

Informatica di base 6/ed

Autori: Dennis P. Curtin, Kim Foley, Kunal Sen e Cathleen Morin

A cura di: Agostino Marengo e Alessandro Pagano



Capitolo 13

La programmazione e lo sviluppo di sistemi

Copyright © 2016

McGraw-Hill Education (Italy) srl





- **Computer come macchina**
 - Alimentata da corrente elettrica è in grado di svolgere vari compiti.
 - I compiti sono svolti grazie a sequenze di istruzioni dette programmi.
- **I programmi sono scritti dai progettisti di software.**
- **Similitudine tra attività quotidiane del mondo reale e algoritmi => programmi.**

Il concetto di programma



- Il calcolatore elettronico è simile ad un qualsiasi elettrodomestico, ma:
 - gli elettrodomestici
 - uso specifico;
 - funzione determinata dall'interazione tra i componenti fisici.
 - i calcolatori:
 - sono macchine universali in grado di svolgere un numero potenzialmente infinito di compiti;
 - per passare da un'operazione all'altra non occorre modificare i componenti fisici;
 - la RAM permette di memorizzare ed eseguire le istruzioni.

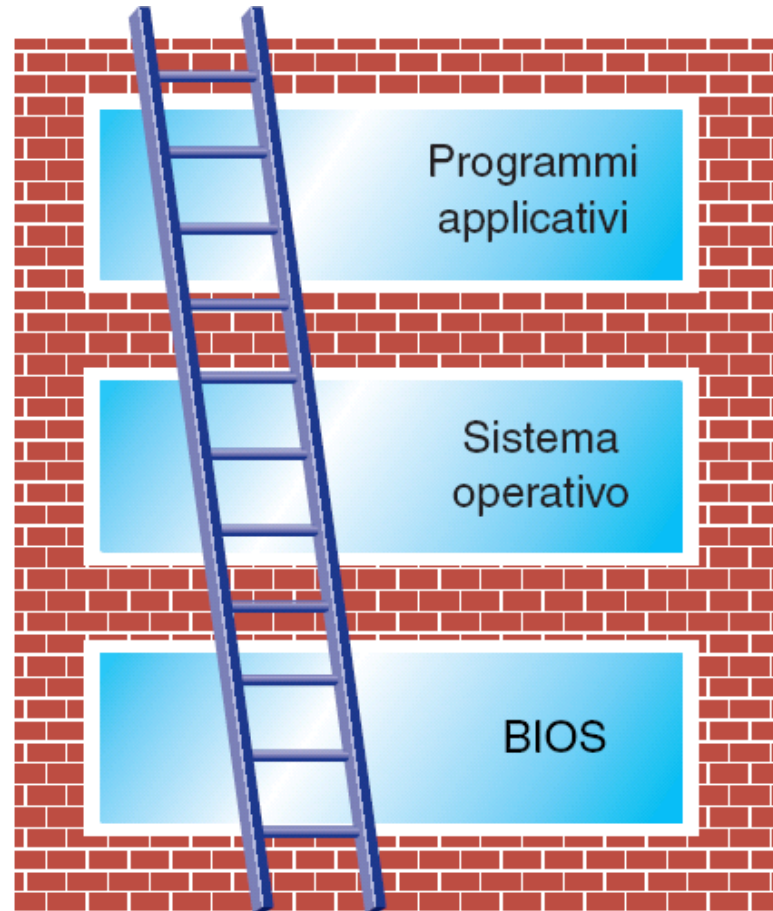
La scala dei programmi



Programmi applicativi:
programmi per svolgere
compiti precisi.

Sistema operativo:
programma per gestire tutte le
risorse del computer.

BIOS:
eseguito all'accensione del
computer, verifica i
componenti presenti e
l'assenza di guasti gravi.



La scala dei programmi

Scrittura di un programma

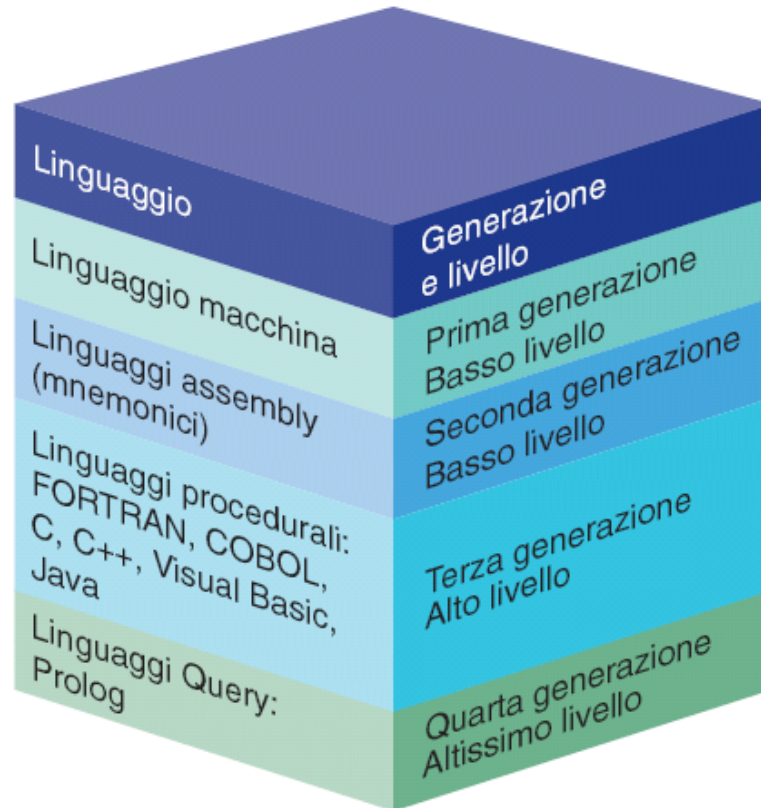


- I **programmatori** convertono le specifiche (descrizioni accurate) di un problema in una sequenza di istruzioni per risolvere una classe di problemi per mezzo del computer.
- Per scrivere (codificare) un programma si utilizza un **linguaggio di programmazione**.
- I linguaggi di programmazione si suddividono in:
 - generazioni;
 - livelli.

Generazioni e livelli



La generazione indica il periodo di ideazione di un linguaggio di programmazione.



Il livello indica il grado di vicinanza con il "linguaggio macchina" (insieme di 0 e 1).

Le generazioni dei linguaggi di programmazione

Prima e seconda generazione



Linguaggio di programmazione = sintassi rigida + parole in numero fisso.

- **I generazione:**

- linguaggio macchina = istruzioni e dati sono solo numeri;
- è l'unico linguaggio che la CPU comprende.

- **II generazione:**

- linguaggi assembletativi = le istruzioni e i dati sono sostituiti da espressioni mnemoniche della loro funzione;
- questo linguaggio va tradotto in linguaggio macchina tramite un *assemblatore*.

Terza generazione



- Linguaggi procedurali = la sintassi e le parole si arricchiscono di elementi che “riassumono” funzioni più complesse in un'unica istruzione.
- Il codice assomiglia sempre più a espressioni del linguaggio naturale.
 - Un traduttore si occupa di sciogliere le espressioni complesse (codice sorgente) in codice oggetto, ossia in linguaggio macchina:
 - il traduttore si chiama **compilatore** se produce una versione del programma in linguaggio macchina che poi viene eseguita;
 - **interprete** se esegue la traduzione in progressione col codice sorgente.

Esempi di linguaggi procedurali



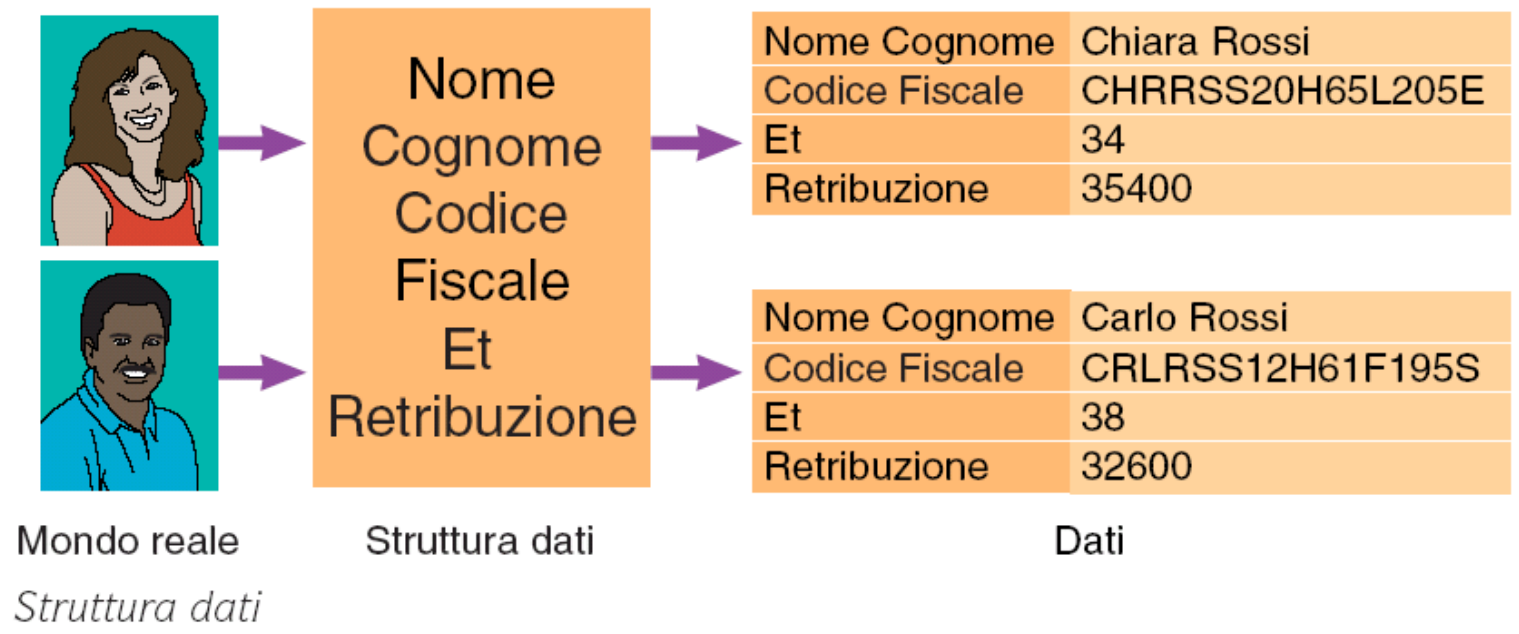
- FORTRAN
- COBOL
- BASIC
- Pascal
- C
- C++
- Java
- Javascript (da inserire in pagine HTML)

Metodologia di programmazione



- Rappresentazione astratta del mondo reale (**struttura dati**).
 - Dati di interesse per risolvere il problema.
- Insieme di passi (algoritmo) per risolvere il problema partendo dai dati di input.
- Migliore sarà l'organizzazione delle strutture dei dati → più semplici e affidabili saranno gli algoritmi.

Struttura dati e algoritmo



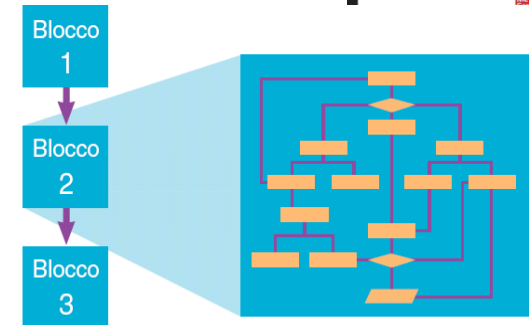
Algoritmo = l'insieme dei passi necessari per eseguire e portare a termine un compito determinato, agendo su determinati dati e strutture dati.

Tipi di programmazione



Programmazione a blocchi:

il codice è scritto in piccoli blocchi con un solo punto d'ingresso e uno d'uscita.



Un programma strutturato a blocchi

Modulo 1



Modulo 2



Modulo 3



Modulo 4



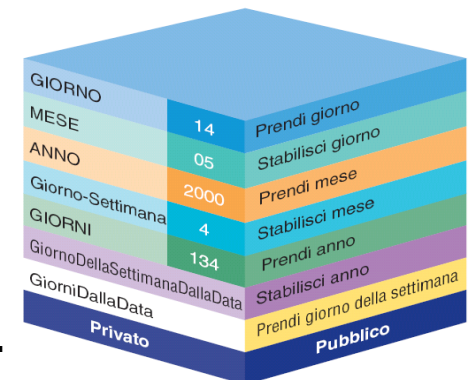
Un programma modulare

Programmazione modulare:

il programma è costituito da parti indipendenti sviluppate da programmatori diversi, che hanno in comune la sola interfaccia esterna.

Programmazione a oggetti:

un "oggetto" è un insieme fatto di una struttura dati e procedure che la elaborano in modo da restituire informazioni.



Un esempio di programmazione a oggetti

Sviluppo del software



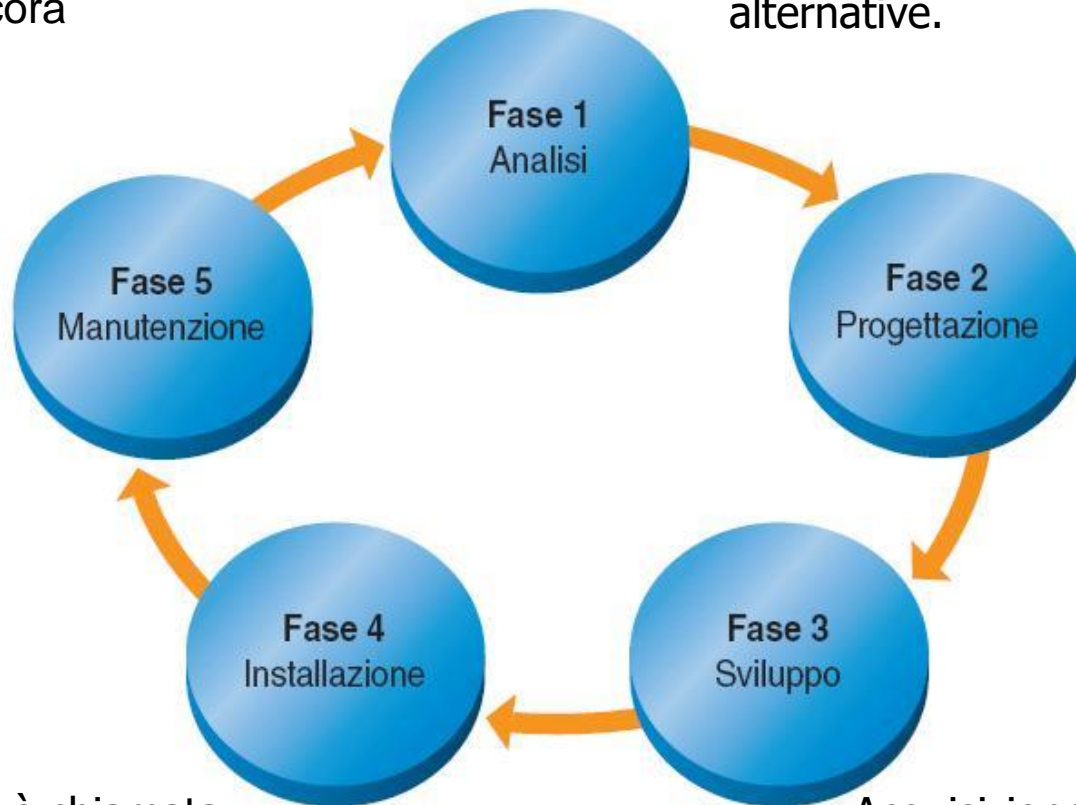
Le fasi di sviluppo del software

1. **Definizione accurata del problema** da risolvere e dei passi necessari per farlo.
2. **Progettazione del programma** con diagrammi di flusso e pseudoodici.
3. **Codifica del programma** in uno dei linguaggi di programmazione possibili.
4. **Verifica dei risultati**, con conseguente caccia al "bug" nascosto.
5. **Scrittura dei documenti** che accompagnano il programma: manuale, doc.tecnici, ecc.

I sistemi in ambito aziendale



La manutenzione dura finché il sistema è ancora operativo. Si risolvono i problemi e si introducono modifiche migliorative.



Analisi del problema dell'azienda e definizione dettagliata dei problemi. Proposte di soluzioni alternative.

Accettata l'analisi e una delle soluzioni proposte, la si progetta collettivamente.

Il consenso e la scelta tra progetti diversi è compito dell'azienda.

Acquisizione dell'hardware e del software e verifica del nuovo sistema.

L'installazione è chiamata conversione se si sostituisce un sistema già esistente.

Fase di installazione: dettagli



- **Installazione diretta**
 - Disattivazione vecchio sistema e avvio di quello nuovo
- **Installazione parallela**
 - Installazione nuovo sistema senza alterare quello vecchio
 - Il carico di lavoro viene portato avanti parallelamente
 - Quando il nuovo sistema è totalmente affidabile il vecchio viene disinstallato
- **Installazione scaglionata**
 - Introduzione nuovo sistema per gradi/fasi
 - Tempi lunghi
- **Installazione pilota**
 - Il nuovo sistema viene utilizzato solo per una piccola parte delle attività aziendali
 - Installazione meno rischiosa

World Wide Web

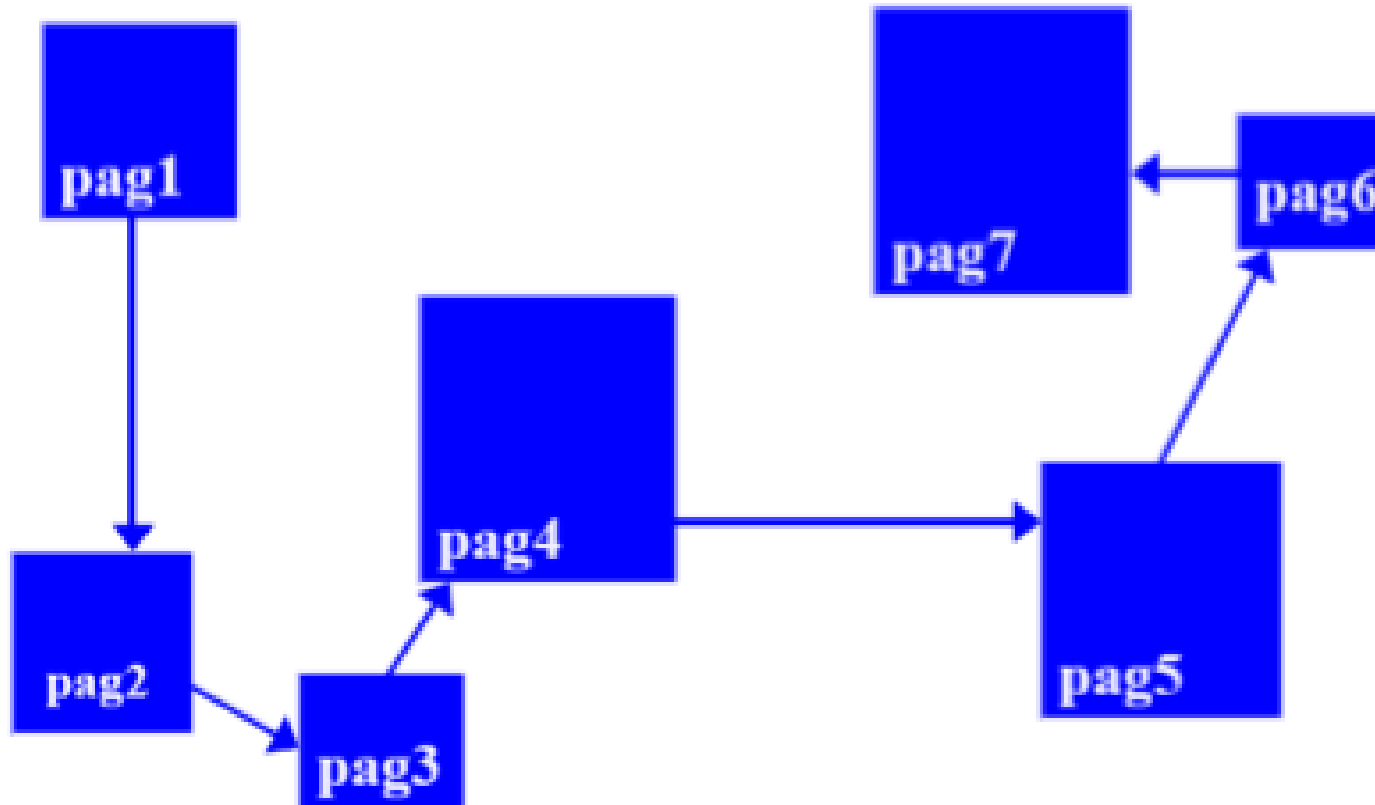


- Si basa sul protocollo http ed è la vera novità degli anni '90.
- Sviluppato presso il CERN di Ginevra (da Tim Berners-Lee) è il più potente mezzo di diffusione telematica di documenti elettronici.
- Mezzo di comunicazione globale, interattivo, multimediale e ipertestuale ha cambiato radicalmente il modo di comunicare e di lavorare.

Organizzazione di un testo



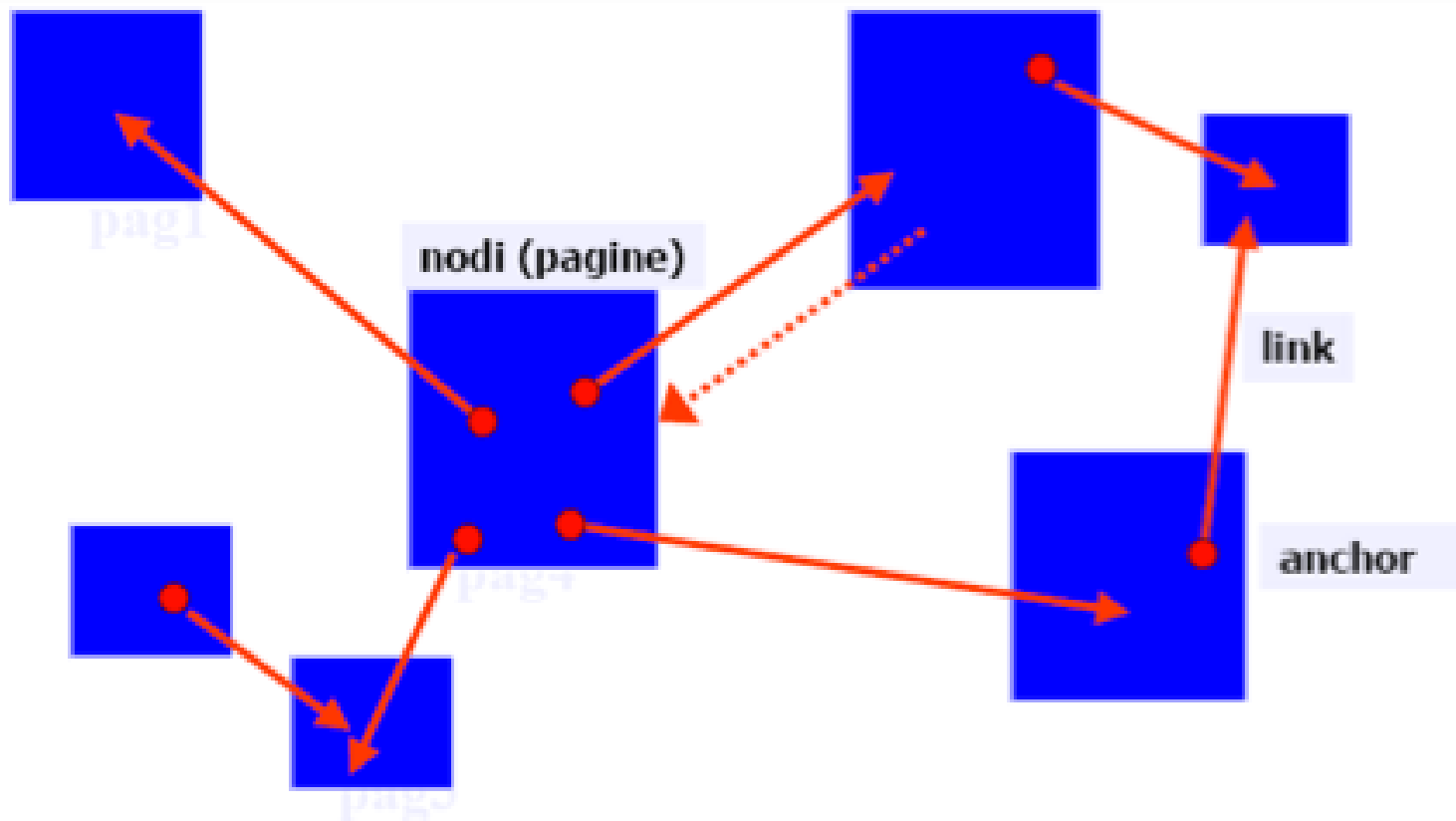
Accesso sequenziale



Organizzazione di un ipertesto



Accesso secondo legami associativi



L'ipertesto è semplicemente un testo che funziona come un link.

Linguaggi di markup (1/2)



- In un documento il markup è il codice che contiene le informazioni per la sua formattazione.

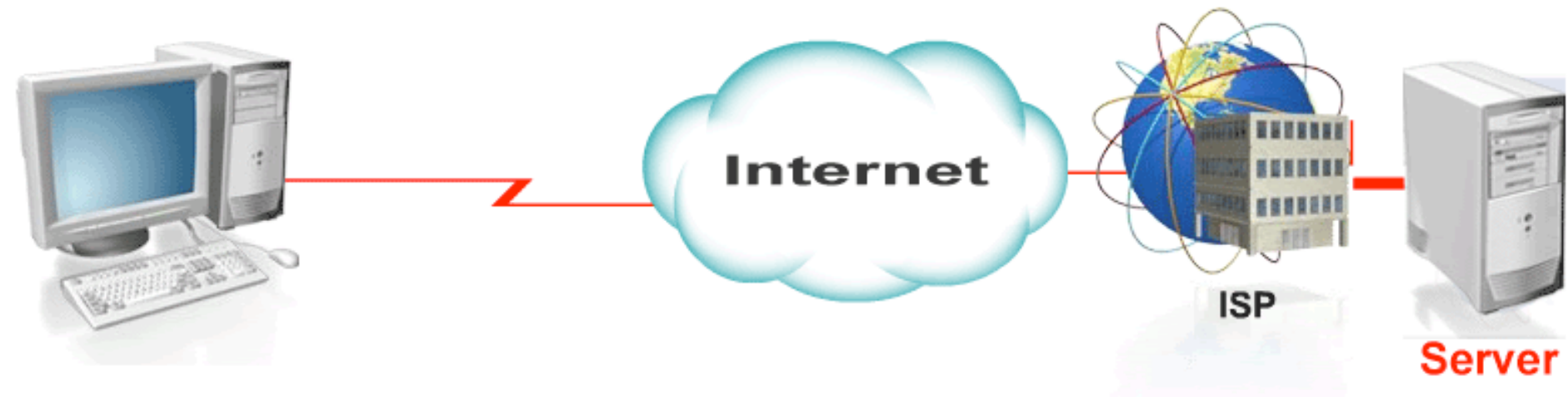


Linguaggi di markup (2/2)



- Nell'ambito dell'elaborazione elettronica dei testi sono stati creati strumenti informatici per automatizzare alcune operazioni per la creazione dei documenti.
- Esistono programmi per l'elaborazione dei testi di tipo diverso.
 - Gli editor **WYSIWYG**.
 - Gli editor che usano il concetto di **markup**.
- Nel caso di editor che usano il concetto di markup, un testo è costituito da due parti.
 - Il contenuto vero e proprio.
 - Le “istruzioni” che specificano come il contenuto deve essere rappresentato sul dispositivo (lo schermo di un PC, ma anche per esempio una stampante).
- In genere si racchiude il testo fra istruzioni chiamate **tag** (o etichette, o codici).
- I tag sono scritti usando i normali caratteri e quindi i documenti possono essere creati usando semplici editor di testo.

Come funziona la navigazione Internet?



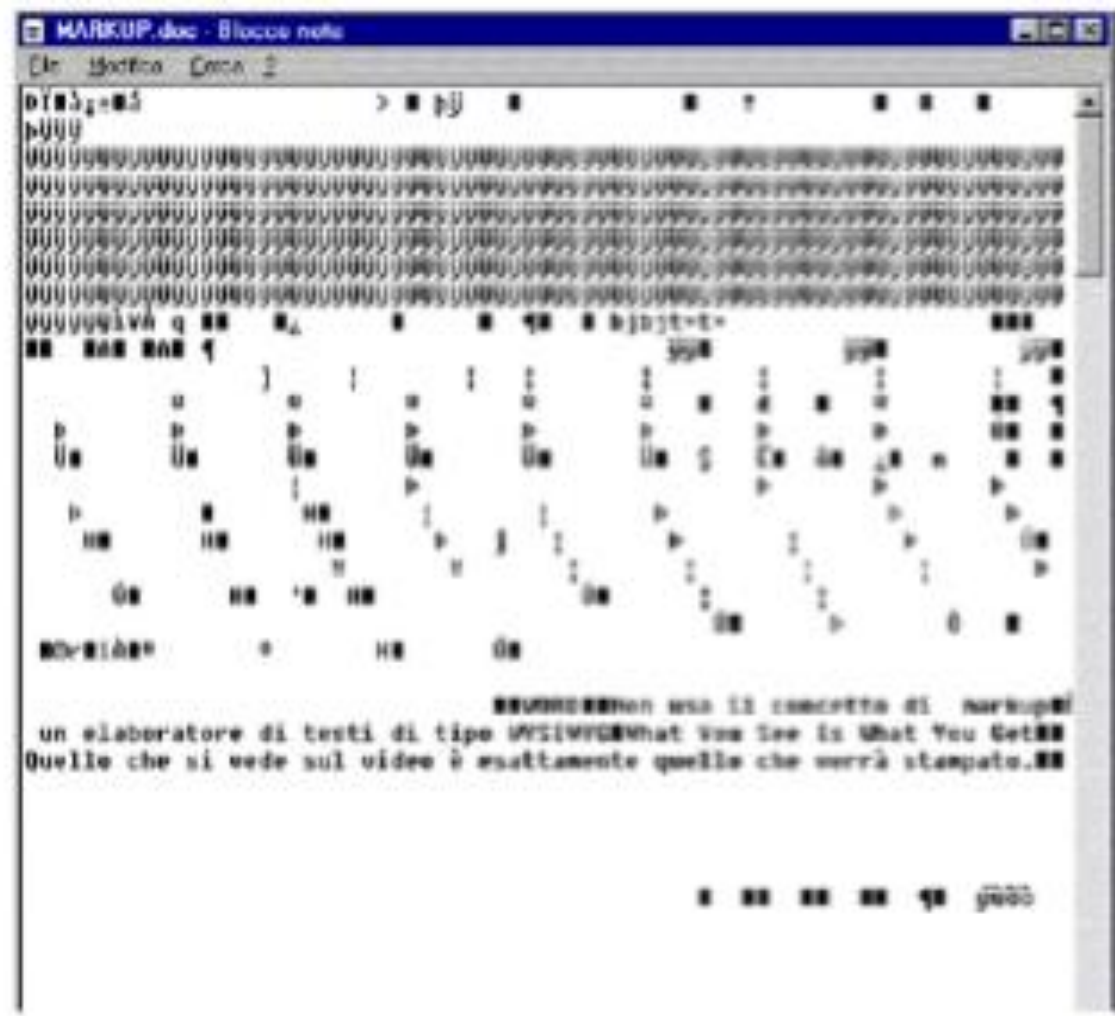
File Word (.doc)



File Word (.doc)



Aprendo il file precedente con Blocco Note si vede il formato interno di Word e non si distinguono markup.



File in formato RTF (.rtf)



RTF è un formato solo testo in cui si distinguono informazioni legate alla presentazione del documento e informazioni che fanno parte del contenuto

```

MARKUP.rtf - Blocco note
File Modifica Cerca ?
Ribaldo Marina}{\operator Ribaldo Marina}
{\creatin\yr2000\mo12\dy18\hr16\min11}{\revtim\yr2000\mo12\dy18\hr16\min11}{\printim\yr2000\mo12\dy18\hr15\min29}{\version2}{\ednins0}
{\nofpages1}{\nofwords26}{\nofchars150}{\*\company Dipartimento di Informatica}{\nofcharsvs184}{\vern113}}
\paperw11906\paperh16838\margl1134\margr1134\margt1417\margb1134
\defstab708\widowctrl\ftnbj\enddoc\hyphhotz283\formshade\viewkind4\viewscale76\viewzk2\pgbrdrhead\pgbrdrfoot \fet0\sectd
\linex0\headery709\footery709\colsx709\endnhere\sectdefaultc1
{\*\pnseclvl1\pnucrn\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl2\pnucltr\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl3\pndec\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl4\pnlcltr\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl5
\pndec\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta
)}}{\*\pnseclvl6\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb
(}{\pntxta
)}}{\*\pnseclvl7\pnlcrn\pnstart1\pnindent720\pnhang{\pntxtb
(}{\pntxta )}}{\*\pnseclvl8\pnlcltr\pnstart1\pnindent720\pnhang
{\pntxtb (}{\pntxta
)}}{\*\pnseclvl9\pnlcrn\pnstart1\pnindent720\pnhang{\pntxtb
(}{\pntxta )}}\pard\plain \qc\widctlpar\adjustright
\fs20\lang1040\cgrid {\fs36
\par
\par }{\b\fs56 WORD
\par }{\b\fs44
\par }\pard \widctlpar\adjustright {\fs44 Non usa il concetto di
markup
\par \c8 un elaboratore di testi di tipo WYSIWYG
\par What You See Is What You Get
\par
\par Quello che si vede sul video \e8 esattamente quello che
verr\ e0 stampato.
    
```


File in formato PostScript (.ps)



Questo è un formato usato per la stampa. Anche in questo caso si distinguono informazioni legate alla presentazione del documento e informazioni che fanno parte del documento stesso.

```

stackup.ps - Blocco note
File Modifica Desc 2
%!-12345XWPJL JOBWPJL SET RESOLUTION = 600WPJL ENTER
LANGUAGE=POSTSCRIPT
%%PS-Adobe-3.0
%%Title: Microsoft Word - MARKUP.rtf
%%Creator: PSCRIPT.DRU versione 4.0
%%CreationDate: 12/18/00 16:12:57
%%BoundingBox: 14 13 581 829
%%Pages: (atend)
%%PageOrder: Special
%%Requirements:
%%documentNeededFonts: (atend)
%%documentSuppliedFonts: (atend)
%%DocumentData: Clean7Bit
%%LanguageLevel: 2
%%endComments

%%BeginProlog
%%BeginProcSet: Pscript_Min_ErrorHandler 1.0 1
/currentpacking where{pop/oldpack currentpacking def/setpacking
where{pop
false setpacking}if}{if/$hekpage 64 dict def $hekpage begin/print{dup
type
/stringtype not{=string cvs}if dup length 6 mul/tx exch def/ty 10
def
currentpoint/toy exch def/tox exch def 1 setgray newpath tox toy 2
sub moveto
0 ty rlineto tx 0 rlineto 0 ty neg rlineto closepath fill tox toy
moveto 0
setgray show}bind def/ni{currentpoint exch pop lmargin exch moveto
0 -10
rmoveto}def/=={/cp 0 def typeprint nl}def/typeprint{dup type

```

HTML



- Gli ipertesti del web sono scritti usando il linguaggio **HyperText Markup Language (HTML)**.
- HTML **non è** un linguaggio di programmazione!
 - HTML non ha meccanismi per prendere delle decisioni o compiere delle iterazioni.
- HTML **è un linguaggio di markup** e permette di descrivere la disposizione di tutti gli elementi presenti all'interno di un documento.
 - Le indicazioni vengono date attraverso degli appositi marcatori, detti "tag".
- I documenti HTML sono dei file in **formato testo** (codice ASCII).
- Si possono creare con degli editor di testo dando l'estensione **.htm** o **.html**.
- I browser leggono i documenti HTML e li visualizzano **interpretando** le specifiche di formattazione in modo sequenziale.

Standard (X)HTML



L'organizzazione che si occupa di standardizzare la sintassi del linguaggio HTML (il W3C: World Wide Web Consortium) ha rilasciato diverse versioni di questo linguaggio (HTML 2.0, HTML 3.2, HTML 4.0); e - da un certo punto in poi - l'HTML si è evoluto in XHTML (si tratta dell'HTML riformulato come linguaggio XML - ne sono già state rilasciate due versioni).

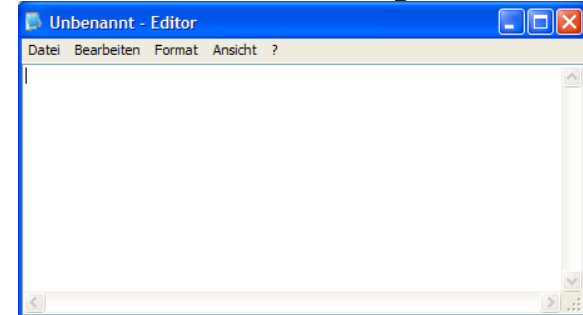
```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2  <html>
3      <head>
4          <title>Example</title>
5          <link href="screen.css" rel="sty
6      </head>
7      <body>
8          <h1>
9              <a href="/">Header</a>
10         </h1>
11         <ul id="nav">
12             <li>
13                 <a href="one/">One</a>
14             </li>
15             <li>
16                 <a href="two/">Two</a>
17             </li>

```

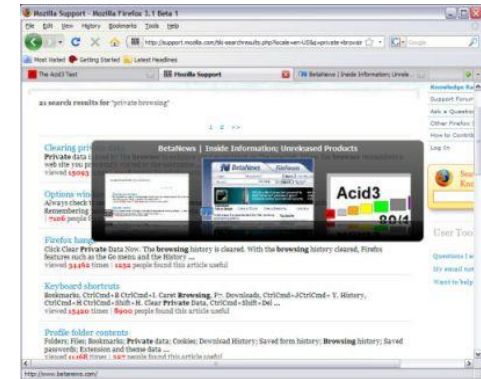


Di cosa abbiamo bisogno



✓ Un editor di testo semplice (Notepad – Blocco note – Wordpad).

✓ Un browser internet (Firefox, Explorer, Safari).



✓ Uno spazio su un server per rendere pubbliche le nostre pagine (opzionale).



HTML: un esempio



```
<html>
```

```
  <head>
```

```
    <title>Pagina HTML di prova</title>
```

```
  </head>
```

```
  <body>
```

Utilizzare un editor di testo: in ambiente Windows WordPad va bene, in ambiente Unix vi per esempio

```
  </body>
```

```
</html>
```

HTML: sintassi (1/2)



```
<nomecomando>
    informazioni
</nomecomando>
```

- Un documento HTML inizia sempre con il tag `<html>` e termina sempre con il tag `</html>`.
- Il linguaggio HTML è **case-insensitive** e quindi `<HTML>`, `<html>`, `<Html>` sono tutti tag leciti (lo stesso vale per gli altri tag).

Suggerimenti:

- e meglio decidere all'inizio come si scriveranno i tag ed essere consistenti;
- per analogia con XHTML (derivato dell'XML e dell'HTML che è case-sensitive) è consigliabile scrivere tutto in minuscolo.

HTML: sintassi (2/2)



<html>

<head>

caratteristiche del documento

</head>

<body>

documento

</body>

</html>

HTML: <head>



<html>

<head>

<title>titolo del documento</title>

<meta name="keywords" content="parole chiave
qui">

<meta name="author" content="nome e cognome
qui">

</head>

.....

.....

</html>

HTML: <head>



- Questa sezione del documento contiene informazioni non immediatamente percepibili, ma che riguardano il modo in cui il documento deve essere letto ed interpretato.
- È la parte del documento dove vengono inseriti i metatag (alcuni sono ad esclusivo beneficio dei motori di ricerca), script JavaScript o VbScript, i fogli di stile, ecc.

HTML: <body>



<html>

<head>

<title>titolo del documento</title>

</head>

<body lista di opzioni>

documento vero e proprio

</body>

</html>

HTML: <body>



<body

 bgcolor="colore sfondo"

 background="nome file con l'immagine"

 text="colore testo"

 link="colore link da visitare"

 vlink="colore link visitato"

 alink="colore link selezionato"

>

NB: l'ordine in cui si scrivono gli attributi non è importante.

HTML: <body>



bgcolor="red"

bgcolor="yellow"

bgcolor="#00ffff"

text="#eeeeee"

link="#ffffff"

background="images/sfondo.gif"

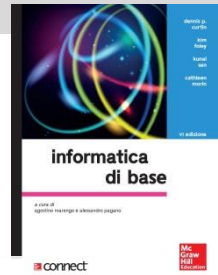
<body text= "white" link="#cc0000" background="images/sabbia.gif">

HTML: RGB (1/2)



- Ogni colore può essere codificato mediante tre numeri compresi tra 0 e 255 che rappresentano le quantità di ROSSO VERDE e BLU presenti nel colore stesso.
- Ogni numero compreso tra 0 e 255 deve essere trasformato nella rappresentazione esadecimale corrispondente (base 16).
- Con questa codifica si possono rappresentare più di 16,7 milioni di colori diversi.

HTML: RGB (2/2)



	decimale	#esadecimale
nero	0 0 0	#000000
blu	0 0 255	#0000ff
verde	0 255 0	#00ff00
rosso	255 0 0	#ff0000
bianco	255 255 255	#ffffff

NB: nei file HTML le codifiche esadecimali vanno scritte senza lasciare spazi vuoti tra le cifre.



- Il documento inserito nel `<body>` viene visualizzato secondo le direttive di formattazione.

Per scrivere **titoli** si possono usare:

```
<h1> </h1>  
<h2> </h2>  
  
<h6> </h6>
```

Per andare **a capo**:

```
<br>  
  
<p>.....</p>
```



Formato dei caratteri:

- `` grassetto ``
- `` grassetto ``
- `<i>` corsivo `</i>`
- `` enfatizzato ``
- `<code>` codice `</code>`
- `` testo ``
- `` ... ``
- `` ... ``



- Allineamento:
 - `<p align="center">`
 - `<p align="right">`
 - `<p align="justify">`
 - `<center>` testo da centrare `</center>`

HTML: elenchi



- ** e (lista non ordinata)**

```
<ul>
```

```
    <li>primo elemento</li>
```

```
    <li>secondo elemento</li>
```

```
</ul>
```

- ** e (lista ordinata)**

```
<ol>
```

```
    <li>primo elemento</li>
```

```
    <li>secondo elemento</li>
```

```
</ol>
```

HTML: altri marcatori



- Linee orizzontali `<hr>`:
 - `<hr width="90%" size=3>`
- Testo lampeggiante (solo per Netscape):
 - `<blink>` questo testo lampeggia `</blink>`
- Testo scorrevole (solo per Explorer):
 - `<marquee>` questo testo scorre `</marquee>`
- Commenti:
 - `<!--` questo testo non viene visualizzato `-->`

HTML: <a href>

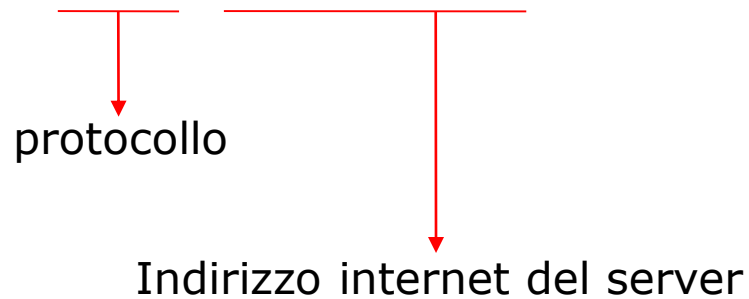


- Link <a>
 - testo del link
- Gli indirizzi per “trovare” i documenti nel web sono detti **Uniform Resource Locator (URL)** e li identificano in modo univoco.
- Protocollo: //indirizzo internet del server/pathname

HTML: <a href>



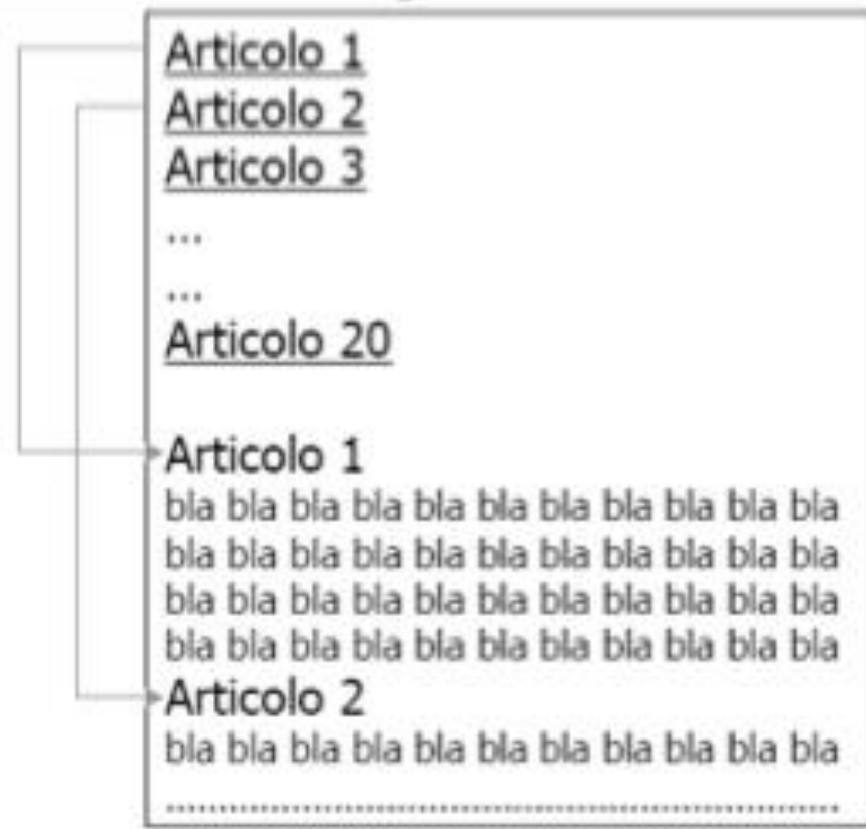
Università di Bari



HTML: link interni (ancore) (1/3)



- Link interni allo stesso documento: si usa l'attributo name del tag <a>





HTML: link interni (ancore) (2/3)

- Si devono marcare quelle porzioni del documento dove si vuole saltare.

```
Articolo 1  
Articolo 2  
Articolo 3  
...  
...  
Articolo 20  
  
Articolo 1 <a name="art1"></a>  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
Articolo 2 <a name="art2"></a>  
bla bla bla bla bla bla bla bla bla bla  
.....
```

HTML: link interni (ancore) (3/3)



- Per i link interni si deve usare nell'attributo href il nome scelto come valore dell'attributo name, preceduto dal simbolo #.

```

Articolo 1 <a href="#art1">Articolo1 </a>
Articolo 2 <a href="#art2">Articolo2 </a>
Articolo 3 <a href="#art3">Articolo3 </a>
...
...
Articolo 20

Articolo 1 <a name="art1"></a>
bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla
Articolo 2 <a name="art2"></a>
bla bla bla bla bla bla bla bla bla bla
.....
    
```


HTML:



NB: l'immagine deve essere su un file a parte in formato GIF, PNG o JPEG (BMP solo per Explorer).

HTML: audio



- Formati audio .au .aiff .mid .wav
 - `clicca qui`
 - `<bgsound src="nome file audio" loop="10">`

NB: bgsound fa partire l'audio quando si entra nella pagina ma funziona solo in Explorer.



- Formati video .mpeg .avi .mov
 - clicca qui
 -

NB: dynsrc fa partire automaticamente il video ma funziona solo in Explorer.

HTML: tabelle



- Per creare una tabella in HTML si usano i tag:

<table> **</table>**

all'interno dei quali si deve definire il contenuto della tabella

- Per definire una tabella si procede nel modo seguente:
 - Si realizza la prima riga facendo uso del tag **<tr>** (table row)
 - Si indicano le varie celle facendo uso:
 - Del tag **<th>** per le intestazioni di cella **</th>** (table header)
 - Del tag **<td>** per i dati della cella **</td>** (table data)
 - Si passa alla seconda riga con le relative celle avendo cura di usare il tag **</tr>**

HTML: tabella



HTML: <table>



<table

width="larghezza della tabella"

cellpadding="numero"

indica la distanza tra il contenuto della cella e il bordo. Se il valore viene indicato con un numero intero, la distanza è espressa in pixel; il cellpadding tuttavia può anche essere espresso in percentuale.

Di default la distanza è nulla.

cellspacing="numero"

specifica la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo. Di default è un pixel, dunque occorrerà sempre azzerarlo esplicitamente, quando non lo si desidera.

border="numero"

specifica la larghezza dei bordi di una tabella (in pixel).

bgcolor="colore sfondo tabella"

Esempio:

```
<table width="80%" cellpadding="5" bgcolor="#ff3300">
```

HTML: <td> <th>



<td (th)

width="dimensione della cella"

colspan="numero"

rowspan="numero"

bgcolor="colore sfondo cella"

align="right" | "left" | "center"

valign="top" | "bottom" | "middle"

nowrap

>

Esempio:

- `<td width="100" bgcolor="#0033aa" align="right">`

HTML: tabelle



- Le tabelle si usano principalmente per organizzare gli elementi all'interno di una pagina.
- Per fare questo non si visualizzano i bordi

```
<table border="0">
```

righe e colonne

```
<\table>
```