

Accademia Statale BBAA, Foggia
Corsi PM, a.a. 2020-21 Docente Mario Chiari
Dispensa nozioni varie per PM/ABTEC40
Versione del 19 ottobre 2020

ATTENTA\O: **DEVI ESPLORARE LE RISORSE LINKATE**

Note generali

A – CONOSCERE LA VOSTRA MACCHINA E COME USARLA

la vostra macchina è il vostro strumento di lavoro, dovete averne cura e conoscenza professionale

asap = as soon as possible / il prima possibile

1. **asap** sai individuare le principali componenti **hardware** della tua macchina, nome, azienda produttrice, caratteristiche principali:

- **cpu,**
- **gpu,**
- **ram,**
- **scheda madre,**
- **monitor,**
- **hard disk,**
- ecc.

Per chi ha un vecchio desktop a casa, apritelo e curiosate, e individuate le diverse componenti (*non aprite il vostro laptop!*)

2. **asap** sai individuare il **sistema operativo** installato sulla tua macchina:

I sistemi operativi sono - a livello consumer – o della famiglia Windows (della Microsoft) o derivati Unix. I Mac della Apple usano oggi una versione di Unix.

Se siete interessati (*lo dovrete essere!*), scoprite cosa sia **Linux** e cosa si intende per **distribuzione Linux** (le più note sono **Fedora**, **Ubuntu**, **Debian**, **Mint**, ma ce ne sono alcune centinaia). Capite come si usa gergalmente il termine **piattaforma**.

Per saperne di più:

- sul mondo Windows/Microsoft, vedi dopo
- su **Unix**. Fate conoscenza con **Dennis Ritchie**, uno dei creatori dell'era digitale. Unix e derivati sono una galassia di sistemi, vedi per esempio questa **Unix Timeline**;
- Su **Linux**, e il suo creatore **Linus Torvalds**;

Extra bonus: leggete la seguente frase e cercate di capire il ruolo delle diverse componenti indicate: *A typical Linux distribution comprises a [Linux kernel](#), [GNU tools and libraries](#), additional software, documentation, a [window system](#) (the most common being the [X Window System](#)), a [window manager](#), and a [desktop environment](#).*

- per le macchine della **Apple**, vedi a [MacOS](#), e prendi nota che: *macOS originally named Mac OS X until 2012, then rebranded to OS X until 2016*. Ricorda anche che fino a circa il 2001, le macchine Apple erano dotate di un sistema operativo, di proprio design, denominato [Classic Mac OS](#), il cui successo fu dovuto soprattutto alla correlata GUI.

AVVERTENZA: Se hai un computer della Apple, devi sapere l'uso da parte delle Apple dei nomi dei felini, e poi dei nomi di alcune località californiane, e sapere la versione precisa della tua macchina.

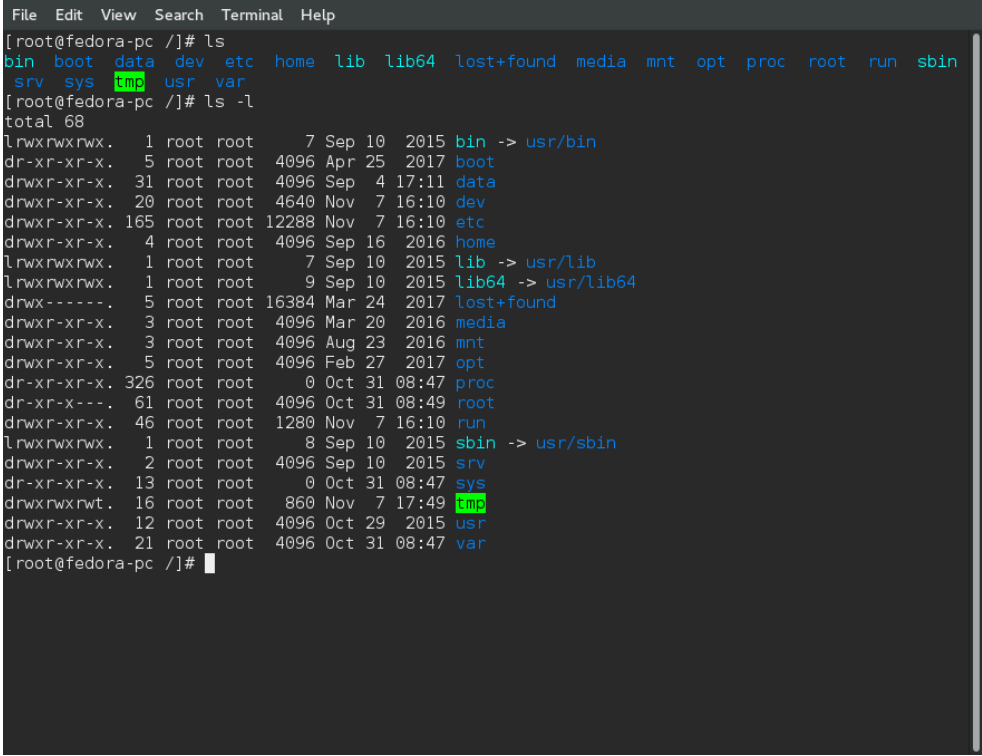
3. sai distinguere tra sistema operativo e l'**interfaccia grafica utente**, GUI

GUI

- per le recenti versioni del sistema operativo commercializzato dall'azienda statunitense Microsoft (quella del noto Bill Gates), la distinzione OS/GUI è attualmente difficile da formulare perchè *Microsoft* ha via via integrato sempre più OS e GUI. Storicamente Microsoft è nota per *MS-DOS* (OS, dal 1981 ai primi anni 2000) e per *Windows* (dal 1984 ad oggi), quest'ultima inizialmente solo una GUI, ma poi via ha assunto il ruolo di un Sistema Operativo Grafico. La storia in verità è un poco più complicata, se interessa vedi le due voci wikipedia [Microsoft Windows](#) e [MS-DOS](#). Vedi anche [Windows 10](#).
- per le macchine della azienda Apple, la GUI credo sia nota come [Aqua](#) (vedi anche [qui](#)).
- infine per le distribuzioni Linux, la GUI è una prodotto ben distinto dal sottostante SO, ve ne sono diverse, sviluppate da comunità diverse di sviluppatori. Tra le più note [Gnome](#) e [KDE](#).

Per rendersi conto di come le cose sono evolute velocemente, vedi questa [storia](#) visuale. E **questa** è un'altra grande risorsa storica.

4. sai aprire il terminale (o *prompt* dei comandi), sia nei sistemi Microsoft (un tempo nella cartella Accessori, oppure con il comando *cmd*) sia nei Mac (in genere cercando il *terminal*), sia nella varie distribuzioni Linux. Il terminale della mia macchina a casa:



```
File Edit View Search Terminal Help
[root@fedora-pc ~]# ls
bin boot data dev etc home lib lib64 lost+found media mnt opt proc root run sbin
srv sys tmp usr var
[root@fedora-pc ~]# ls -l
total 68
lrwxrwxrwx. 1 root root 7 Sep 10 2015 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 Apr 25 2017 boot
drwxr-xr-x. 31 root root 4096 Sep 4 17:11 data
drwxr-xr-x. 20 root root 4640 Nov 7 16:10 dev
drwxr-xr-x. 165 root root 12288 Nov 7 16:10 etc
drwxr-xr-x. 4 root root 4096 Sep 16 2016 home
lrwxrwxrwx. 1 root root 7 Sep 10 2015 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Sep 10 2015 lib64 -> usr/lib64
drwx----- 5 root root 16384 Mar 24 2017 lost+found
drwxr-xr-x. 3 root root 4096 Mar 20 2016 media
drwxr-xr-x. 3 root root 4096 Aug 23 2016 mnt
drwxr-xr-x. 5 root root 4096 Feb 27 2017 opt
dr-xr-xr-x. 326 root root 0 Oct 31 08:47 proc
dr-xr-xr-x. 61 root root 4096 Oct 31 08:49 root
drwxr-xr-x. 46 root root 1280 Nov 7 16:10 run
lrwxrwxrwx. 1 root root 8 Sep 10 2015 sbin -> usr/sbin
drwxr-xr-x. 2 root root 4096 Sep 10 2015 srv
dr-xr-xr-x. 13 root root 0 Oct 31 08:47 sys
drwxrwxrwt. 16 root root 860 Nov 7 17:49 tmp
drwxr-xr-x. 12 root root 4096 Oct 29 2015 usr
drwxr-xr-x. 21 root root 4096 Oct 31 08:47 var
[root@fedora-pc ~]#
```

Nel corso dell'anno vi sarà chiesto di eseguire qualche semplice comando a riga di comando. Le stesse funzionalità sono eseguite con comandi simili, ma non purtroppo gli stessi, in dos/win e rispettivamente in linux (+macOS).

Ho trovato questa per come far eseguire le funzionalità principali nei due casi.
(la url da cui l'ho presa non mi sembra più attiva, ma se cercate ne trovate varie versioni on line, epr esempio [qui](#))

Command's Purpose	MS-DOS	Linux	Basic Linux Example
Copies files	copy	cp	cp <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Moves files	move	mv	mv <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Lists files	dir	ls	ls
Clears screen	cls	clear	clear
Closes shell prompt	exit	exit	exit
Displays or sets date	date	date	date
Deletes files	del	rm	rm <i>thisfile.txt</i>
"Echoes" output to the screen	echo	echo	echo <i>this message</i>
Edits text files	edit	gedit([a])	gedit <i>thisfile.txt</i>
Compares the contents of files	fc	diff	diff <i>file1 file2</i>
Finds a string of text in a file	find	grep	grep <i>word or phrase</i> <i>thisfile.txt</i>
Formats a diskette	format a: (if diskette is in A:)	mke2fs	/sbin/mke2fs /dev/fd0 (/dev/fd0 is the Linux equivalent of A:)
Displays command help	<i>command</i> /?	man or info	man <i>command</i>
Creates a directory	mkdir	mkdir	mkdir <i>directory</i>
Views contents of a file	more	less([b])	less <i>thisfile.txt</i>
Renames a file	ren	mv([c])	mv <i>thisfile.txt thatfile.txt</i>
Displays your location in the file system	chdir	pwd	pwd
Changes directories with a specified path (<i>absolute path</i>)	cd <i>pathname</i>	cd <i>pathname</i>	cd / <i>directory/directory</i>
Changes directories with a <i>relative path</i>	cd..	cd ..	cd ..
Displays the time	time	date	date
Shows amount of RAM in use	mem	free	free

Notes:

- Gedit** is a graphical text editor; other editors you can use in place of **Gedit** include **nano** and **vi**.
- The `more` pager can also be used to page through a file one screen at a time.
- The `mv` command can both move a file and, if you want to rename a file in the same directory, "move" that file to the same directory with a new name.

B – CONOSCERE INTERNET

cercate di capire l'idea di base della comunicazione telematica Internet. Cercate di capire – consultando wikipedia – la distinzione

Commutazione di circuito vs. Commutazione di pacchetto

(nelle voci wikipedia, leggi i paragrafi iniziali, per capire di che cosa si stia parlando, il resto sono nozioni per ingegneri della tele-comunicazione. Può essere utile domandarsi di quale precedente infrastruttura tecnologica Internet sia l'evoluzione, di quale? Perché? Cercate di confrontare la struttura della comunicazione Internet con quelle fornite dal sistema postale, da radio/televisione, dalla telefonia e dalla stampa quotidiana (Internet è in particolare l'evoluzione di una di queste, quale?)

Qui trovate la storia di Internet scritta dai protagonisti-inventori.

TASK: dovete saper riconoscere un indirizzo IP e, quando la vostra macchina è collegata a Internet, dovete sapere come individuare quale indirizzo IP gli è stato assegnato.

Quando siete all'interno di una LAN (come quando siete in aula C02), o comunque collegati a Internet, aprite il terminale e digitate il comando **ifconfig** (sistemi basati su Unix) o **ipconfig** (sistemi Win). Vi apparirà -tra altre informazioni- l'indirizzo IP, una stringa composta da quattro numeri, ciascuno tra 1 e 254, con cui la vostra macchina è identificata nella LAN, o in Internet; dovrebbe essere qualcosa del tipo 192.168.50.123.

Cercate la definizione e l'uso dei principali **protocolli** Internet

- TCP-IP
- SMTP
- FTP
- HTTP (HTML) ← questo è il protocollo per comunicare trasmettere pagine Web

P è sempre per *Protocol*; **T** è per *Transfer*. Sappiate spiegare gli acronimi. Sappiate indicare di ciascuno la data di quando sono stati formulati e divenuti di uso comune.

ATTENZIONE:

noi **non** (No! No! No!) confondiamo Internet con il Web, e tantomeno Internet con Google o Facebook

C – TROVARE INFORMAZIONI, RISORSE e TOOLS IN RETE

- uso di motori di ricerca, ovviamente soprattutto **Google** (ma ce ne sono altri)
Vi consiglio di cercare in Inglese, ovvero impostate Google indicando *Inglese* come lingua, e *Stati Uniti* come area geografica.
Inoltre, utilizzate *search word* chiare e tecnicamente appropriate.
Potete cercare anche per immagini (provare per credere)

- **wikipedia** è una *enciclopedia* importante, soprattutto per tutto ciò che riguarda l'informatica. Molto, molto, molto meglio la **versione inglese**!

Extra bonus: Se vi ritenere competenti in un qualche settore, potreste tentare di registrarvi per scrivere un paio di voci di wikipedia, e fare pratica delle modalità di scrittura, revisione, discussione di wikipedia. Una voce ben scritta e accettata vale uno, anche due voti, anche tre!!!!
voti in più ai miei esami!!

- **www.w3c.org** Sviluppo protocolli Web e tecnologie collegate. Estremamente ricco, ci vuole un po' per imparare ad orientarsi, e trovare le url dei tutorial che possono servire, alcuni anche elementari e molto ben scritti;
- **www.php.net** Sito ufficiale del linguaggio PHP. Documentazione molto utile anche per iniziare con piccoli esempi <http://www.php.net/manual/en/>;
- **www.mysql.com** sito ufficiale di MySQL, documentazione a <http://dev.mysql.com/doc/>;
- **www.mozilla.org/foundation/** *The Mozilla Foundation*, sviluppatori di Firefox, e di molto altro. Per noi utili le sezioni su **JavaScript**, e sul **DOM**;
- **www.ietf.org** Sviluppo protocolli Internet. Questo è per informatici, ma è sempre utile vedere cosa si intende per protocollo, per esempio vedi l'ultima versione della formulazione del [protocollo per la posta elettronica](#) mentre qui trovate una versione, ora superata, del [protocollo HTTP](#). Lo IETF pubblica dei documenti indicati come RFC (*Request For Comments*), che contengono la descrizione dei protocolli (quelli proposti da qualcuno, e quelli adottati secondo una complessa procedura)

Per esempio nella RFC2616, si trovano le definizioni di cosa si deve intendere per *client* e per *server*:

client

A program that establishes connections for the purpose of sending requests.

server

An application program that accepts connections in order to

service requests by sending back responses. Any given program may be capable of being both a client and a server; our use of these terms refers only to the role being performed by the program for a particular connection, rather than to the program's capabilities in general.[...]

- [stackexchange](#) forse meno noto, ma estremamente utile: un sito estremamente performante che organizza comunità on line di Q&A (*questions and answers*). Dovete iscrivervi a una delle comunità tematiche, e poi porre domande o suggerire risposte. Molto interessante il meccanismo di calcolo della vostra reputazione.
- [w3schools](#), una delle innumerevoli scuole o siti con tutorial, guide, lezioni interattive, video, podcast, ecc. su tutto ciò che stiamo cercando di capire, di qualsiasi livello, dall'introduttivo all'*advanced*. Cercate pure in rete, e divertitevi.

D – CAVI E PRESE

banale ma utile e necessario

dovete sapere canale ma utile e necessario collegare la vostra macchina alla rete elettrica di alimentazione, alla presa [Ethernet](#) per accedere a Internet, all'accesso wi-fi, e ai proiettori disponibili nell'aula, e quindi saper distinguere uscite/prese e cavi corrispondenti.

Semplifica la vita conoscere gli standard più comuni per cavi e prese: **consultate [qui](#) e [qui](#)** (o simili risorse in rete)

Ricordatevi:

per le presentazioni durante il corso, **e in sede di Esame**, siete voi a dovervi preoccuparvi di presentarvi con cavi e spinotti per collegare la Vostra macchina al proiettore messo a disposizione in aula

NOTA: per alcune macchine Apple con uscite non standard, dovrete munirvi (comprare!) i necessari connettori.

E – IL NOSTRO GERGO

ACRONIMI

- CPU, GPU, RAM
- OS, GUI, CLI, BIOS
- PSTN
- ITU, ISO/OSI, IETF, ICANN, RIR, ISOC
- TCP-IP, FTP, SMTP, HTTP-HTML, RFC, DNS, URL, URI, CSS
- LAN, WAN
- PHP
- SQL

- **(L)(M)(W)AMP**
- **GNU GPL, CC**
- **IDE, WYSIWYG**
- **DVI, HDMI, VGA, USB, Ethernet, RCA, DisplayPort,**
- (*altri??*)

per ognuno degli acronimi precedenti, consultare la corrispondente [voce di wikipedia](#)

NOZIONI INFORMATICHE

preoccupatevi di saper distinguere gli ambiti semantici delle seguenti nozioni e adoperarle in modo appropriato:

- **Protocollo**
- **Standard**
- **Algoritmo**
- **Programma**
- **Linguaggio (di programmazione)**
- **(pacchetto) Applicativo**
- **Macchina (computazionale)**

(controllate come si traducono questi termini nella vostra lingua madre e in Inglese)

Fate attenzione all'uso corretto di alcuni **termini italiani** (alcuni forse desueti, o poco usati), consultate un buon dizionario, e le seguenti voci Treccani:

- [Informatica](#)
- [Telematica](#)
- [Cibernetica](#) (io non lo uso)

•

F – COME RENDERE LA VOSTRA MACCHINA OPERATIVA PER IL CORSO ABTEC40

1) Installate un *software bundles* (L)(M)(W)AMP

dove

- **(L)(M)(W)** = (Linux)(Mac)(Windows), il vostro sistema operativo;
- **A** = Apache, il web (http) server;
- **M** = mySQL il programma di gestione database;
- **P** = l'interprete di PHP. PHP è il linguaggio di programmazione orientato al web.

Per ***software bundle*** si intende in genere un pacchetto di applicativi composto da più pacchetti indipendenti (per produzione, funzionalità e possibilità d'uso) ma che sono di fatto interdipendenti nei loro utilizzi più comuni e che quindi merita installare in modo coordinato. Un buon *software bundle* comprende inoltre qualche applicativo ausiliario, in genere per installare e/o configurare i pacchetti principali, per il loro utilizzo e monitoraggio, nonché una “pulsantiera”^{\$} (che in genere si installa sulla barra dei comandi, talvolta da una pagina web dedicata) per lanciare o chiudere e i diversi tool (talvolta detti servizi) principali (e quindi non confondete il lanciare la pulsantiera stessa con il lanciare i servizi principali). Ogni *bundle* è in genere sviluppato per uno specifico OS, ovvero devi installare la versione adatta al tuo OS, e anche al tuo hw, se 32bit o 64bit.

(se siete interessati: provate a installare Apache, mySQL e PHP separatamente, e vedrete il numero di difficoltà da superare).

Di seguito elenco i *bundle* AMP che ho visto usare negli ultimi anni:

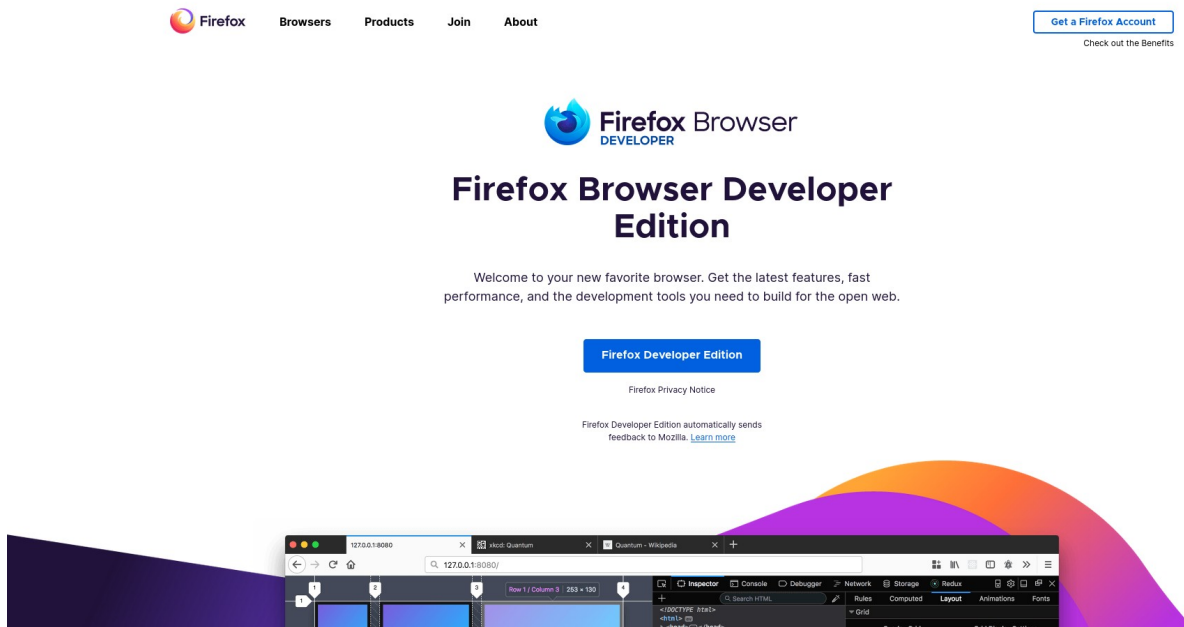
- MAMP, www.mamp.info
FORSE IL PIÙ POPOLARE, ora anche versione per Win: MAMP per Win
- XAMPP, <http://www.apachefriends.org/en/xampp.html>
- easyPHP, <http://www.easyphp.org/> **(interessante e di facile uso)**
- WAMPServer, <http://www.wampserver.com/en/>
- AMPPS <http://www.ampps.com/>
- *altri??*

Se usate una distribuzione Linux, i programmi AMP dovrebbero essere già presenti. Anche su Windows e Mac potrebbero essere già installati, ma installare una opportuna *bundle* è più pratico (per voi e per me).

^{\$} *Pulsantiera* un termine di *mia* invenzione, per intenderci.

In teoria, potete avere più installazioni di AMP sulla vostra macchina, ma nel caso dovete fare un poco di attenzione a che non entrino in conflitto (in genere basta una qualche attenzione alle porte su cui sono in ascolto).

2) **Installare il *browser* (http-client) Firefox, versione *developer* .** **(OBBLIGATORIO = Fortemente Consigliato!)**



+ installare gli ***add-on*** (componenti aggiuntivi), dal menù Strumenti/*Tools* di Firefox:

WebDeveloper, estremamente utile

ColorZilla, per gestire i colori, utile e divertente

Nimbus, per prendere screenshot di una pagina web

nelle versioni recenti un tool per screenshot è già incorporato in Firefox

Curiosate e sperimentate qualcuno dei numerosi *add-on*, magari ne trovate qualcuno che ci è utile, non tutti sono performanti, e alcuni entrano in conflitto con altri. Si tratta di provare e semmai disinstallare. *Controlla anche che non vi siano stati installati add-on che non utilizzi, e che comunicano a tua insaputa con qualche servizio esterno.*

3) **installare un *Syntax Highlighter Editor***

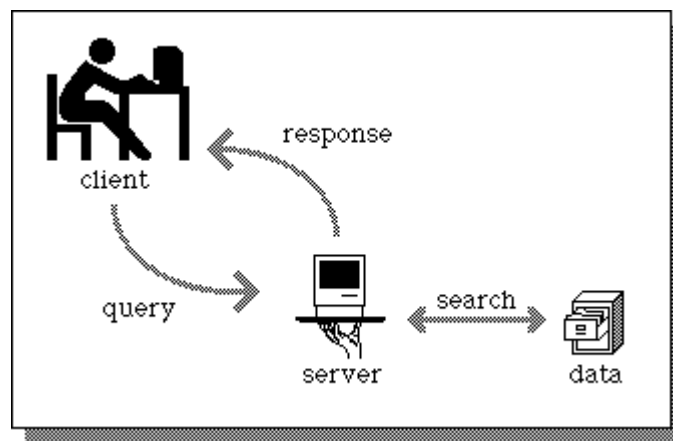
Inoltre avete bisogno di un tool che ***highlight*** il codice mentre lo scrivete, ovvero che colori le diverse categorie sintattiche di cui sono composti i linguaggi che imparerete a conoscere (html+css, php, mysql, JS) con colori diversi. Per quanto apparentemente banale, questo permette una scrittura estremamente più facile e veloce, ed aiuta anche a capire come sono strutturati i diversi linguaggi. I *Syntax Highlighter Editor* sono in

genere diversi per OS diversi (alcuni hanno versioni multiplatforma), in particolare potete usare:

- **BEdit** per OSX (Apple)
- **Notepad++**: "a free source code editor which supports several programming languages running under the MS Windows environment", quindi per Wins
- **Gedit**, per distribuzioni Linux.
- **Atom** (multiplatforma). Forse il migliore.
ATTENZIONE: sto cercando di capire se Atom ci possa aiutare alla **scrittura di codice condiviso da remoto** (vedi questo [post](#)). Se si, probabilmente chiederò a tutti di installare e usare Atom.

QUELLO CHE SEGUE È MOLTO IMPORTANTE (E SEMPLICE!!!!)

G – CLIENT – SERVER



Per poter svolgere i compiti richiesti da questo corso è essenziale capire i ruoli di *client* e di *server* tra entità che comunicano, in genere rispetto a un certo protocollo di comunicazione. Quindi, per ogni protocollo, dovete saper individuare e operare con gli applicativi che svolgono i ruoli di *client* e rispettivamente di *server*, in particolare per almeno i seguenti protocolli: **http-html**, **ftp**, **smtp**. Per il corso, il caso importante è ovviamente il primo, ma utilizzeremo anche un client ftp.

IMPORTANTE: le entità che comunicano, nel ruolo di *client* o di *server*, sono in genere collocate su macchine diverse, a indirizzi diversi della rete Internet, come viene raffigurato nel disegno sopra. Ma le entità che comunicano sono programmi (applicativi)

installati su una macchina, non macchine (non confondere). E - **nota bene** – un programma *client* e un programma *server* possono comunicare tra di loro anche se sono installati sulla stessa macchina! È importante che capiate come far interagire tale coppie di programmi su una vostra macchina, perché questo è quello che dovrete fare.

Si utilizza il termine *browser* per indicare un qualche *http client*. Gli applicativi che sono dei client *http* sono i noti *Firefox*, *IE*, *Safari*, *Opera*, *Chrome*, *SeaMonkey* e altri meno noti. Il *server http* più diffuso e più noto (ma non l'unico) è *Apache*.

Quindi: da un punto operativo dovete assai velocemente imparare a distinguere tra **operare lato client** e **operare lato server**:

MOLTO IMPORTANTE:

dovete aver ben chiaro la distinzione tra operare lato Server e operare lato Client

dop aver installato la suite AMP

CONTROLLATE CHE TUTTO FUNZIONI

e che avete capito come muoversi tra il lato client e il lato server.

IMPARATE LA SEGUENTE PROCEDURA

1. Individuate la cartella nel vostro hd in cui porre tutto ciò che costituirà il vostro sito (*nei Wins, viene in genere creata la cartella **www**; con MAMP in genere la cartella **htdocs** (nei Mac apribile da Applicazioni, sotto la cartella MAMP); nel mio Linux Fedora, la trovo al percorso **/var/www/html/**). Per intenderci chiameremo tale cartella '**cartella server**'. Quando modificate il contenuto di questa cartella state agendo lato Server.*
2. Lanciate il servizio Apache (preferibilmente dalla pulsantiera della vostra bundle; in alcuni bundle apparirà una spia verde=acceso, o un checkbox spuntato)
3. Create un file primo.html con le seguenti righe di codice html e salvatelo nella cartella server:

```
<html>
<head>
<title> mio primo file </title>
</head>
<body>
<p> Hello <b>World</b>! </p>
</body>
</html>
```

(ATTENZIONE: se avete Win: prestate attenzione a se il file è salvato con estensione esattamente .html: Win ha spesso delle configurazioni di salvataggio automatico che aggiungono estensioni non volute e non visibili).

4. Dalla barra di navigazione del vostro browser, interrogate/chiamate il vostro server con la URL (indirizzo Internet) <http://127.0.0.1/primo.html>, o (nome di dominio) <http://localhost/primo.html>. Se vedete il testo: Hello **World!** (ma non le tag <p>,), allora il tutto funziona. OTTIMO!

in alcuni casi dovete indicare anche la porta di connessione, per es.

<http://localhost:8888/primo.html>

Quando, con un browser, eseguite una tale chiamata http, state agendo lato Client.

5. **ATTENZIONE**, se invece digitate nella barra di navigazione qualcosa tipo **file://percorso--a/www** allora state utilizzando il vostro browser semplicemente come visualizzatore di codice html, senza connessione al *server*

SE CONFONDETE QUESTI DUE CASI LO CONSIDERO UN ERRORE GRAVE!!

