

Laboratorio di Introduzione alla Matematica
Computazionale – A.A. 2021/2022
03 - HTML, e \LaTeX

Fabio Durastante <fabio.durastante@unipi.it>,
Sergio Steffé <steffe@cs.dm.unipi.it>

– Basato sulle slide dell'A.A. 2020/21 di L. Robol e S. Steffé –

9 Novembre 2021

Albori di Internet

- ▶ nel 1986 Il CNR a Pisa era la prima organizzazione italiana a collegarsi ad Internet.
- ▶ per collegarsi e trasferire informazioni erano in uso telnet, ftp, e la e-mail.
- ▶ il primo *social* erano le usenet NEWS
- ▶ nel programma per apple hypercard venivano usati gli hyperlink (1987)
- ▶ nel 1991 aveva un certo successo il gopher (che però dal 1993 fu frenato da problemi di copyright)
- ▶ tentativi di creare un sistema di ricerca globale furono archie e veronica (con scarso successo).
- ▶ nel 1991 in Italia le macchine collegate stabilmente ad Internet erano poche migliaia.

Storia del WWW

- ▶ Nel 1991, Tim Berners Lee, lavorando al CERN, creò un sistema che permetteva di ospitare delle pagine con collegamenti anche a elementi ospitati da altri server (ipertesti);
- ▶ Il sistema era un'evoluzione di quanto era stato sviluppato fino ad allora, ma proiettato decisamente al futuro.
- ▶ all'epoca caricare una pagina web poteva richiedere parecchi minuti se non si disponeva di rete e workstation allo stato dell'arte !

Storia del WWW

- ▶ Nel 1991, Tim Berners Lee, lavorando al CERN, creò un sistema che permetteva di ospitare delle pagine con collegamenti anche a elementi ospitati da altri server (ipertesti);
- ▶ Il sistema era un'evoluzione di quanto era stato sviluppato fino ad allora, ma proiettato decisamente al futuro.
- ▶ all'epoca caricare una pagina web poteva richiedere parecchi minuti se non si disponeva di rete e workstation allo stato dell'arte !
- ▶ Con il tempo, sono nati vari browser, sono stati aggiunti supporti multimediali alle pagine, e la tecnologia si è evoluta. Ma il protocollo di trasmissione ha ricevuto pochi cambiamenti.
- ▶ Il CERN rinunciò ai diritti e perciò il protocollo è aperto e il W3C è l'ente delegato a seguirne la standardizzazione.

Collegamenti ipertestuali

- ▶ Le grosse novità del WWW erano l'essere un protocollo aperto e la possibilità di **passare da una pagina all'altra anche su altri server** tramite "collegamenti".
- ▶ Questo assicurava la possibilità di continuo sviluppo e permetteva di **catalogare** facilmente i contenuti!
- ▶ I motori di ricerca catalogano infatti le pagine web.
- ▶ Un enorme passo in avanti è stato fatto poi dal **pagerank**, che riesce a determinare quale pagina è importante e quale no essenzialmente risolvendo un sistema di equazioni (come? è un problema di autovalori e autovettori).

Pagine web

- ▶ Quando visitiamo una pagina web, il PC deve richiederne il contenuto attraverso la rete.
- ▶ Un **URL** specifica esattamente che cosa si vuole ottenere.
esempio: `http://unimap.unipi.it/:80`
- ▶ I primi passi usano le tecnologie che abbiamo già discusso (DNS, TCP/IP, ...).
- ▶ Nei passi seguenti è necessario che il server e il client si accordino su come interpretare i dati, ovvero come descrivere il contenuto della pagina.

WWW e HTML

- ▶ Generalmente, HTTP (o la sua versione criptata HTTPS) sono i protocolli più usati per navigare in Internet.
- ▶ Altri protocolli sono riconosciuti dai vari Browser ma molto meno usati.
- ▶ negli URL si trovano normalmente come nomi di protocolli http, https, file, più raramente ftp, ma sono presenti anche protocolli più esotici come per esempio per konqueror il fish (sftp).
- ▶ il domain name può essere anche localhost
- ▶ Se non specificato il numero di porta di default è 80 per http, 443 per https, 21 per ftp.

Browser

- ▶ Per visualizzare la pagine web, c'è bisogno di un browser.
- ▶ Potete scegliere il vostro preferito, Firefox, Chrome (google)/Chromium, Edge(Microsoft), Opera(Opera), konqueror, iceweasel, lynx, ...
- ▶ Linux dispone di alcuni comandi per scaricare dati dal web :
 - ▶ `wget https://www.google.com` scarica la home page di Google sul vostro PC.
 - ▶ `curl https://www.google.com` effettua la stessa operazione; CURL ha molte opzioni, e permette anche di **spedire** dati, oltre che scaricarli.
- ▶ E' normale che la stessa pagina appaia diversa in diversi Browser !

Conversazione HTTP

- ▶ GET / HTTP/1.0: “vorrei vedere la home-page”
- ▶ Il server risponde con il codice HTML; questo potrebbe contenere link ad altri file da caricare (ad esempio, immagini).
- ▶ Il client richiede i file ausiliari:
 - ▶ GET /images/image1.png HTTP/1.0
 - ▶ GET /images/image2.png HTTP/1.0
 - ▶ ...

Nei prossimi laboratori vedremo come creare una pagina web. Gli studenti di matematica gestiscono autonomamente il server `poisson.phc.dm.unipi.it` su cui ospitano le pagine degli studenti. Il server web poisson esiste da **prima** che il dipartimento avesse un suo server web !

(Andare su archive.org, cercare nella WayBack Machine `poisson.dm.unipi.it` del 17 ottobre 1997...)

Pagine web “statiche” e “dinamiche”

Per servizi come l'e-banking o l'e-commerce occorre che l'interazione tra server e client sia più complessa del semplice trasferimento di un file html:

- ▶ pagina statica: il server manda semplicemente una copia della sua pagina html al client.
- ▶ pagina dinamica lato client: il server manda al client una pagina html e del codice eseguibile che viene eseguito dal client.
- ▶ pagina dinamica lato server: un programma eseguito sul server dialoga col client e crea al volo un codice html personalizzato che manda al client.
- ▶ nell'ultimo caso ovviamente è anche possibile spedire anche codice eseguibile lato client !

Pagine web “statiche” e “dinamiche”

Per servizi come l'e-banking o l'e-commerce occorre che l'interazione tra server e client sia più complessa del semplice trasferimento di un file html:

- ▶ pagina statica: il server manda semplicemente una copia della sua pagina html al client.
- ▶ pagina dinamica lato client: il server manda al client una pagina html e del codice eseguibile che viene eseguito dal client.
- ▶ pagina dinamica lato server: un programma eseguito sul server dialoga col client e crea al volo un codice html personalizzato che manda al client.
- ▶ nell'ultimo caso ovviamente è anche possibile spedire anche codice eseguibile lato client !
- ▶ Questo apre la possibilità a problemi di sicurezza: il vostro computer esegue codice scaricato da un server, senza chiedervi conferma.
- ▶ Per questo, i browser utilizzano strategie particolare (e.g., sandboxing).

Server Proxy

A volte è preferibile non utilizzare una connessione diretta al server web, ma passare attraverso un “proxy”. Ad esempio, questo permette a chi ha l'accesso alla macchina ssh2 di accedere alla riviste scientifiche per cui l'università paga l'abbonamento:

- ▶ `ssh -N -D 8080 ssh2.dm.unipi.it`: Apertura di un proxy attraverso SSH;
- ▶ Configurare il proxy sul proprio PC (tipicamente Impostazioni → Rete → Proxy, oppure direttamente dal browser). Il tipo di protocollo si chiama SOCKS.
- ▶ Navigare su `mathscinet.ams.org`.
- ▶ Questa è una alternativa all'uso di un VPN.

Privacy

Con il passare del tempo, e l'aumento della popolarità di Internet, si è cominciato a preoccuparsi della privacy degli utenti:

- ▶ Quante informazioni vengono condivise dal mio PC quando navigo in Internet?
- ▶ A che scopo possono essere utilizzate queste informazioni (ad esempio, per la pubblicità?).
- ▶ Posso negare il consenso?
- ▶ Non sempre è un male: a volte il tracciamento dell'utente serve ad impedire truffe.
- ▶ Se possibile usare https che impedisce lettura e modifiche del codice http da parte di "man in the middle".

Privacy in locale

Lo stesso problema si applica ai dati locali:

- ▶ Cosa succede se mi rubano il PC e/o riescono ad accederci da remoto?
- ▶ Esiste un sistema per prevenire il furto dei dati?
- ▶ Ce ne sono alcuni: cifratura della home / container / “trusted-platform”, con diversi pro e contro.

Identificazione in rete

- ▶ Quali dati possono essere utilizzati quando navigo?
- ▶ Ci sono stati diversi tentativi di legiferare su questo argomento da parte dell'UE (Cookies, GDPR).
- ▶ Se vogliamo essere (ragionevolmente) sicuri che i nostri dati non siano utilizzati per la pubblicità – possiamo aprire una scheda in incognito.
- ▶ tenere separate la finestra per gli URL da visitare da quella per le ricerche ! Altrimenti regalate al motore di ricerca la storia di tutti i siti che vistate !
- ▶ Alcuni dati però sono ancora disponibili: IP, geolocation, e se mi loggo perdo l'anonimato.
- ▶ Può servire anche avere diversi accounts sul proprio PC ed usare diversi VPN o tool appositi come Tor.

Lavoro sugli HTML

- ▶ Il linguaggio HTML è un linguaggio di **markup**.
- ▶ Nel Browser possiamo sempre vedere il **codice sorgente delle pagine** che visitiamo!
- ▶ Il sito del W3C ospita un validatore online con cui possiamo vedere se il nostro codice ha errori !
- ▶ Esistono varie versioni di HTML.
- ▶ I files html sono files di puro testo da editare con un editore pulito e non con word processors.
- ▶ esistono editori grafici ma le sorgenti html generate sono di solito poco comprensibili.
- ▶ Nei CMS (per esempio WordPress) le pagine vengono create dinamicamente agendo su un data base.

Come creare una pagina HTML?

Una pagina base semplice ha questo aspetto:

```
<!DOCTYPE html>
<html>
  <head>
    <title>La mia pagina personale</title>
  </head>

  <body>
    <h1>Home page</h1>
    <p>Benvenuti sulla mia pagina ...</p>
  </body>
</html>
```

- ▶ Tag: hanno parentesi angolate (<p>, <head>) e si devono “chiudere” (</p>, </head>). Possono contenere dell'altro HTML al loro interno, e/o avere **attributi**.

Aggiungiamo degli attributi

```
<!DOCTYPE html>
<html>
  <head>
    <title>La mia pagina personale</title>
  </head>

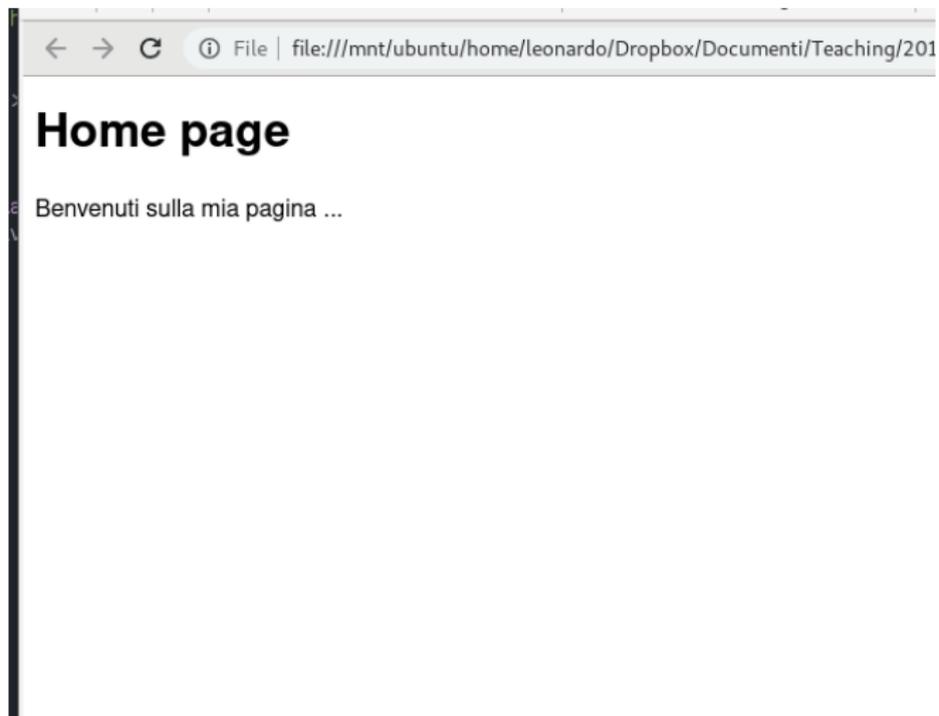
  <body>
    <h1 class="titolo" id="titolo-principale">Home page</h1>
    <p>Benvenuti sulla mia pagina ...</p>
  </body>
</html>
```

- ▶ `<h1>`, ..., `<h6>` sono tag per i titoli.
- ▶ `<p>` corrisponde ad un paragrafo.
- ▶ Ogni tag può avere una classe, ed un id. L'ID deve essere unico! Ci sono molti altri attributi possibili.

Alcuni esempi

- ▶ `Open google.com`: un collegamento a Google (tag “anchor”).
- ▶ ``: Un'immagine di un gattino: notate la sintassi finale `/>`: questo è equivalente ad aprire e chiudere un tag, ovvero a scrivere ``.
- ▶ Molti altri tag che vedremo in laboratorio.

Uno sguardo alla nostra pagina



- ▶ La pagina funziona, ma è decisamente triste e spoglia.
- ▶ Per ovviare a questo problema, possiamo usare i **fogli di stile** (Cascading Style Sheets).
- ▶ Possiamo usare **classi** e **id** per specificare l'aspetto di varie parti della pagina, oppure cambiare l'aspetto di tutti i tag di un certo tipo.
- ▶ Il linguaggio dei CSS è diverso dall'HTML:

```
/* Il selettore .titolo matcha la class titolo */  
.titolo {  
    color: blue;  
}
```

La nuova pagina

Home page

Benvenuti sulla mia pagina ...

- ▶ Il risultato non è ancora entusiasmante, potremmo cambiare lo sfondo modificando il tag `body` – che rappresenta tutto il corbo del testo. Magari possiamo anche aggiungere un bordo al paragrafo.

```
/* Il selettore .titolo matcha la class titolo */  
.titolo {  
    color: blue;  
}  
/* Si possono specificare anche nomi di tag */  
body {  
    background-color: red;  
}  
p {  
    border: 1px solid green;  
}
```

La nuova pagina

Home page

Benvenuti sulla mia pagina ...

- ▶ Le possibilità sono infinite: bordi, margini, padding, caratteri, animazioni, ...
- ▶ È possibile selezionare anche elementi utilizzando il loro id, ad esempio `<h1 id="titolo-principale">Home page</h1>` si può selezionare con

```
#titolo-principale {  
    ...  
}
```

- ▶ Altre possibilità per i più esperti: selezionare tag che sono all'interno di certi altri tag, oppure solo in "condizioni" particolari, ad esempio solo quando il mouse è sopra di loro:

```
/* Colore rosso solo quando il mouse ci passa sopra */  
#titolo-principale:hover {  
    color: red;  
}
```

TEX e L^AT_EX

- ▶ Donald Knuth è l'autore di "The Art of Computer programming" (leggete la dedica!).
- ▶ Quando, alla fine degli anni 70, ottenne le bozze della seconda ristampa, rimase inorridito dalla bassa qualità tipografica.
- ▶ ... e decise che doveva esistere qualche metodo più intelligente di affrontare il problema.
- ▶ Così è nato il TEX.



T_EX e L^AT_EX

- ▶ Donald Knuth è l'autore di “The Art of Computer programming” (leggete la dedica !).
- ▶ Quando, alla fine degli anni 70, ottenne le bozze della seconda ristampa, rimase inorridito dalla bassa qualità tipografica.
- ▶ ... e decise che doveva esistere qualche metodo più intelligente di affrontare il problema.
- ▶ Così è nato il T_EX.



Ogni matematico ad un certo punto incontra il T_EX – e certamente toccherà anche a voi quando dovrete scrivere la tesi oppure presentare un seminario.

Un po' di storia

- ▶ Knuth cominciò a lavorare al T_EX nel 1977, progettando di finirlo nel seguente anno.

Preliminary preliminary description of T_EX

D Knuth, May 13, 1977

In this memo I will try to explain the T_EX system for preparing publishable documents and how it is proposed to implement the system. Even though I don't understand T_EX very well myself yet, I think the best way for me to get started is by trying to explain it to somebody else.

*T_EX is for technical text. Insiders pronounce the X as a Greek Chi (cf. the Scottish 'ch' sound in 'Loch Ness') since the word 'technical' stems from a Greek root meaning art as well as technology. I am preparing the system primarily for use in publishing my series *The Art of Computer Programming*— the initial system will be tuned for my books, **but it will not be difficult to extend it for other purposes if anybody wants to do so.***

Un po' di storia

- ▶ Knuth cominciò a lavorare al $\text{T}_{\text{E}}\text{X}$ nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 ($\text{T}_{\text{E}}\text{X}$ 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Ad esempio, l'ultima release del $\text{T}_{\text{E}}\text{X}$ è la versione 3.14159265. Il numero di versione verrà settato a π — e lo sviluppo congelato — dopo la morte di Knuth.
- ▶ In maniera simile, il sistema di compilazione dei caratteri usati dal sistema $\text{T}_{\text{E}}\text{X}$ (MetaFont) ha raggiunto la versione 2, e da allora converge a $e = 2.71828\dots$

Un po' di storia

- ▶ Knuth cominciò a lavorare al $\text{T}_{\text{E}}\text{X}$ nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 ($\text{T}_{\text{E}}\text{X}$ 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Ad esempio, l'ultima release del $\text{T}_{\text{E}}\text{X}$ è la versione 3.14159265. Il numero di versione verrà settato a π — e lo sviluppo congelato — dopo la morte di Knuth.
- ▶ In maniera simile, il sistema di compilazione dei caratteri usati dal sistema $\text{T}_{\text{E}}\text{X}$ (MetaFont) ha raggiunto la versione 2, e da allora converge a $e = 2.71828\dots$
- ▶ Knuth offriva **premi in denaro** alle persone che **trovavano** e **segnalavano** un **bug** in $\text{T}_{\text{E}}\text{X}$. Il premio per bug è iniziato a 2,56 \$ e raddoppiato ogni anno fino a quando non è stato congelato al suo valore attuale di 327,68 \$.

Un po' di storia

- ▶ Knuth cominciò a lavorare al $\text{T}_{\text{E}}\text{X}$ nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 ($\text{T}_{\text{E}}\text{X}$ 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Ad esempio, l'ultima release del $\text{T}_{\text{E}}\text{X}$ è la versione 3.14159265. Il numero di versione verrà settato a π — e lo sviluppo congelato — dopo la morte di Knuth.
- ▶ In maniera simile, il sistema di compilazione dei caratteri usati dal sistema $\text{T}_{\text{E}}\text{X}$ (MetaFont) ha raggiunto la versione 2, e da allora converge a $e = 2.71828\dots$
- ▶ Knuth offriva **premi in denaro** alle persone che **trovavano** e **segnalavano** un **bug** in $\text{T}_{\text{E}}\text{X}$. Il premio per bug è iniziato a 2,56 \$ e raddoppiato ogni anno fino a quando non è stato congelato al suo valore attuale di 327,68 \$.
- ▶ Knuth ha perso relativamente pochi soldi in quanto sono stati rivendicati pochissimi bug... e i destinatari incorniciano il loro assegno come prova di aver trovato un bug in $\text{T}_{\text{E}}\text{X}$ piuttosto che incassarlo...

Funzionamento del T_EX

Il T_EX è un software di typesetting – ma anche un linguaggio di programmazione ad alto livello.

Generalmente nessuno usa direttamente il programma T_EX ma usa dei grossi pacchi (plain tex, latex) di comandi (in realtà macro precompilate).

In origine il plain T_EX e il L^AT_EX creano da un file sorgente (.tex) un file con la descrizione precisissima della pagina indipendente dal device di stampa o visualizzazione (.dvi), che poi unito alle fonts e ad altre sorgenti (figure) viene stampato o visualizzato da altri programmi appositi come xdvi o dvips o dvi_{pdf} o dvi_{hp} o altri.

Oggi si passa direttamente dai files sorgente tex a un pdf con pdftex o pdf_latex, usando fonts scalabili OpenType o TrueType, e incorporando eventuali figure.

Esempio minimale

Al solito occorre usare un editore di puro testo, come per scrivere programmi.

Proviamo a compilare un documento contenente

Definiamo $y = x^2 - ab$.

`\bye`

e otteniamo

$$Definiamo y = x^2 - ab$$

Esempio minimale

Al solito occorre usare un editore di puro testo, come per scrivere programmi.

Proviamo a compilare un documento contenente

```
Definiamo $y = x^2 - ab$.  
\bye
```

e otteniamo

$$\text{Definiamo } y = x^2 - ab$$

Oppure possiamo usare **comandi** T_EX, che cominciano con il simbolo `\`:

```
\[  
  \frac{\partial u(x,t)}{\partial t} =  
  \int_0^1 u(x,t) \varphi(x) dx  
\]
```

che produce

$$\frac{\partial u(x,t)}{\partial t} = \int_0^1 u(x,t) \varphi(x) dx$$

Macro

- ▶ In $\text{T}_\text{E}\text{X}$ è possibile definire delle **macro**, ovvero “comandi aggiuntivi”.
- ▶ Queste si possono usare per estendere il $\text{T}_\text{E}\text{X}$ – e fornire funzionalità aggiuntive. Tipicamente queste funzionalità vengono raggruppate in **pacchetti**.
- ▶ Ad esempio, c'è un pacchetto per le lettere gotiche, un pacchetto per scrivere del codice, un pacchetto per le slide (Beamer, che ho usato per questi lucidi), un pacchetto per disegnare, per fare grafici, ecc.

Un elenco di pacchetti disponibili si trova su:

The Comprehensive $\text{T}_\text{E}\text{X}$ Archive Network

<https://www.ctan.org/>

- ▶ Il $\text{T}_\text{E}\text{X}$ è un linguaggio di programmazione, e come spesso succede andare a capo non è troppo diverso da inserire uno spazio; per andare a capo nel testo bisogna usare (in $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$) `\\`, o lasciare una riga vuota.

- ▶ Nell'uso di tutti i giorni, è necessario avere un set di regole preconfezionate per il layout di pagina – altrimenti è facile incorrere in errori tipografici.
- ▶ Per questa ragione, è stato sviluppato il L^AT_EX, un set di **macro** T_EX che permette all'utente di descrivere la struttura del testo, senza preoccuparsi (troppo) della formattazione.
- ▶ Il L^AT_EX fornisce degli “ambienti”.
- ▶ Le case editrici distribuiscono inoltre dei pacchetti con particolari “stili” adatti ai loro libri e alle loro riviste, che personalizzano ulteriormente le pagine.
- ▶ Impareremo anche questo in laboratorio.

Documento L^AT_EX di esempio

```
\documentclass{article}

\title{Il mio documento}
\author{Johann Friedrich Carl Gau\ss}

\begin{document}
  \maketitle

  \section{Introduzione}
  Questo documento presenta la principali caratteristiche
  del \LaTeX; possiamo scrivere matematica in linea
  ( $a = 2 \cdot b$ ), o in equazioni
  \begin{equation}
    \frac{1}{2\pi i} \int_{\Gamma} f(z) dz = 0.
  \end{equation}
\end{document}
```

Ambienti

La differenza essenziale fra TEX e $\text{L}\text{A}\text{T}\text{E}\text{X}$ è la presenza di **ambienti**, che si aprono con il comando `\begin` e si chiudono con `\end`:

- ▶ `\begin{document} ... \end{document}` contiene tutto il testo.
- ▶ `\begin{equation} ... \end{equation}` contiene un'equazione numerata.
- ▶ ...

Automatismi

L^AT_EX si occupa automaticamente di molte cose:

- ▶ La gestione delle sezioni, e i riferimenti; aggiusta anche la spaziatura come necessario secondo le regole tipografiche.
- ▶ La numerazione di equazioni, teoremi, ecc.
- ▶ La gestione della bibliografia.
- ▶ Il posizionamento ottimale di figure e tabelle.
- ▶ Liste puntate, numerate, ecc.

Possiamo usare i comandi `\label` e `\ref` per marcare delle sezioni / teoremi / equazioni, e per creare dei riferimenti.

Riferimenti

```
\begin{equation} \label{eq:uno}
```

$$y = 2x$$

```
\end{equation}
```

Come visto in equazione~\eqref{eq:uno},
 y è il doppio di x .

Risulta nel seguente output:

$$y = 2x \tag{1}$$

Come visto in equazione (1), y è il doppio di x .

Lucidi

Generare lucidi è simile.

Queste slide sono fatte in \LaTeX usando la classe beamer.

Il codice di questa slide è più o meno:

```
\begin{frame}{Lucidi}
```

Il codice di ogni slide è
più o meno:

```
[...] blocco di codice qui [...]
```

```
\end{frame}
```

Ambienti Integrati

Per editare file tex di progetti complessi (molti files) invece di limitarsi ad usare un semplice editore, si può fare ricorso ad ambienti integrati come per esempio:

1. kile
2. TeXStudio
3. TexMaker
4. MacTeX (per macintosh)
5. MiKTeX (per Windows)
6. emacs
7. overleaf (online, per esempio su tex.cs.dm.unipi.it, usando e-mail e credenziali di ateneo)