill();  
→ rect(25,25,50,50);
```

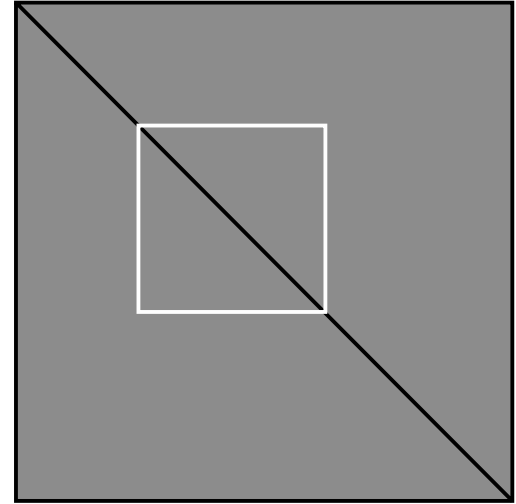
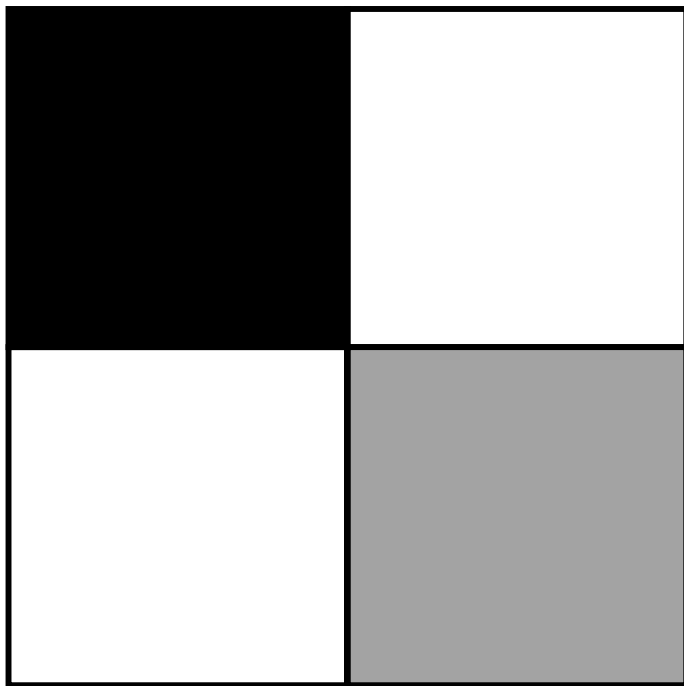


fig. 1.17

ESERCIZIO

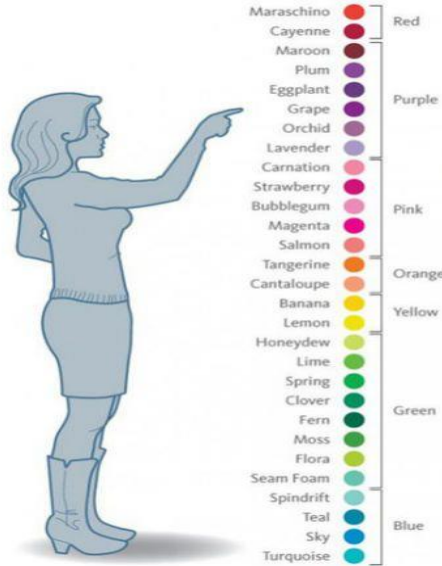


- Scrivere le istruzioni per creare il diagramma a sinistra



HOW MANY COLOR?

Female



Male

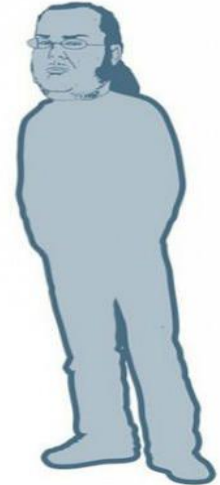


Dog



Programmer

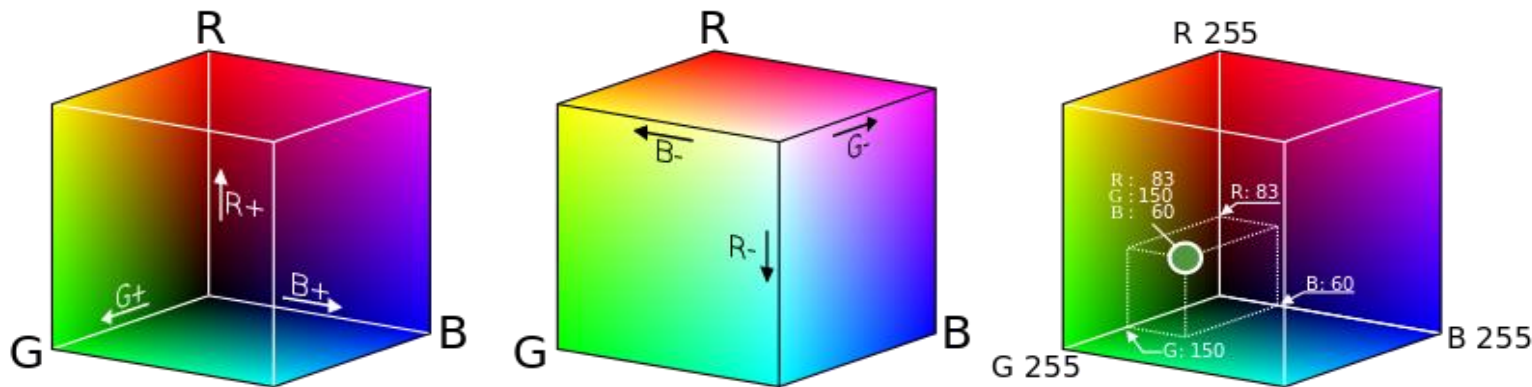
Gray #f94433
Gray #ac203b
Gray #85343d
Gray #874994
Gray #663c84
Gray #8c2590
Gray #a16799
Gray #af99c7
Gray #f38da3
Gray #d2157b
Gray #ec90b7
Gray #e90086
Gray #f57d7e
Gray #f27727
Gray #fc9b7b
Gray #f7d305
Gray #f1e311
Gray #ccdf62
Gray #68bd46
Gray #00a04f
Gray #006965
Gray #005704
Gray #3ba246
Gray #abcf37
Gray #68c3b2
Gray #8bccd0
Gray #0687a7
Gray #078dca
Gray #0fb8b5



We KnowMemes

RGB COLOR MODEL

- Basato sui colori primari
- È un modello additivo: il colore risultante è la somma dei componenti di base



RGB COLOR MODEL

- I valori R,G,B possono essere espressi come numeri nell'intervallo $[0,1]$
- Alcune applicazioni (e anche processing) usano l'intervallo $[0,255]$
- Non è un modello intuitivo. Come definire i valori per un marrone chiaro?



COLORI RGB

Example 1-3: RGB color

```
background(255);  
noStroke();  
  
fill(255,0,0);  
ellipse(20,20,16,16);  
  
fill(127,0,0);  
ellipse(40,20,16,16);  
  
fill(255,200,200);  
ellipse(60,20,16,16);
```

Bright red

Dark red

Pink (pale red).



fig. 1.18

TRASPARENZA

Example 1-4: Alpha transparency

```
background(0);  
noStroke();
```

```
fill(0,0,255);  
rect(0,0,100,200);
```

No fourth argument means 100% opacity.

```
fill(255,0,0,255);  
rect(0,0,200,40);
```

255 means 100% opacity.

```
fill(255,0,0,191);  
rect(0,50,200,40);
```

75% opacity

```
fill(255,0,0,127);  
rect(0,100,200,40);
```

50% opacity

```
fill(255,0,0,63);  
rect(0,150,200,40);
```

25% opacity

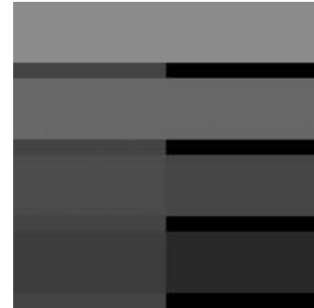


fig. 1.20

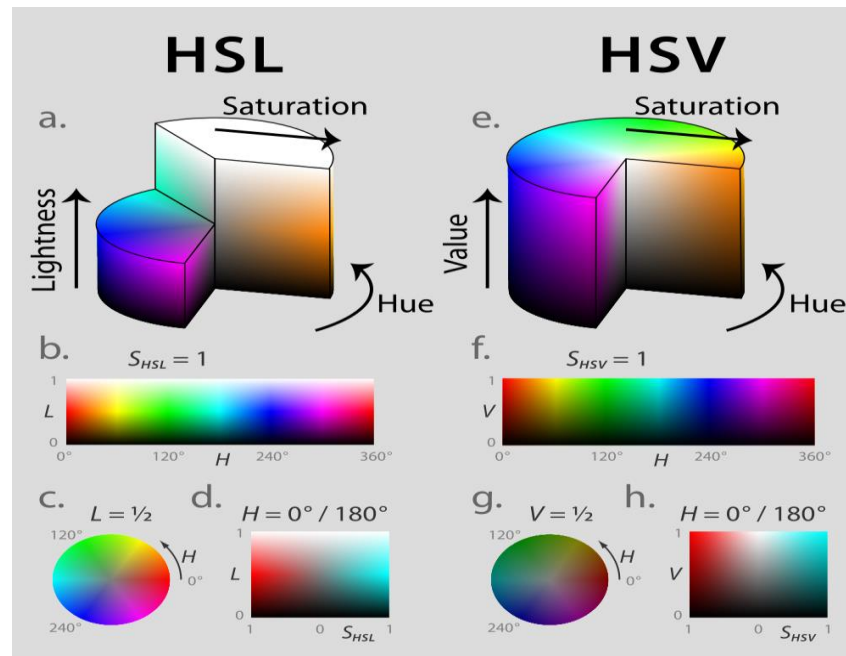
COLOR MODE

- I range di colori sono riconosciuti solo come intervalli tra 0 e 255
- E' possibile definire una mappa diversa utilizzando l'istruzione `colorMode`
 - `colorMode(RGB,100)`
- Alternative color model: HSB
 - Hue: il tipo di colore da usare (varia tra 0 e 360)
 - Saturation: vivicità del colore (varia tra 0 e 100)
 - Brightness: luminosità (varia tra 0 e 100)



HSV COLOR MODEL

- Basato sui concetti intuitivi di
 - tinta
 - saturazione
 - Intensità o valore



"Hsl-hsv models" by Jacob Rus - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons -
http://commons.wikimedia.org/wiki/File:Hsl-hsv_models.svg#/media/File:Hsl-hsv_models.svg

ESERCIZIO

Descrivere una creature usando semplici forme geometriche

Example 1-5: Zoog

```
ellipseMode(CENTER);
```

```
rectMode(CENTER);
```

```
stroke(0);
```

```
fill(150);
```

```
rect(100,100,20,100);
```

```
fill(255);
```

```
ellipse(100,70,60,60);
```

```
fill(0);
```

```
ellipse(81,70,16,32);
```

```
ellipse(119,70,16,32);
```

```
stroke(0);
```

```
line(90,150,80,160);
```

```
line(110,150,120,160);
```

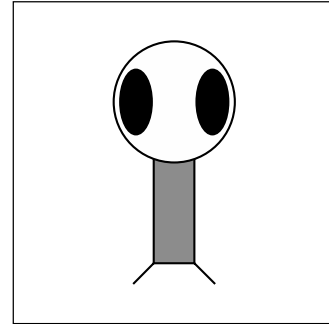
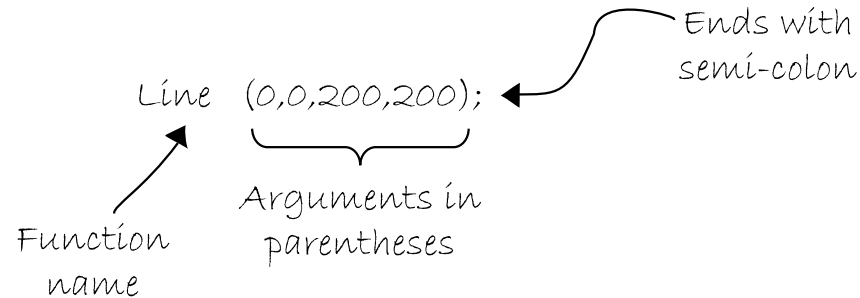


fig. 1.21

CODICE IN PROCESSING

- Possiamo scrivere tre tipi di istruzioni
 - Chiamare delle funzioni
 - Assegnare un valore ad una variabile
 - Strutture di controllo



The diagram shows the code `Line (0,0,200,200);` with handwritten annotations. An arrow points from the word "Line" to the label "Function name". A bracket under the parentheses and their contents is labeled "Arguments in parentheses". Another arrow points from the semicolon to the label "Ends with semi-colon".

Line (0,0,200,200);

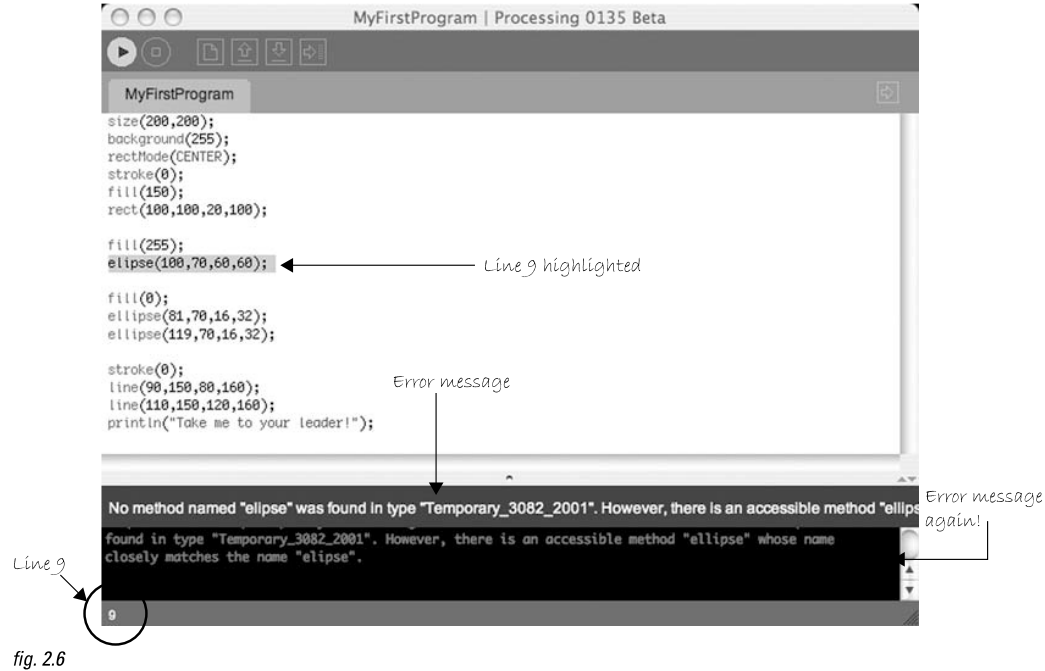
Function name

Arguments in parentheses

Ends with semi-colon

fig. 2.4

MESSAGGI DI ERRORE



ESERCIZIO: TROVA L'ERRORE



Exercise 2-6: Fix the errors in the following code.

```
size(200,200);
```

```
background();
```

```
stroke 255;
```

```
fill(150)
```

```
rectMode(center);
```

```
rect(100,100,50);
```
