

Metalinguaggi di marcatura – 2006/2007

Docente: Andrea Bertagnoli

Email: andrea.bertagnoli@uniroma2.it

Sito Internet: www.uniroma2.it/didattica/metalinguaggi

Indice

1. INTRODUZIONE.....	3
1.1 SGML - STANDARD GENERALIZED MARKUP LANGUAGE	3
2. HTML - HYPERTEXT MARKUP LANGUAGE	4
2.1 SCRITTURA DEL CODICE HTML	8
2.2 ELEMENTI E TAG	9
2.3 ELEMENTI E ATTRIBUTI	12
2.4 STRUTTURA DI UN DOCUMENTO HTML	14
2.5 LA SEZIONE HEAD	16
2.5.1 Titolo del documento.....	16
2.6 COMMENTI.....	17
2.7 ELEMENTO H	18
2.8 PARAGRAFI	18
2.9 LINEE ORIZZONTALI	19
2.10 LINEE DI INTERRUZIONE.....	21
2.11 FORMATTAZIONE DEI CARATTERI.....	26
Famiglie di caratteri	30
Colori	31
2.12 TAG DI STILE FISICI E LOGICI	34
2.13 ELENCHI.....	37
Esercizio con variazione di attributi.....	39
2.14 TABELLE	41
2.15 SPAZI	ERRORE. IL SEGNA LIBRO NON È DEFINITO.
2.16 CARATTERI SPECIALI.....	46
2.16 IMMAGINI.....	47

1. Introduzione

Fino a non molti anni fa, quando si passava un testo ad un tipografo per la composizione, si scrivevano a margine delle annotazioni per indicare come si voleva che il testo fosse composto: in che modo doveva essere giustificato, con che carattere, corpo, se in tondo, corsivo, grassetto eccetera. Un redattore italiano chiamerebbe questo lavoro caratterizzazione; gli americani usano i termini markup, e tags per le annotazioni riportate a margine.

Questo approccio è stato utilizzato per realizzare un linguaggio (o più correttamente un metalinguaggio) standard da utilizzare all'interno di un sistema informativo elettronico per descrivere la struttura logica di un qualsiasi documento.

1.1 SGML - *Standard Generalized Markup Language*

SGML è un codice che serve a descrivere altri codici: in questo senso è un metalinguaggio, cioè un linguaggio che parla di un altro linguaggio. Niente di strano: è come quando si parla (usando il codice del linguaggio verbale) di musica (che è a sua volta un linguaggio codificato). SGML non fa altro che stabilire quali annotazioni (markup) si possono mettere in un linguaggio di markup, quali devono esserci necessariamente, e come si fa a distinguere le annotazioni (markup) dal testo vero e proprio.

È un'idea di grande semplicità, tenere separato lo stile di presentazione dalla struttura logica del documento, eppure è proprio questa semplice trovata che consente molteplici utilizzazioni della stessa informazione: con la marcatura descrittiva è possibile utilizzare lo stesso documento per la stampa, oppure per la ricezione in linea, per la pubblicazione su cd-rom oppure per il riuso del documento integrandovi altro testo. L'organizzazione di un documento non è espressa usando la codifica dei sistemi di scrittura, che è finalizzata alla presentazione grafica, ma sono evidenziate le parti in cui è strutturato il documento (ad esempio paragrafi, capitoli) insieme ad altre particolarità del testo (come note, tabelle, intestazioni). La marcatura descrive la struttura logica di un documento usando un modello gerarchico che riflette l'interrelazione nel testo di elementi ed entità.

Esempio:

```
<antologia>
<poesia><titolo>San Martino</titolo>
<strofa>
<verso>La nebbia a gl'irti colli</verso>
<verso>Piovigginando sale,</verso>
<verso>E sotto il maestrale</verso>
<verso>Urla e biancheggia il mar;</verso>
</strofa>
...
</poesia>
<!-- seguono altre poesie -->
</antologia>
```

Per completezza va detto che un documento SGML è sempre accompagnato da un DTD (Document Type Definition) nel quale vengono aggiunte ulteriori specifiche sulle marcature presenti nel documento.

2. HTML - Hypertext Markup Language

HTML, acronimo di *HyperText Markup Language*, è il linguaggio standard per descrivere e definire il contenuto e l'aspetto delle pagine (risorse) sul World Wide Web. Può essere definito anche come un metodo di descrizione del formato delle pagine.

HTML non è un linguaggio di *programmazione* né un linguaggio di *descrizione dei dati* (come invece è XML), ma è un linguaggio di *markup* (marcatura o contrassegnazione). Ciò significa che tramite HTML si contrassegnano specifiche parti di testo, seguendo la sintassi prevista dal linguaggio, attribuendo così a tali parti un determinato significato. La parola hypertext dell'acronimo si riferisce alla caratteristica del linguaggio di poter creare in determinati punti del testo collegamenti (hyperlink)

ad altre risorse (altre parti dello stesso testo, altre pagine, altri siti, ecc.). Possono inoltre contenere testi, immagini, suoni, filmati, collegamenti (hyperlink ad altre pagine o fonti informative) che danno luogo agli *ipertesti*, nei quali si possono includere più fonti di informazione in un unico schema documentale. Le pagine HTML contengono comandi incorporati che forniscono informazioni sulla struttura, l'aspetto e il contenuto della pagina stessa, denominati *tag*.

Esistono dei programmi, i cosiddetti browser, che interpretano il linguaggio HTML cioè utilizzano le informazioni fornite dai tag per determinare come visualizzare la pagina. In sintesi, ciò è analogo a quanto avviene con gli elaboratori di testo nei quali, tramite gli opportuni comandi, si applica al testo la *formattazione*, come il tipo e la dimensione con cui rappresentare i caratteri, l'allineamento del testo, la spaziatura tra le righe, ed altro ancora, determinando in tal modo l'applicazione di uno *stile*. Nei programmi di elaborazione di testi, però, la formattazione avviene tramite istruzioni di marcatura proprie dei differenti programmi (si dice che si tratta di istruzioni di tipo *proprietario*). Di conseguenza si viene a determinare una stretta dipendenza tra programmi e documenti, rendendo necessaria una conversione di formato per lavorare sugli stessi documenti con differenti programmi applicativi.

HTML, invece, assicura l'indipendenza della rappresentazione dei dati da tecnologie proprietarie. Come abbiamo già detto sopra la formattazione del testo, in HTML, si realizza mediante l'utilizzo dei *tag* e solitamente vengono indicati con il nome di un formato incluso tra parentesi angolari.

Esempio

`<b>` è un tag che indica di formattare il testo in grassetto. (b sta per bold)

Di seguito si riportano due righe di codice HTML che si riferiscono, rispettivamente, a una stringa di testo in carattere normale e ad un'altra in carattere grassetto.

```
stringa in carattere normale
<b>stringa in carattere grassetto</b>
```

Visualizzando questo codice mediante un browser si ottiene la schermata di Fig. 1.

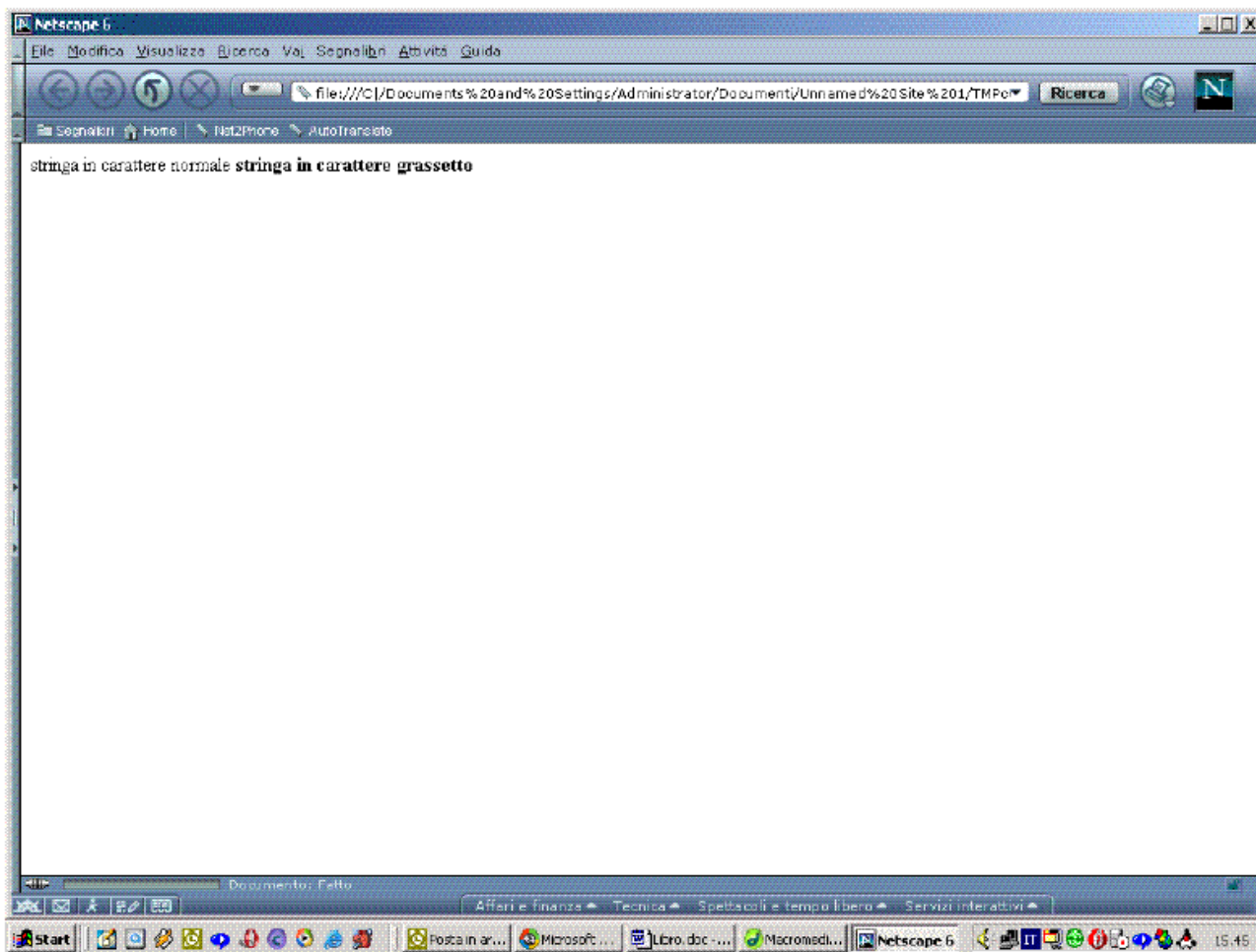


Fig. 1: Stringhe in carattere normale e in carattere grassetto

In questa semplicissima pagina Web si nota che la prima stringa di caratteri è scritta in modalità normale mentre la seconda, che è la “zona” influenzata dall’effetto del tag `<b>`, è invece mostrata in modalità grassetto. Si osservi, intanto, che mentre il codice è scritto su due differenti righe di testo la rappresentazione a video del codice avviene su un’unica riga. Ciò è del tutto normale ed è una delle caratteristiche del linguaggio HTML che verranno successivamente espone.

I tag HTML non sono sensibili alle differenze tra caratteri maiuscoli e minuscoli per cui, ad esempio, scrivere `<b>` è del tutto equivalente allo scrivere `<B>`.⁶ Va comunque tenuto presente che il World Wide Web Consortium (W3C, www.w3c.org), il consorzio internazionale composto da oltre 500 enti ed aziende, e preposto alla regolamentazione delle tecnologie Web, nelle proprie raccomandazioni HTML 4 *suggerisce* l’uso di tag scritti in caratteri minuscoli mentre XHTML (la successiva

generazione HTML) *richiede* necessariamente che i tag siano in caratteri minuscoli. In gergo tecnico si dice che i tag di HTML sono “case-insensitive” mentre quelli di XHTML sono “case-sensitive”.

E’ buona norma, quindi, abituarsi il prima possibile a scrivere i tag con caratteri minuscoli. Gli standard HTML sono creati dal W3C, e sono stati aggiornati più volte. Tuttavia le diverse versioni del linguaggio sono state implementate in modo differente dai diversi browser. Inoltre alcuni produttori hanno introdotto caratteristiche valide solo per i browser di propria produzione (caratteristiche *proprietarie*).

Come abbiamo appena visto nell’esempio precedente, la formattazione del testo è eseguita dai browser solo attraverso la lettura e l’interpretazione dei tag presenti nel documento HTML. Per questo motivo gli spazi o le righe in bianco eventualmente contenute in un testo HTML non vengono rappresentate nel modo in cui appaiono nel codice HTML. Sia le linee lasciate in bianco sia gli spazi nel documento HTML vengono rappresentati e visualizzati come un unico spazio. Per visualizzare il testo con una determinata formattazione, quindi, non vanno mai utilizzati spazi e righe in bianco ma si deve ricorrere all’utilizzo di particolari tag che consentono la visualizzazione del testo esattamente come è rappresentato nel codice. La caratteristica appena discussa consente di scrivere il codice HTML aggiungendo liberamente spazi in bianco e indentazioni nel codice, rendendolo più facilmente leggibile senza alterarne il significato. A ciò va comunque aggiunto che, per diversi motivi, non è possibile sapere in anticipo come verrà esattamente visualizzato un testo in un browser. I principali motivi di questa apparente anomalia derivano dalla differente grandezza degli schermi usati dagli utenti, dalla diversa dimensione che l’utente può impostare per la dimensione dei caratteri e dal fatto che, di norma, l’utente può ridimensionare liberamente la pagina Web che sta visualizzando trascinando con il mouse i bordi della finestra.

Per quanto riguarda i browser, possono essere suddivisi in due principali categorie: browser grafici e browser testuali. I browser grafici sono in grado, in particolare, di visualizzare immagini e di riprodurre suoni e rappresentano quindi strumenti di tipo multimediale. I browser testuali, invece, non riproducono immagini e suoni, e per questo motivo sono generalmente più veloci di quelli grafici. In proposito si rammenta che è opportuno che tutti i documenti HTML siano verificati dagli autori o dal team di collaudo sia in modalità grafica sia in modalità testo. Oltre alle due categorie menzionate si stanno diffondendo recentemente anche browser per non vedenti, dotati della capacità di interpretare le istruzioni HTML e di convertirle in impulsi vocali e/o in codice Braille.

2.1 Scrittura del codice HTML

Il codice HTML può essere scritto usando semplici programmi di editazione di testo come il blocco note di Windows o Wordpad. Esistono, comunque, strumenti di creazione di codice HTML estremamente sofisticati, come Dreamweaver di Macromedia e FrontPage di Microsoft, in grado, tra l'altro, di generare automaticamente il testo HTML partendo dal disegno a video della pagina che l'autore ha eseguito. In questo modo il processo di creazione delle pagine risulta drasticamente abbreviato. Tuttavia nel momento in cui si inizia ad apprendere il linguaggio HTML l'uso di tali sofisticati strumenti, per diversi motivi, non è consigliabile. Tra questi motivi rientra la semplicità degli esempi didattici normalmente usati all'inizio del percorso di apprendimento e che non richiede il ricorso a tali strumenti. Inoltre è opportuno focalizzare la propria attenzione sulle caratteristiche e sul funzionamento dei diversi comandi previsti dal linguaggio, anche apportando deliberatamente variazioni al codice per verificarne l'effetto all'atto della visualizzazione. Inoltre gli strumenti automatici aggiungono solitamente righe e parti di codice supplementari rispetto a quanto strettamente necessario per il corretto funzionamento degli esempi, rendendoli, come si dice, più “pesanti” e aumentando la difficoltà di lettura del codice da parte dell'autore. Cercare di apprendere i principi della scrittura dei documenti HTML partendo direttamente dagli strumenti di generazione automatica può essere dunque paragonabile all'utilizzo di traduttori automatici per una lingua straniera: non conoscendo ancora la nuova lingua non si è in grado di comprendere la versione tradotta e, di conseguenza, non è possibile riconoscere e correggere gli eventuali errori di traduzione commessi dallo strumento automatico. Infine, oltre ai possibili errori che gli strumenti automatici potrebbero generare, potrebbero rendersi maggiormente necessarie modifiche e personalizzazioni alle pagine costruite automaticamente. Tra i semplici editor di testo, come il blocco note di Windows, e gli strumenti automatici per la generazione e la gestione di siti web, come Dreamweaver di Macromedia e Frontpage di Microsoft, si collocano altri strumenti in grado di leggere il codice e di verificarne la sintassi. Un esempio è dato dal programma di utilità Tidy, disponibile gratuitamente sul sito del W3C. Un'altra categoria di strumenti per la scrittura automatica di codice HTML è data dai cosiddetti *convertitori*. Si tratta di funzionalità software, incluse nei programmi di videoscrittura, che consentono il salvataggio di file di testo in formato HTML, in genere mediante l'opzione “Salva come HTML”. Anche i convertitori, però, aggiungono solitamente tag non necessari e generano un codice molto più complesso rispetto a quello minimo necessario per ottenere l'effetto desiderato.

Un eccellente modo di iniziare a scrivere codice HTML consiste nell'esaminare ciò che è già stato scritto. HTML consente l'esame del codice di qualsiasi pagina visualizzata nel browser e questo costituisce una fonte preziosa per l'apprendimento, che si aggiunge alle risorse presenti su Internet e destinate specificatamente alla didattica.

2.2 Elementi e tag

Gli elementi e i tag costituiscono concetti fondamentali di HTML. Il codice HTML si basa su coppie di *tag* di apertura e di chiusura che definiscono ogni elemento di una pagina Web, come ad esempio un paragrafo di testo, una tabella o un'immagine. Il linguaggio HTML contrassegna quindi i componenti di una pagina Web con un insieme di *elementi* predefiniti. Gli elementi sono costituiti da intestazioni, paragrafi, elenchi, tabelle, immagini e collegamenti. Ogni elemento è costituito da un *tag* iniziale formato da una parte di testo preceduto da una parentesi angolare sinistra (<) e seguita da una parentesi angolare destra (>). Il tag finale (in qualche caso non è obbligatorio) è simile al tag iniziale con la differenza che include una barra rovesciata (/). Il testo compreso tra il tag iniziale e quello finale è il *contenuto*. L'elemento può pertanto essere definito come la sequenza di tag di apertura, contenuto e tag di chiusura. La figura 2 riassume i componenti sinora esaminati.

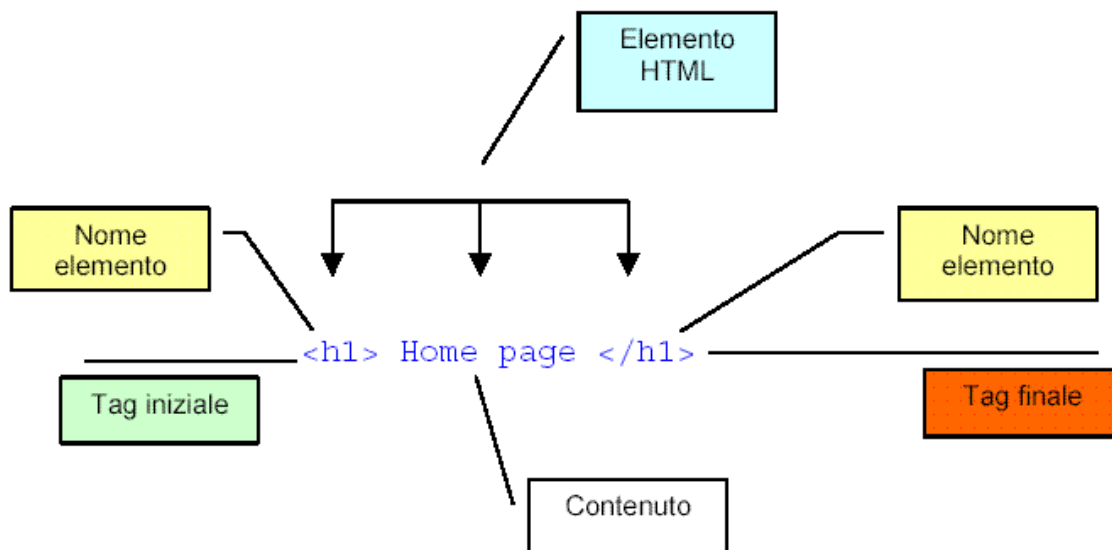


Fig. 2: Elementi e tag

Nella figura 2 è rappresentata una semplice riga di codice HTML. La pagina visualizzata tramite il browser presenterà la scritta “Home page” (contenuto dell’elemento) con la formattazione prevista per il tag *h1* (elemento). Quando il browser legge questa riga di istruzione il tag iniziale “attiva” l’elemento al quale si riferisce (nell’esempio: *h1*, ovvero intestazione di primo livello) e il tag finale “disattiva” l’elemento. In altri termini il contenuto dell’elemento, costituito dal testo compreso tra il tag iniziale e il tag finale, è reso sulla pagina Web con la modalità che il browser prevede per il tag stesso.

Nella figura 3 è mostrato l’effetto che questa semplice istruzione produce quando è visualizzata attraverso un browser.

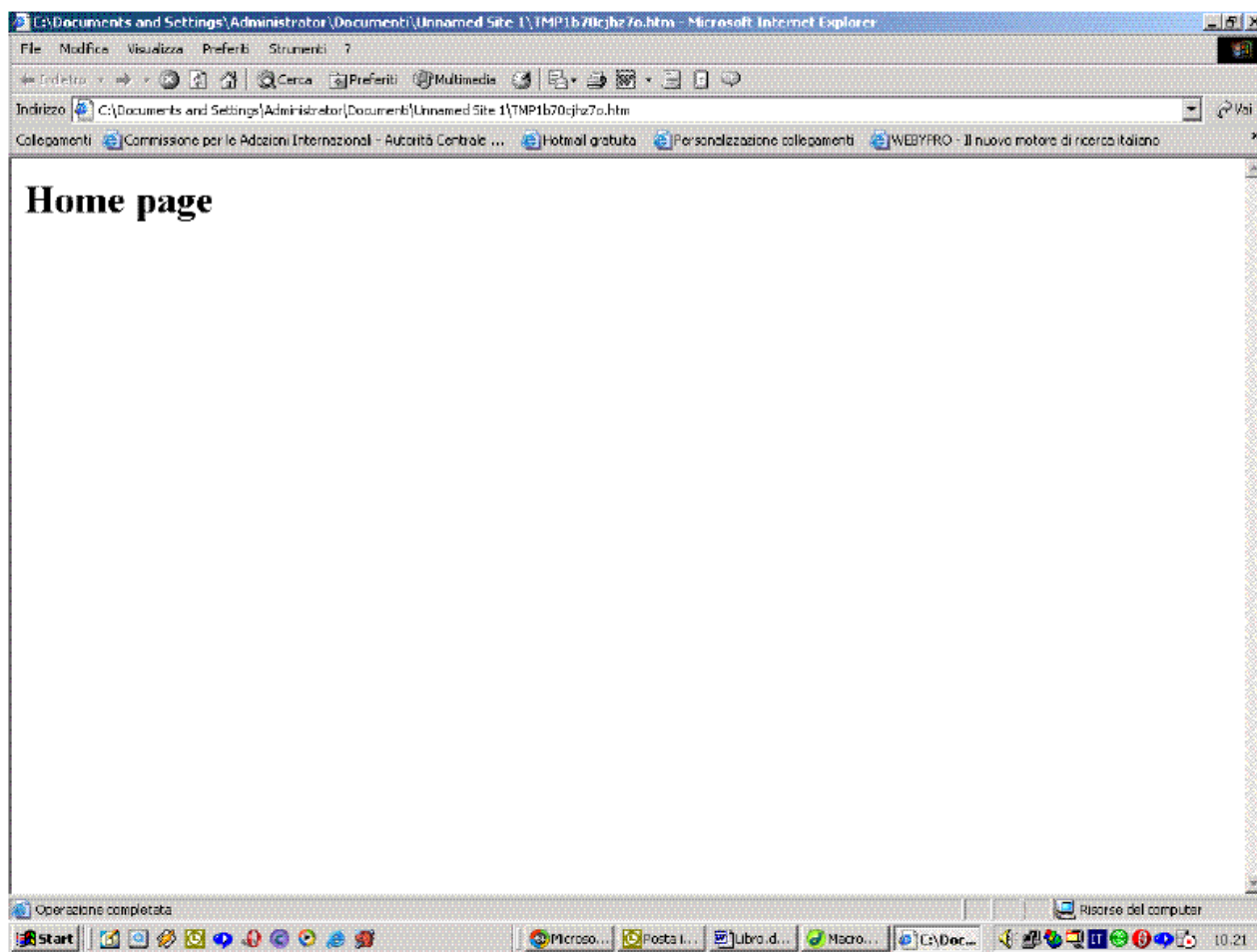


Fig. 3: Un'intestazione di tipo h1

Non tutti gli elementi prevedono un tag di chiusura e che, in tal caso, sono definiti elementi vuoti. Quando è presente la coppia di tag di apertura e chiusura i tag stessi sono anche definiti bilaterali, mentre quando è presente il solo tag di apertura il tag è anche detto monolaterale.

In proposito è bene precisare che il linguaggio HTML prevede due categorie di tag: tag *vuoti* e tag *contenitori*. I tag vuoti hanno lo scopo di formattare la pagina, non hanno testo al proprio interno e, come già affermato, non prevedono il tag di chiusura, mentre possono essere corredati da attributi (gli attributi verranno descritti nei paragrafi successivi). Un esempio di tag vuoto è dato dal tag `<img>` (immagine) che, richiamando un file di tipo immagine, fa apparire l'immagine stessa nella pagina Web, nella posizione e con le modalità indicate dal tag. I tag contenitori sono invece costituiti da un tag iniziale, con la presenza di eventuali attributi, da un testo e da un tag finale che ha lo scopo di segnalare il punto in cui deve terminare l'effetto previsto dal tag.

Un esempio di tag finale è dato da `</html>` che rappresenta il tag con il quale deve terminare ogni documento HTML.

Si osservi che la combinazione di più tag deve essere scritta annidandoli e non sovrapponendoli ovvero “chiudendo” prima quelli più interni e via via quelli più esterni.

```
<h1><body></h1></body>
```



Fig. 4: Esempio di tag sovrapposti (errato)

```
<h1><body></body></h1>
```



Fig. 5: Esempio di tag annidati (corretto)

Dalle figure 6 e 7 si vede che i tag devono essere chiusi in ordine inverso a quello con il quale sono stati aperti.

Sebbene i browser siano generalmente “tolleranti” nei confronti di errori nella codifica del linguaggio HTML e che pertanto siano in grado di rappresentare correttamente anche pagine contenenti combinazioni di tag non rispondenti alla regola appena indicata è buona norma scrivere le aperture e le chiusure dei tag in base alla regola indicata. Infatti i futuri sviluppi delle specifiche del linguaggio prevedono che la codifica avvenga secondo questa regola.

Nella tabella successiva si riportano alcuni elementi e la parte di pagina che ognuno di essi contrassegna.

<i>Elemento HTML</i>	<i>Descrizione</i>
Html	tag che segna l'inizio di un documento html
Head	informazioni relative all'intestazione
Title	titolo (appare nella finestra del browser)
Body	corpo principale del testo
h1	intestazione di primo livello
h2	intestazione di secondo livello
h3	intestazione di terzo livello
P	paragrafo
img	immagine
A	ancoraggio per il collegamento ad altra pagina o parte di pagina
B	blocco di testo in grassetto
I	blocco di testo in corsivo
ol	elenco ordinato (ordered list)
Li	singolo elemento di un elenco
table	tabella

In seguito questi ed altri elementi verranno discussi ed usati in diversi esempi. Il browser è in grado di formattare e visualizzare gli elementi, secondo le caratteristiche previste per ciascuno di essi.

2.3 Elementi e attributi

Abbiamo visto che gli elementi identificano parti specifiche di un testo HTML ed assumono un determinato significato. Gli elementi HTML possono essere (e in genere sono) accompagnati da attributi, che rappresentano un modo per assegnare proprietà aggiuntive ed aggiungere informazioni supplementari all'elemento cui si riferiscono. In termini non rigorosi si può affermare che un tag dice al browser di *fare* qualcosa, un attributo del tag dice al browser *come* farlo. Ad esempio il tag <body> può essere accompagnato da un attributo *bghcolor* (colore di background – sfondo) che comunica al browser il colore di sfondo con il quale la pagina deve essere visualizzata. Gli attributi si presentano

come una coppia costituita dal nome dell'attributo e dal suo valore, uniti da un segno di uguale, e vengono aggiunti al tag iniziale di un elemento HTML. E' possibile, per uno stesso tag, definire più di un attributo ed in questo caso gli attributi sono separati l'uno dall'altro con uno spazio. L'ordine con il quale si specificano i diversi attributi non è rilevante. Ci si potrebbe chiedere, a questo punto, quali sono gli attributi che è possibile specificare per ciascun elemento. La risposta è che ogni elemento possiede una lista di attributi ammissibili e di relativi valori, definiti nelle specifiche HTML. I valori degli attributi possono essere racchiusi indifferentemente tra apici o tra doppi apici. Se, ad esempio, il valore dell'attributo contiene apici al suo interno per delimitare tale valore devono essere usati i doppi apici, e viceversa.

I nomi degli attributi sono *case-insensitive* ma spesso i loro valori sono invece *case-sensitive*.

Nella figura 6 è illustrato un esempio di sintassi per l'utilizzo di un attributo.

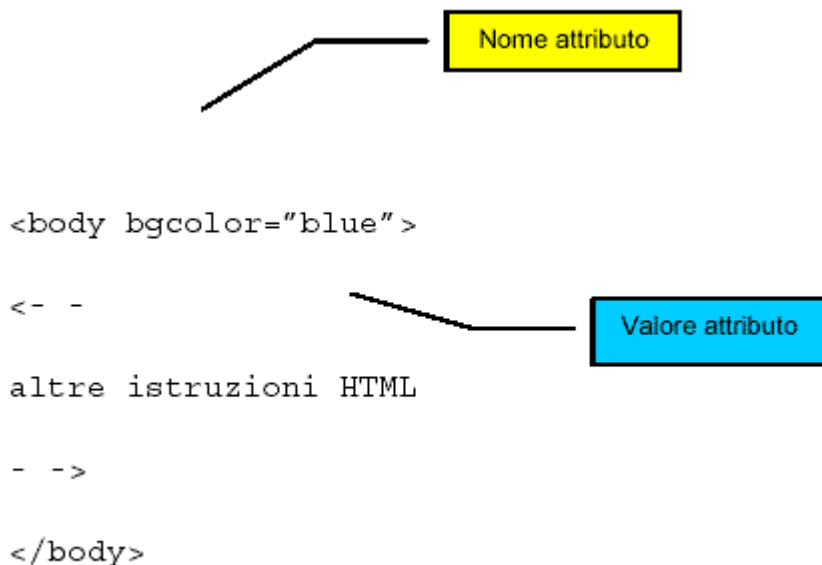


Fig. 6: La valorizzazione dell'attributo bgcolor

Nell'esempio appena mostrato è stato aggiunto al tag iniziale `body` l'attributo `bgcolor` seguito dal valore `blue` che è uno dei possibili valori previsti per l'attributo stesso. In questo modo il colore di sfondo della sezione `body` della pagina visualizzata avrà il colore blu, che è stato indicato quale valore dell'attributo.

2.4 Struttura di un documento HTML

Tutti i documenti HTML hanno la stessa struttura di base, come riportato di seguito:

- intestazione: contiene informazioni di controllo usate dal browser e dal server;
- body;
- contenuto;
- tag che controllano la formattazione del contenuto.

Esempio

```
<html>
  <head>
    <title>Primo documento HTML </title>
  </head>
  <body>
    <h1> Esempio intestazione di primo livello</h1>
    <h2> Esempio intestazione di secondo livello</h2>
    <p>Esempio di un primo paragrafo</p>
    <p>Esempio di un secondo paragrafo</p>
  </body>
</html>
```

Visualizzando il codice attraverso un browser si ottiene il risultato presentato nella figura 7.

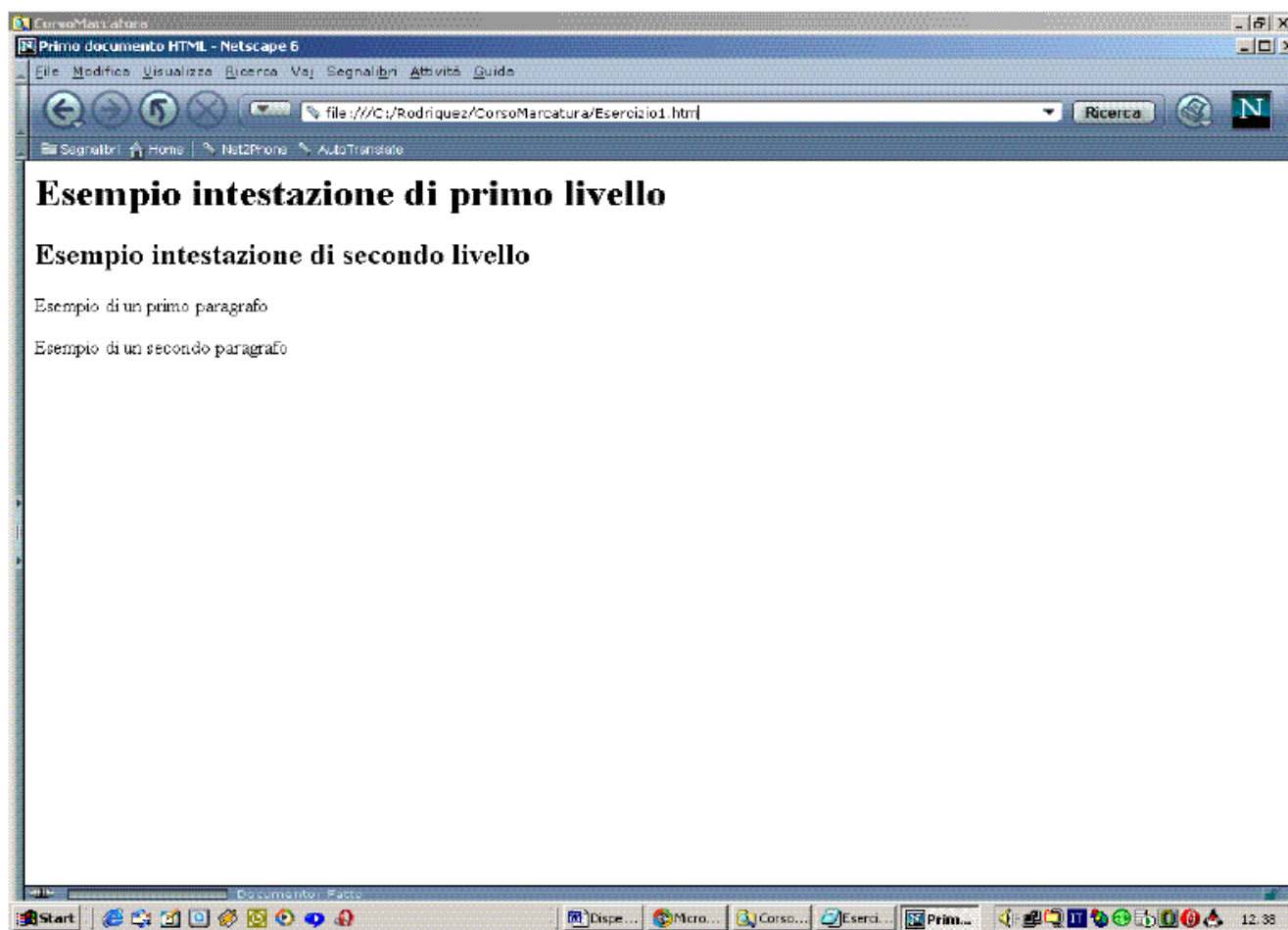


Fig. 7: Esempi di intestazioni e paragrafi

Esaminando in dettaglio il codice esemplificativo appena illustrato osserviamo che:

- i tag iniziale `<html>` e finale `</html>` comunicano al browser, rispettivamente, l'inizio e la fine del documento HTML;
- la porzione di testo compresa tra il tag iniziale `<head>` ed il tag finale `</head>` costituisce la sezione di intestazione, e non viene visualizzata nella finestra del browser;
- il contenuto del tag `title` appare sulla riga del titolo della finestra del browser;
- la porzione di testo compresa tra il tag iniziale `<body>` ed il tag finale `</body>` costituisce la sezione che viene visualizzata nella finestra del browser;
- il contenuto dei tag `h1` e `h2`, che rappresentano righe di intestazione appare, rispettivamente, con enfasi maggiore e minore;
- il contenuto dei tag `p` appare uguale in entrambe le righe e rappresenta i paragrafi nei quali è articolata la pagina.

I file HTML possono essere salvati sui supporti di memorizzazione (dischi) sia con estensione (l'estensione di un file è la parte del nome del file che segue il punto) .htm sia con estensione .html.

L'estensione .htm è più diffusa e probabilmente tale maggiore diffusione è dovuta al fatto che i sistemi operativi del recente passato consentivano di denominare i file attribuendo all'estensione del nome la lunghezza massima di tre caratteri.¹³

2.5 La sezione head

La sezione head di una pagina HTML, detta intestazione del documento, è sempre compresa tra il tag <html> e il tag <body>. A causa della posizione “obbligata” nella quale questo tag deve trovarsi molti browser sono in grado di stabilire automaticamente la sua posizione e per questo motivo, spesso, sono presenti sul Web documenti senza l'esplicita presenza di questo tag. In base alle specifiche HTML nella sezione head possono essere inclusi solo determinati tag, come TITLE e META. Tutto ciò che fa parte di questa sezione¹⁴ non è visualizzato dal browser, ad eccezione di quanto previsto dal tag <title>. In realtà, se nella sezione head vengono introdotti tag che generano una visualizzazione, come ad es. h1 oppure p, molti browser visualizzano comunque tali elementi. In ogni caso le informazioni previste in questa sezione rivestono notevole importanza. La presenza della sezione head è facoltativa secondo le specifiche HTML 3.2 ma torna ad essere un prerequisito nella versione delle specifiche 4.01. Questa parte verrà trattata più approfonditamente nei paragrafi successivi.

2.5.1 Titolo del documento

Il titolo della pagina è un elemento particolarmente importante. Definisce e descrive in maniera sintetica il contenuto della pagina stessa (ciò che l'utente si aspetta di trovare) e pertanto dovrebbe essere al tempo stesso esplicativo e sufficientemente breve. Il titolo è definito mediante il tag <title> ... </title> che fa apparire la stringa di caratteri contenuta in tale elemento sulla riga di intestazione della finestra del browser. Il contenuto di questo tag, inoltre, dà il nome al collegamento nel momento in cui si aggiunge la pagina nei siti “preferiti” del browser. Alcuni motori di ricerca indicizzano il contenuto di questo tag. Queste due ultime caratteristiche del tag <title> devono essere opportunamente temperate. Se da un lato, infatti, è utile inserire numerose parole per una accurata descrizione del sito e per una conseguente idonea indicizzazione da parte dei motori di ricerca, dall'altro lato una descrizione troppo lunga impedisce un'agevole e rapida aggiunta del sito agli indirizzi preferiti dell'utente. Inoltre un titolo troppo lungo, all'atto della visualizzazione, potrebbe

essere troncato. Un'altra caratteristica del titolo è che si tratta del primo elemento che appare quando il browser inizia a caricare una pagina Web. Ciò significa che, in presenza di una connessione Internet lenta, la visualizzazione del titolo della pagina, di fatto, comunica all'utente che la pagina richiesta è stata trovata e che è in corso il caricamento. Per ciascun documento HTML è previsto un solo titolo.

2.6 Commenti

Come negli altri linguaggi anche per l'HTML è prevista la possibilità di inserire commenti nel codice sorgente. I commenti sono informazioni inserite a fini di documentazione e quindi non incidono sul funzionamento delle pagine. L'autore di pagine HTML dovrebbe sempre inserire nelle pagine commenti per documentare opportunamente il codice che scrive. I commenti vengono ignorati dal browser e sono visibili solo accedendo al codice sorgente della pagina. Proprio perché chiunque, esaminando il codice sorgente, è in grado di leggere i commenti inseriti non andrebbe scritto nulla che non possa essere pubblicamente letto. I commenti sono utili per fornire chiarimenti o informazioni aggiuntive, finalizzate alla manutenzione del codice HTML e sono particolarmente importanti nel caso di pagine complesse e/o se il lavoro è svolto da più sviluppatori. In quest'ultimo caso servono a comunicare efficacemente a tutti i componenti del gruppo di sviluppo le ragioni ed i procedimenti logici che hanno portato a compiere determinate scelte in fase di codifica. I commenti possono essere usati anche in fase di verifica e correzione degli errori (*debugging*), eliminando temporaneamente parti del codice senza necessità di cancellare effettivamente ed eventualmente riscrivere il codice eliminato. Di seguito si riportano alcuni esempi di commenti che può essere utile inserire nelle proprie pagine.

- nome dell'applicazione;
- descrizione dello scopo dell'applicazione;
- autore della pagina;
- data di prima creazione;
- data ultimo aggiornamento;
- numero versione della pagina;
- inizio e/o fine di parti di pagina;
- fonte delle immagini originali pubblicate nella pagina.

I commenti possono essere scritti in qualunque punto del documento HTML e possono essere costituiti da una o più righe. Sono racchiusi tra i tag “<!--” e “-->”.

Esempio:

```

istruzioni HTML
<!-- commento
articolato
su tre righe -->
altre istruzioni HTML

```

Si noti che il punto esclamativo va inserito prima della parentesi angolare di apertura ma non prima della parentesi angolare di chiusura.

2.7 Elemento *h*

Tramite l'elemento *h* (*h* sta per *headin*:intestazione). è possibile creare nelle pagine intestazioni di diverse dimensioni e quindi di diverso impatto visivo. Queste intestazioni, o titoli, non vanno confuse con il titolo della pagina HTML generato mediante il tag `title`. Si tratta infatti di componenti che svolgono funzioni simili a quelle svolte dai titoli di una qualsiasi pubblicazione, nella quale separano parti di testo e riassumono gli argomenti trattati nelle diverse parti, e nelle quali ad argomenti più importanti corrispondono, di norma, titoli più grandi. In HTML esistono 6 livelli di intestazione che appaiono con diversa formattazione. Il primo livello (*h1*) è quello che appare con caratteri più grandi rispetto a quelli degli altri livelli, ed è quindi contraddistinto da una maggiore enfasi. Usando, ad esempio, il tag `<h3>` comunichiamo al browser una specifica formattazione che può essere costituita da una combinazione di caratteri e colori da applicare, analogamente a quanto avviene con gli elaboratori di testo. Si osservi che nel visualizzare la pagina, HTML aggiunge una riga bianca prima e dopo ciascuna tag di intestazione.

Di seguito si riporta un esempio di sintassi con l'uso dell'attributo *align*, che regola l'allineamento del testo.

```

<h1 [align="left"|"center"|"right"]> .... </h1>
<h2 [align="left"|"center"|"right"]> .... </h2>
<h3 [align="left"|"center"|"right"]> .... </h3>

```

Per vedere l'effetto prodotto a video dai tag di intestazione si veda la figura 7.

2.8 Paragrafi

I paragrafi di una pagina sono definiti ricorrendo al tag `<p>`. I tag di paragrafo sono utili per assicurarsi che il browser visualizzi il testo con la formattazione desiderata in quanto, come sappiamo, gli spazi in bianco presenti nei testi non vengono considerati. Il tag paragrafo dispone di un tag di apertura e di un

tag di chiusura, anche se il tag di chiusura in genere è omissivo. E' diffusa l'abitudine di applicare spazi in bianco tra blocchi di testo ricorrendo a una successione di tag `<p>`, senza il corrispondente tag di chiusura. Questa pratica, però, non risulta particolarmente valida in quanto, a seconda dei browser usati, è anche possibile che all'atto della visualizzazione venga considerato solo il primo dei tag `p`, con la conseguenza che non verrà visualizzato tutto lo spazio desiderato. Come per i tag di intestazione (`h`) anche in questo caso nel visualizzare la pagina l'HTML aggiunge una riga bianca prima e dopo ciascun paragrafo.

Di seguito si riporta un esempio di sintassi con l'uso dell'attributo *align* (allineamento del testo).

```
<p [align="left"|"center"|"right"]> .... </p>
```

2.9 Linee orizzontali

La leggibilità di un testo può essere migliorata suddividendolo in sezioni tramite apposite linee orizzontali, tracciate sul dispositivo di rappresentazione della pagina. In HTML le linee orizzontali si ottengono utilizzando il tag `<hr>` (horizontal rule: righello). Per questo elemento è previsto il solo tag di apertura in quanto non deve contenere nulla al proprio interno. Le linee tracciate con questo tag non occupano necessariamente tutta la lunghezza della pagina che è invece definita tramite un apposito attributo previsto per il tag.

Di seguito è mostrato un esempio di sintassi con l'utilizzo di alcuni attributi.

```
<hr [align="left"|"center"|"right"] [size="n"] [noshade] [width="nn%"]>
```

L'attributo *align*, così come per altri tag, regola l'allineamento orizzontale dell'elemento. L'attributo *size* imposta l'altezza della riga in pixel (il valore di partenza è 1) mentre l'attributo *width* determina la larghezza della riga. Width può essere impostato come valore fisso in pixel (es. `size="50"`) oppure come valore percentuale (es. `size="50%"`). Nel caso del valore fisso la lunghezza della riga corrisponde al valore impostato mentre nel caso del valore espresso in percentuale l'estensione della riga all'interno della finestra del browser corrisponde al valore percentuale assegnato, indipendentemente dalle effettive dimensioni della finestra. In altri termini il valore in percentuale dell'estensione della riga viene mantenuto anche nel caso in cui la finestra del browser è modificata manualmente dall'utente. Impostando un valore percentuale pari a 100 la linea si estende per tutta la larghezza della finestra del browser. Per default la linea viene tracciata sul dispositivo di emissione con un effetto tridimensionale, particolarmente evidente quando la linea ha un notevole spessore (a partire da 5 o 6 pixel). Usando l'attributo *noshade* la riga è riprodotta senza l'effetto predefinito di ombreggiatura. Questo attributo non richiede la presenza di un segno di uguale e, per disattivare l'impostazione predefinita, è richiesta

solo la sua presenza. Gli attributi di questo tipo, che attivano o disattivano le impostazioni predefinite semplicemente con la loro presenza, sono definiti commutatori (*toggle*). Si tenga presente, infine, che gli attributi possono essere elencati in qualsiasi ordine.

Esempio

```
<html>
  <head>
    <title>Esempi di righe orizzontali</title>
  </head>
  <body>
    <hr>
    <hr align="center" size="5" width="50">
    <hr align="center" size="5" width="50%">
    <hr align="center" size="10" width="50%">
    <hr align="center" size="10" width="50%" noshade>
  </body>
</html>
```

Il codice produce a video il risultato illustrato nella figura 8.

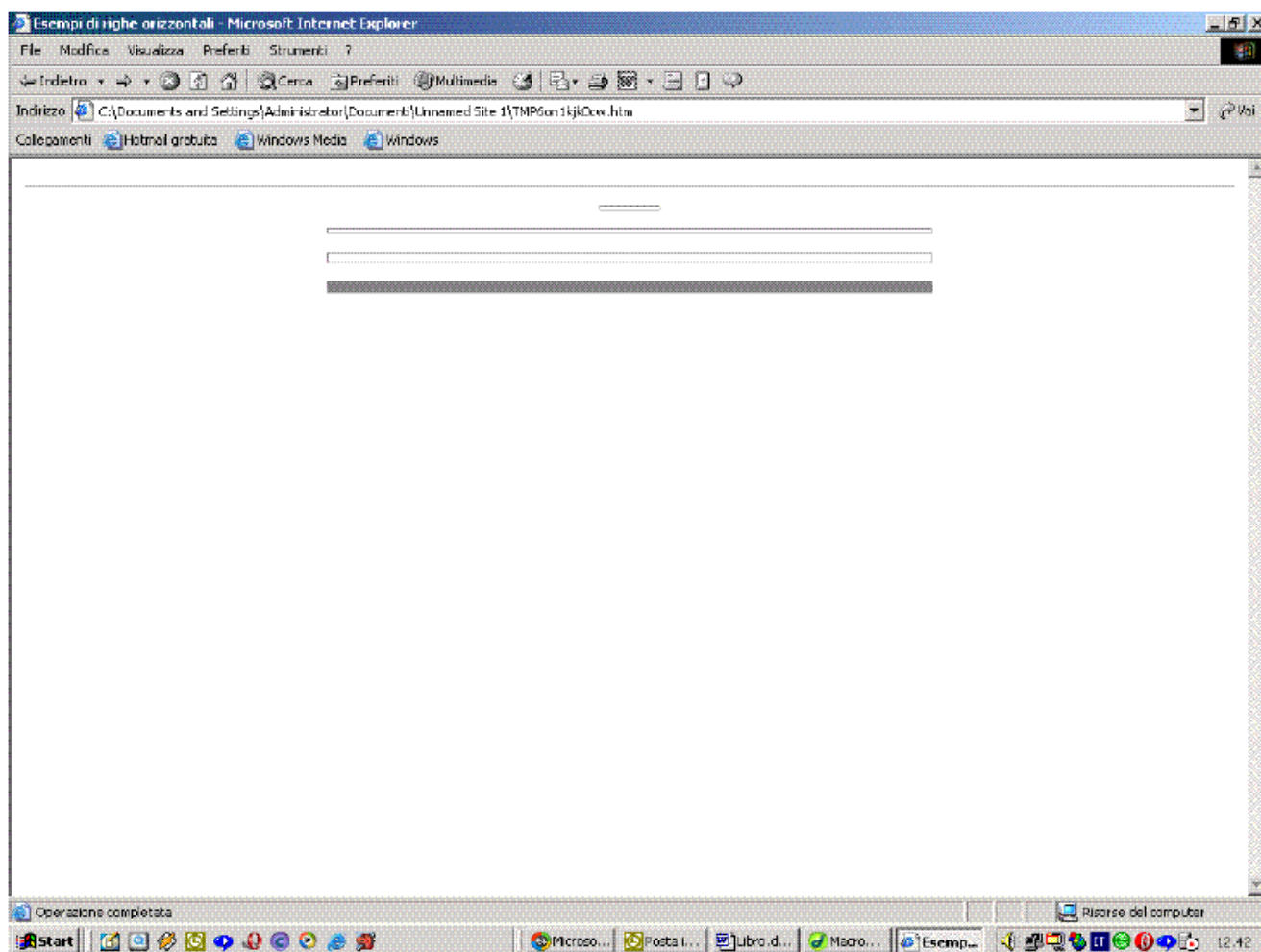


Fig. 8: Esempi di linee orizzontali ottenute con il tag hr

2.10 Linee di interruzione

Talvolta si desidera iniziare una nuova riga di testo senza creare un nuovo paragrafo con il contestuale inserimento di una riga in bianco. In HTML per far cominciare il testo su una nuova riga si usa il tag `<br>`. La digitazione del tasto “Invio” nel codice sorgente non produce alcun effetto in quanto i punti in cui le righe si interrompono sono determinati dalla larghezza della finestra del browser, che di norma l’utente può ridimensionare a suo piacimento.

Con il tag `br` il browser viene forzato a visualizzare una linea di interruzione. Questo tag non prevede un tag di chiusura (si tratta quindi di un elemento vuoto). E’ inoltre possibile utilizzare successioni di tag `br` che in diversi browser vengono resi con blocchi di spazio in bianco più o meno estesi a seconda del numero di tag `br` inseriti.

Esempio

```
<html>
  <head>
    <title>Tag br</title>
  </head>
  <body>
    <p>Linea di testo nella quale qui
    è stato premuto il tasto Invio durante la digitazione</p>
    <p>Linea di testo nella quale qui<br />è stato inserito il tag
    br</p>
    <p>Si può osservare che la prima linea di testo non si interrompe
    mentre la seconda sì</p>
  </body>
</html>
```

Il codice produce a video il risultato illustrato nella figura 9.

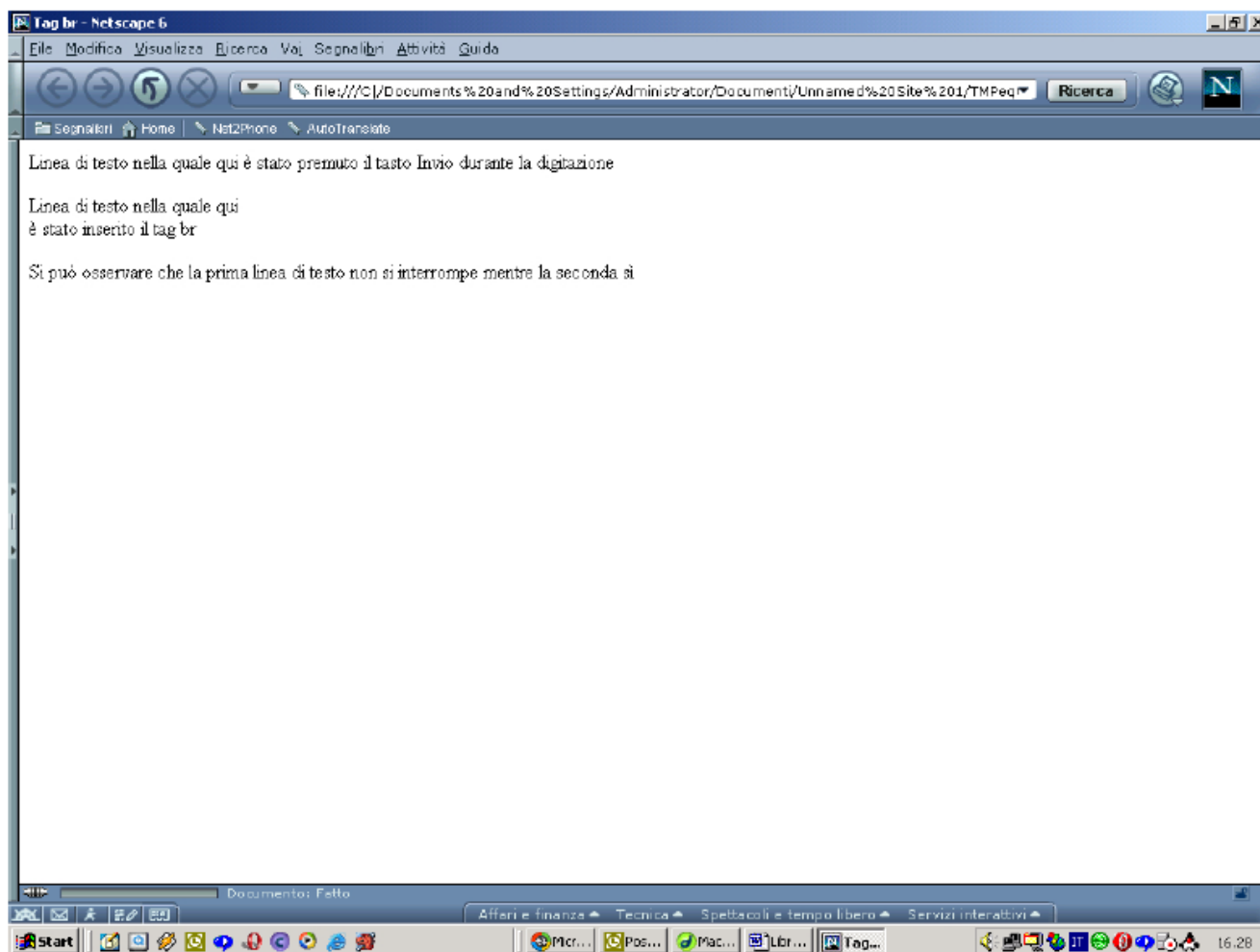


Fig. 9: Interruzioni di linea ottenute con il tag br

Un attributo del tag `br` che è opportuno esaminare più in dettaglio è *clear*, usato per gestire la disposizione del testo attorno alle immagini. In questo caso l'interruzione di linea che si desidera introdurre può comportare la prosecuzione del testo su una nuova riga sia a fianco dell'immagine sia al di sotto della stessa. Con l'attributo *clear* si determina tale comportamento.

L'esempio riportato di seguito dovrebbe chiarire meglio il contesto in cui l'attributo agisce.

```
<html>
  <head>
    <title>Attributo clear del tag br</title>
  </head>
  <body>
    
    <p>Questo testo è adiacente all'immagine presente nella pagina e
    tutte le parole in esso presenti continuano ad essere adiacenti,
    tranne nel caso di ridimensionamento della finestra che potrebbe
    far spostare le parole al di sotto dell'immagine</p>
    <p>In questo testo è usato l'attributo clear a partire da qui <br
    clear="left" /> ed il testo continua al di sotto dell'immagine a
    partire dal margine sinistro della prima riga disponibile</p>
  </body>
</html>
```

Il codice produce a video il risultato illustrato nella figura 10.

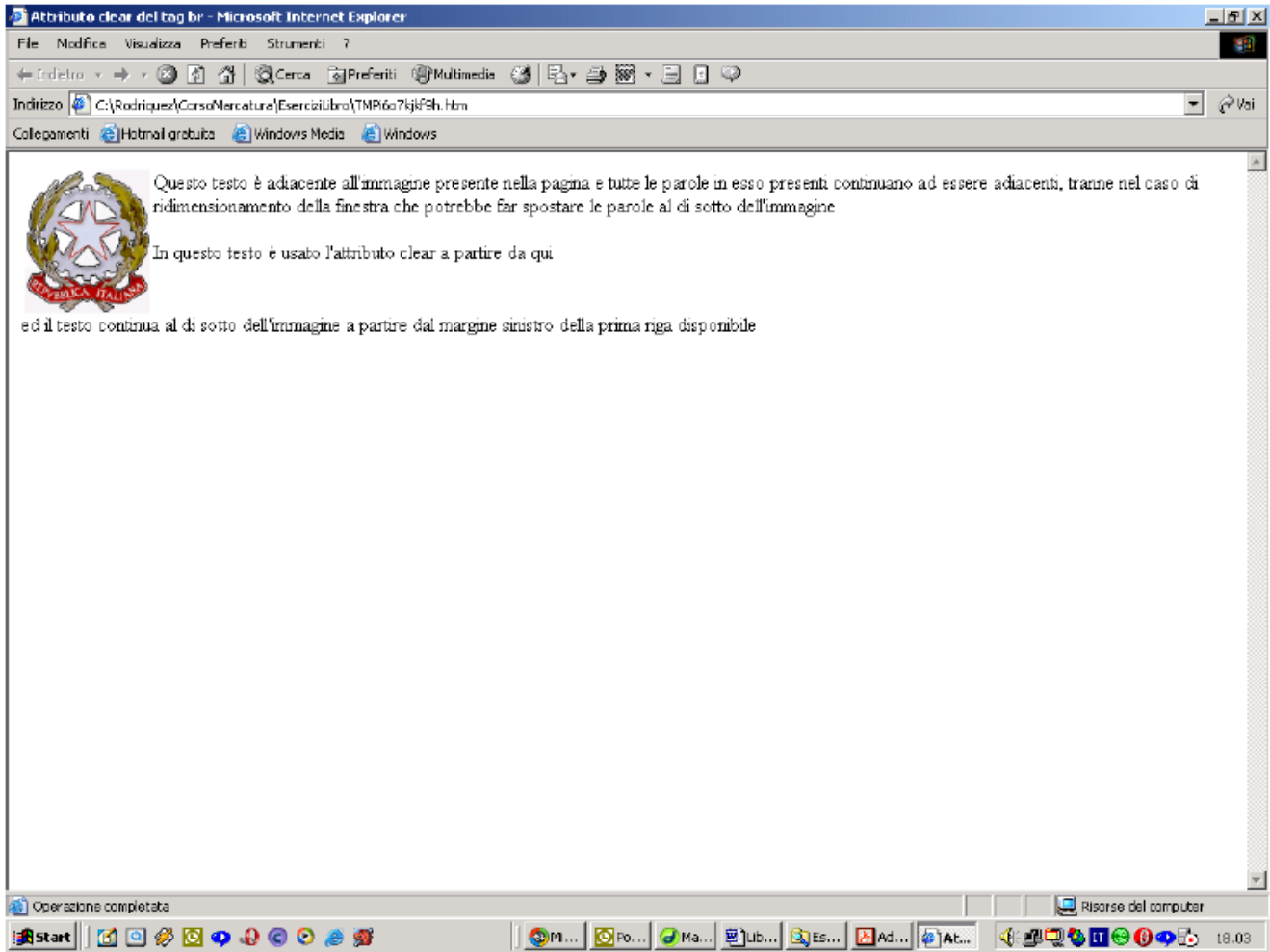


Fig. 10: Effetto dell'attributo clear in presenza di un'immagine

I valori ammissibili dell'attributo clear sono *left*, *right*, *all* e *none*. L'effetto del valore *left* è stato appena illustrato. *Right* fa apparire il testo sul margine destro della prima riga disponibile mentre *all* richiede che sia il margine sinistro sia il margine destro siano disponibili per far continuare il testo. Infine con *none*, che è anche il valore di default, il testo continua a scorrere nella riga successiva. Si propone ora il codice seguente che riepiloga alcuni dei tag sinora esaminati

```
<html>
  <head>
    <title>Università ...</title>
  </head>
  <body>
    <h1>Università ...</h1>
    <h2> Informazioni...</h2>
    <p>Questo sito fornisce informazioni sui servizi offerti a docenti,
```

```
    studenti e terze parti interessate.</p>
    <hr>
    <h3>Corsi</h3>
    <p align="center"> Elenco corsi</p>
    <hr width="50%">
  </body>
</html>
```

Il codice genera il risultato illustrato nella figura 11.

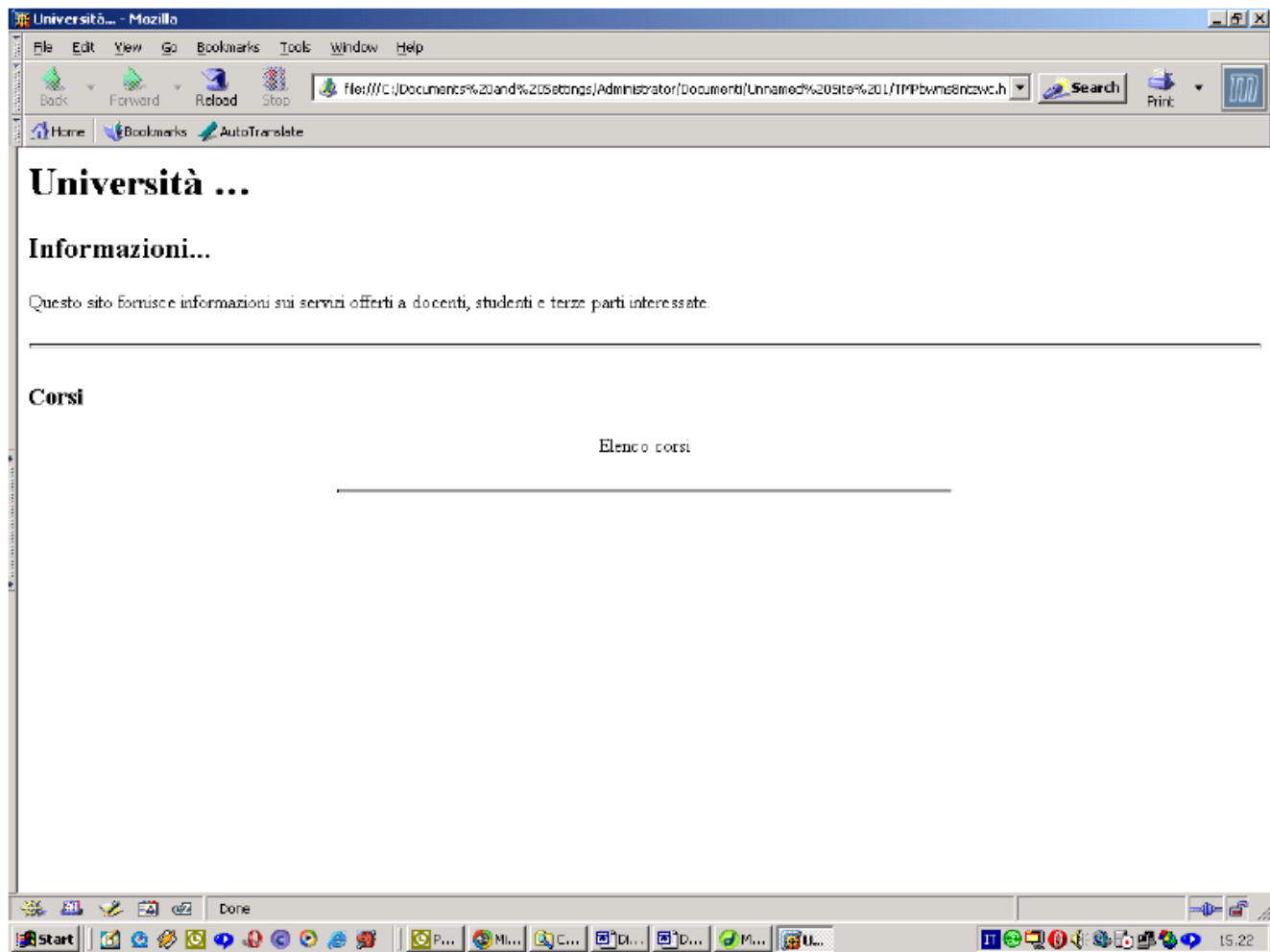


Fig. 11: Sezioni head e body con altri tag

Nell'esempio si noti, in particolare, il tag *hr* usato sia senza alcun valore sia con un valore espresso in percentuale. Nel primo caso la riga occupa sempre l'intera larghezza della finestra del browser mentre

nel secondo caso la riga occupa sempre il 50% della finestra, anche nel caso in cui questa venga ridimensionata manualmente dall'utente.

2.11 Formattazione dei caratteri

La formattazione dei caratteri, unitamente ad altre caratteristiche, assume notevole importanza nel determinare il livello di efficacia ottenibile attraverso la comunicazione della parola scritta. Un semplice esempio è offerto dalla decisione in merito alle dimensioni da attribuire ai caratteri poiché a minori dimensioni corrisponde, a parità di spazio disponibile sulla pagina, un maggior numero di informazioni rappresentabili, al quale si accompagna, di contro, una possibile maggiore difficoltà di lettura. HTML è corredato di opportuni tag volti a definire e modificare nei modi più diversi le modalità di visualizzazione del testo. Ad esempio, oltre alla possibilità di scegliere il tipo font da utilizzare per la visualizzazione del testo, l'autore della pagina HTML può impostarne le dimensioni ed il colore così come può far apparire determinate stringhe o parti di testo in grassetto, corsivo, sottolineato, ecc. Questi tag possono essere usati in combinazione fra loro, per cui una stringa può ad esempio essere visualizzata contestualmente in corsivo ed in grassetto.

Il tag per applicare un determinato tipo di carattere alla pagina è *font*. Prevede diversi attributi tra i quali qui si menzionano:

- *Size*, per indicare la dimensione dei caratteri; la sintassi è `size="N"` in cui N può variare da 1 (minima dimensione) a 7 (massima dimensione). Si osservi che nel caso siano indicati valori minori o maggiori, rispettivamente, di quelli minimi e massimi consentiti l'effettiva dimensione sarà ricondotta al ed espressa con il valore minimo o massimo consentito. Il valore di default dell'attributo size è 3. È anche possibile usare espressioni del tipo `size="+2"` o `size="-2"` che genera la rappresentazione dei caratteri, rispettivamente, aumentandola o diminuendola di due corpi rispetto alla precedente dimensione.
- *Face*, per specificare il nome del carattere o del gruppo di caratteri. In genere si elencano più nomi di caratteri ed il browser userà il primo dell'elenco, se installato sul computer dell'utente, passando poi a usare gli altri nello stesso ordine con il quale i caratteri sono elencati, nel caso in cui i caratteri non siano installati. Il processo continua sino a quando non si individua un carattere compreso nell'elenco e presente sul computer. In caso negativo il browser utilizzerà il proprio carattere di default. Per questo motivo risulta preferibile utilizzare nelle proprie pagine font abbastanza comuni, disponibili sui computer degli utenti, in modo che le pagine siano

visualizzate con le caratteristiche di formattazione previste dall'autore. La sintassi è `face="NomeCarattereA, NomeCarattereB, NomeCarattereC"`

- *Color*, per indicare il colore con il quale il carattere deve essere rappresentato. Il colore può essere indicato attraverso il suo nome o tramite il suo corrispondente valore esadecimale.

Esempio

```
<html>
  <head>
    <title>Elemento font</title>
  </head>
  <body>
    <p>
      <font size="6" face="Verdana, Arial, Helvetica, sans-serif"
      color="blue">L</font>a prima lettera di questo paragrafo è più
      grande degli altri caratteri ed appare in colore blu.
    </p>
  </body>
</html>
```

Il codice produce il risultato illustrato nella figura 12.

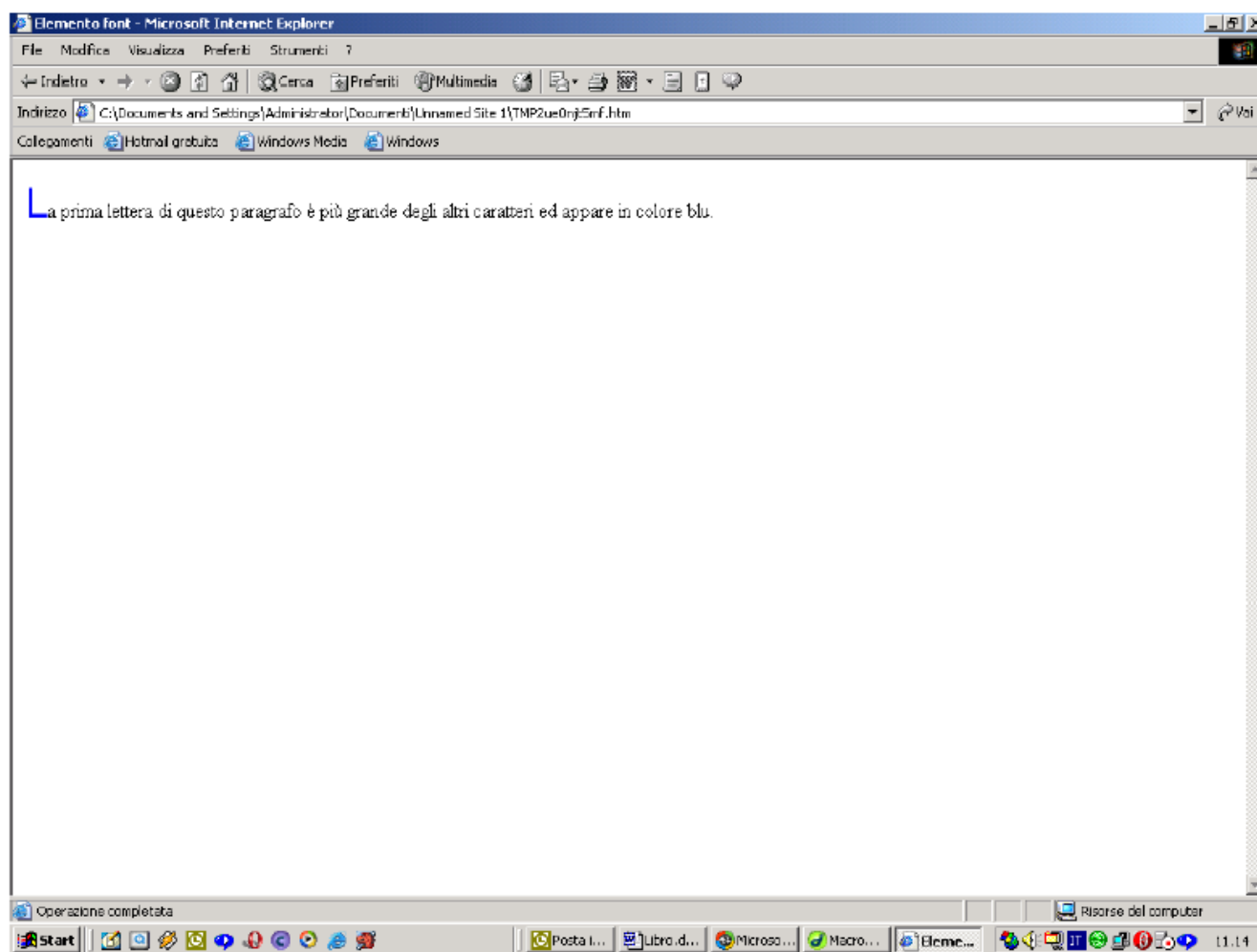


Fig. 12: Esempio di utilizzo del tag font

Anziché lavorare con singole porzioni di testo è possibile attribuire ad un intero documento HTML una dimensione predefinita dei caratteri tramite l'elemento *basefont*, i cui principali attributi sono gli stessi illustrati per il tag *font*. Il tag *basefont* non dispone di un tag finale poiché le informazioni occorrenti per le funzioni che deve svolgere sono già tutte comprese all'interno del tag iniziale. E' possibile utilizzare utilmente, nello stesso documento, sia il tag *font* sia il tag *basefont*, impostando tramite *font* le dimensioni dei caratteri in maniera *relativa* rispetto a quanto specificato da *basefont*. Il seguente codice esemplificativo dovrebbe risultare utile per chiarire il meccanismo.

```
<html>
  <head>
    <title>Elementi basefont e font</title>
  </head>
  <body>
    <basefont size="4" face="Arial, Helvetica, sans-serif">
    <p>Testo basato sul tag basefont con attributo size="4"
    </p>
    <p>
      <font size="-1"> Testo ridotto di una misura rispetto a basefont(3)
    </font>
    </p>
    <p>
      <font size="+1"> Testo aumentato di una misura rispetto a basefont(5)
    </font>
    </p>
    <p>
      <font size="+2"> Testo aumentato di due misure rispetto a basefont(6)
    </font>
    </p>
    <p>
      <font size="+3"> Testo aumentato di tre misure rispetto a basefont(7)
    </font>
    </p>
    <p>
      <font size="+5"> Testo aumentato di cinque misure rispetto a
      basefont(9)e ricondotto al massimo (7)
    </font>
    </p>
  </body>
</html>
```

Il codice produce il risultato illustrato nella figura 13.

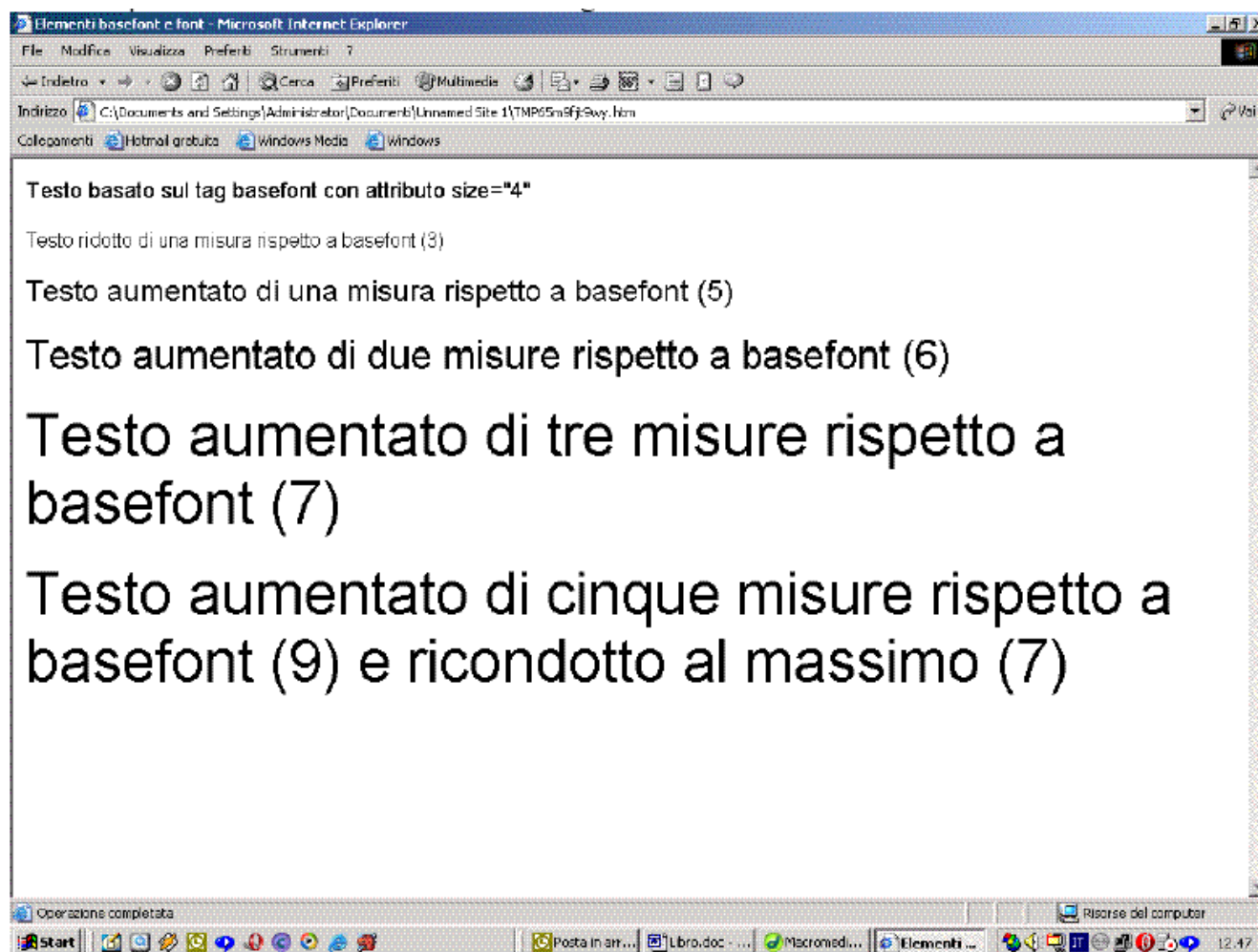


Fig. 13: Ridimensionamento del testo con i tag basefont e font

Si osservi che anche nel caso in cui le dimensioni dei caratteri siano impostate in modalità relativa non possono comunque ricadere al di fuori dei limiti minimi (1) e massimi (7) previsti. Per i valori assegnati all'attributo *face* si veda quanto esposto al paragrafo “Famiglie di caratteri”.

2.11.1 Famiglie di caratteri

Abbiamo già discusso dell'opportunità, all'interno di una stessa pagina HTML, di indicare più nomi di caratteri che saranno usati dal browser nello stesso ordine in cui sono elencati, sino a trovare un carattere disponibile sul computer dell'utente. Un'ulteriore possibilità di far visualizzare le pagine nel modo più aderente possibile rispetto a quello progettato è data dall'utilizzo di nomi di *famiglie generiche di caratteri*. Tali famiglie generiche sono quattro, come elencato di seguito.

- *serif*: è resa sempre con caratteri con grazie;
- *sans-serif*: è resa sempre con caratteri senza grazie;
- *cursive*: è resa sempre con caratteri in corsivo;
- *monospace*: è resa sempre con caratteri monospaziati, nei quali ad ogni carattere è riservata una medesima larghezza.

In quanto ultima alternativa normalmente le cosiddette famiglie di caratteri sono posizionate alla fine della lista dei nomi dei caratteri nelle istruzioni HTML. In questo modo se ad esempio si vuole rappresentare un testo o una parte di esso con caratteri con grazie, oltre agli specifici nomi di caratteri aventi questa caratteristica, si terminerà la lista con il nome della famiglia di caratteri *serif*.

L'effettivo carattere usato dipende dal sistema operativo e dal browser con il quale il documento è visualizzato ma, con questa tecnica, si è certi che la rappresentazione avverrà con uno dei caratteri appartenenti alla famiglia indicata.

2.11.2 Colori

I colori, contribuendo a guidare l'utente attraverso la struttura di un sito e le diverse sezioni di cui è costituito, rappresentano un altro fondamentale elemento della progettazione delle pagine Web.

Di seguito si propongono i principali termini usati per la descrizione delle proprietà dei colori.

Tinta – E' il nome di un colore dello spettro (giallo, rosso, blu, ecc.);

Saturazione – E' l'intensità di un colore;

Valore – E' la luminosità o lo scuro relativo di un colore;

Ombra – E' la gradazione di colore ottenuta tramite l'aggiunta di nero;

Colori primari – sono i tre colori principali di un modello cromatico;

Colori secondari – sono i colori prodotti attraverso la combinazione di due colori primari.

La rappresentazione dei colori negli schermi televisivi e nei monitor dei computer è ottenuta tramite il modello cromatico *RGB* (Red, Green, Blue) che è un modello *additivo*, vale a dire che i colori si ottengono componendo le onde luminose, sommando i componenti principali dei colori per ottenere tutte le possibili tonalità. In questo modello i colori primari sono rosso, verde e blu mentre i colori secondari sono ciano, magenta e giallo. Nelle pagine HTML i colori possono essere indicati attraverso il loro nome o mediante un valore esadecimale a sei cifre. I nomi dei colori sono più semplici da utilizzare e devono essere scelti da un'insieme di nomi contenuti nelle specifiche HTML o supportati

dai diversi browser. Delle sei cifre del valore esadecimale le prime due cifre si riferiscono al colore rosso, le seconde due al colore verde e le ultime due al colore blu.

L'assenza di colore è indicata con il valore 00 mentre la presenza del colore alla sua massima intensità è espressa con il valore FF. I colori si ottengono quindi combinando diversamente i valori di ciascuna coppia indicanti l'intensità dei tre colori primari da usare per l'ottenimento del colore finale. Se tutti i valori sono espressi con la massima intensità possibile si ottiene il colore bianco mentre l'assenza di tutti e tre i valori determina la riproduzione del colore nero. La sintassi di HTML prevede che le tre coppie di valori siano precedute dal simbolo “#”. Tale simbolo (detto *octothorp*) ha lo scopo di comunicare al browser che il valore che segue è il numero del colore espresso in esadecimale e non il nome del colore stesso. In HTML i colori sono espressi utilizzando combinazioni dei colori primari rosso, verde e blu, in base al modello RGB. Ognuno dei tre colori primari incide in base ad un valore compreso tra 0 e 255, per complessivi 256 valori. Numerosi editor grafici HTML consentono di selezionare i colori direttamente da una cosiddetta tavolozza che appare sullo schermo durante l'editazione della pagina, inserendo automaticamente nel codice sorgente i valori esadecimali associati alle scelte cromatiche effettuate dall'autore.

Infine, ma l'argomento non è meno importante, deve essere considerato che ai colori sono in genere associati significati diversi, sia positivi sia negativi, e che possono generare reazioni emotive di diversa intensità nei visitatori. A ciò deve essere aggiunto il diverso significato che uno stesso colore può avere in relazione alle differenti etnie e culture. Non dimentichiamo, infatti, che i siti Web sono “vetrine” aperte 24 ore su 24 e 7 giorni su 7, accessibili da qualsiasi postazione connessa ad Internet, da qualsiasi parte del mondo.

Di seguito è illustrato un esempio di codice che imposta il colore di sfondo della pagina in blu ed il colore del testo in bianco. Per modificare il colore di default dello sfondo si usa l'attributo *bgcolor* mentre per modificare il colore di default di un testo si utilizza l'attributo *text*, entrambi utilizzabili all'interno della sezione *body*.

```
<html>
  <head>
    <title>Esempio di colori caratteri e sfondo</title>
  </head>
  <body bgcolor="#0000FF" text="#FFFFFF">
    Questa riga di testo è in bianco su sfondo blu
  </body>
</html>
```

Il codice produce il risultato illustrato nella figura 14.

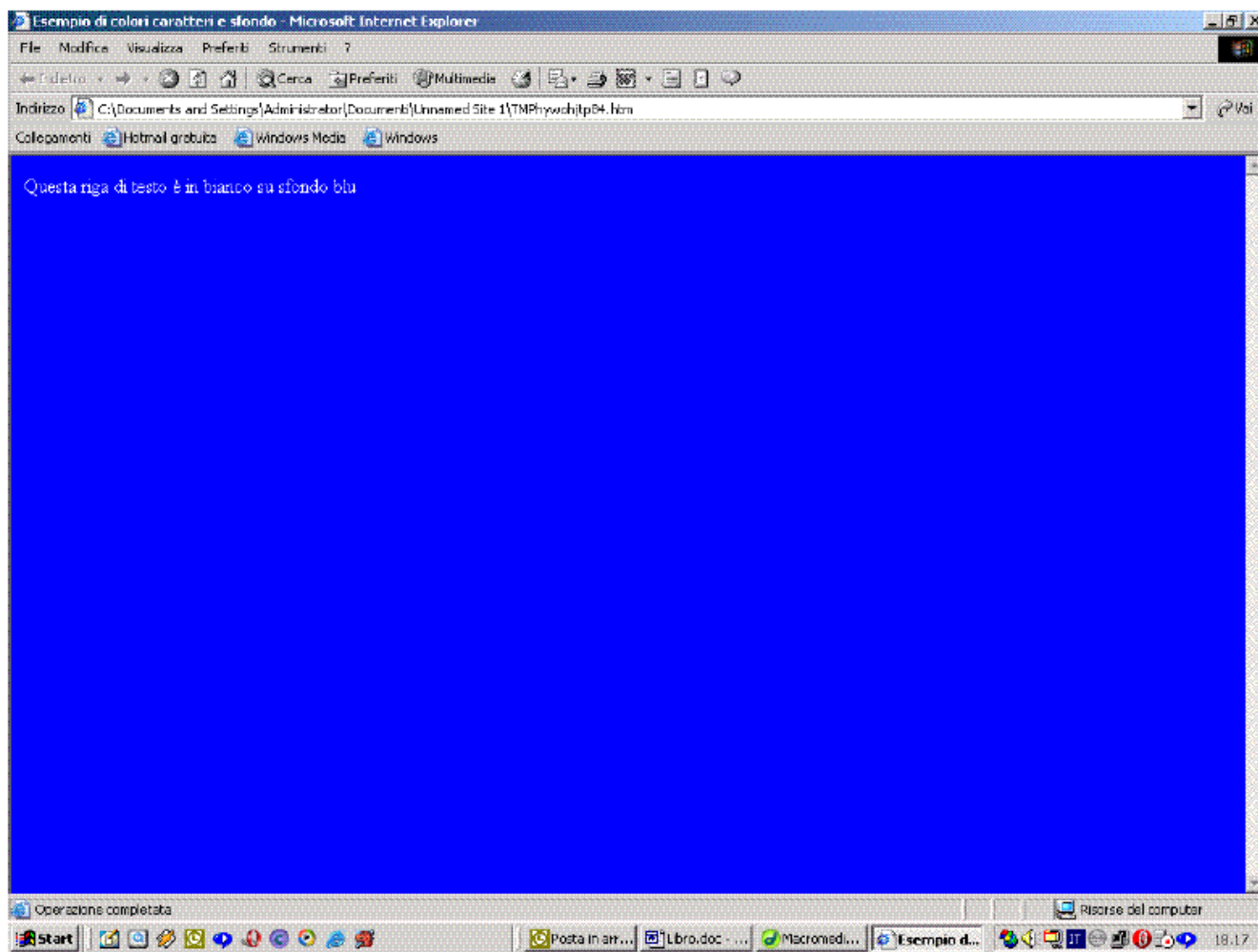


Fig. 16: Variazione dei colori nella sezione body con alcuni suoi attributi

Si può ottenere lo stesso effetto indicando i colori attraverso il loro nome anziché i valori esadecimali, come nella riga seguente:

```
<body bgcolor="blue" text="white">
```

2.12 Tag di stile fisici e logici

Oltre agli interventi di formattazione sui caratteri, un testo può essere modificato da numerosi altri tag che possono essere suddivisi concettualmente in tag *fisici* e tag *logici*.

Per comprendere questi due concetti può essere opportuno riferirsi ad un esempio. Si supponga di voler evidenziare una parte di testo utilizzando, ad esempio, il carattere corsivo. Ciò può essere ottenuto ricorrendo al tag `<i>` oppure al tag `<em>`. Il primo, che è un tag fisico, rende in corsivo la parte di testo contrassegnata, indipendentemente dal browser e dalla piattaforma di rappresentazione utilizzata. Il secondo, che rientra invece nella categoria dei tag logici, evidenzia la parte di testo contrassegnata, *generalmente*, in corsivo. Nel secondo caso, in altri termini, si ottiene comunque l'effetto di evidenziazione ma non si è certi del modo in cui ciò avviene. A questo punto possiamo affermare che un tag fisico ha lo scopo di contrassegnare le parti del documento per definirne la struttura eseguendo contestualmente l'opportuna formattazione mentre un tag logico definisce la struttura prescindendo dallo specifico aspetto. Volendo fornire un altro esempio, il tag `<b>` visualizza in grassetto il testo incluso nell'elemento ed ha pertanto lo scopo di formattare opportunamente il testo. Si tratta perciò di un tag fisico.

Il concetto sottostante l'utilizzo dei tag logici sta nell'ottimizzazione della rappresentazione delle pagine in rapporto alla piattaforma usata per tale rappresentazione. Ritorniamo a considerare il tag `em` prima esemplificato. Sappiamo che evidenzia il testo contrassegnato. Il punto è che le pagine possono essere accedute e rappresentate tramite diversi strumenti, tra i quali rientrano ad es. i sistemi vocali che leggono le pagine trasformando le informazioni digitali in messaggi vocali. Nel caso del tag `em` la parte di testo evidenziata verrà, in genere, visualizzata in corsivo se la pagina è acceduta tramite browser e verrà, in genere, riprodotta con un volume maggiore rispetto alla restata parte del testo, nel caso venga acceduta tramite un sistema vocale. L'evidenziazione ottenuta tramite il tag `<i>` sarebbe quindi valida solo nel primo caso ma nel secondo non avrebbe avuto effetto. Non esiste un modo per pronunciare un corsivo! Il tag logico viene pertanto reso a seconda di come è interpretato dalla piattaforma di rappresentazione effettivamente utilizzata. In conclusione i tag logici dovrebbero essere usati al posto di quelli fisici nel caso in cui si ritiene che le pagine siano accedute attraverso dispositivi di diversa tipologia.

Nell'esempio successivo si riportano alcuni tag, fisici e logici, per la formattazione dei caratteri.

`<html>`

```

<head>
  <title>Formattazione caratteri</title>
</head>
<body>
  <h1>Esempi di formattazione caratteri</h1>
  <p>Testo normale</p>
  <p><b>Testo in grassetto</b></p>
  <p><i>Testo in corsivo</i></p>
  <p>Testo in<sup>apice</sup></p>
  <p>Testo in<sub>pedice</sub></p>
  <p><tt>Testo monospaziato</tt></p>
  <p>Testo<s>sbarrato</s></p>
  <p>Testo con formattazione<pre>
p
r
o
p
r
i
a</pre>
</p>
</body>
</html>

```

Il codice produce a video quanto rappresentato nella figura 15.

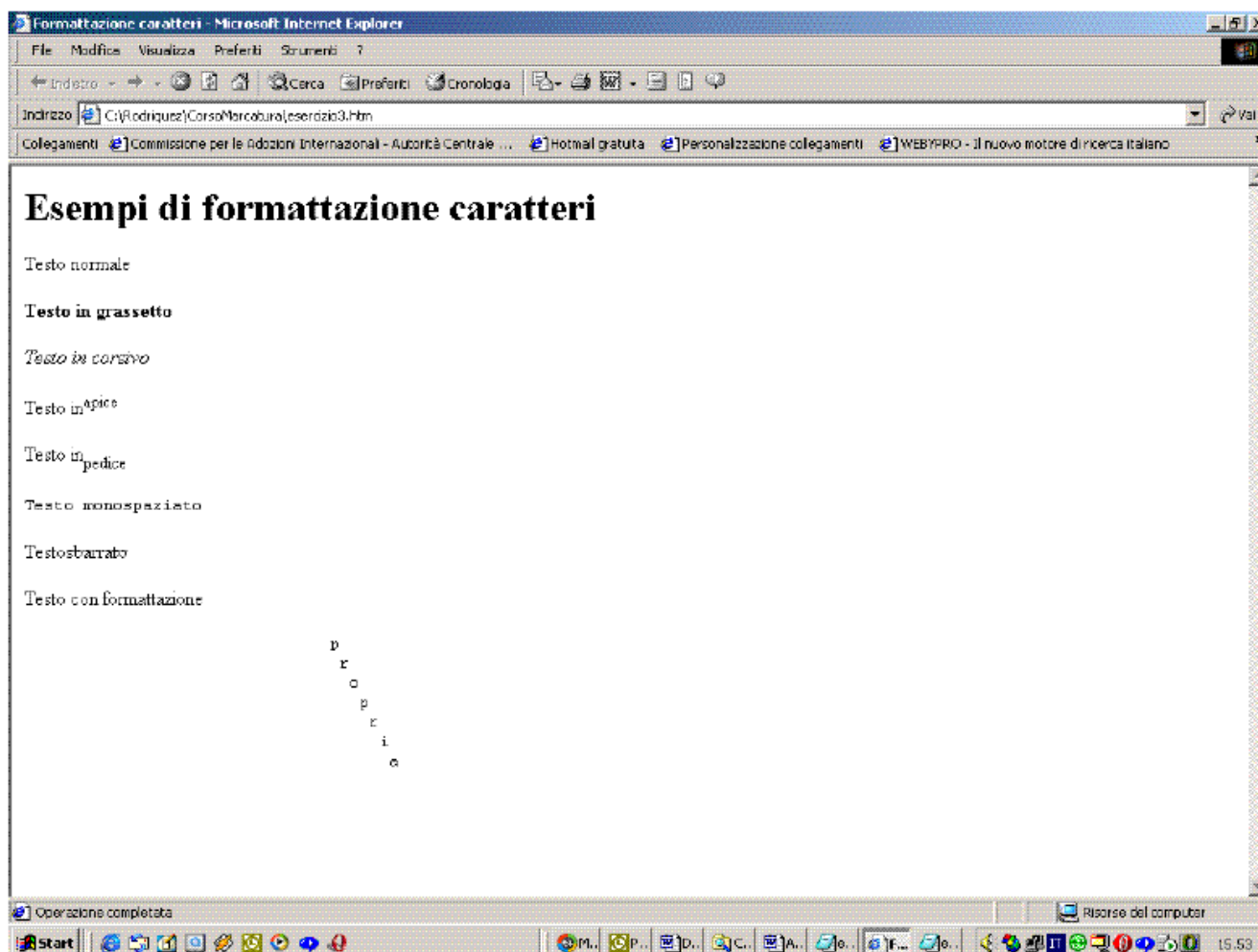


Fig. 15: Esempi di tag per la formattazione dei caratteri

Dall'immagine risultante a video si può osservare che:

- il tag (logico) <sup> visualizza il testo in apice;
- il tag (logico) <sub> visualizza il testo in pedice;
- il tag (fisico) <s> trasforma il testo in un testo sbarrato, tracciando su di esso una linea orizzontale;
- il tag (fisico) <tt> attribuisce al testo un carattere in cui ad ogni lettera o spazio è riservata una medesima dimensione (carattere monospaziato);
- il tag <pre> visualizza il testo con la sua formattazione predefinita, nella quale rimangono invariati gli spazi e le interruzioni di riga.

2.13 Elenchi

Con HTML è possibile generare tre tipi di elenco:

- a punti (non ordinato)
- numerati
- di definizioni

Gli elenchi possono contenere al loro interno altri elenchi.

Tag utilizzati:

`<ul>...</ul>` (unordered list – elenco a punti)

Questo tag definisce il contenuto di un elenco puntato. E' disponibile un attributo *type* per specificare il tipo di punto da visualizzare, che può assumere i seguenti valori:

- *square* per un piccolo quadrato
- *circle* per un punto vuoto circolare
- *disc* (simbolo predefinito) per un punto pieno circolare

`<li>...</li>`

Questi tag contengono le voci degli elenchi puntati e numerati. Si noti che all'interno di ciascuna voce è possibile inserire interruzioni di riga, paragrafi, link, altri elenchi, ecc.

`<ol>...</ol>` (ordered list – elenco ordinato)

Questi tag definiscono il contenuto di un elenco numerato. Esiste un attributo *type* per specificare lo schema di numerazione da usare (es. da quale numero iniziare). I valori dell'attributo *type* sono:

- *I* per i numeri romani maiuscoli
- *i* per i numeri romani maiuscoli
- *a* per lettere minuscole
- *A* per lettere maiuscole

Si noti che il *type* predefinito visualizza numeri arabi.

L'attributo *start* stabilisce il numero o la lettera dalla quale l'elenco deve avere inizio. Ad es. *start=5* farà iniziare l'elenco dal numero 5 (arabo o romano) o dalla lettera E (quinta lettera dell'alfabeto).

L'attributo *start* accetta solo valori numerici.

`<dl>...</dl>` (definition list)

Questo tag indica un elenco di definizioni, ovvero una lista di termini e delle spiegazioni relative a ciascun termine.

`<dt>...</dt>` (definition terms)

Questo tag contiene l'elenco delle voci da definire (nell'esempio successivo: Prima voce elenco definizioni)

`<dd>...</dd>` (definitions)

Questo tag contiene le definizioni delle voci contenute nell'elenco

Esercizio

`<html>`

`<body>`

`<h2>Esempio elenchi disponibili</h2>`

`<h3>Elenco a punti</h3>`

`<ul>`

`<li>Prima voce elenco</li>`

`<li>Seconda voce elenco</li>`

`<li>Terza voce elenco</li>`

`</ul>`

`<h3>Elenco numerato</h3>`

`<ol>`

`<li>Prima voce elenco num.</li>`

`<li>Seconda voce elenco num.</li>`

`<li>Terza voce elenco num.</li>`

`</ol>`

`<h3>Esempio elenco di definizioni</h3>`

`<dl>`

`<dt>Prima voce elenco definizioni</dt>`

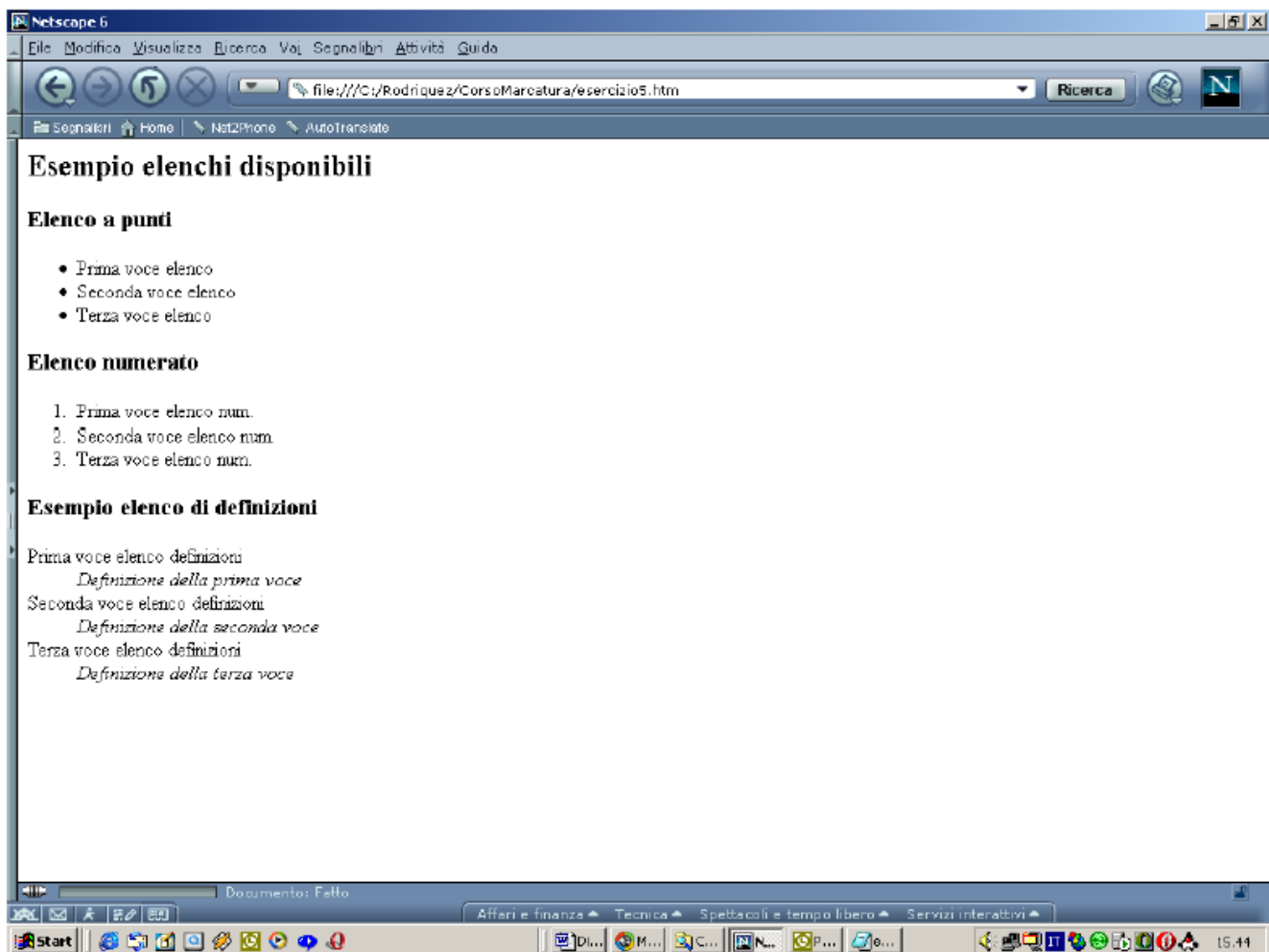
`<dd><i>Definizione della prima voce</i></dd>`

`<dt>Seconda voce elenco definizioni</dt>`

`<dd><i>Definizione della seconda voce</i></dd>`

```
<dt>Terza voce elenco definizioni</dt>
<dd><i>Definizione della terza voce</i></dd>
</dl>
</body>
</html>
```

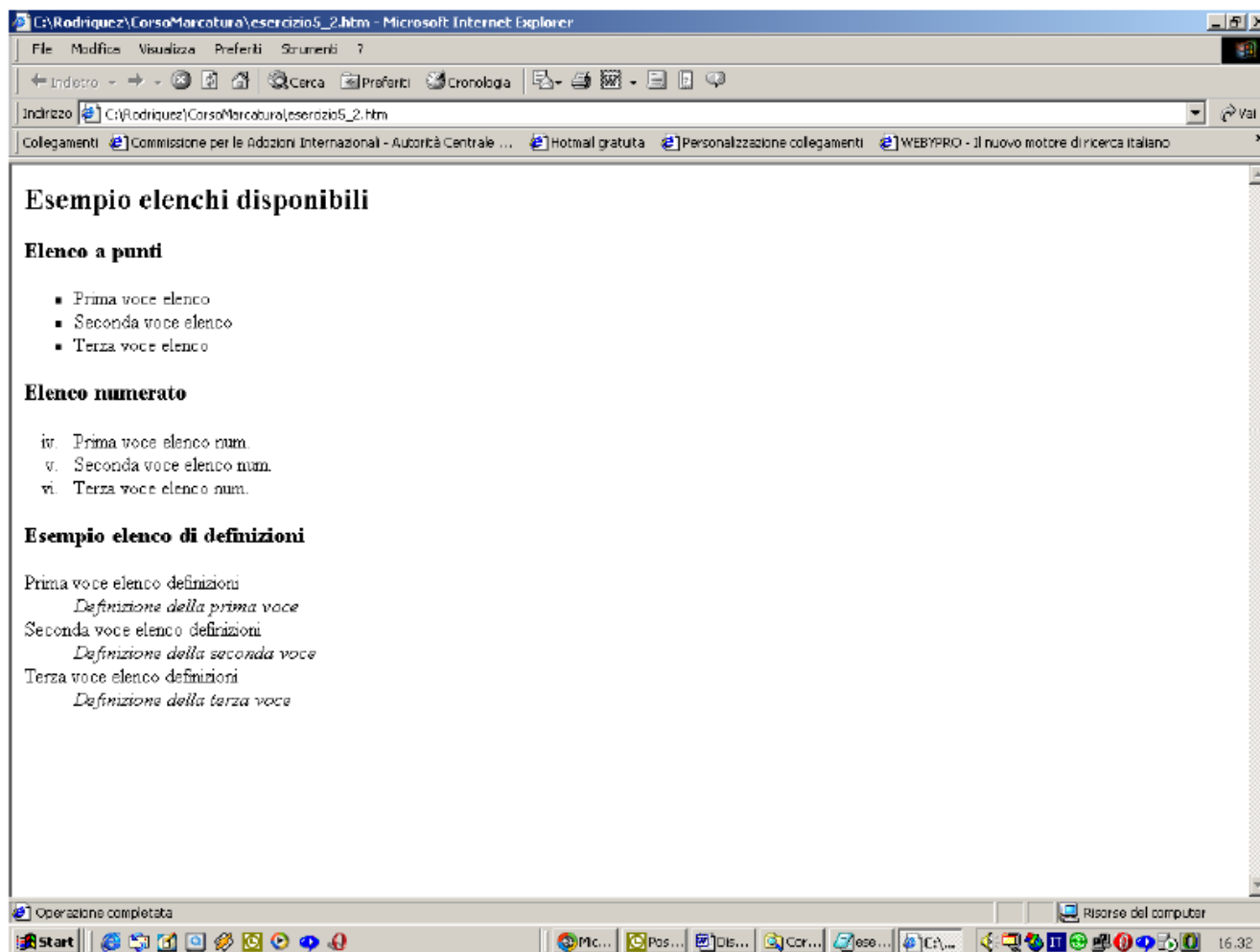
Il codice genera il seguente risultato:



2.13.1 Esercizio con variazione di attributi

```
<html>
<body>
<h2>Esempio elenchi disponibili</h2>
```

```
<h3>Elenco a punti</h3>
<ul type="square">
  <li>Prima voce elenco</li>
  <li>Seconda voce elenco</li>
  <li>Terza voce elenco</li>
</ul>
<h3>Elenco numerato</h3>
<ol type="i" start="4">
  <li>Prima voce elenco num.</li>
  <li>Seconda voce elenco num.</li>
  <li>Terza voce elenco num.</li>
</ol>
<h3>Esempio elenco di definizioni</h3>
<dl>
  <dt>Prima voce elenco definizioni</dt>
  <dd><i>Definizione della prima voce</i></dd>
  <dt>Seconda voce elenco definizioni</dt>
  <dd><i>Definizione della seconda voce</i></dd>
  <dt>Terza voce elenco definizioni</dt>
  <dd><i>Definizione della terza voce</i></dd>
</dl>
</body>
</html>
```



2.14 Tabelle

Una tabella può essere definita una griglia (insieme di celle), formata da righe e colonne, che contiene informazioni. Ciascuna cella può contenere testo, righe, immagini, elenchi, form, e anche altre tabelle (tabelle innestate). Le tabelle sono estremamente utili per disporre i dati sulla pagina in maniera ordinata e sistematica. A tal fine sono spesso usate per mostrare informazioni disposte in formato tabellare senza visualizzare la corrispondente griglia bordata. Un esempio può essere dato dal formato simile alle colonne giornalistiche, nel quale il testo appare incolonnato e privo di bordi. Una tabella può essere usata anche per contenere parti di un'immagine completa per ridurre il tempo di download.

Sintassi:

```
<table [align="center"|"left"|"right"] [border [= "n"]]
[cellpadding="n"] [width="nn%"] [cellspacing="n"]> ...
</table>
```

Gli elementi *table* contengono la tabella. Tutto ciò che è compreso tra questi tag fa parte della tabella.

Si noti che gli attributi di questo tag si riferiscono all'intera tabella e non alle singole celle.

L'attributo *border* esprime l'ampiezza del bordo in pixel. Se non è specificato la tabella risulterà priva di bordo.

Relativamente alla spaziatura delle tabelle esistono gli attributi *cellpadding* e *cellspacing*. L'attributo *cellpadding*, espresso in pixel, può essere usato per specificare lo spazio tra il contenuto della cella e il bordo interno (a valori numerici crescenti dell'attributo aumenta lo spazio tra celle e bordi).

L'attributo *cellspacing*, espresso in pixel, specifica invece lo spazio vuoto tra le celle. L'attributo *width* determina lo spazio sullo schermo destinato alla tabella. Il valore dell'attributo può essere espresso in pixel o come un valore percentuale dello spazio disponibile tra i margini sinistro e destro. Usando un valore percentuale lo spazio è variato proporzionalmente nel caso in cui l'utente modifichi manualmente la dimensione della finestra del browser.

```
<tr [align="left"|"center"|"right"]  
[valign="top"|"center"|"bottom"] > ... </tr>
```

L'elemento *tr* (table row) contiene ogni riga della tabella.

```
<td >contenuto1, contenuto2,..., contenutoN</td>
```

L'elemento *td* (table data) indica il contenuto di ciascuna cella

Nel caso in cui il contenuto di determinate celle sia vuoto è probabile che il browser non visualizzi correttamente la cella, producendo sull'intera tabella una cattiva rappresentazione visiva. Per evitare ciò è preferibile, nelle celle vuote, usare sempre il carattere speciale ` ` che sta ad indicare il carattere "spazio".

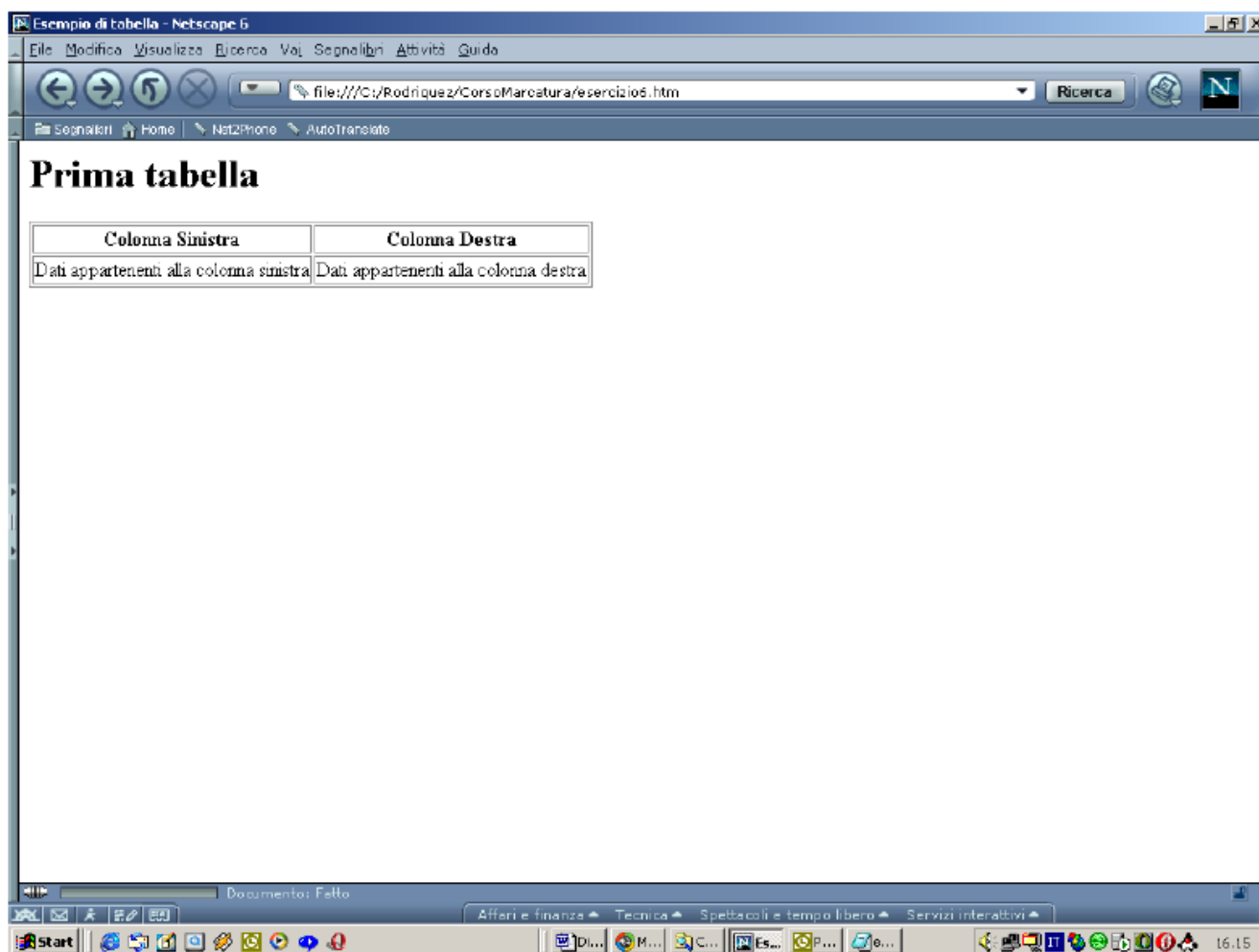
Le intestazioni delle tabelle sono specificate con il tag `<th>`.

L'impostazione di default del browser centra l'elemento *th* (intestazioni) e allinea a sinistra gli elementi *td* (celle). L'allineamento di default può essere modificato con i valori "left", "center", "right". Per centrare il contenuto in verticale è disponibile l'attributo *valign* con i possibili valori "top", "middle", "bottom", utilizzabile sia per le righe complete sia per singole celle. Il browser per default usa il valore "middle" per l'intestazione e il valore "top" per il contenuto di ogni singola cella.

Esercizio:

```
<html>
  <head>
    <title>Esempio di tabella</title>
  </head>
  <body>
    <h1>Prima tabella</h1>
    <table border="1">
      <tr>
        <th>Colonna Sinistra</th>
        <th>Colonna Destra</th>
      </tr>
      <tr>
        <td>Dati appartenenti alla colonna sinistra</td>
        <td>Dati appartenenti alla colonna destra</td>
      </tr>
    </table>
  </body>
</html>
```

Il codice genera il seguente risultato:



Sintassi

```
<th [align="left"|"center"|"right"]
[ valign="top"|"center"|"bottom"]
[ nowrap] [ colspan="n" ] [ rowspan="n" ] > ... </th>
```

L'elemento *th* è usato per le celle che contengono l'intestazione di tabella.

Il tag `<br>` può essere usato per forzare l'"a capo" del testo nelle celle.

Gli attributi *colspan* e *rowspan* consentono di espandere lo spazio destinato ad una cella, rispettivamente, in orizzontale e in verticale

```
<td [align="left"|"center"|"right"]
[ valign="top"|"center"|"bottom"]
[ nowrap] [ colspan="n" ] [ rowspan="n" ] > ... </td>
```

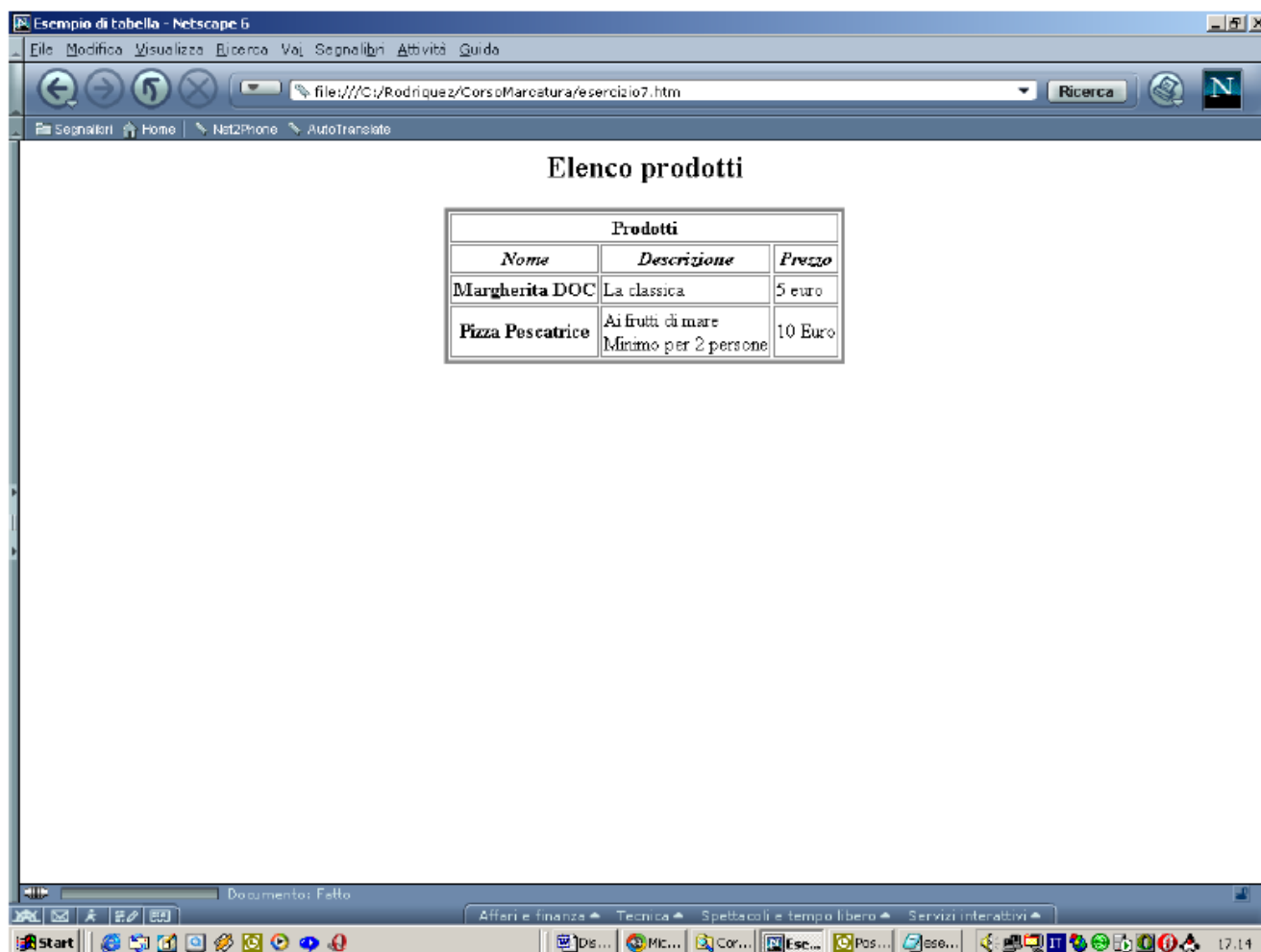
Il tag *td* si riferisce al contenuto delle celle

Esercizio

```

<html>
  <head>
    <title>Esempio di tabella</title>
  </head>
  <body>
    <h2 align="center">Elenco prodotti</h2>
    <table border="3" align="center">
      <tr>
        <th colspan="3" align="center">Prodotti </th>
      </tr>
      <tr>
        <th><i>Nome</i></th>
        <th><i>Descrizione</i></th>
        <th><i>Prezzo</i></th>
      </tr>
      <tr>
        <th>Margherita DOC</th>
        <td>La classica</td>
        <td>5 euro</td>
      </tr>
      <tr>
        <th>Pizza Pescatrice</th>
        <td>Ai frutti di mare<br>Minimo per 2 persone<br></td>
        <td>10 Euro</td>
      </tr>
    </table>
  </body>
</html>

```



2.15 Caratteri speciali

Dal momento che in HTML alcuni caratteri rivestono un particolare significato (es. “<”), la visualizzazione di tali caratteri richiede l'utilizzo di *entità o caratteri speciali*.

Il carattere “&” indica l'inizio di un carattere speciale e il carattere “;” ne indica la fine. Al centro è contenuto il nome dell'entità oppure il carattere “#” ed il numero di un'entità. Ad esempio la visualizzazione del carattere “<” è effettuata scrivendo < oppure <. La differenza tra i due modi illustrati è che il primo, pur essendo più facile da usare in quanto l'uso di nomi e abbreviazioni è più agevole rispetto all'utilizzo dei numeri, non è supportato dai browser meno recenti mentre il secondo è supportato dai browser in maniera pressoché completa.

Di seguito è riportato un elenco dei principali caratteri speciali espressi con il nome dell'entità:

 (non-breaking space; spazio)

< (minore)

> (maggiore)

& (&)

" (virgolette)

­ (trattino)

© Copyright

™ Trademark

2.16 Immagini

Il tag per inserire un'immagine è: `<img>`. Si tratta di un tag vuoto (empty) ovvero non richiede tag di chiusura e contiene solo attributi.

Sintassi:

```

```

Il valore dell'attributo *src* (*src* sta per *source*) contiene l'URL del file immagine che si vuole visualizzare (URL è un generico indirizzo internet o un'immagine presente sul proprio computer; nei prossimi paragrafi verrà spiegato in modo più approfondito). La maggior parte dei browser riconosce le immagini in formato JPEG , GIF e PNG. Questi tre formati supportano il “progressive rendering” (l'immagine scaricata viene rifinita progressivamente).

Gli attributi *height* e *width*, espressi in pixel, impostano lo spazio sulla pagina da destinare all'immagine. Non sono obbligatori ma se valorizzati contribuiscono alla riduzione dei tempi di creazione della pagina web (i documenti HTML arrivano sul browser “a pezzi” e sono posizionati quando arrivano anche le sezioni circostanti). Inoltre il testo di commento dell'immagine (attributo *alt*) sarà subito disponibile. L'attributo *alt* (Alt sta per alternate -text-) contiene del testo alternativo utilizzato per descrivere l'immagine. Il browser visualizza la stringa di caratteri quando il mouse è posizionato sull'immagine o se l'immagine non è caricata dal browser (ad es. perché l'utente ha inibito sul browser la visualizzazione delle immagini).

Si noti che l'uso dell'attributo *alt* è particolarmente importante per consentire ai programmi (in genere utilizzati da persone disabili) di riprodurre acusticamente il testo delle pagine web o nel caso che l'utente utilizzi un browser di tipo testuale. Per questi motivi ogni immagine dovrebbe sempre includere un attributo *alt* coerentemente valorizzato.

Va inoltre ricordato che i tempi occorrenti per lo scaricamento delle immagini sul browser suggeriscono di ricorrere alle immagini con moderazione.

L'attributo *usemap* è usato per definire una mappa immagine suddivisa in aree cliccabili, ad es. nel caso di cartine geografiche cliccabili su specifiche regioni.

```
<body background="URL"> ... </body>
```

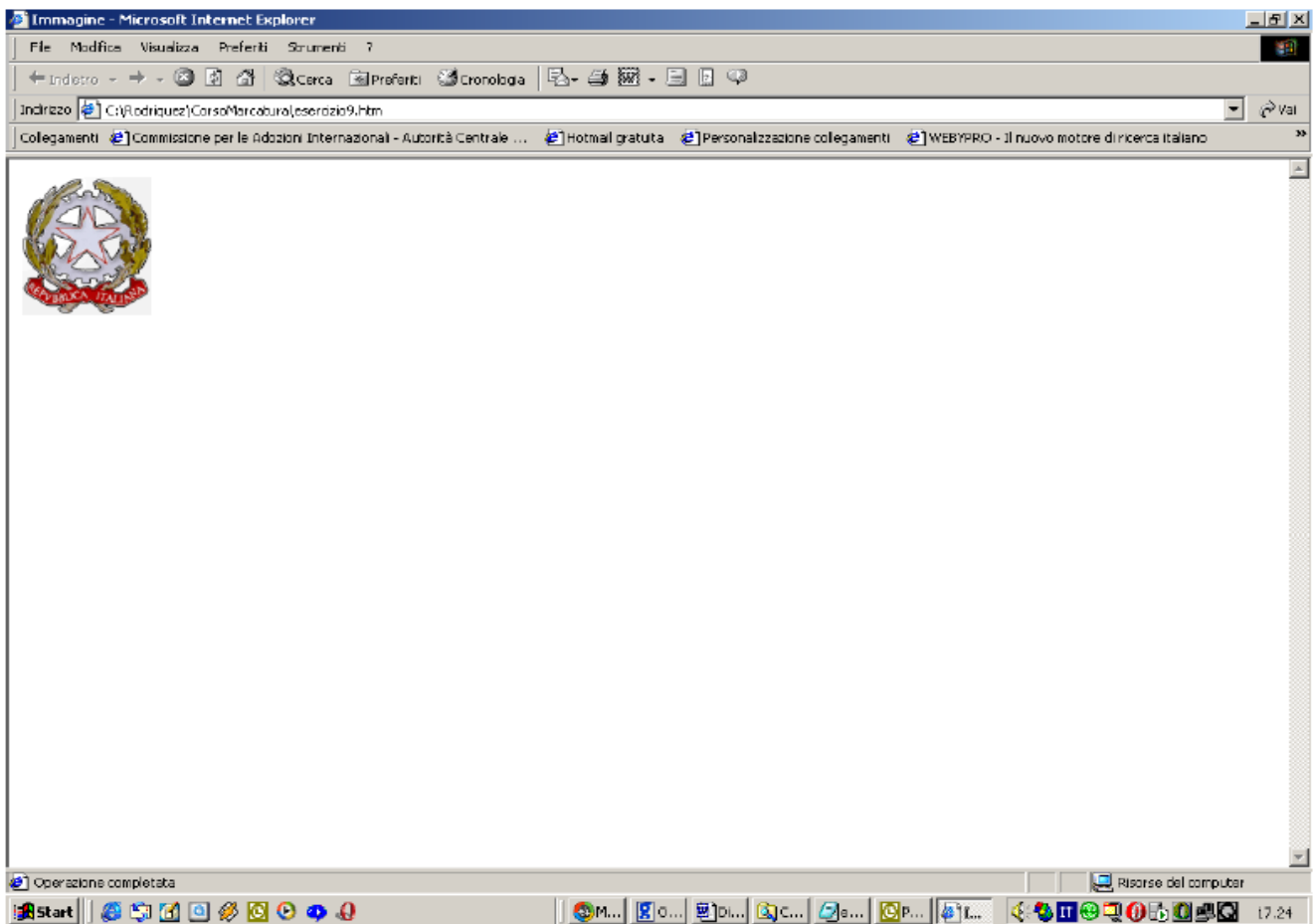
Questo tag imposta l'immagine da usare come sfondo della pagina, che a partire dall'angolo superiore sinistro della finestra è ripetuta automaticamente fino a riempire la pagina. Tale processo è definito *tiling*.

Trattando le immagini di sfondo non deve essere dimenticato che queste interferiscono sempre con il testo che si trova in primo piano. Ogni pagina Web che include immagini di sfondo deve essere pertanto attentamente verificata per impedire che la lettura del testo risulti in qualche misura compromessa.

Esercizio

```
<html>
  <head>
    <!--Esempio immagine -->
    <title>Immagine</title>
  </head>
  <body bgcolor="white">
    
  </body>
</html>_
```

Il codice genera il seguente risultato:



Si noti la differenza tra gli attributi *bgcolor* e *background* del tag `<body>`.

L'attributo `bgcolor` imposta il colore di sfondo mentre l'attributo `background` imposta un'immagine di sfondo. L'attributo `background` è valorizzato tramite l'URL dell'immagine da visualizzare. Si tenga presente che nel caso in cui l'immagine è di dimensioni minori rispetto alla finestra del browser sarà automaticamente ripetuta sino a riempire l'intera finestra.