

HTML (Hyper Text Markup Language)

- **Linguaggio** per descrivere una **pagina di ipertesto**
- Specifica **come dovrà apparire** quando sarà visualizzata, quali **collegamenti** contiene e **dove portano**
- **Comando in HTML**: riguarda una **porzione di testo**, tag all'inizio e alla fine
- Es.: ` testo ` per il grassetto

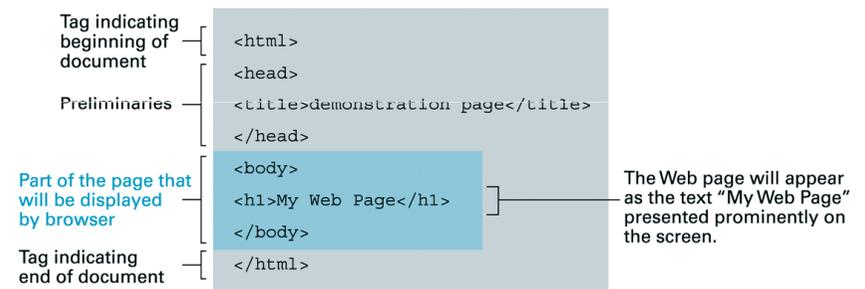
Struttura di un documento HTML

```
<HTML>  
<HEAD>  
<TITLE>  
Titolo della pagina (viene usato dai motori di ricerca)  
</TITLE>  
</HEAD>  
<BODY>  
Quello che sarà visualizzato dal browser  
</BODY>  
</HTML>
```

Comandi di HTML

- Per definire un paragrafo `<p>`
- Per far apparire un testo in grassetto `` o corsivo ``
- Per centrare un pezzo di testo `<CENTER>`
- Per realizzare una lista:
``
 `` primo elemento
 `` secondo elemento
``
- Per definire il colore dello sfondo:
`<BODY COLOR = "#xxxyyzz">` (codice RGB)
- Per inserire un collegamento: ` hotword `

Altro esempio di HTML



Motori di ricerca -- 1

- Per poter accedere ad un **sito Internet**, dobbiamo sapere il suo **URL**
- Troppi siti, e nuovi siti ogni giorno → non ci sono indici di tutti i siti
- I **motori di ricerca** permettono di specificare alcune **parole chiave**, e trovano i siti piu' rilevanti per queste parole
- Alcuni motori di ricerca:
 - www.google.com
 - www.yahoo.com
 - www.lycos.com
 - www.virgilio.it
 - www.arianna.it

Motori di ricerca -- 2

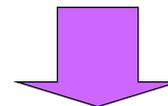
- Le pagine **non vengono cercate in quel momento**:
 - I motori di ricerca **navigano continuamente** su Internet, e a mano a mano che trovano nuove pagine le **associano a certe parole chiave** (es. quelle nel titolo)
 - Quando arriva una parola chiave, hanno gia' le pagine relative a questa parola → **risposta in breve tempo**
- Diverse politiche che ordinano i risultati ottenuti
- **Esempio: Google → PageRank**
 - A pagina puntata dalle pagine **B1...Bn**
 - $C(B_i)$ = numero di pagine puntate da B_i
 - d = fattore di decadimento ~ 0.85
 - $PR(A) = d + d \times (PR(B_1)/C(B_1) + \dots + PR(B_n)/C(B_n))$

Protocolli di rete -- 1

- Come **collegare in rete elaboratori diversi**?
- **Anni '70**:
 - Elaboratori tutti **uguali**
 - **Stesso sistema operativo**
 - **Un solo mezzo** trasmissivo

Protocolli di rete -- 2

- In seguito, elaboratori **diversi** ma **stesso software di base**
- Ma se si collegano **sistemi diversi**?



Necessita' di uno **standard architetturale** che regoli la realizzazione del **software di comunicazione**

Protocolli di rete -- 3

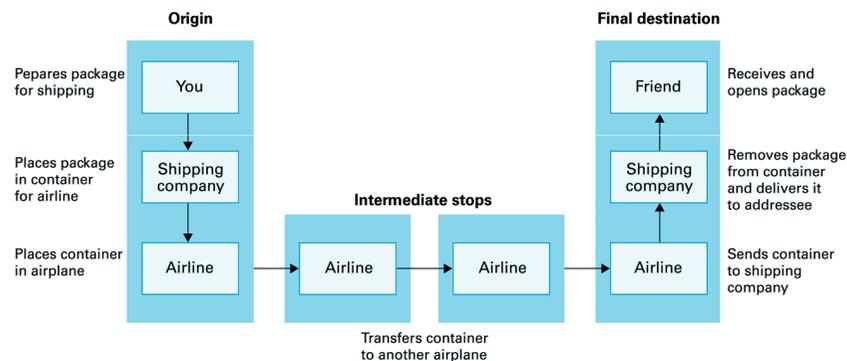
- Nel 1978 l'ISO (International Standard Organization) propose il **modello ISO/OSI** (Open System Interconnection) per **specificare il modo in cui elaboratori diversi comunicano tramite la rete**
- Architettura di rete strutturata in **sette livelli**

- **Modello TCP/IP** per Internet
- Architettura di rete strutturata in **quattro livelli**

Esempio: spedizione di un pacco

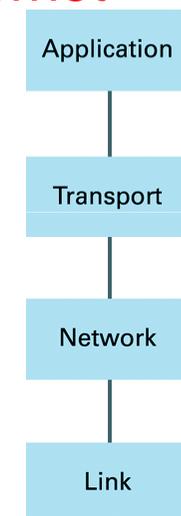
- **Alla partenza:**
 - **Noi:** Scriviamo l'indirizzo e diamo il pacco ad una ditta di spedizioni
 - **Ditta di spedizioni 1:** mette il pacco insieme ad altri in un container e lo da' ad una compagnia aerea
 - **Compagnia aerea:** carica il container sull'aereo, fa il viaggio aereo
- **All'arrivo:**
 - **Compagnia aerea:** scarica il container e lo mette negli spazi della ditta di consegne 2
 - **Ditta di consegne 2:** estrae il pacco dal container e lo consegna all'indirizzo
 - **Destinatario:** riceve il pacco

Schema a tre livelli per l'esempio



Comunicazione su Internet

- **Quattro livelli di software:**
 - Applicazione (application)
 - Trasporto (transport)
 - Rete (network)
 - Connessione dati (data link)



Livello **applicazione**

- Software di utilita' per **richiedere** la comunicazione via Internet
- Es.: **FTP** (software per trasferire file tra calcolatori)
- Es.: **telnet o ssh** (software per accedere ad un calcolatore in rete come se fossimo utenti locali)
- Analogo di noi che **facciamo il pacco e scriviamo l'indirizzo giusto**: deve specificare l'indirizzo IP del calcolatore con cui comunicare

Livello **trasporto**

- Accetta messaggi dal livello applicazione
- Li divide in pacchetti e **spedisce i pacchetti**
 - Nei punti di smistamento di Internet, un messaggio lungo costringerebbe tutti gli altri messaggi ad aspettare la sua trasmissione completa
- Ad ogni pacchetto aggiunge:
 - dei bit per riconoscerli e riassemblare il messaggio alla destinazione
 - l'indirizzo IP della destinazione
- Passa i pacchetti cosi' trasformati al livello di rete

Livello di **rete**

- Manda ogni pacchetto come un messaggio a se' stante
 - Anche **percorsi diversi** per **pacchetti dello stesso messaggio**
- Controlla che i pacchetti arrivino a destinazione
- Ad ogni nodo intermedio, **decide** l'instradamento del pacchetto, cioe' **l'indirizzo di destinazione intermedia**
- **Aggiunge questo indirizzo** al pacchetto e lo passa al livello di connessione

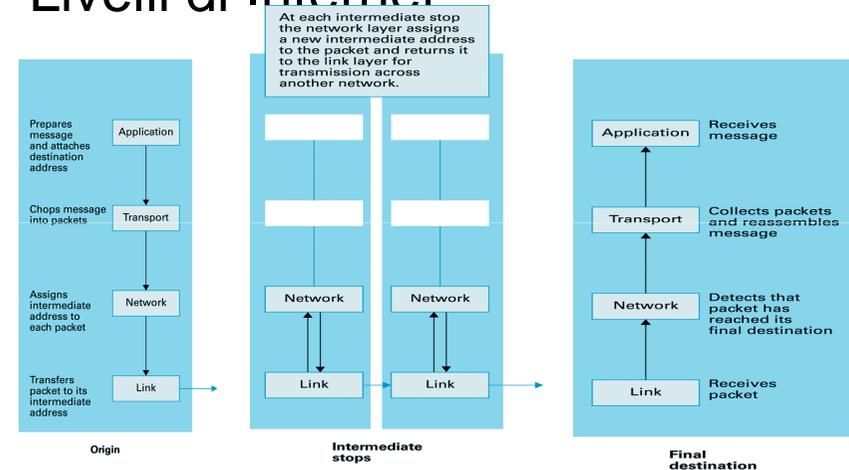
Livello **connessione**

- Gestisce la **trasmissione affidabile dei dati** tra **nodi adiacenti** della rete (cioe' fisicamente collegati da un mezzo trasmissivo)

All'arrivo ...

- Ogni pacchetto viene ricevuto dal livello **connessione**
- Lo passa al livello di **rete**
- **Se destinazione finale = indirizzo locale** →
 - Lo passa al livello **trasporto**
 - Quando **tutti** i pacchetti arrivano al livello **trasporto**, **ricostruisce il messaggio e lo passa al livello applicazione**
- **Altrimenti**, aggiunge un nuovo **indirizzo intermedio** e lo ripassa al livello **connessione**
- Solo livello **connessione** e **rete** nelle fermate intermedie

Livelli di Internet



Riassumendo

- **Livello applicazione:** messaggi e indirizzi per un software di utilità
- **Livello trasporto:** da messaggi a pacchetti e viceversa
- **Livello di rete:** instradamento pacchetti attraverso Internet
- **Livello connessione:** trasmissione pacchetti tra computer all'interno di una rete
- Tempo totale: **millisecondi**