



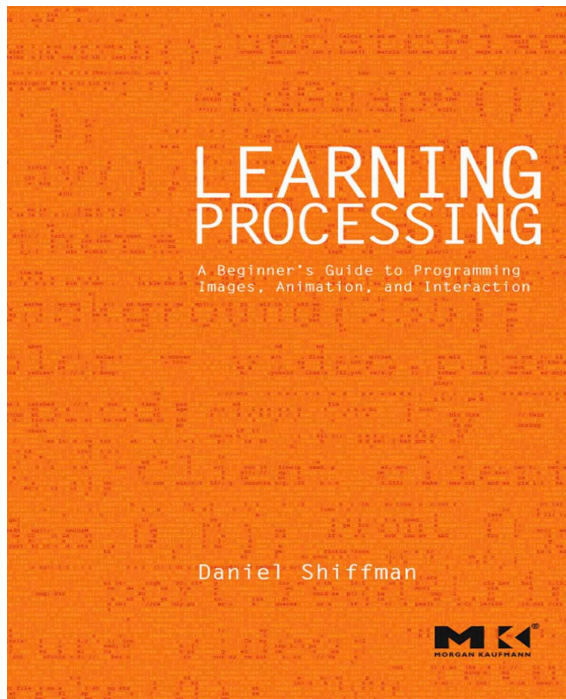
FONDAMENTI DI INFORMATICA

Alma Artis

Salvatore Rinzivillo (ISTI, CNR)

Variabili e Istruzioni Condizionali

LIBRI E RIFERIMENTI



- Introduzione
- Capitolo 1-3
- Registrarsi al sito:
<http://almaartis.rinziv.it/>

Learning Processing– Second Edition

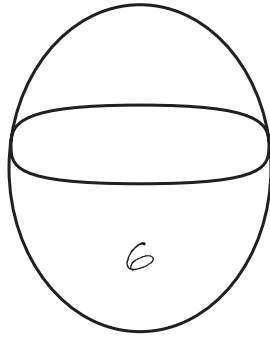
Daniel Shiffman

Available here: <http://learningprocessing.com/>

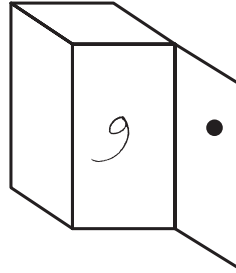


VARIABILI

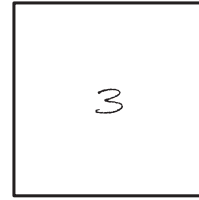
COSA È UNA VARIABILE?



variable
bucket



variable
locker



variable
post-it

fig. 4.1

Un nome per indirizzare una locazione della memoria del computer

USO DI VARIABILI

Jane's Score	Billy's Score
5	10
30	25
53	47
65	68
87	91
101	98

fig. 4.3

- Oltre a memorizzare un valore, una variabile permette di seguire il cambiamento del suo contenuto

ESERCIZIO

- Definire le variabili per il gioco Pong



DICHIARAZIONE DI VARIABILI

- Una variabile può memorizzare valori primitivi o riferimenti ad array e oggetti
- Una variabile deve essere dichiarata prima di essere usata
- Per dichiararla, bisogna specificare
 - il tipo: quale dato dovrà contenere
 - Il nome: come chiamiamo la locazione che conserva il dato

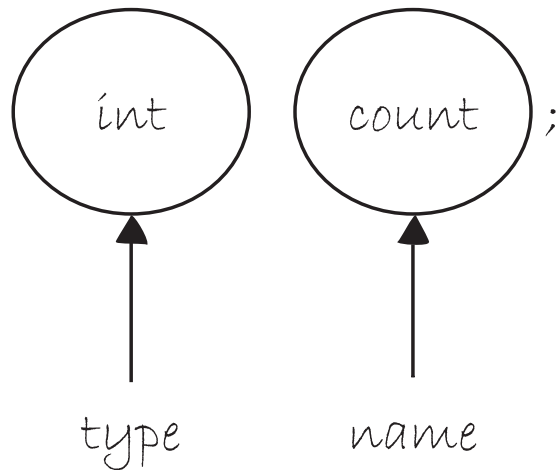


fig. 4.4

TIPI DI VARIABILE

- boolean: true or false
- char: un carattere, 'a', 'b', 'c', etc.
- byte: un numero piccolo, -128 to 127
- short: un numero un poco più grande, -32768 to 32767
- int: un numero intero, -2147483648 to 2147483647
- long: un numero estremamente grande
- float: numero con cifre decimali, such as 3.14159
- double: numero con tante cifre decimali



ASSEGNAZIONE DI UN VALORE

```
int count;  
count = 50;
```

Declare and initialize a variable in two lines of code.

```
int count = 50;
```

Declare and initialize a variable in one lines of code.

ESEMPIO

Example 4-1: Variable declaration and initialization examples

```
int count = 0;           // Declare an int named count, assigned the value 0
char letter = 'a';       // Declare a char named letter, assigned the value 'a'
double d = 132.32;       // Declare a double named d, assigned the value 132.32
boolean happy = false;   // Declare a boolean named happy, assigned the value false
float x = 4.0;            // Declare a float named x, assigned the value 4.0
float y;                  // Declare a float named y (no assignment)
y = x + 5.2;              // Assign the value of x plus 5.2 to the previously declared y
float z = x*y + 15.0;     // Declare a variable named z, assign it the value which
                          // is x times y plus 15.0.
```

USO DI VARIABILI

```
// tra poco aggiungeremo le nostre  
variabili
```

```
void setup(){  
  size(200,200);  
}
```

```
void draw(){  
  background(255);  
  stroke(0);  
  fill(175);  
  ellipse(100,100,50,50);  
}
```



USO DI VARIABILI

```
// tra poco aggiungeremo le nostre  
variabili
```

```
void setup(){  
  size(200,200);  
}
```

```
void draw(){  
  background(255);  
  stroke(0);  
  fill(175);  
  ellipse(mouseX,mouseY,50,50);  
}
```



USO DI VARIABILI

```
// dichiariamo le nostre variabili
int circleX = 100;
int circleY = 100;

void setup(){
    size(200,200);
}

void draw(){
    background(255);
    stroke(0);
    fill(175);
    ellipse(circleX,circleY,50,50);
}
```

ASSEGNAZIONE DI UN NUOVO VALORE

```
// dichiariamo le nostre variabili
int circleX = 0;
int circleY = 100;

void setup(){
  size(200,200);
}

void draw(){
  background(255);
  stroke(0);
  fill(175);
  ellipse(circleX,circleY,50,50);

  circleX = circleX + 1;
}
```

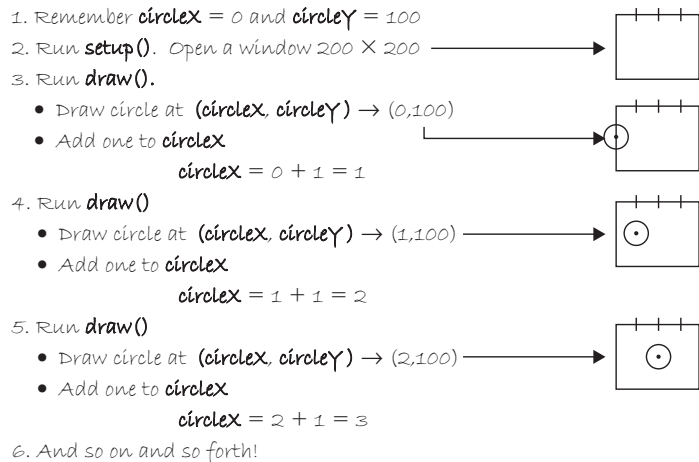


fig. 4.5

ESERCIZIO

- Modifica l'esempio precedente per fare in modo che il cerchio cresca di dimensione

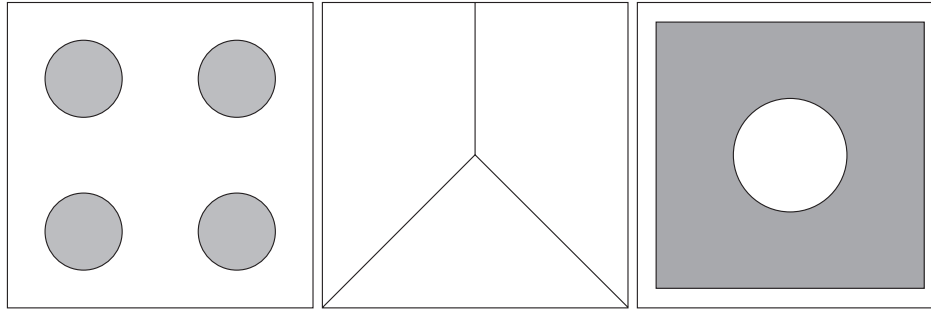


ESERCIZIO



Exercise 4-4

- Step 1:** Write code that draws the following screenshots with hard-coded values. (Feel free to use colors instead of grayscale.)
- Step 2:** Replace all of the hard-coded numbers with variables.
- Step 3:** Write assignment operations in **draw()** that change the value of the variables. For example, “`variable1 = variable1 + 2;`”. Try different expressions and see what happens!



VARIABILI DI SISTEMA

- Processing crea delle variabili prima dell'inizio della esecuzione
- Abbiamo già visto un esempio con mouseX e mouseY
- Ecco un elenco di variabili utili:
 - width: la larghezza della finestra
 - height: la altezza della finestra
 - frameCount: il numero di refresh della finestra dall'inizio dell'esecuzione
 - frameRate: numero di frame disegnati al secondo
 - screen.width: larghezza dello schermo
 - screen.height: altezza dello schermo
 - key: codice dell'ultimo tasto premuto
 - keyPressed: valore boolean, true se viene premuto un tasto



ESEMPIO

Example 4-5: Using system variables

```
void setup() {  
  size(200,200);  
  frameRate(30);  
}
```

```
void draw() {  
  background(100);  
  stroke(255);  
  fill(frameCount/2);  
  rectMode(CENTER);  
  rect(width/2,height/2,mouseX+10,mouseY+10);  
}
```

frameCount is used to color a rectangle.

```
void keyPressed() {  
  println(key);  
}
```

The rectangle will always be in the middle of the window if it is located at (width/2, height/2).



NUMERI CASUALI

NUMERI RANDOM

- La funzione `random()` è una funzione speciale che ritorna un valore casuale
- Rispetto alle altre funzioni viste finora (`ellipse`, `line`, `point`), questa risponde con un valore numerico
- La funzione ha bisogno di due numeri: viene ritornato un numero compreso tra i due
- La funzione ritorna un numero con la virgola: float
 - `float w = random(1,100);`
`rect(100,100,w,50);`



RANDOM ELLIPSES

Example 4-7: Filling variables with random values

```
float r;  
float g;  
float b;  
float a;  
  
float diam;  
float x;  
float y;  
  
void setup() {  
  size(200,200);  
  background(0);  
  smooth();  
}  
  
void draw() {  
  // Fill all variables with random values  
  r = random(255);  
  g = random(255);  
  b = random(255);  
  a = random(255);  
  diam = random(20);  
  x = random(width);  
  y = random(height);  
  
  // Use values to draw an ellipse  
  noStroke();  
  fill(r,g,b,a);  
  ellipse(x,y,diam,diam);  
}
```

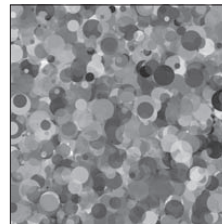


fig. 4.8

Each time through *draw()*, new random numbers are picked for a new ellipse.


VARIABLE ZOOG





ISTRUZIONI CONDIZIONALI

CONTROLLO DEL FLUSSO

- Quando un programma è formato da più istruzioni, queste vengono eseguite in ordine dall'alto verso il basso 
- Il seguente programma è composto da due istruzioni

```
int theNumber = int(random(10, 100));  
println("Your number is " + theNumber);
```



ISTRUZIONI CONDIZIONALI

- Tutti i programmi vengono eseguiti in sequenza
- Queste istruzioni di controllo consentono di scegliere due possibili percorsi, sulla base di un valore booleano
- Sintassi:

```
if (espressione)  
    istruzione1  
else  
    istruzione2
```



- Espressione è una espressione booleana
- Istruzione1 rappresenta il ramo eseguito se la valutazione ritorna true
- Istruzione2 rappresenta il ramo eseguito se la valutazione ritorna false

ESEMPIO

```
if(mouseX < width / 2){  
  fill(255);  
  rect(0,0,width/2,height);  
}
```

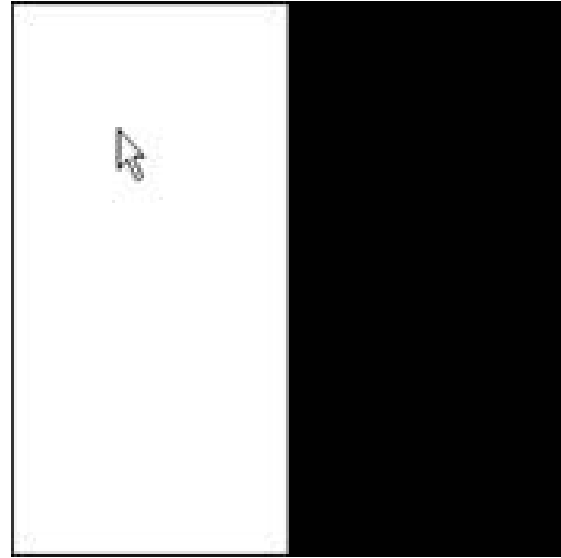


fig. 5.1

ESEMPIO

```
if(mouseX < width/2){  
  background(255);  
}else{  
  background(0);  
}
```

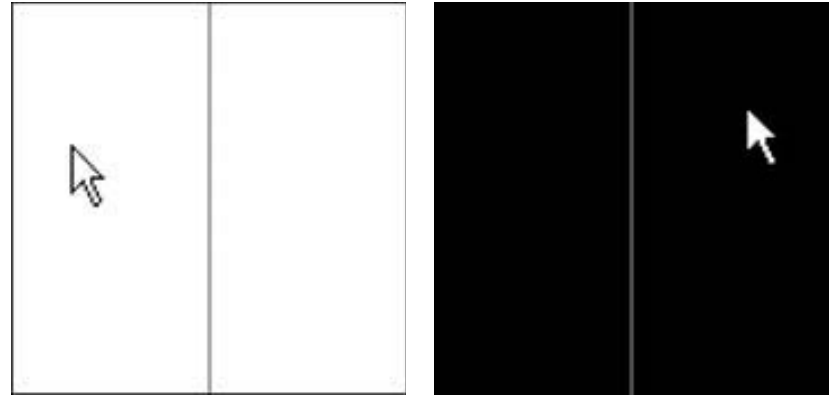


fig. 5.2

CONDIZIONI MULTIPLE

```
if (boolean expression #1) {  
  // code to execute if boolean  
  expression #1 is true  
} else if (boolean expression #2) {  
  // code to execute if boolean  
  expression #2 is true  
} else if (boolean expression #n) {  
  // code to execute if boolean  
  expression #n is true  
} else {  
  // code to execute if none of the  
  above  
  // boolean expressions are true  
}
```

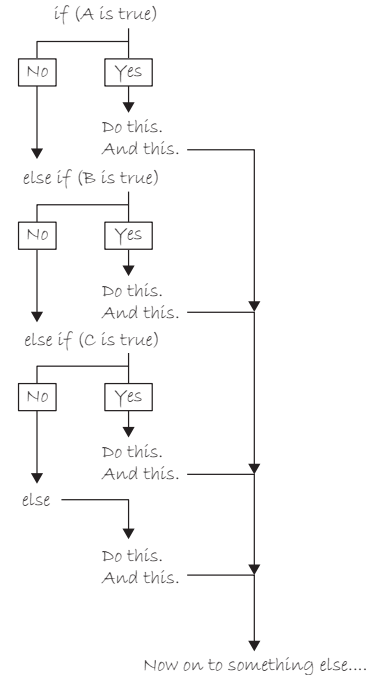


fig. 5.3

ESEMPIO

```
if (mouseX < width/3) {  
    background(255);  
} else if (mouseX < 2*width/3) {  
    background(127);  
} else {  
    background(0);  
}
```

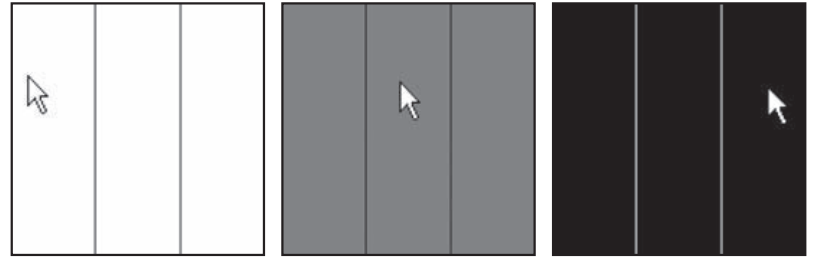


fig. 5.4

SKETCH CONDIZIONALI

Example 5-1: Conditionals

```
float r = 150;  
float g = 0;  
float b = 0;
```

1. Variables.

```
void setup() {  
  size(200,200);  
}
```

```
void draw() {  
  background(r,g,b);  
  stroke(255);  
  line(width/2,0,width/2,height);
```

2. Draw stuff.

```
if(mouseX > width/2) {  
  r = r + 1;  
} else {  
  r = r - 1;  
}
```

3. "If the mouse is on the right side of the screen" is equivalent to "if **mouseX** is greater than width divided by 2."

```
if (r > 255) {  
  r = 255;  
} else if (r < 0) {  
  r = 0;  
}  
}
```

4. If *r* is greater than 255, set it to 255.
If *r* is less than 0, set it to 0.

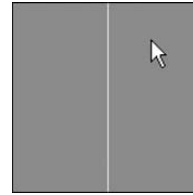


fig. 5.5

FUNZIONE constrain()

```
if (r > 255) {  
    r = 255;  
} else if (r < 0) {  
    r = 0;  
}
```

Constrain with an “if” statement.

```
r = constrain(r, 0, 255);
```

Constrain with the ***constrain()*** function.

