*Uso delle variabili*

Ecco l'introduzione di una variabile intera di nome x a cui si assegna inizialmente il valore 0 con la *dichiarazione*

int x = 0

e dopo il disegno di ogni linea è incrementata di 10 unità con il *comando di assegnamento*

x=x+10 .

*Una dichiarazione ha il compito di creare una variabile stabilendo il nome e l'insieme dei valori che la variabile può assumere (ovvero il tipo della variabile, in questo caso di tipo int che denota i numeri interi), nonché di riservare nella memoria del calcolatore lo spazio necessario per memorizzare il valore corrente della variabile.*

In Processing come in Java e tanti altri linguaggi, una variabile non può essere usata se non è prima dichiarata. Invece vedremo che in JavaScript la dichiarazione di una variabile non è necessaria.

L'effetto finale del programma è sempre lo stesso, ma, a differenza del programma precedente, il codice mette bene in evidenza la relazione tra le coordinate delle 5 linee.



**Iterazione**

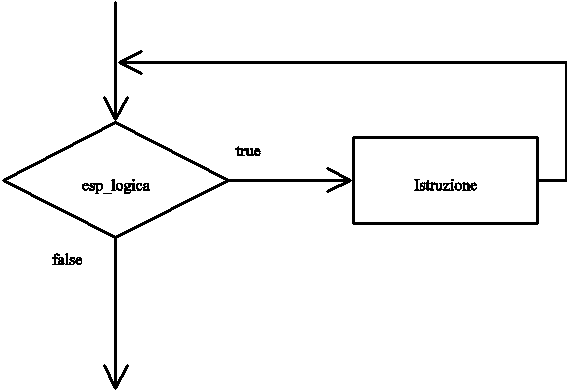
I costrutti di iterazione eseguono un certo numero di volte un'istruzione, detta *corpo*. Un costrutto di iterazione è anche detto *istruzione di ciclo*, o semplicemente *ciclo*.

Noi esamineremo due tipi di costrutti di iterazione:

*comando while*

while (<espressione logica> ) I

L'esecuzione di un comando while causa l'esecuzione dell'istruzione I (il *corpo* del while) se l’espressione logica (la *guardia* del while) è vera, poi si valuta di nuovo l’espressione logica: se è vera si esegue di nuovo l’istruzione, e così via finché l’espressione logica diventa falsa.



*Esempio di uso del comando while*

Osservando il codice degli esempi precedenti, le cinque righe parallele sono disegnate incrementando di 10 unità la prima componente dei punti estremi di ogni retta.

Si ha quindi una ripetizione (5 volte) dell'esecuzione del comando di tracciatura delle linee.

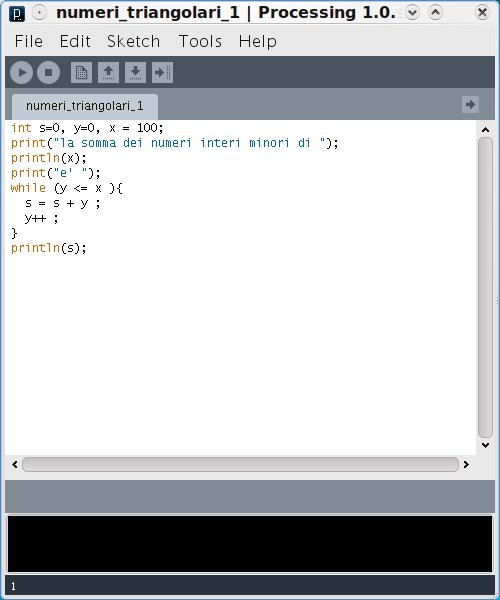
La tracciatura delle linee può avvenire senza scrivere 5 volte lo stesso comando, usando il comando while.



Il corpo del ciclo while disegna una linea ed incrementa il valore della variabile intera x di 10 unità.

Pertanto, la variabile x assume i valori 0, 10, 20, 30, 40, in corrispondenza dei quali una linea viene disegnata. Quando x assume il valore 50, la guardia del ciclo while diventa falsa e l'esecuzione del ciclo termina.

La figura seguente mostra come il ciclo while possa essere usato per calcolare i numeri triangolari, ovvero i numeri interi positivi ottenuti come somma dei numeri interi compresi tra 1 e un intero positivo n.



Il programma funziona in modo analogo al diagramma a blocchi discusso in una lezione precedente (teoria degli algoritmi).

*comando for*

for (I1 ; E ; I2 ) I3

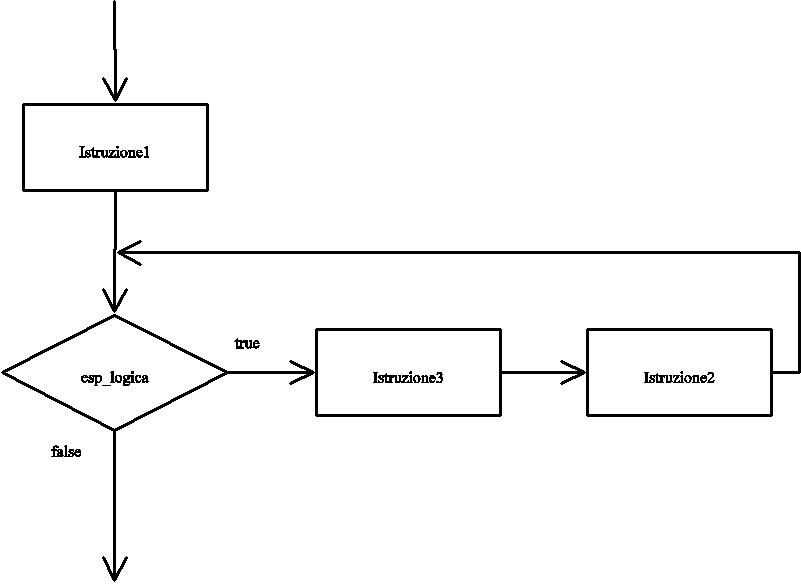
I1 è l’istruzione di inizializzazione;

E è la guardia del for ;

I2 è l’istruzione di incremento o aggiornamento;

I3 è il corpo del for ;

l'esecuzione di un comando for avviene come segue: innanzi tutto è eseguita l'istruzione I1 di inizializzazione (che di solito assegna un valore ad una variabile, detta variabile di controllo del ciclo for); se la guardia è vera si esegue il corpo I3 e l’aggiornamento I2 , poi si valuta di nuovo la guardia: se è vera si esegue di nuovo il corpo e l’aggiornamento, finché la guardia diventa falsa.



*Esempio di uso del comando for*

Questo esempio è una riscrittura del programma precedente usando il ciclo for. Si noti che l'istruzione che assegna 0 alla variabile x è l'istruzione di inizializzazione del ciclo for, mentre l'istruzione che incrementa di 10 unità il valore di x è l'istruzione di aggiornamento del ciclo for. Il corpo del for è costituito dalla sola istruzione di tracciatura di una linea e viene ripetuto 5 volte.



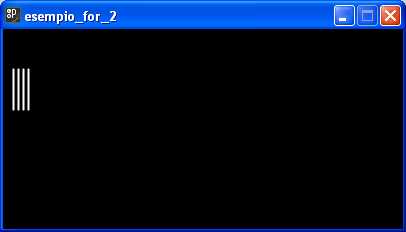
*Attenzione: le istruzioni di ciclo non sono solo dei semplici sostituti del comando di sequenzializzazione che permettono di abbreviare i programmi, ma permettono di fare molto di più. Se non usiamo i cicli e vogliamo disegnare 10 righe anziché 5, dobbiamo modificare il programma, aggiungendo 5 nuovi comandi di disegno.*

Il prossimo esempio mostra come con le istruzioni di iterazione sia possibile disegnare un numero di linee che dipende dal numero di secondi del tempo in cui si lancia l'esecuzione del programma: non è pertanto stabilire a priori quante linee si devono disegnare e un tale programma non si può realizzare con la sola sequenzializzazione di comandi.



Alla variabile intera s è assegnato il numero dei secondi letti dall'orologio del calcolatore.

Nell'immagine precedente il testo che appare nella finestra del terminale indica che il valore assegnato ad s è 4. Il programma disegnerà pertanto quattro righe, come mostrato nella figura seguente:



Il corpo

line(5\*x+10, 40, 5\*x+10, 80);

del comando for successivo che esegue la tracciatura di una linea verticale è eseguito un numero di volte pari al valore della variabile s: infatti, la variabile x di conteggio del for è inizializzata a 0 e fintanto che il suo valore è minore di quello di s, si traccia una linea e si incrementa il valore di x.

*In generale, se s è una variabile intera con un valore definito, l'esecuzione del ciclo*

for(int x = 0; x<s; x=x+1){  
 /\* corpo del for \*/  
};

*provoca l'esecuzione ripetuta s volte del corpo del for.*

Con le istruzioni di ciclo, si può incappare nel cosidetto "fencepost error" (errore dello steccato e dei pali), che consiste nell'eseguire il corpo del ciclo una volta in più o in meno del numero corretto di iterazioni. Per essere sicuri che il programma funzioni correttamente, provate ad assegnare dei valori piccoli alla variabile s, ad es. 0 o 1 e vedete cosa succede: se le prove hanno successo, allora il programma *dovrebbe* essere corretto.

Per concludere l'analisi di quest'esempio, si noti il *commento multiriga* in fondo al codice. Si tratta di un tipo di commento che inizia con i simboli "/\*" , termina con i simboli "\*/" e può occupare un numero arbitrario di righe di testo.

**Selezione**

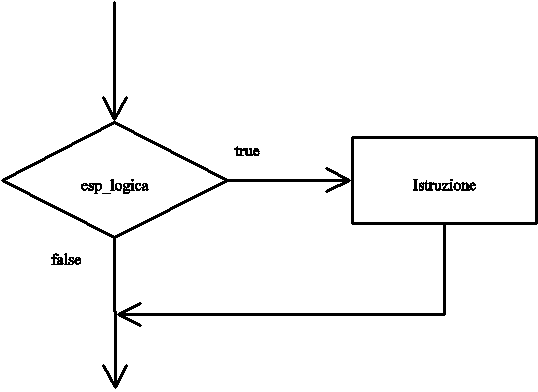
I costrutti di selezione scelgono di eseguire un'istruzione piuttosto che un'altra in base al valore di un'espressione logica ( espressione che assume solo due valori, *vero* o *falso*, e pertanto una tale espressione è anche detta *espressione booleana*, da George Boole, matematico e logico del diciannovesimo secolo).

Noi esamineremo due tipi di costrutti di selezione:

*comando if*

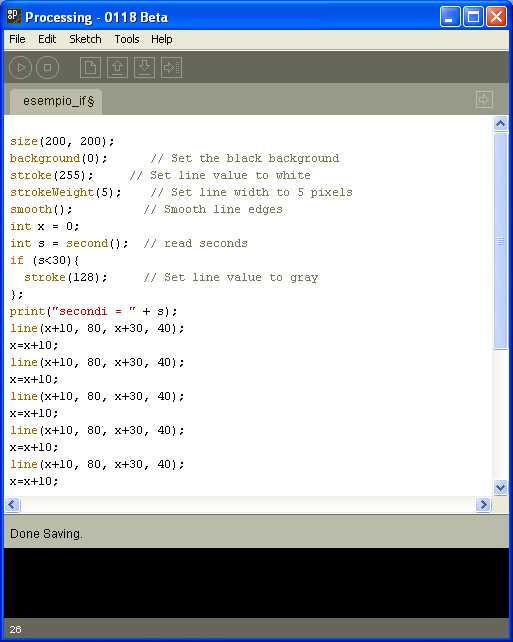
if (<espressione logica> ) I1

L'esecuzione di tale comando causa l'esecuzione dell'istruzione I1 se l’espressione logica (detta *guardia*) è vera



*Esempio di uso del comando if*

Il seguente programma disegna 5 righe parallele di colore grigio se, al momento dell'esecuzione del programma, l'orologio del calcolatore indica un'orario in cui il valore dei secondi è inferiore a 30; in caso contrario, le righe sono di colore bianco. La decisione di quale colore applicare è fatta con il costrutto if.



Il colore delle linee da tracciare viene preimpostato con la terza riga di codice

stroke(255)

che sceglie il colore bianco. Il successivo comando if

if (s<30){

stroke(128); // Set line value to gray

};

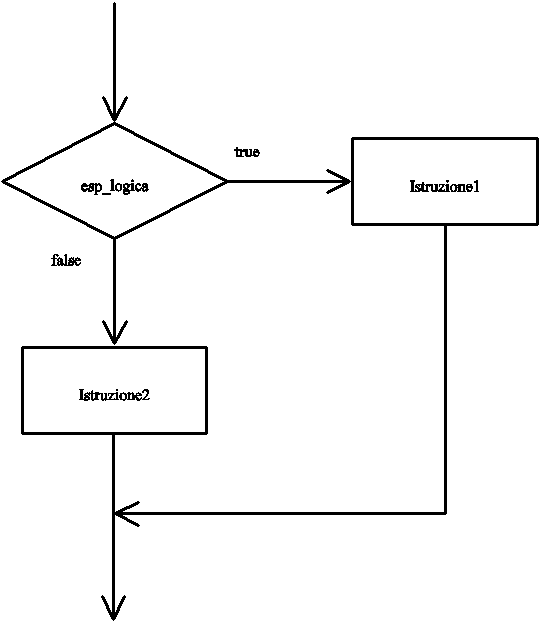
imposta con l'istruzione stroke(128)il colore grigio come colore di tracciatura se il numero dei secondi, contenenuto nella variabile s è minore di 30.

*comando if-else*

if (<espressione logica> ) I1

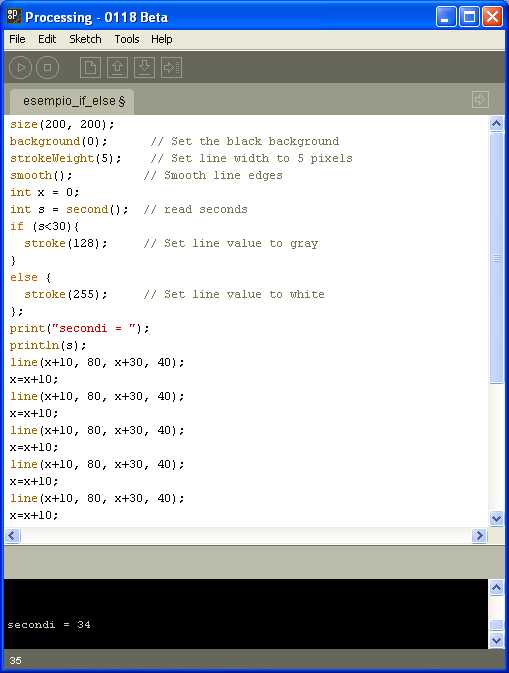
else I2

L'esecuzione di tale comando causa l'esecuzione dell'istruzione I1 se l’espressione logica è vera; dell'istruzione I2 in caso contrario.



*Esempio di uso del comando if-else*

Il seguente programma disegna 5 righe parallele di colore grigio se, al momento dell'esecuzione del programma, l'orologio del calcolatore indica un'orario in cui il valore dei secondi è inferiore a 30; in caso contrario, le righe sono di colore bianco. La decisione di quale colore applicare è fatta con il costrutto if-else



A differenza dell'esempio precedente, il colore delle linee da tracciare non viene preimpostato, ma viene determinato dal comando if-else

if (s<30){

stroke(128); // Set line value to gray

}

else {

stroke(255); // Set line value to white

};

che imposta con l'istruzione stroke(128)il colore grigio come colore di tracciatura se il numero dei secondi, contenenuto nella variabile s è minore di 30, mentre in caso contrario il colore di tracciatura è impostato al colore bianco con l'istruzione stroke(255).