

Come già scritto nello scorso "note", gli **algoritmi** sono una serie di passi finiti che possono essere trasformati in una nuova informazione, in maniera più semplice, "un insieme di passi finiti nel tempo, che uno sviluppatore deve compiere per arrivare a un certo risultato".

Ovviamente i "passi" di un algoritmo vanno eseguiti in maniera **ordinata**, partendo da un insieme di **dati iniziali** per arrivare alla fine, cioè a un **insieme di dati finali**(risultato) ;

Esempio di un'algoritmo: se fuori piove, prendo l'ombrello, altrimenti non piove non prendo l'ombrello;

Potrà sembrare un esempio stupido, **ma questo è un algoritmo**, semplice ma sempre un'algoritmo.

Possiamo quindi affermare che un algoritmo è un **ragionamento** compiuto per arrivare a un risultato ben definito;

Inoltre ogni algoritmo può essere **rappresentato** con i **diagrammi di flusso** o **diagrammi a blocchi** o **flow chart**, i diagrammi di flusso utilizzano **quattro figure**(ovale, rombo, rettangolo, parallelogramma) ognuna di queste figure indica una determinata "funzione":

Input e output:

Operazioni **aritmetiche** e di **assegnazione:**

Condizioni:

Inizio e fine algoritmo:

Quindi, infine, possiamo dire che un **algoritmo** è un **ragionamento compiuto** ben definito per arrivare a un **determinato risultato** e che quest'ultimo **si può rappresentare attraverso i diagrammi di**

CONDIZIONI

All'interno di un **algoritmo** è spesso **necessario eseguire istruzioni diverse** a seconda dei dati. Per questo avremo il bisogno di **verificare** se delle **condizioni** sui nostri dati siano **soddisfatte** o meno;

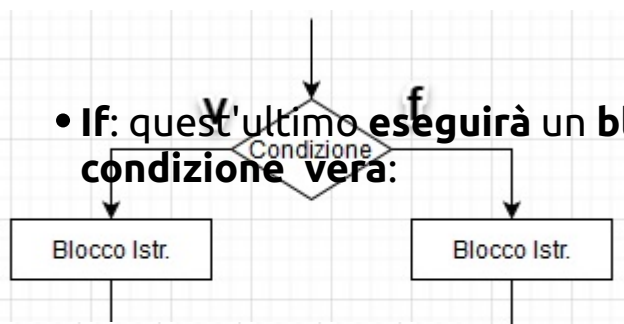
Come già detto in precedenza, per **rappresentare** tramite un diagramma di flusso una **condizione**, si usa il **rombo**;

Quest'ultimo eseguirà (a livello del digramma) un blocco istruzioni se la condizione sarà vera, e un altro blocco quando quest'ultima sarà falsa, in base alla struttura utilizzata;

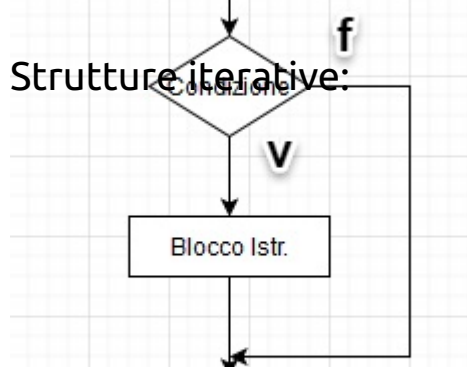
Le strutture esistenti in ogni linguaggio di programmazione si dividono in:

Strutture selettive o condizionali:

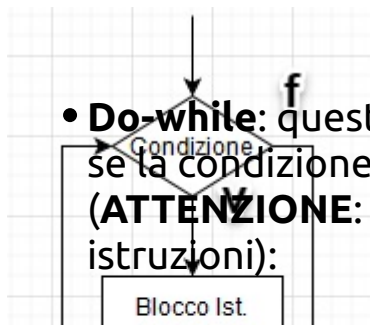
- **If-else**: quest'ultimo **eseguirà** un **blocco istruzioni** quando la **condizione** sarà **falsa** o **vera**:



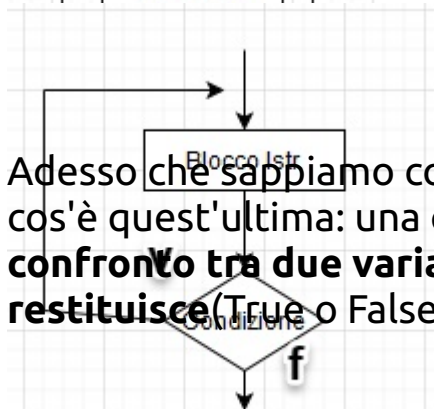
Strutture iterative:



- **While:** quest'ultima eseguirà un blocco istruzioni fin che la condizione sarà vera:



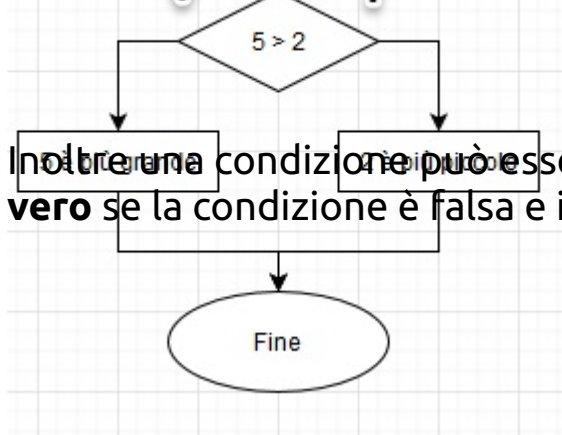
- **Do-while:** quest'ultima eseguirà prima il blocco istruzioni, poi, se la condizione è vera lo ripeterà fin che non sarà falsa; (**ATTENZIONE:** nel vero non si può eseguire un blocco istruzioni):



Adesso che sappiamo come utilizzare una condizione spieghiamo cos'è quest'ultima: una **condizione** può essere definita come un **confronto tra due variabili o non**, che in base al **risultato** che **restituisce** (True o False), **cambierà** il corso del **programma**;

Esempio di una condizione:

In questo caso se $5 > 2 = \text{true}$, visualizza "5 è più grande" invece se $5 > 2 = \text{false}$, visualizza "2 è più piccolo";



Inoltre una condizione può essere **invertita** con **(!)** così che esegue il **vero** se la condizione è falsa e il **falso** se la condizione è vera;