Laboratorio di Comunicazione mediante Calcolatore

Leonardo Robol <leonardo.robol@unipi.it>

12 Novembre, 2018

Dati e protocolli

- La trasmissione di dati in Internet avviene tramite scambio di pacchetti di dati ("protocolli di trasmissione").
- Al di sopra di questo livello, le applicazioni devono accordarsi su un linguaggio per interpretare i dati scambiati ("protolli delle applicazioni").

Dati e protocolli

- La trasmissione di dati in Internet avviene tramite scambio di pacchetti di dati ("protocolli di trasmissione").
- Al di sopra di questo livello, le applicazioni devono accordarsi su un linguaggio per interpretare i dati scambiati ("protolli delle applicazioni").

Esempio: con il `telnet` in laboratorio abbiamo parlato ad un server web.

Email e protocolli

Lo scambio di email viene gestito in maniera simile, con diversi protocolli:

- SMTP: "Simple Mail Transfer Protocol", gestisce la spedizione delle e-mail. Questo è il vero protagonista dello smistamento delle e-mail.
- IMAP, POP3, sono invece protocolli che permettono agli utenti di consultare la propria casella e-mail.
- Una mail, nella forma più semplice, è sostanzialmente un file di testo con un'intestazione.
- L'intestazione (header) contiene i metadati del messaggio: data e ora, oggetto, mittente, destinatario, ecc.

Come viene spedita una email?

Supponiamo che Alice voglia scrivere a Bob.

Come viene spedita una email?

Supponiamo che Alice voglia scrivere a Bob.

1. Alice digita il messaggio sul suo calcolatore.
2. Tramite un software, Alice affida il messaggio ad un server SMTP (il relay), che lo prende in carico.
3. Tramite uno o più passaggi, il server consegna il messaggio al server in carico di gestire le mail di Bob (ricordate `host -t MX bob.com`?
4. Bob accede alla sua casella (con il suo software preferito, o una webmail), e legge il messaggio di Alice.

Come viene spedita una email?

Supponiamo che Alice voglia scrivere a Bob.

1. Alice digita il messaggio sul suo calcolatore.
2. Tramite un software, Alice affida il messaggio ad un server SMTP (il relay), che lo prende in carico.
3. Tramite uno o più passaggi, il server consegna il messaggio al server in carico di gestire le mail di Bob (ricordate `host -t MX bob.com`?
4. Bob accede alla sua casella (con il suo software preferito, o una webmail), e legge il messaggio di Alice.

La trasmissione dei messaggi avviene tramite SMTP.

Il protocollo SMTP

Assumiamo di voler scrivere a `leonardo.robol@unipi.it`.

```
leonardo@georg:~$ host -t MX unipi.it
unipi.it mail is handled by 50 emailsecurity.unipi.it.
```

Ora abbiamo determinato chi gestisce la posta per `@unipi.it`.
Colleghiamoci con `telnet`.

Il protocollo SMTP (continua)

```
leonardo@georg:~$ telnet emailsecurity.unipi.it 25
Trying 131.114.142.96...
Connected to emailsecurity.unipi.it.
Escape character is '^]'.
220 esra2.unipi.it ESMTP SonicWall (9.1.3.3987)
```

Il protocollo SMTP (continua)

```
leonardo@georg:~$ telnet emailsecurity.unipi.it 25
Trying 131.114.142.96...
Connected to emailsecurity.unipi.it.
Escape character is '^]'.
220 esra2.unipi.it ESMTP SonicWall (9.1.3.3987)

helo georg
250 esra2.unipi.it
```

Il protocollo SMTP (continua)

```
leonardo@georg:~$ telnet emailsecurity.unipi.it 25
Trying 131.114.142.96...
Connected to emailsecurity.unipi.it.
Escape character is '^]'.
220 esra2.unipi.it ESMTP SonicWall (9.1.3.3987)

helo georg
250 esra2.unipi.it

mail from: <leonardo@georg.cs.dm.unipi.it>
250 2.1.0 MAIL ok
```

Il protocollo SMTP (continua)

```
leonardo@georg:~$ telnet emailsecurity.unipi.it 25
Trying 131.114.142.96...
Connected to emailsecurity.unipi.it.
Escape character is '^]'.
220 esra2.unipi.it ESMTP SonicWall (9.1.3.3987)

helo georg
250 esra2.unipi.it

mail from: <leonardo@georg.cs.dm.unipi.it>
250 2.1.0 MAIL ok

rcpt to: <leonardo.robol@unipi.it>
250 2.1.5 <leonardo.robol@unipi.it> ok
```

Il protocollo SMTP (continua)

```
data
354 3.0.0 End Data with <CR><LF>.<CR><LF>
Subject: Email di prova
[ contenuto del messaggio qui ]
.
250 2.6.0 Message Accepted
```

Il protocollo SMTP (continua)

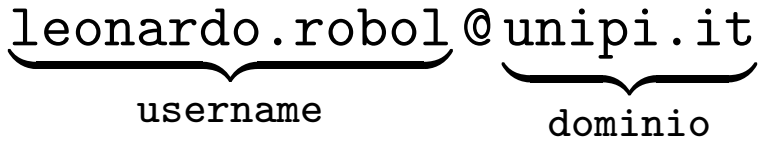
```
data
354 3.0.0 End Data with <CR><LF>.<CR><LF>
Subject: Email di prova
[ contenuto del messaggio qui ]
.
250 2.6.0 Message Accepted

quit
221 2.0.0 esra2.unipi.it says goodbye; [...]
Connection closed by foreign host.
```

Il protocollo SMTP (continua)

```
data
354 3.0.0 End Data with <CR><LF>.<CR><LF>
Subject: Email di prova
[ contenuto del messaggio qui ]
.
250 2.6.0 Message Accepted

quit
221 2.0.0 esra2.unipi.it says goodbye; [...]
Connection closed by foreign host.
```

- Per una serie di ragioni, questo messaggio sarà sicuramente finito nello SPAM.
- Il protollo SMTP è leggermente più complesso al giorno d'oggi: firme crittografiche permettono di controllare (in parte) che la catena di trasmissione sia corretta – noi abbiamo ignorato tutto questo.

Struttura di un indirizzo email



username è la parte dell'indirizzo che (solitamente) descrive l'utente locale.

dominio invece determina dove dev'essere recapitata l'email (ancora, utilizzando `host -t MX unipi.it`).

Struttura di un indirizzo email

$$\underbrace{\texttt{leonardo.robol}}_{\texttt{username}}\texttt{@}\underbrace{\texttt{unipi.it}}_{\texttt{dominio}}$$

username è la parte dell'indirizzo che (solitamente) descrive l'utente locale.

dominio invece determina dove dev'essere recapitata l'email (ancora, utilizzando `host -t MX unipi.it`).

Esistono molte varianti di questa sintassi, in particolare per lo username – sono perlopiù in disuso. Una che può tornare utile è:

`leonardo.robol+keyword@unipi.it`

- Alcuni caratteri non sono ammessi.
- Ultimamente, è stato esteso il set di caratteri utilizzabile anche per i domini (UTF8).

Struttura di un messagio

Un'email è sostanzialmente un file di testo, con questa struttura:

```
Subject: Messaggio
From: Leonardo Robol <leonardo.robol@unipi.it>
To: Leonardo Robol <leonardo.robol@unipi.it>

Contenuto del messaggio qui
```

- La prima parte si chiama header, e contiene linee del tipo `Chiave: valore`.
- La seconda è il corpo del messaggio (body).
- Sono separate da una linea vuota.

Dissezione di un header

Consideriamo questa e-mail che mi sono auto-mandato:

Dissezione di un header

Consideriamo questa e-mail che mi sono auto-mandato:

Dal programma di posta è possibile aprire il sorgente della e-mail, ed ispezionarne il contenuto.

Dissezione in un header

```
Return-Path: <leonardo.robol@unipi.it>

Received: from mx3.unipi.it (mx3.unipi.it [131.114.21.49])
by mbox5.unipi.it (Postfix) with ESMTP id 8F11DE02B8
for <a019485@mbox5.unipi.it>;
Fri,  9 Nov 2018 08:01:59 +0100 (CET)

Received: from localhost (localhost [127.0.0.1])
by mx3.unipi.it (Postfix) with ESMTP id 95B11C02E3
for <leonardo.robol@unipi.it>;
Fri,  9 Nov 2018 08:01:59 +0100 (CET)

[...] (altri 5 passaggi intermedi)

Received: from georg (dhcp05.cs.dm.unipi.it [131.114.10.165])
(Authenticated User)
by smtp.unipi.it (Postfix) with ESMTPSA id B4A8B40DA1
for <leonardo.robol@unipi.it>;
Fri,  9 Nov 2018 08:01:58 +0100 (CET)
```

Dissezione di un header (parte 2)

```
[...]

Date: Fri, 09 Nov 2018 08:02:19 +0100
From: Leonardo Robol <leonardo.robol@unipi.it>
Subject: Messaggio di prova
To: LEONARDO ROBOL <leonardo.robol@unipi.it>
Message-Id: <1541746939.31612.2@smtp.unipi.it>
X-Mailer: geary/0.12.4
```

Dissezione di un header (parte 2)

```
[...]

Date: Fri, 09 Nov 2018 08:02:19 +0100
From: Leonardo Robol <leonardo.robol@unipi.it>
Subject: Messaggio di prova
To: LEONARDO ROBOL <leonardo.robol@unipi.it>
Message-Id: <1541746939.31612.2@smtp.unipi.it>
X-Mailer: geary/0.12.4


Questo e' un messaggio di prova.

-- Leonardo.
```

MIME

- In realtà, raramente i messaggi hanno un corpo così semplice e leggibile;
- Molti software utilizzando un formato più complesso per avere un documento con più parti nel testo (ad esempio, testo formattato, poi uno o più allegati, ...).
- A volte il contenuto viene codificato in modo particolare (base64) – rendendolo di fatto illeggibile ad un essere umano.

Server di posta

- Normalmente, noi affidiamo le nostre e-mail ad un server di posta (SMTP) che gira su qualche server (`smtp.unipi.it`, ad esempio).
- Su sistemi Linux è però (abbastanza) comune avere un server su ogni macchina.
- Questo permette di spedire e-mail utilizzando comandi appositi: `sendmail`, `mail`, ...
- Le macchine dell'Aula 4 hanno un server SMTP ciascuna: questi comandi sono disponibili!
- In realtà, questi server redirigono semplicemente tutte le e-mail ad un altro mail server del dipartimento.

Il comando mail

Con il comando `mail` possiamo scrivere e-mail dalla linea di comando:

```
$ mail -s "Subject" utente@gmail.com
Messaggio di prova
.
Cc:
```

Il comando mail

Con il comando `mail` possiamo scrivere e-mail dalla linea di comando:

```
$ mail -s "Subject" utente@gmail.com
Messaggio di prova
.
Cc:
```

Mini-esercizio: come possiamo utilizzare questo comando per spedire una stessa e-mail a moltissime persone?

Una mailing list “fatta in casa”

```
$ cat indirizzi.txt
utente1@gmail.com
utente2@libero.it
[...]

$ cat email.txt
From: Me <me@unipi.it>

Ciao,
[...]

$ for i in $(cat indirizzi.txt); do
    mail -s "Saluto" $i < email.txt
  done
```

SPAM

- La spedizione di e-mail è libera: chiunque può mandarmi un messaggio.
- Questo causa un'abbondanza di messaggi di spam nelle nostre caselle, in particolare se il nostro indirizzo viene pubblicato online.
- La maggior parte dei mailserver utilizza dei filtri di vario tipo per cercare di limitare queste dinamiche.
- Due tipo di filtri principali: blocking list e filtri statistici / Bayesiani.

Blocking list

- Idea molto semplice: mantenere una lista di PC che sono noti per spedire mail di spam (e.g., PC infetti, o server tipicamente utilizzati da spammer).
- La lista viene continuamente aggiornata, se il vostro PC è infetto ci potete facilmente finire per sbaglio!
- Esiste una procedura di removal per chiedere di essere rimossi dalla lista.

Filtri Bayesiani

Consideriamo $\mathcal{E}$ l'insieme di tutte le e-mail, partizionato come

$$\mathcal{E} = \mathcal{S} \cup \mathcal{M}, \qquad \mathcal{S} \cap \mathcal{M} = \emptyset,$$

e dove $\mathcal{S}$ sono le e-mail di spam, e $\mathcal{M}$ quelle "regolari".

- Idea: l'occorrenza di varie parole è diversa in $\mathcal{S}$ ed in $\mathcal{M}$;
- Dato un sample di emails in $\mathcal{M}$ ed $\mathcal{S}$, possiamo studiare le probabilità che una nuova email e stia in $\mathcal{S}$:

$$\mathbb{P}(e \in \mathcal{S} \mid e \in W) = \frac{\mathbb{P}(e \in W \mid e \in \mathcal{S}) \cdot \mathbb{P}(\mathcal{S})}{\dots}$$

dove W è un'insieme di e-mail contenente certe parole (e la parte omessa è fatta di cose note o "facilmente stimabili").

Ma da dove viene il termine SPAM?

Ma da dove viene il termine SPAM?

Client di posta

Ci sono svariati client di posta:

- Webmail (GMail, Roundcube (email di ateneo), . . .)
- Client desktop (Outlook, Apple Mail, Thunderbird, . . .)
- Client per smartphone

Mailing list

- Spesso, si vuole comunicare con un insieme di persone (o discutere di qualcosa).
- Per questo, sono state inventate le mailing-list.
- Quando si spedisce ad una mailing list, tutti ricevono il messaggio; rispondendo alla mailing list si continua la discussione.
- Voi siete già iscritti a varie mailing list: Studenti, Galois, ...

WWW e HTML

- La scorsa settimana abbiamo visto come funziona un collegamento HTTP, che permette di scaricare una pagina web.
- Generalmente, HTTP (o la sua versione criptata HTTPS) sono i protocolli più usati per navigare in Internet.
- Altri protolli (gopher, e in qualche misura anche FTP), sono più in disuso.

Storia del WWW

- Nel 1991, Tim Berners Lee, lavorando al CERN, creò un sistema che permetteva di ospitare delle pagine con collegamenti (ipertesti);
- Il sistema era un evoluzione di quanto aveva sviluppato negli anni prima; fino ad allora, internet veniva utilizzata per le email ma non esisteva un modo semplice di scambiare file.

Storia del WWW

- Nel 1991, Tim Berners Lee, lavorando al CERN, creò un sistema che permetteva di ospitare delle pagine con collegamenti (ipertesti);
- Il sistema era un evoluzione di quanto aveva sviluppato negli anni prima; fino ad allora, internet veniva utilizzata per le email ma non esisteva un modo semplice di scambiare file.
- Con li tempo, sono nati dei browser, è stata aggiunta una componente "grafica" alle pagine, e la tecnologia si è evoluta. Ma il protocollo di trasmissione ha ricevuto pochi cambiamenti.

Collegamenti ipertestuali

- La grossa novità del WWW era la possibilità di passare da una pagina all'altra tramite "collegamenti".
- Questo permetteva di catagolare facilmente i contenuti!
- Un enorme passo in avanti è stato fatto poi dai motori di ricerca, che riescono a determinare quale pagina è importante e quale no (come? è un problema di autovalori e autovettori).

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`
  - `...`

Nel prossimo laboratorio vedremo come creare una pagina web.

Server Proxy

A volte è preferibile non utilizzare una connessione diretta al server web, ma passare attravero un "proxy". Ad esempio, questo permette di accedere alla riviste scientifiche per cui l'università paga l'abbonamento:

- `ssh -D 8080 ssh1.dm.unipi.it`: Apertura di un proxy attraverso SSH;
- Configurare il proxy sul proprio PC (tipicamente Impostazioni $\rightarrow$ Rete $\rightarrow$ Proxy, oppure direttamente dal browser).
- Navigare su `mathscinet.com`.

Browser

- Per visualizzare la pagine web, c'è bisogno di un browser.
- Potete scegliere il vostro preferito, Firefox, Chrome/Chromium, Edge, Opera, ...
- Linux dispone di alcuni comandi per scaricare dati:
  - `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.

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?

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 (ragionevolemente) sicuri che i nostri dati non siano utilizzati per la pubblicità – possiamo aprire una scheda in incognito.
- Alcuni dati però sono ancora disponibili: IP, geolocation, e se mi loggo perdo l'anonimato.

Come creare una pagina HTML?

Il linguaggio HTML è un linguaggio di markup. Una pagina base 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 conterere dell'altro HTML al loro interno, e/o avere attributi.

Aggiungiamo degli attributi

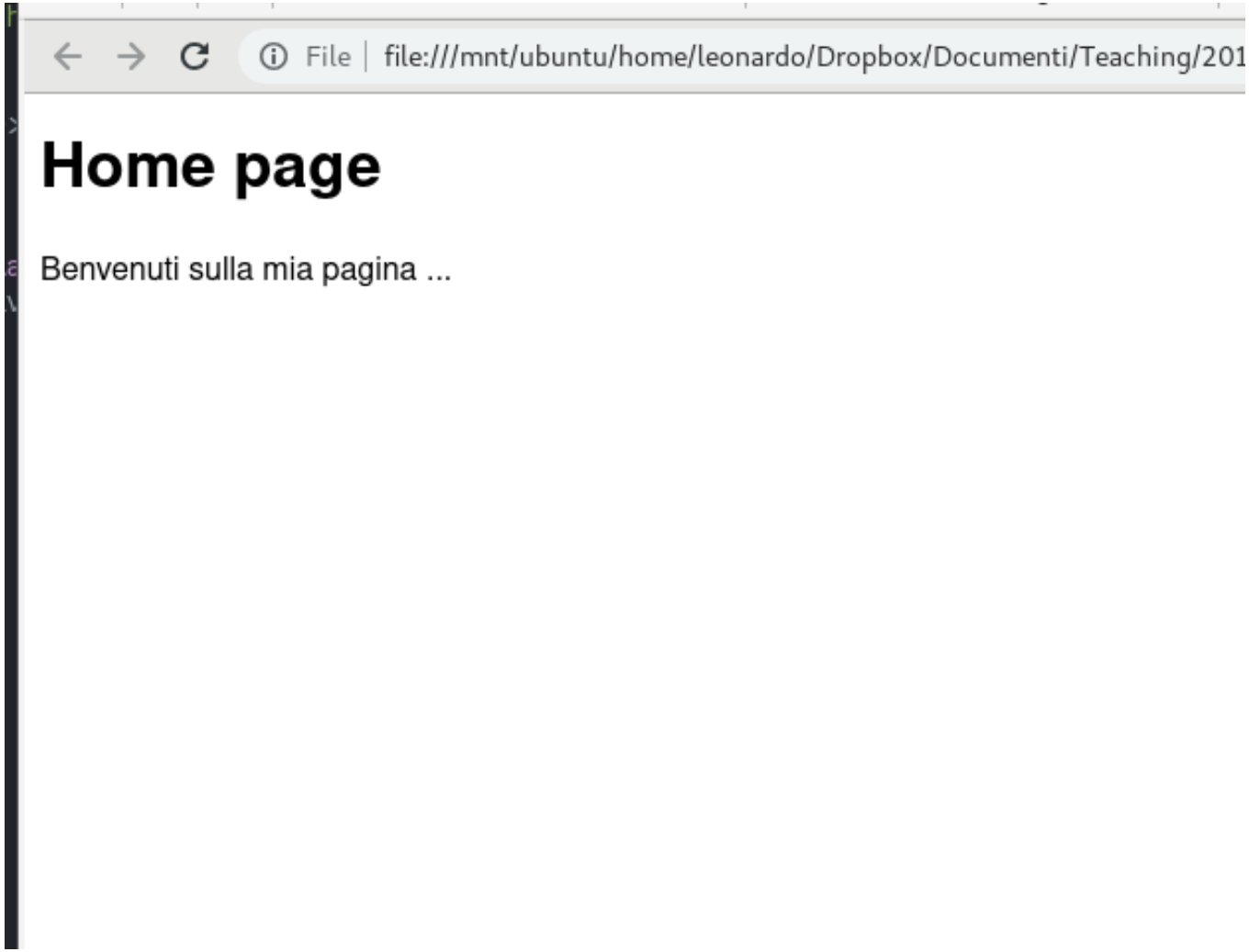
```
<!DOCTYPE html>
<html>
  <head>
    <title>La mia pagina personale</title>
  </head>

  <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

- `<a href="http://www.google.com">Open google.com</a>`: un collegamento a Google (tag "anchor").
- `<img src="images/cat.png" />`: Un'immagine di un gattino: notate la sintassi finale `/>`: questo è equivalente ad aprire e chiudere un tag, ovvero a scrivere `<img src="images/cat.png"></img>`.
- Molti altri tag che vedremo in laboratorio.

Uno sguardo alla nostra pagina

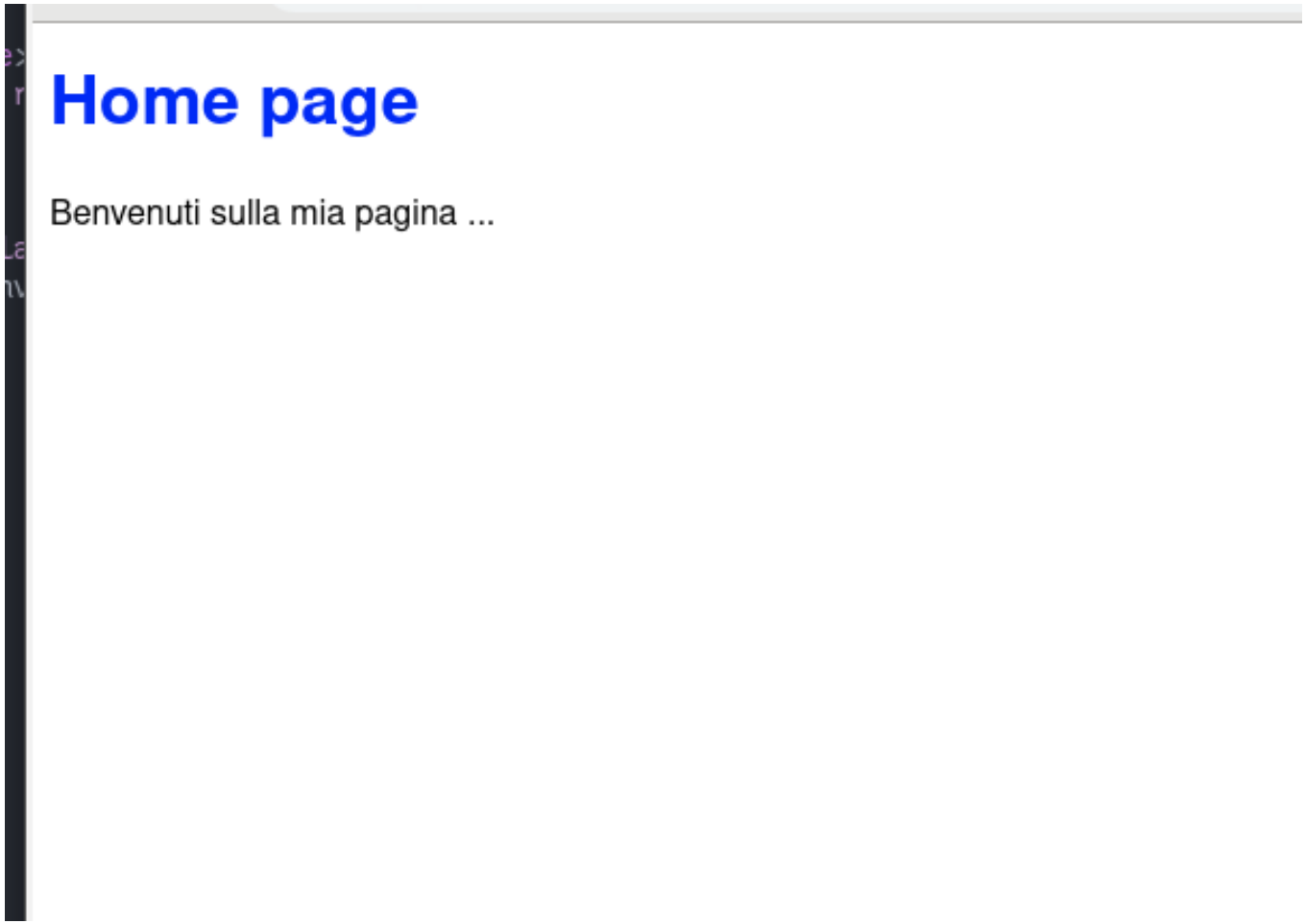


CSS

- 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 pagine, 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



CSS

- 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

CSS

- 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 LATEX

- Donald Knuth è l'autore di "The Art of Computer programming".
- 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.

TEX e LATEX

- Donald Knuth è l'autore di "The Art of Computer programming".
- 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.

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

Un po' di storia

- Knuth cominciò a lavorare al TEX nel 1977, progettando di finirlo nel seguente anno.
- In realtà, la versione finale fu rilasciata solo nel 1989 (TEX 3); da quel momento le nuove versioni hanno un numero che converge a π.
- Ad esempio, l'ultima release del TEX 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 TEX (MetaFont) ha raggiunto la versione 2, e da allora converge a $e = 2.71828\ldots$.

Funzionamento del TEX

Il TEX è un software di typesetting – ma anche un linguaggio di programmazione; dato un file in linguaggio TEX, compilarlo con il comando `tex` genera in output un file PDF.

- L'interprete TEX legge un file `esempio.tex`, tramite il comando `pdftex`.
- Valuta i vari comandi, e produce dei file ausiliari, insieme ad un file `esempio.pdf`.

Esempio minimale

Proviamo a compilare un documento contenente

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

e otteniamo

Definiamo $y = x^2 - ab$

Esempio minimale

Proviamo a compilare un documento contenente

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

e otteniamo

Definiamo $y = x^2 - ab$

Oppure possiamo usare comandi TEX, che cominciano con il simbolo \:

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

che produce

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

Macro

- In TEX è possibile definire delle macro, ovvero "comandi aggiuntivi".
- Queste si possono usare per estendere il TEX – 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.
- Il TEX 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 \\, 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 LaTeX, un set di macro TeX che permette all'utente di descrivere la struttura del testo, senza preoccuparsi (troppo) della formattazione.
- Il LaTeX fornisce degli "ambienti".
- Impareremo anche questo a laboratorio.

Documento LaTeX di esempio

```
\documentclass{article}

\title{Il mio documento}
\author{Leonardo Robol}

\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_{\partial \Gamma} f(z) dz = 0.
  \end{equation}
\end{document}
```

Ambienti

La differenza essenziale fra TEX e LATEX 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

LaTeX 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: il codice di questa slide è più o meno:

```
\begin{frame}{Lucidi}
  Generare lucidi è simile: il codice di questa slide è
  più o meno:

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

\end{frame}
```