

OCaml: Guida all'installazione e Uso - 1

La nostra piattaforma (Linux, Mac, Windows, ...) deve avere installato il pacchetto di strumenti di Ocaml che [comprendono](#)):

- **ocaml**: Top Level Interpreter (di tipo REPL)
- **ocamlrun**: RunTime System
- **ocamlc**: Compilatore su bytecode
- **ocamlopt**: Compilatore su codice nativo

L'ultima release di OCaml è la 4.12.0 (beta)

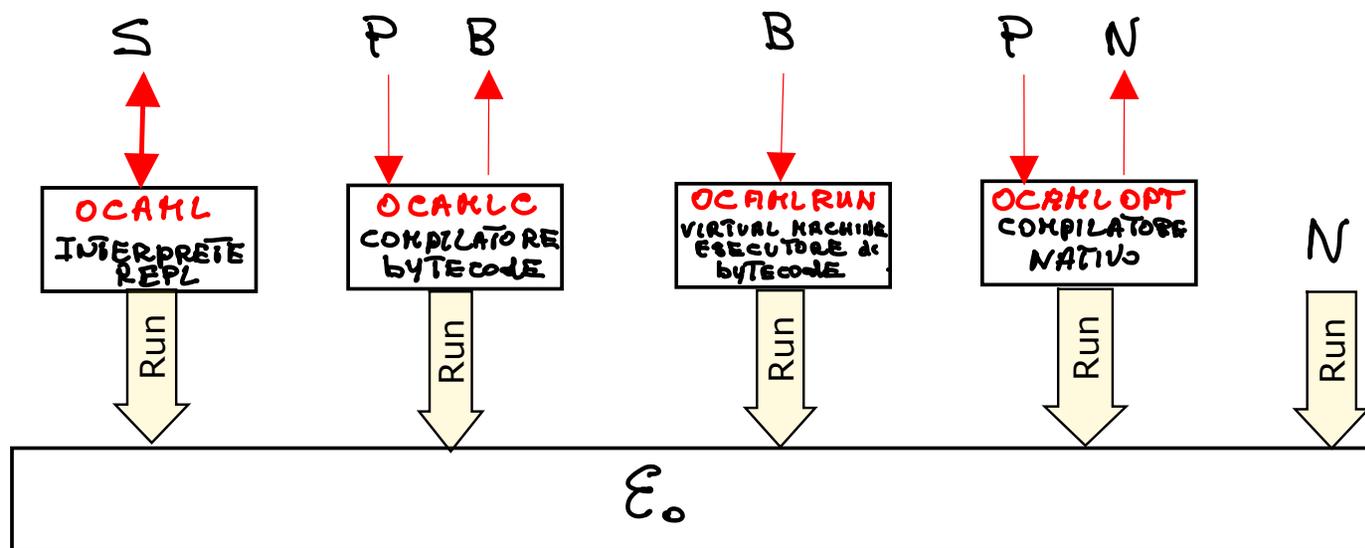
Chi ha già OCaml: Per noi è sufficiente una qualunque versione successiva alla 4.0.0

Chi non ha OCaml: Può installare la release 4.11.1 (stabile) (vedi [InstalloCaml](#))

Chi non sa: Apra una shell o prompt e usi il comando (which, where, locate, find, ...) di ricerca pacchetti (OPAM, Cygwin, OCaml)

OCaml: Documentation and User's Manual.

- Accedere alle Pagine del Corso (Materiale in Organizzazione Corso).
- ocaml.org



- S: Espressioni linguaggio OCAML
- P: Programmi linguaggio OCAML
- B: Programmi linguaggio OCAML-bytecode
- N: Programmi linguaggio macchina di Ho con esecutore E_0

OCaml: Guida all'installazione e Uso - 2

● Download e Installazione: Guida Essenziale.

- (a) Connettersi a ocaml.org: Sito di riferimento per sviluppatori e utenti.
- (b) Selezionare "Install OCaml"
- (c) Vediamo installazioni diverse per:
 - **Avvertenza**, Disinserire Controlli/Protezioni su Download o Accessi Internet
Reinserire ad installazione completata.
Se l'installazione è interrotta occorre rimuovere directories e file installati.
 - **Linux, macOS:**
 - Disporre di [Installer](#) di pacchetti:
Comandi brew, port (in macOS) e apt, apt-get, pac-mac,... (in Linux)
Altrimenti installarli: [Homebrew](#)
 - Installazione di OPAM (package manager di OCaml)
 - Procedere come in [Up and Running with OCaml](#)
 - aprire una shell
 - (macOS) eseguire: `brew install gpatch`
 - (macOS) eseguire: `brew install opam`
 - (ubuntu) eseguire: `add-apt-repository ppa:avsm/ppa`
 - (ubuntu) eseguire: `apt update`
 - (ubuntu) eseguire: `apt install opam`
 - (tutti) eseguire: `opam init`
 - (tutti) eseguire: `eval 'opam env'`
 - (tutti) eseguire: `opam switch create 4.11.1`
 - (tutti) eseguire: `eval 'opam env'`
 - **Windows:**
 - Installazione di Cygwin (emulatore unix per window) e di OPAM per Cygwin.
 - Procedere come in [Graphical Installer](#)

OCaml: Guida all'installazione e Uso - 3

● Download e Installazione: Guida Essenziale.

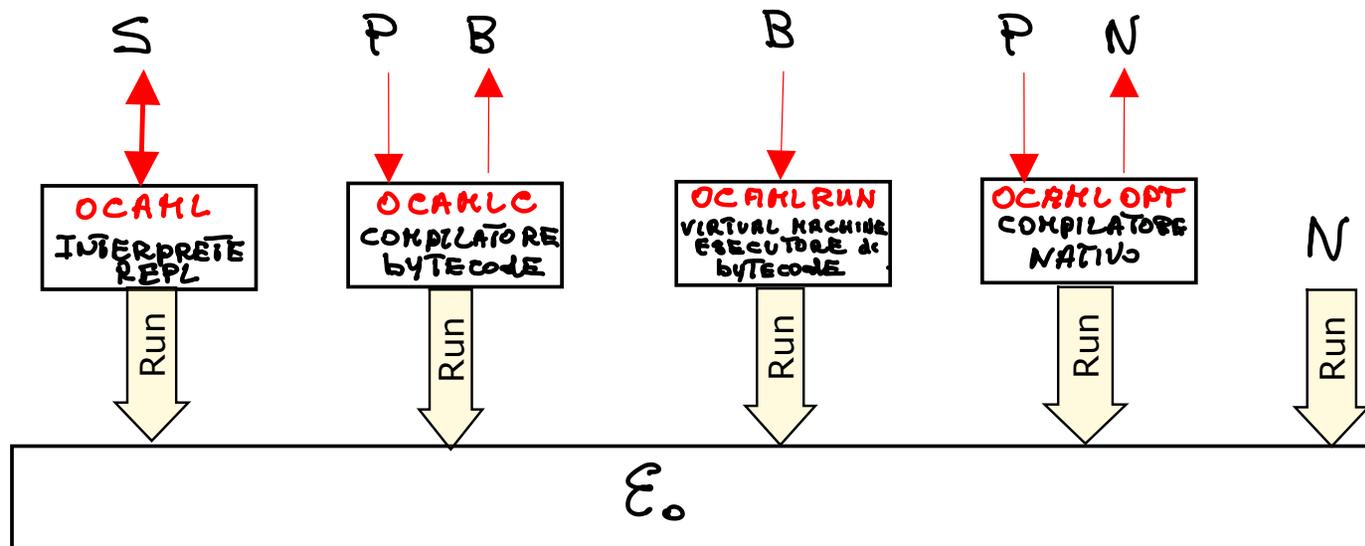
- (a) Connettersi a ocaml.org: Sito di riferimento per sviluppatori e utenti.
- (b) Selezionare "Install OCaml"
- (c) Vediamo installazioni diverse per:
 - **Avvertenza**, Disinserire Controlli/Protezioni su Download o Accessi Internet
Reinserire ad installazione completata.
Se l'installazione è interrotta occorre rimuovere directories e file installati.
 - **Linux, macOS:**
 - **Windows:**
 - Installazione di Cygwin (emulatore Unix per Windows) e di OPAM per Cygwin.
 - Procedere come in [Graphical Installer](#)
 - Download **32-bit** o **64-bit** secondo propria architettura
 - Launch **Ocaml32.exe** o **Ocaml64.exe** scaricato - segui indicazioni
 - Appaiono le icone per avviare finestre (grafiche) per:
 - shell bash Cygwin per installare **ulteriori pacchetti**
 - shell bash OCaml per usare la AM OCaml con i suoi tools
 - OCaml è ora installato, configurato e pienamente usabile
 - All'indirizzo **C:/OCaml34/home/xxxx** (con leftslash, e xxxx = nome_utente)
creare folder **Programmi** con subfolder **Small121**
da usare per riporre files testo generati con editor di Windows ed eseguire in OCaml

OCaml: Guida all'installazione e Uso - 4

● Download e Installazione.

● Uso.

- (a) apertura di una session in REPL (Interprete Read-Eval-Print-Loop)
 - L'interprete è invocato a "linea di comando" (da shell in Mac, linux, Windows);
 - che apre una session interattiva con prompt `#` (vedi session allegata);
 - interpreta (esegue) ogni termine racchiuso tra il prompt e il simbolo `';;'` (doppio punto-e-virgola);
 - stampa il valore calcolato (termine irriducibile) nella linea successiva;
 - stampa il prompt nella successiva linea e rimane in attesa;
 - Ogni termine è interpretato nell'ambiente della sessione definito dalle valutazioni precedenti;
- (b) chiusura di una session
 - usare espressione `exit(n);;` (con `n` intero), o comando `#quit;;`;
- (c) I termini di una session:
 - Espressioni di Ocaml
 - Comandi per controllo interprete: `#use`(caricamento di file di programma), `#cd` (per directory),
- (d) Editing del programma:
 - Il codice di un Programma può coinvolgere molti termini, essere scritto in più linee e conservato in più files.
 - I files di codice ocaml devono essere file testo ed avere suffisso `'.ml'`.
 - I file possono essere editati con un qualunque editor per file testo (della shell o della piattaforma).
- (e) Caricamento di un file in una session:
 - `# #use "A.ml";;`
- (f) Prova: Accesso Sessione, Scrittura ed Esecuzione di un programma
- (g) Prova: Esecutori - `ocaml`, `ocamlc`, `ocamlopt`



- S: Espressioni linguaggio OCAML
- P: Programmi linguaggio OCAML
- B: Programmi linguaggio OCAML-bytecode
- N: Programmi linguaggio macchina di Ho con esecutore E_0

OCaml: Guida all'installazione e Uso - 4

- **Download e Installazione.**
- **Uso.**
 - (a-e)....
 - (f) Prova: Accesso Sessione, Scrittura ed Esecuzione di un programma
 - (g) Prova: Esecutori - ocaml, ocamlc, ocamlpt

Esecuzione ad Interprete di tipo REPL

```
Marco-Bellias-MacBook-Air:~ marcob$ ocaml
OCaml version 4.08.1

# (* Iniziamo una sessione interattiva in REPL di OCaml *);;
# ;;
# open Printf;;
# let rec gcd x y =
    if x=y then x
    else if x>y then gcd (x-y) y else gcd x (y-x);;
val gcd : int -> int -> int = <fun>
# gcd 3 17;;
- : int = 1
# printf "applichiamo gcd\n    gcd 64 6 = %d\n" (gcd 64 6);;
applichiamo gcd
    gcd 64 6 = 2
- : unit = ()
# (* terminiamo la sessione interattiva in REPL *);;
# #quit;;
Marco-Bellias-MacBook-Air:~ marcob$
```