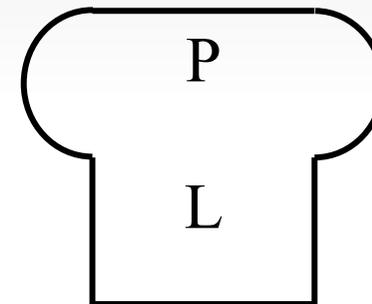


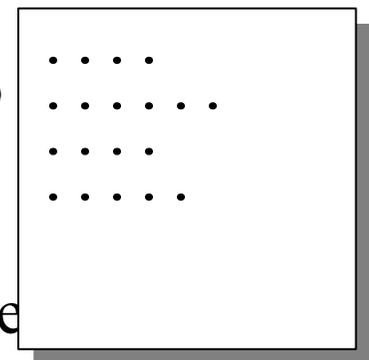
# Programmi

- I linguaggi di programmazione servono per scrivere programmi.
- Il codice di un programma si dice anche codice sorgente (source code)

Un programma P scritto nel linguaggio L



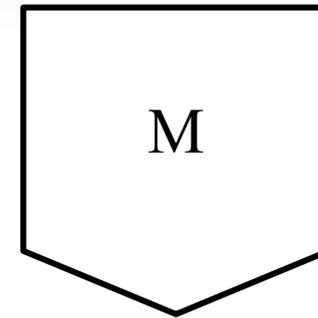
Si tratta di codice sorgente di P e' scritto nel linguaggio L che in genere e' memorizzato su un file



# Il linguaggio macchina

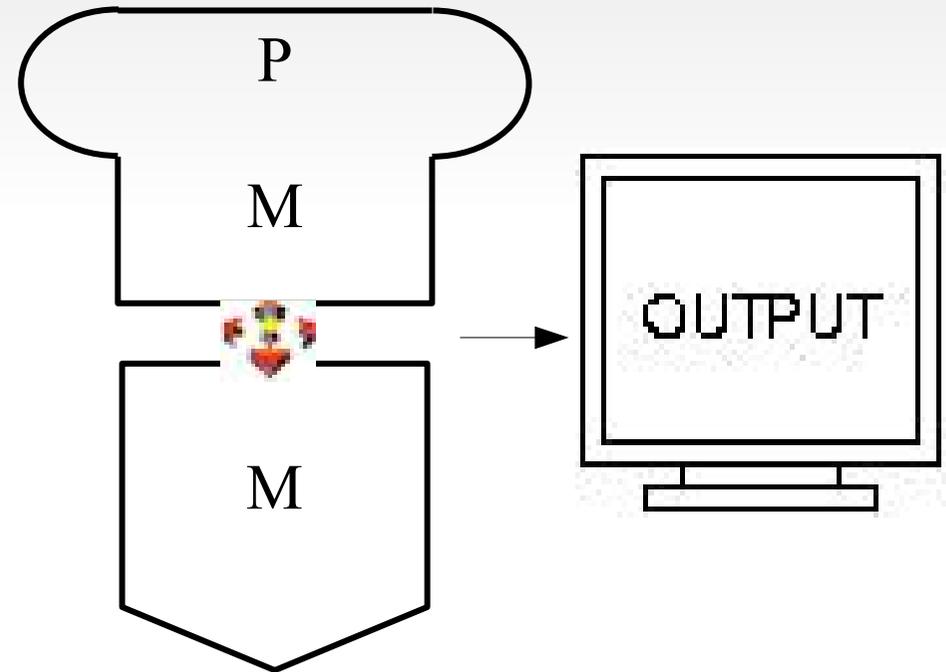
- Solo il linguaggio macchina e' eseguibile direttamente da un computer
- Ogni computer ha un suo linguaggio macchina

Un computer con linguaggio macchina M



# Esecuzione

- Ogni computer e' in grado di eseguire programmi solo nel suo linguaggio macchina



# Problematiche

- Difficile e dispendioso scrivere programmi in linguaggio macchina e in generale con i linguaggi a basso livello.
- Accadeva una volta con i primi computer.
- Ora i linguaggi a basso livello si usano solo in poche applicazioni molto specializzate.

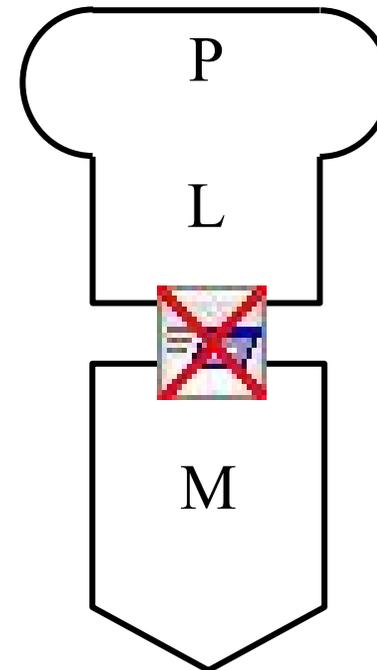
# Vantaggi dei linguaggi ad alto livello

- E' piu' semplice programmare con i linguaggi ad alto livello.
- I programmi si scrivono piu' velocemente.
- I programmi scritti con linguaggi ad alto livello sono **portabili** (eseguibili su computer diversi con poche o nessuna modifica).
- Mentre un programma in linguaggio macchina gira solo sul un certo tipo di computer.

# Pero'

- I programmi scritti nei linguaggi ad alto livello non si possono eseguire direttamente sui computer.

Un programma P scritto nel linguaggio ad alto livello L non può essere eseguito su un computer con linguaggio macchina M.



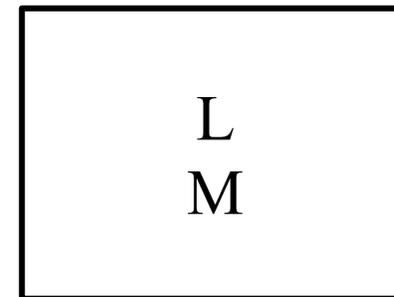
# Esecutori per linguaggi di programmazione

- Sono necessari dei programmi di supporto per eseguire il linguaggi ad alto livello:
  - Interpreti.
  - Compilatori.

# Interpreti

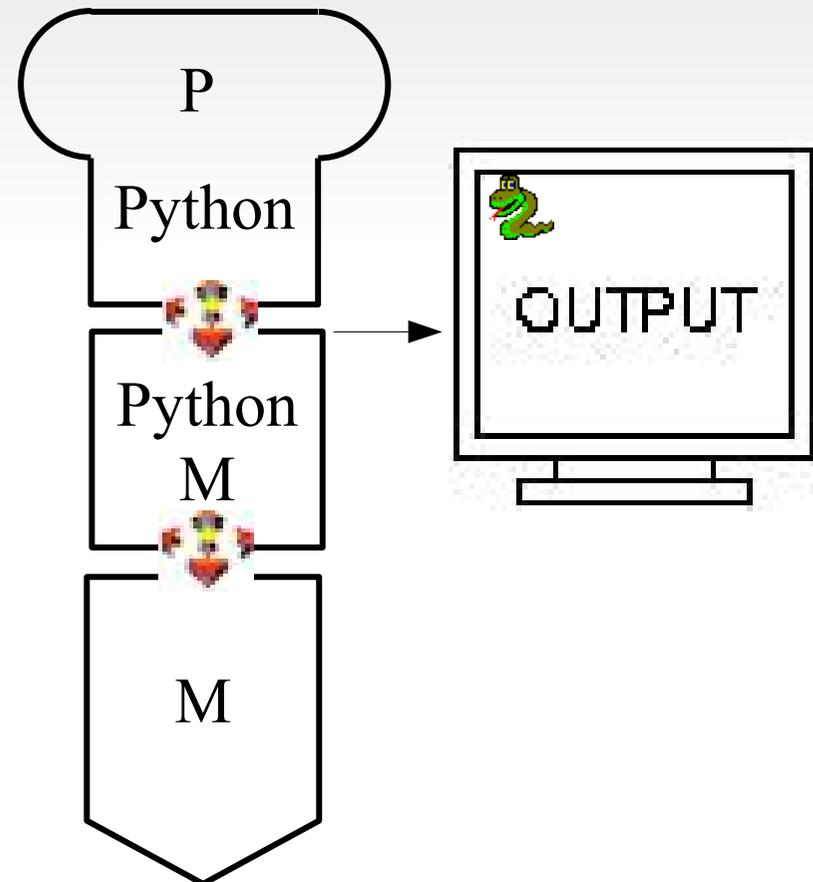
- Un interprete per un linguaggio L permette di eseguire programmi scritti nel linguaggio L.
- Un interprete per un linguaggio L e' un programma che prende come ingresso un programma sorgente scritto nel linguaggio L e lo esegue istruzioni dopo istruzione.
- Un interprete può essere scritto in linguaggio macchina.

Un interprete per il linguaggio L scritto in M.



# Esecuzione di programmi python

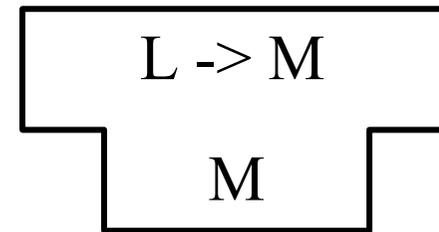
- L'interprete python permette di eseguire un programma P scritto in python in una macchina M.
- **NB. Python e' un linguaggio interpretato, in altre parole un linguaggio eseguito con un interprete.**



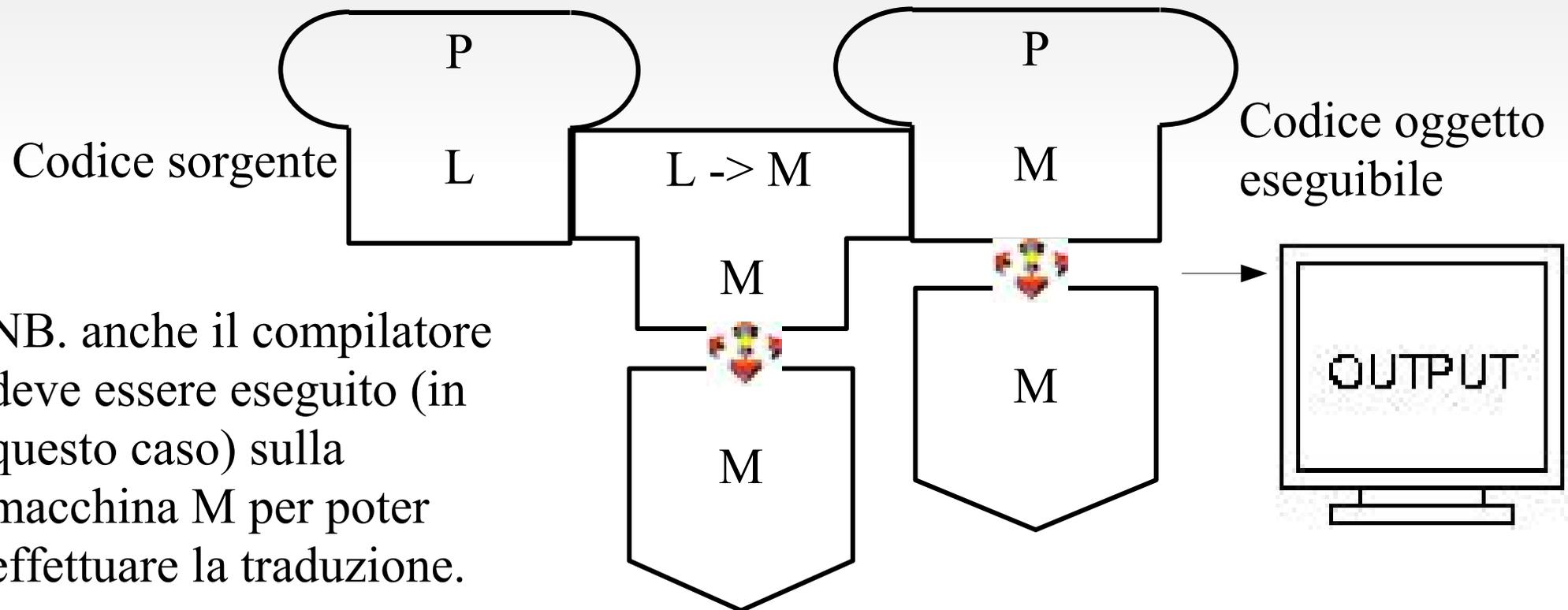
# I compilatori

- L'alternativa all'interprete è tradurre i programmi sorgenti in linguaggio macchina.
- Questo è il ruolo del compilatore.
- Anche il compilatore è un programma che traduce codice sorgente di un linguaggio ad alto livello in un linguaggio a basso livello (in genere linguaggio macchina).

Un compilatore che traduce **codice sorgente** scritto nel linguaggio L in **codice oggetto** del linguaggio macchina M.



# Un programma scritto in un linguaggio ad alto livello L una volta tradotto da un compilatore diventa eseguibile.



# Ma cosa è un programma?

- Un programma è una sequenza di istruzioni che specifica come eseguire un certo calcolo detto anche **computazione**.
- Una computazione può essere di tipo matematico ma può essere anche di tipo simbolico. Ad esempio cercare e rimpiazzare alcune frasi in un documento di testo.

# Alcune tipologie di istruzioni di base comuni a più linguaggi di programmazione

- Input
- Output
- Matematiche
- Esecuzioni condizionali
- Ripetizioni (cicli)
- La programmazione si può vedere come un processo che scompone i problemi da risolvere in tanti **sotto problemi** fino a quando questi sono risolvibili con una di queste istruzioni di base.

# Bugs e Debugging!

- Bug (o baco in italiano)



- Debugging (o debuggare! in italiano)



- Si tratta di una capacita' necessaria per poter programmare.
- NB. Non esistono mai programmi senza errori!
- Se un programma non funziona....si parte con il debugging!



# Tipologie di bugs (errori)

- **Errori sintattici** --> “syntax error”  
il programma e' scritto male e non puo' essere eseguito. Se c'e' un errore sintattico l'interprete python termina immediatamente.
- **Errori a tempo di esecuzione** (runtime) sono spesso chiamati eccezioni (**exceptions**). Rari nei programmi piccoli.
- **Errori semantici**: il programma viene eseguito con successo e termina ma non fa' quello per cui era stato scritto. Questo vuol dire che il significato del programma, **la semantica**, e' sbagliato.

# Debugging

- Task creativo come quello di un investigatore.
- Come mai un programma non funziona?
- In genere si procede per ipotesi e si prova... il programma corretto.
- Se l'ipotesi era sbagliata bisogna cercarne un'altra fino ad escludere tutte le possibilità'.

# Linguaggi formali e naturali

- **Linguaggi naturali:** sono le lingue normalmente parlate nelle varie nazioni.
- **Linguaggi formali:** linguaggi progettati per applicazioni specifiche. Per esempio i linguaggi della matematica o della logica.
- **I linguaggi di programmazione** sono linguaggi formali progettati per esprimere calcoli (computazioni)

# Linguaggi Formali: la sintassi

- Regole strette rispetto alla sintassi.
- Sintassi organizzata a due livelli:
  - **token**: singoli elementi che costituiscono un programma ad esempio: l'istruzione print, il numero 1, l'operatore +, l'operatore =, etc etc....
  - **strutture di frase**: regole grammaticali che dicono come comporre i token per realizzare un programma.
- Il **parsing** e' quel processo che ci permette date le regole grammaticali di “comprendere” la struttura di una frase.

# Differenze tra linguaggi formali e naturali

- **Ambiguità:** Il linguaggio naturale è pieno di ambiguità, mentre i linguaggi formali sono stati progettati per essere quasi o completamente non ambigui.
- **Ridondanza:** tipicamente i linguaggi naturali sono molto ridondanti, ci sono molti modi per dire le stesse cose. I linguaggi formali sono meno ridondanti e più concisi.
- **Nonletteralità:** I linguaggi naturali hanno molti idiomi e metafore che non sono presenti nei linguaggi formali.

# Linguaggi formali semantica

- Semantica: esprime il significato dei costrutti del linguaggio.
- Può essere informale (ad esempio un manuale) o formale (espressa tramite formalismo matematico).
- Diverse tipologie di semantica:
  - Operazionale: descrive in modo preciso il funzionamento dei costrutti del linguaggio.
  - Dichiarativa: esprime il significato dei costrutti del linguaggio dicendo cosa esprimono ma non come funzionano.

# Classificazione dei Linguaggi di Programmazione

- Linguaggi imperativi
  - Sequenze di istruzioni: la computazione si ottiene eseguendo le istruzioni una dopo l'altra (macchina di Von Neumann).
- Linguaggi dichiarativi:
  - Funzionali: la computazione e' espressa tramite il calcolo di funzioni.
  - Logici: la computazione e' espressa come inferenza logica.