LEZIONE 01

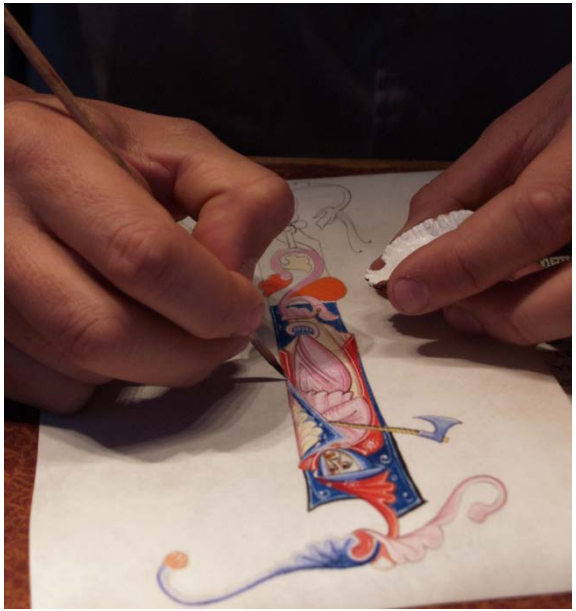
bruno@magicbusmultimedia.it

https://www.facebook.com/bruno.migliaretti

INTRODUZIONE
RAPPRESENTAZIONE DIGITALE DELLE INFORMAZIONI

PRIMA DI GUTEMBERG

AMANUENSI



COME

- Ogni pagina (testo e figure) viene copiata a mano

PRO

- Assoluta libertà creativa: ogni copia è unica

CONTRO

- Alti costi: il lavoro di ogni amanuense produce una sola copia
- Possibilità di errori umani

PRIMA DI GUTEMBERG

XILOGRAFIA

COME

- Ogni pagina viene ancora copiata a mano, ma incidendo una matrice di legno inversa.

PRO

- Una volta realizzata la matrice il processo di copia è automatico.

CONTRO

- Ogni matrice ha la vita di una pagina. Terminata la stampa non si può riutilizzare nulla.

DOPO GUTEMBERG

CARATTERI COMPONIBILI

COME

- La matrice della pagina viene realizzata affiancando caratteri di piombo preesistenti.

PRO

- I caratteri di piombo possono riutilizzati molte volte.

CONTRO

- Viene persa una parte di espressività artistica..

IL PROCESSO

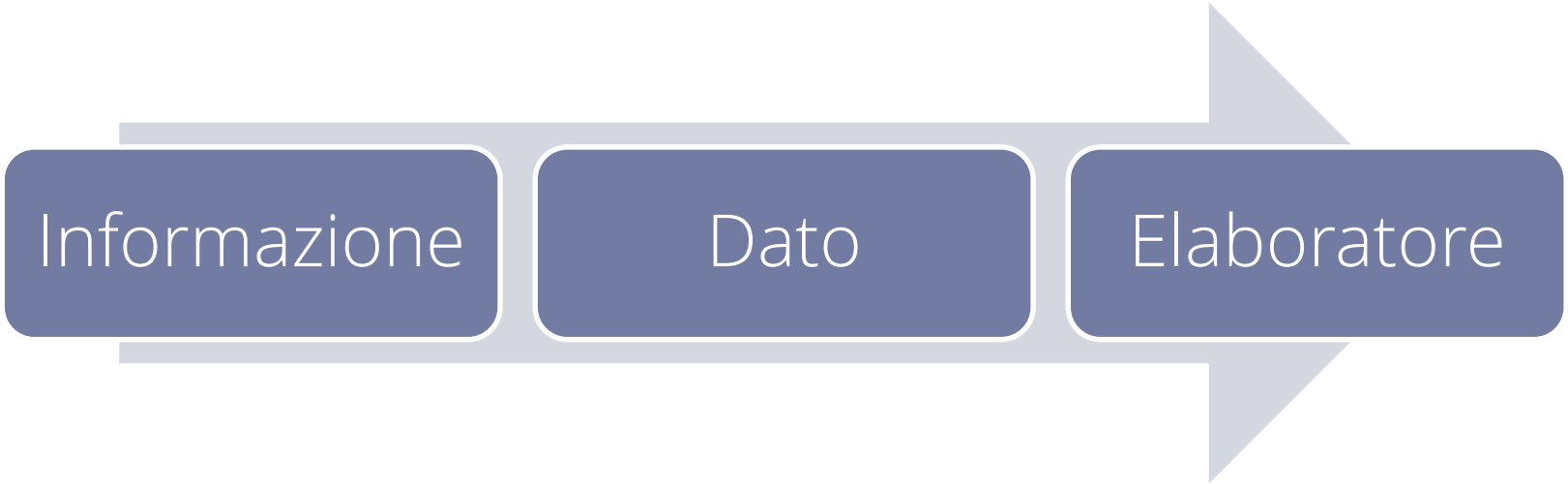
- Ogni pagina viene scomposta in singole unità informative (i caratteri)
- Ad ogni carattere viene sostituita la versione che la macchina di stampa è in grado di elaborare (il carattere componibile di piombo)
- La pagina può essere data in pasto alla macchina che può stamparne migliaia di copie.

IL PROCESSO

- Informatica è la disciplina che studia l'elaborazione automatica di informazioni

DEFINIZIONE DI INFORMATICA

- Informatica è la disciplina che studia l'elaborazione automatica di informazioni



CODIFICA

- *Codifica è* l'operazione con cui rendiamo disponibili le informazione per l'elaborazione automatica. Ogni informazione viene tradotta in un sistema simbolico su cui l'elaboratore può operare.

CODIFICA

- A secondo del tipo di informazione da trattare di solito ho regole di codifica diverse.
- I caratteri componibili sono ottimi per stampare i testi ma non mi servono a nulla per stampare immagini o disegni.
- Per inserire disegni in una pagina stampata con il metodo di Gutemberg posso creare delle incisioni da affiancare ai caratteri componibili.

PROGRAMMABILITÀ

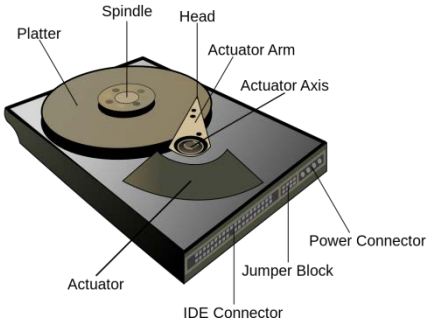
- Un torchio da stampa è una macchina estremamente semplice e specializzata in grado di svolgere un unico compito.
- Ci sono poi macchine che anche se dedicate ad un compito specifico, lo possono svolgere in maniera diversa a seconda dei programmi che eseguono (la macchine industriali a controllo numerico, ma anche le macchine da cucire da famiglia).
- Infine ci sono i computer che, nati come macchine per eseguire calcoli, hanno (soprattutto nelle varie versioni personal) perso ogni specificità e sono in grado, eseguendo i programmi adatti, di elaborare qualsiasi tipo di informazione.

CODIFICA E COMPUTER

- Il computer è l'elaboratore programmabile più complesso e completo oggi esistente.
- Come succede per tutti gli elaboratori, le informazioni devono essere correttamente codificate perché il computer possa elaborarle.
- Il computer è una macchina elettronica e quindi deve usare dei dispositivi elettrici per elaborare le informazioni codificate.
- Come base del sistema di codifica si è deciso di usare dei micordispositivi in grado di avere due stati elettrici definiti che nell'elaborazione rappresentano le cifre **0** e **1**

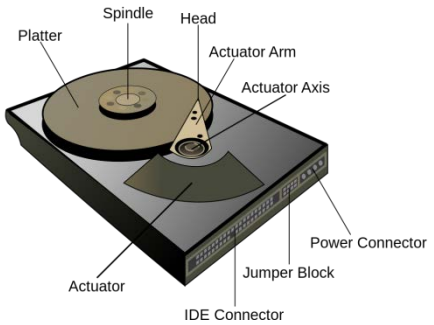
CODIFICA E COMPUTER

- Nelle schedine che vedete qui a destra sono collocate centinaia di miliardi di microdispositivi che possono avere due stati elettrici definiti (potete pensare ad acceso/spento o a carico/scarico)
- Sulla superficie magnetica di un hard-disk abbiamo decina di migliaia di miliardi di particelle in grado di avere due stati magnetici definiti.
- Nel primo caso abbiamo la cosiddetta **RAM**, memoria temporanea che il computer usa per elaborare le informazioni. Nel secondo caso tipo di **memoria di massa** dove i dati vengono memorizzati in maniera permanente.
- In entrambi i casi i due stati rappresentano le cifre **0** e **1**.



BIT BINARY DIGIT

- Letteralmente la parola bit significa cifra binaria
- In generale un bit e' una unità che può assumere un valore tra due possibili (normalmente si parla di 0 e 1)
- La rappresentazione fisica di un bit richiede un qualsiasi dispositivo in grado di trovarsi in uno di due possibili stati
- Interruttore (accesso/spento)
- Un condensatore (carico/scarico)
- Una bandiera (alzata/abbassata)
- Una particella magnetica (Nord/Sud)



La formula per convertire un nume

SISTEMA BINARIO

- Il sistema numerico binario è un sistema numerico posizionale in base 2. Esso utilizza solo due simboli, di solito, indicati con 0 e 1, invece delle dieci cifre utilizzate dal sistema numerico decimale. I numeri espressi nel sistema numerico binario sono chiamati numeri binari.
- Un numero binario è una sequenza di cifre binarie (dette bit). Il valore della cifra nella posizione **n** (contando da destra verso sinistra iniziando da 0) si ottiene moltiplicando la cifra per $\mathbf{2^n}$, anziché per $\mathbf{10^n}$, come avviene nella numerazione decimale.

Binario	Esadecimale	Decimale
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

La formula per convertire un nume

SISTEMA ESADECIMALE

- Il sistema numerico esadecimale (spesso abbreviato come **esa** o **hex**) è un sistema numerico posizionale in base 16, cioè che utilizza 16 simboli invece dei 10 del sistema numerico decimale tradizionale. Per l'esadecimale si usano in genere simboli da **0** a **9** per le prime dieci cifre, e poi le lettere da **A** a **F** per le successive sei cifre, per un totale di 16 simboli.
- Il sistema esadecimale è molto usato in informatica, per la sua relazione diretta tra una cifra esadecimale e quattro cifre binarie.

Binario	Esadecimale	Decimale
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

1125

base 10

Posizione	Calcolo	Valore decimale
0	$10^0 * 5$	5
1	$10^1 * 2$	20
2	$10^2 * 1$	100
3	$10^3 * 1$	1000
Totale		1125

1101

base 2

Posizione	Calcolo	Valore decimale
0	$2^0 * 1$	1
1	$2^1 * 0$	0
2	$2^2 * 1$	4
3	$2^3 * 1$	8
Totale		13

A1F8

base 16

Posizione	Calcolo	Valore decimale
0	$16^0 * 8$	8
1	$16^1 * 15$	240
2	$16^2 * 1$	256
3	$16^3 * 10$	40960
Totale		41464

ESADECIMALE <> BINARIO

11111111 10001000 00000000

ARANCIONE

ESADECIMALE <> BINARIO

1111 1111 1000 1000 0000 0000

F F 8 8 0 0

16746496

CODIFICA

- La codifica e' l'operazione che consente trasformare le informazioni in dati numerici che calcolatori elettronici possono leggere ed elaborare.
- Un bit puo' assumere solo due valori (0 e 1)
- Per rappresentare insiemi costituiti da piu' di due stati/simboli si usano serie di bit.
- Una stringa di bit e' costituita da un certo numero di bit (normalmente 8 o multipli di 8) ed e' detta parola (*word*).

CODIFICA

- La codifica si articola in due fasi:
  - Divisione dell'informazioni in unità informative
  - Assegnazione ad ogni unità di un valore **NUMERICO** che la codifica
- La precisione con cui sarò in grado di rappresentare l'insieme delle informazioni che codifico dipenderà dal rapporto tra il numero delle unità informative diverse da codificare e numero dei codici di cui dispongo.
- Con *n* bit si possono rappresentare 2^n valori diversi e quindi si possono rappresentare 2^n informazioni diverse
- La lunghezza della parola, quindi, definisce quante informazioni possono essere codificate.

NUMERO DI COMBINAZIONI

parola di 8 cifre decimali

10^8 = 100.000.000 conbinazioni

da 0 a 99.999.999

parola di 8 cifre binarie

2^8 = 256 conbinazioni

da 0 a 11111111

CODIFICA ESATTA E CODIFICA APPROSSIMATA

CODIFICA

- Una codifica esatta a **n** bit è possibile solo quando l'insieme delle informazioni da codificare è finito e di dimensione inferiore o uguale al massimo del valore che posso rappresentare con una parola di una lunghezza **n**.
- I calcolatori sono oggetti finiti che elaborano e memorizzano un numero finito di bit.
- Se l'insieme da codificare ha una contiene un numero di informazioni maggiore di $\mathbf{2^n}$ se ne puo' dare solo una rappresentazione approssimata o parziale. Questa limitazione avviene in due modi:

➢ **Operazioni di limitazione**

➢ **Operazioni di partizionamento**

CODIFICA DEL TESTO

ILTESTO

- Un testo e' una sequenza di caratteri alfabetici, separatori e caratteri speciali
- Nella codifica ad ogni carattere viene assegnato un numero binario
- Più è lunga la parola che utilizzo per codificare il testo più saranno i caratteri che sarò in grado di rappresentare.

STANDARD ASCII

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

- La codifica ASCII prevede l'utilizzo di 128 caratteri diversi
- Ogni carattere è associato perciò ad un numero espresso da una parola di 7 bit (0-127)
- La codifica ASCII si è piuttosto limitata. Modellata sulle lingua inglese non comprende le lettere con modificatori quali accenti, dieresi, tilde ecc.

CODICE ASCII ESTESO

- Per superare le limitazioni del codice ASCII standard:
  - Utilizzo di 8 bit invece di sette (256 simboli invece di 128)
  - I primi 128 uguali alla versione standard
  - Gli altri 128 personalizzati per gruppi di lingue
- Per interpretare correttamente un testo sono necessarie due informazioni:
  - Codifica ASCII
  - Tabella che viene utilizzate per i simboli > 127

STANDARD **UNICODE**

- Lo standard ISO10646/Unicode si basa su una codifica a 32 bit che consente oltre due miliardi di possibili caratteri
- Così possono essere rappresentati tutti i caratteri attualmente usati nel mondo
- Lo spazio occupato dal testo quadruplica.

STANDARD **UTF**

UNIVERSAL CHARACTER SET TRANSFORMATION FORMAT

- Sono codifiche che usano un numero variabile di bit per definire un carattere e risparmiare così spazio
- In UTF 8, ad esempio, si usano 8 , 16 , 24, 32 bit a seconda dei caratteri da rappresentare. Il decodificatore decodifica il testo leggendo un byte (8 bit) alla volta. Se il byte è un codice inferiore a 128 il carattere corrisponde all'ASCII standard e viene definito da quell'unico byte, altrimenti il decodificatore controlla il byte successivo, se questo è inferiore 1100000 il carattere viene definito da due byte, altrimenti viene controllato il terzo byte e così via.

NUMERI INTERI

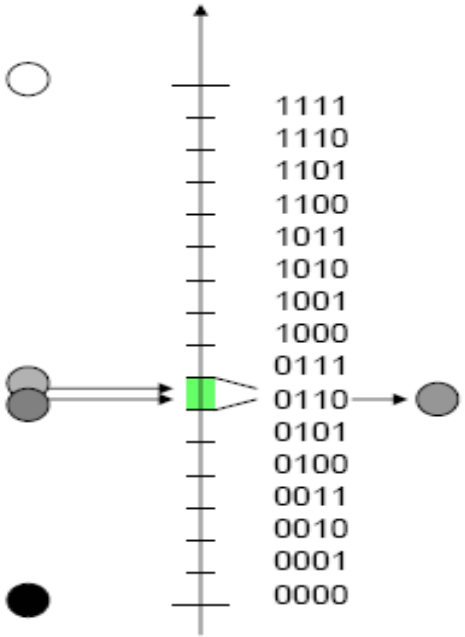
- I numeri interi sono un insieme discreto illimitato.
- Per poter essere codificati devono essere limitati.
- Sottoinsieme simmetrico rispetto allo 0.
- Si usa *1* bit per rappresentare il segno e i restanti $a = n-1$ per rappresentare il modulo.
- Il massimo numero rappresentabile è (in modulo) 2^{a-1}.
- Se il risultato di un'operazione è maggiore di $+2^{a-1}$ o minore di -2^{a-1} non potrà essere codificato.

CODIFICA DELLE IMMAGINI

CODIFICA DELLE IMMAGINI

- Le immagini sono informazioni continue in tre dimensioni: due spaziali ed una colorimetrica.
- Per codificarle occorre operare tre discretizzazioni.
  - Due discretizzazioni spaziali riducono l'immagine ad una matrice di punti colorati, detti pixel.
  - La terza discretizzazione limita l'insieme di colori che ogni pixel può assumere.

ESEMPIO: LIVELLI DI GRIGIO



- La codifica associa un unico codice ad un intervallo di livelli di grigio
- Tutti i livelli di grigio all'interno dell'intervallo vengono codificati allo stesso modo comportando una perdita di informazione
- Il livello di grigio originale non può essere ricostruito in maniera esatta dal codice binario

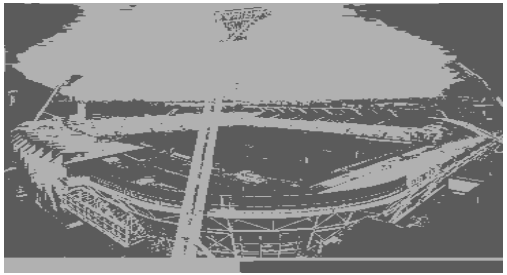
LIVELLI DI GRIGIO

8 bit

5 bit

4 bit

3 bit



1 bit

Immagini RGB

Nelle immagini a 24 bit tre byte definiscono i livelli dei colori fondamentali.

Rosso	Verde	Blu
10011110	10111101	11011110

Nelle immagini a 16 bit si usano 5 bit per definire i livelli dei colori fondamentali.

Rosso	Verde	Blu
10011	10111	11011

Nelle immagini a 32 bit tre byte definiscono i livelli dei colori fondamentali il quarto il livello di trasparenza (alpha) del pixel.

Rosso	Verde	Blu	Alpha
10011110	10111101	11011110	11111111

Il colore del pixel è definito direttamente da 2 o più byte che ne specificano la composizione in termini di Red, Green, Blue.

Compressione delle immagini

- Nella compressione delle immagini si usano sia tecniche di tipo *lossy* che tecniche di tipo *lossless*
- GIF, PNG → lossless
- JPEG → lossy
- La compressione di tipo *lossy* da ottimi risultati per quanto riguarda la dimensione del file prodotto apportando, in alcuni casi, perdite non visibili.
- La compressione di tipo lossless da buoni risultati per immagini a colori piatti (poco efficiente su immagini molto complesse e sfumate)

RIDUZIONE DEI COLORI

- La codifica è composta da due elementi distinti:
  - Una tabella di colori (palette) in cui vengono definiti fino ad un massimo di 256 colori
  - I punti (pixel) di cui è composta l'immagine il cui colore è definito da un byte (8 bit) che indica quale colore usare tra quelli definiti nella tabella.

Immagini a 256 colori

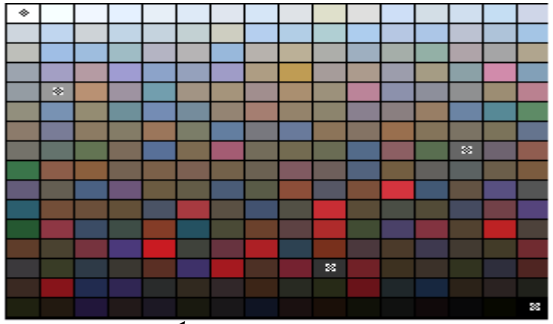
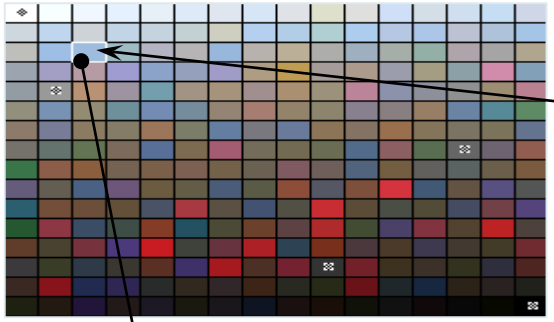


Tabella dei colori (palette) composta da 256 colori numerati da 0 a 255. Ogni colore viene definito per il suo contenuto di Rosso, Verde e Blu.

Immagine. Il colore di ogni punto (pixel) viene definito da un numero da 0 a 255 (8 bit). Viene utilizzato il colore definito nella palette all'indice corrispondente.

Immagini a 256 colori



Ogni colore viene definito nella palette specificando i tre colori fondamentali.

Indice	Rosso	Verde	Blu
00100010	10011110	10111101	11011110

Il colore del pixel è definito dal numero 00100010 (34 decimale) che rappresenta l'indice della palette.

Formato GIF
(Graphic Interchange Format)

- Max 256 colori → profondita' di colore 8 bit.
- Se l'immagine originale contiene un numero più elevato di colori è necessario effettuare una riduzione con conseguente perdita di qualità
- Integra una compressione di tipo LZW (ZIP)
- I colori sono memorizzati in una 'tavolozza', una tabella che associa un numero ad un certo valore di colore.
- Supporta il formato *interallacciato*
- consente anche di definire un colore come *trasparente*.

Formato JPEG/JPG
(Joint Photographic Expert Group)

- 24 bit = 16,8 milioni di colori.
- Tecnica di compressione basata di una codifica dell' immagine *percettiva* in cui viene distinta la luminosità dei pixel dal loro colore.
- Per ogni pixel viene codificata la componente della luminosita' mentre il colore viene codificato a blocchi di 4 pixel; (codificando il colore medio dei quattro)
- Per 4 pixel → 6 valori (4 di luminosità e 2 di colore) anziché 12 valori come in 24 bit (RGB)
- E' possibile scegliere il grado di compressione/qualita'
- Il formato JPEG progressivo emula l'interlacciamento

Formato PNG
(Portable Network Graphic)

- 8, 16, 24 bit = 256, 65536, 16.8 milioni di colori.
- Non è in grado di raggiungere l'efficienza del formato JPEG per quanto riguarda il fattore di compressione
- Supporta sia l'effetto trasparenza che l'interlacciamento
- Puo' incorporare del testo all'interno dell'immagine (come stringa) utile per classificarla e per fare ricerche sui contenuti

IMMAGINI VETTORIALI

- La grafica vettoriale scompone in gruppi logici di componenti (linee, cerchi, rettangoli, ecc.)
- Le forme vengono memorizzate in termini di coordinate e colori dei vari elementi geometrici che le compongono
- Durante la visualizzazione, coordinate e colori vengono utilizzati per ricreare l'immagine
- La grafica vettoriale e' comunemente usata nei disegni, disegni animati e nella grafica lineare in generale

OGGETTO LINEARE

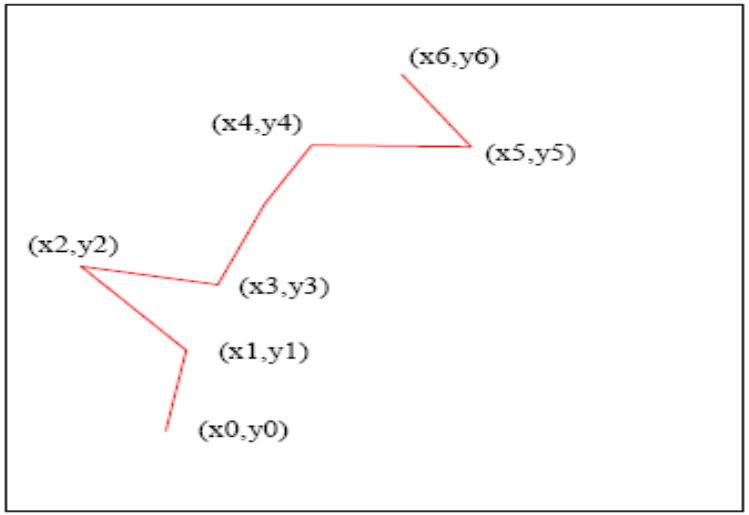
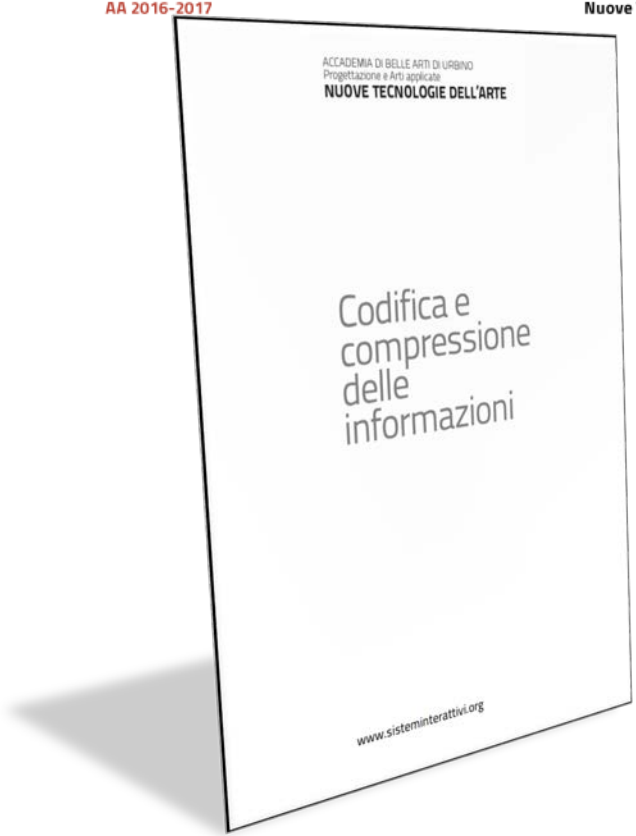


Immagine = x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,rosso



Scarica la dispensa:
http://www.sisteminterattivi.org//userfiles/file/docs/Dispense 2012/Rappresentazione digitale delle informazioni.pdf

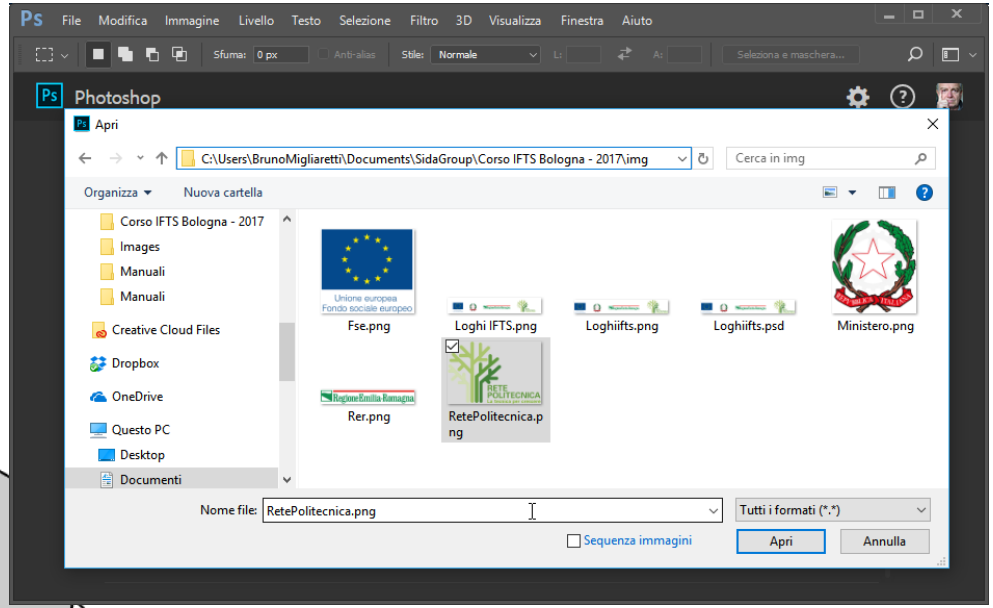
COME FUNZIONA INTERNET

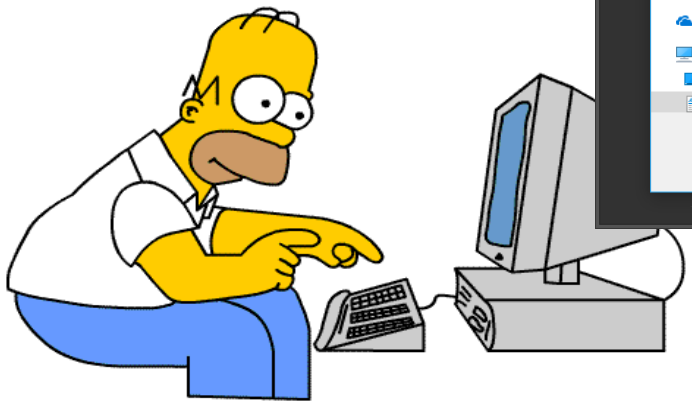
CARICAMENTO DI UN FILE

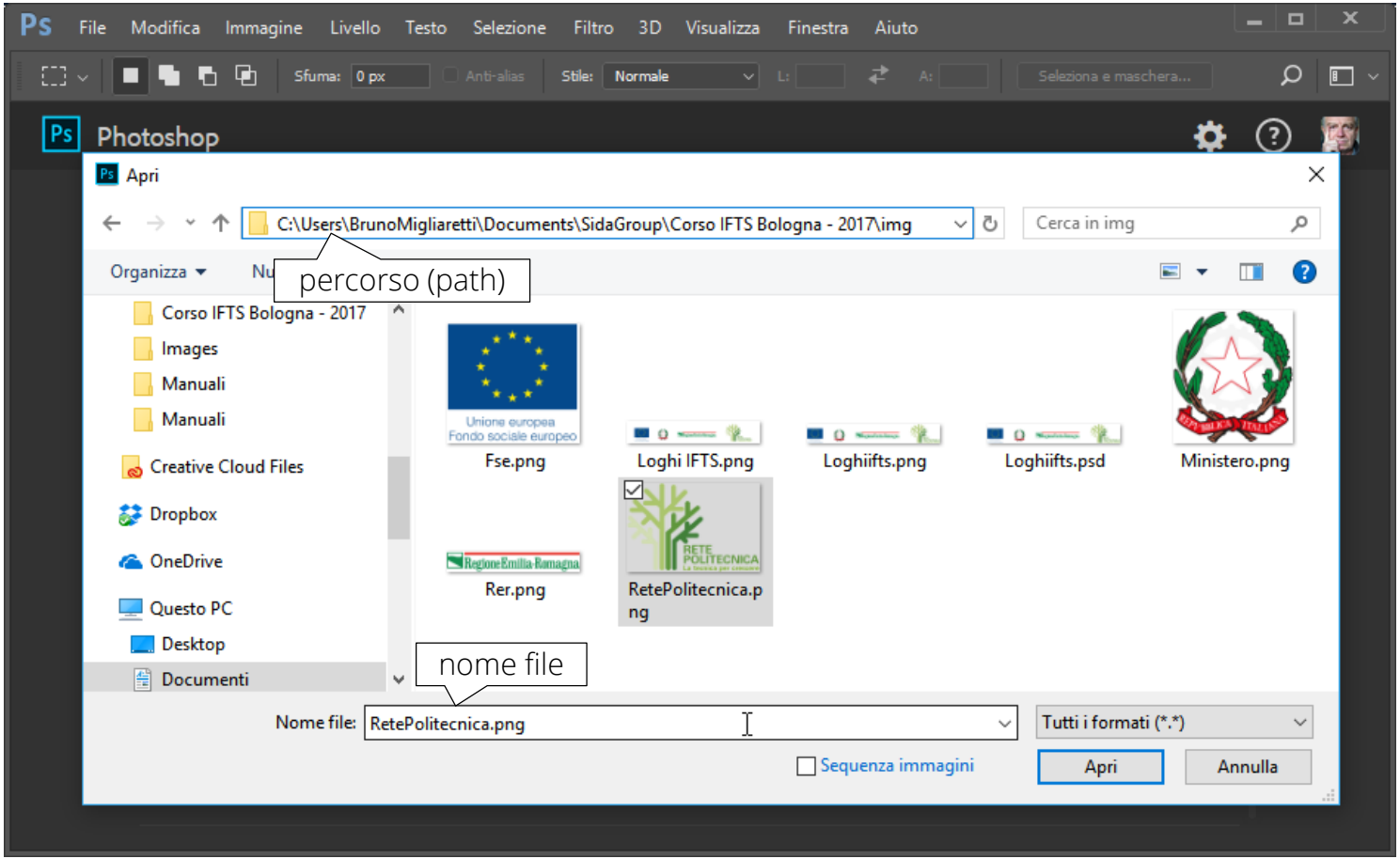
Se voglio caricare un'immagine
In Photoshop...

CARICAMENTO DI UN FILE

...la cerco sul disco del mio computer






File Modifica Immagine Livello Testo Selezione Filtro 3D Visualizza Finestra Aiuto
Sfuma: 0 px
Anti-alias
Stile: Normale
Seleziona e maschera...
Photoshop
Apri
C:\Users\BrunoMigliaretti\Documents\SidaGroup\Corso IFTS Bologna - 2017\img
Cerca in img
Organizza
percorso (path)
Corso IFTS Bologna - 2017
Images
Manuali
Manuali
Creative Cloud Files
Dropbox
OneDrive
Questo PC
Desktop
Documenti
Unione europea
Fondo sociale europeo
Fse.png
Loghi IFTS.png
Loghiifts.png
Loghiifts.psd
Ministero.png
RETE POLITECNICA
RetePolitecnica.png
Regione Emilia-Romagna
Rer.png
nome file
Nome file: RetePolitecnica.png
Tutti i formati (*.*)
Sequenza immagini
Apri
Annulla

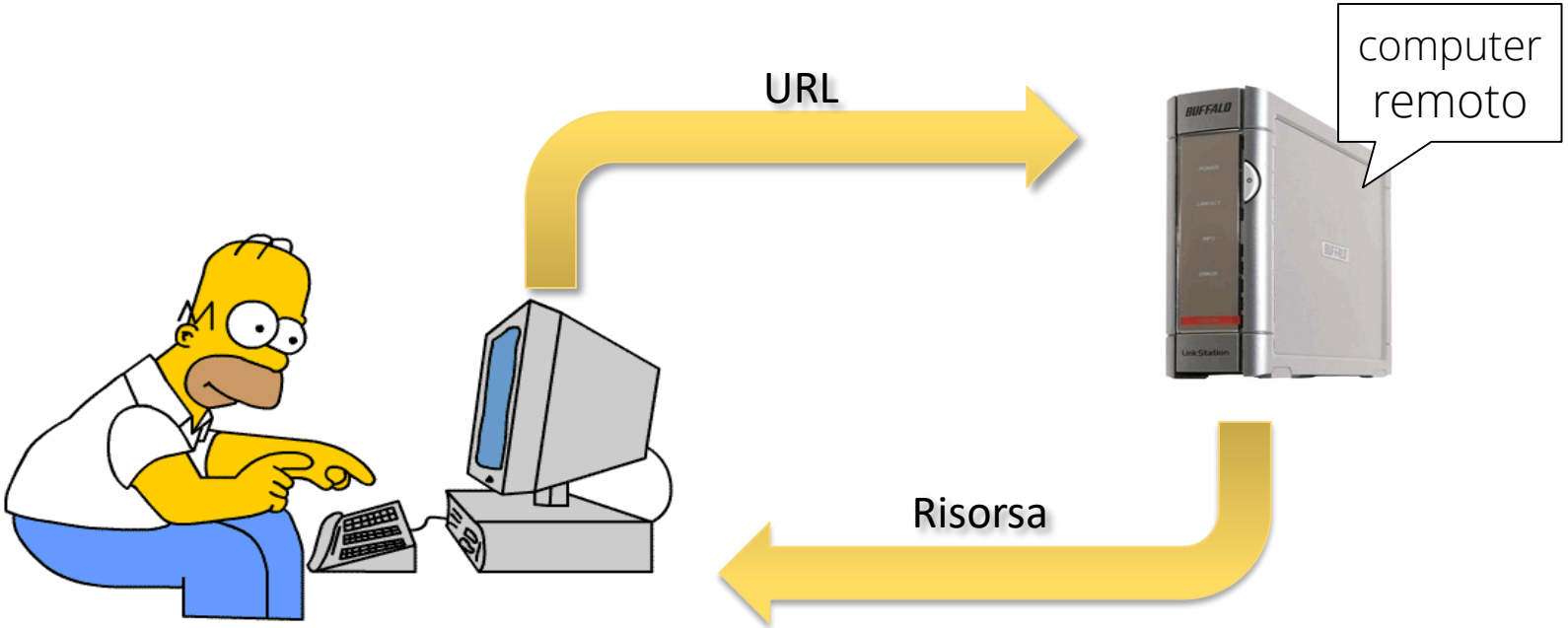
IDENTIFICATIVO UNIVOCO

- **`percorso + nome file`** costituiscono un modo univoco per identificare la risorsa sul mio pc
- è grazie a questa identificazione univoca che posso utilizzare i miei file

```
C:\Users\BrunoMigliaretti\Documents\SidaGroup\Corso IFTS Bologna – 2017\RetePolitecnica.png
```

- è un nome unico nel mio personal computer

CARICAMENTO RISORSA SU INTERNET



URL E URI

Uniform Resource Identifier (URI, acronimo più generico rispetto ad "URL") è una stringa che identifica univocamente una risorsa generica che può essere un indirizzo Web, un documento, un'immagine, un file, un servizio, un indirizzo di posta elettronica, ecc. L'URL è un URI che indica una risorsa internet.

Un Uniform Resource Locator o URL è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, come un documento o un'immagine.

STRUTTURA DELL'URL

- La tipica struttura di un URL è:

protocollo://indirizzo_risorsa

- In un URL **indirizzo_risorsa** è composto da informazioni aggiuntive, alcune obbligatorie, altre opzionali:

nomehost `[:porta][/percorso][?querystring]`

- Se non specifico alcun protocollo il browser aggiungerà all'URL **http** che è il protocollo di default per comunicazione su Internet.

ESEMPI DI URL SEMPLICI

protocollo

percorso

http://brunomigliaretti.com/userfiles/image/Bruno_256.png

host

http://rendera.herokuapp.com

protocollo

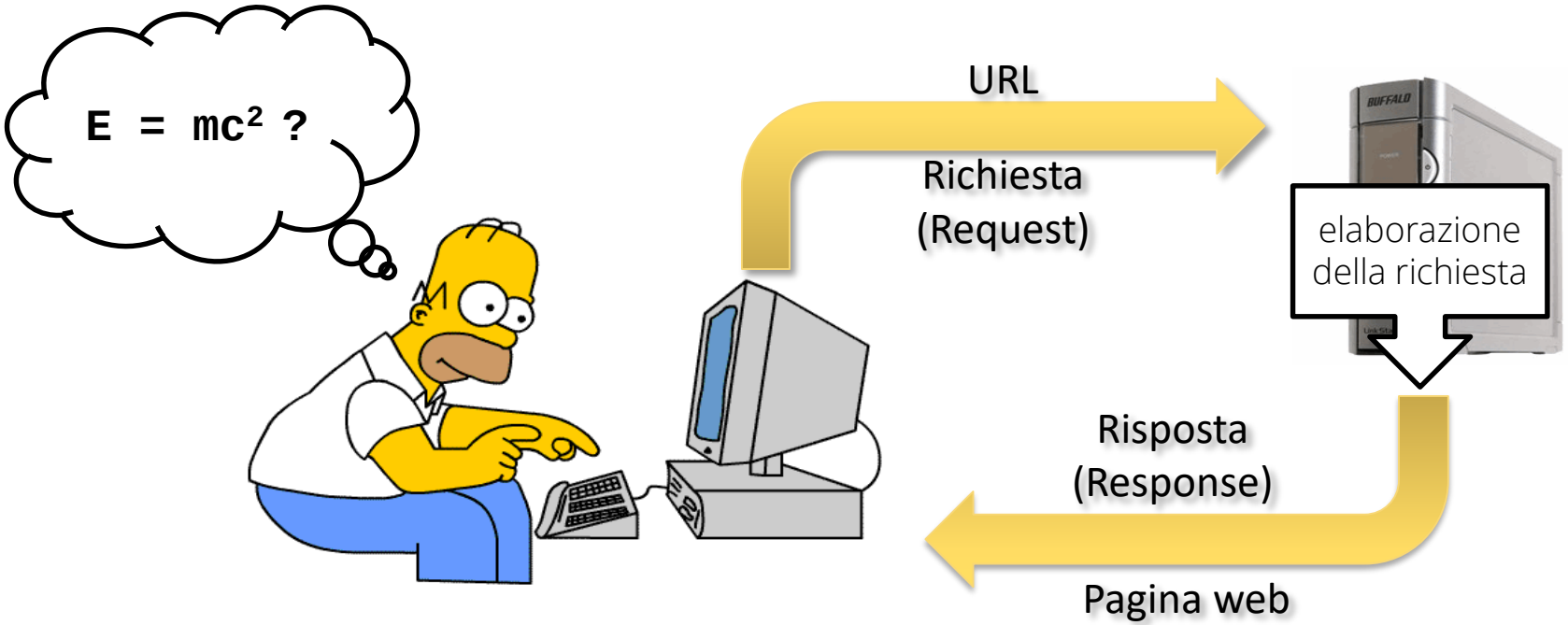
host

ESEMPI DI URL "SICURE"

ARCHITETTURA CLIENT-SERVER

- Server
  - Programma *in ascolto* su una porta (punto di accesso)
  - Quando arriva una richiesta da un client, il server analizza questa richiesta (eventualmente con l'aiuto di altri programmi), elabora una risposta (anche in questo caso, eventualmente con l'aiuto di altri programmi) e la invia al client.
  - Un server, generalmente, può servire più client contemporaneamente
- Client
  - Un client è un programma che si connette ad un server, fa una richiesta, aspetta una risposta e la utilizza per preparane un nuova richiesta al server o per fornire un output (un risultato) all'utente.

ARCHITETTURA CLIENT SERVER



WEB SERVER

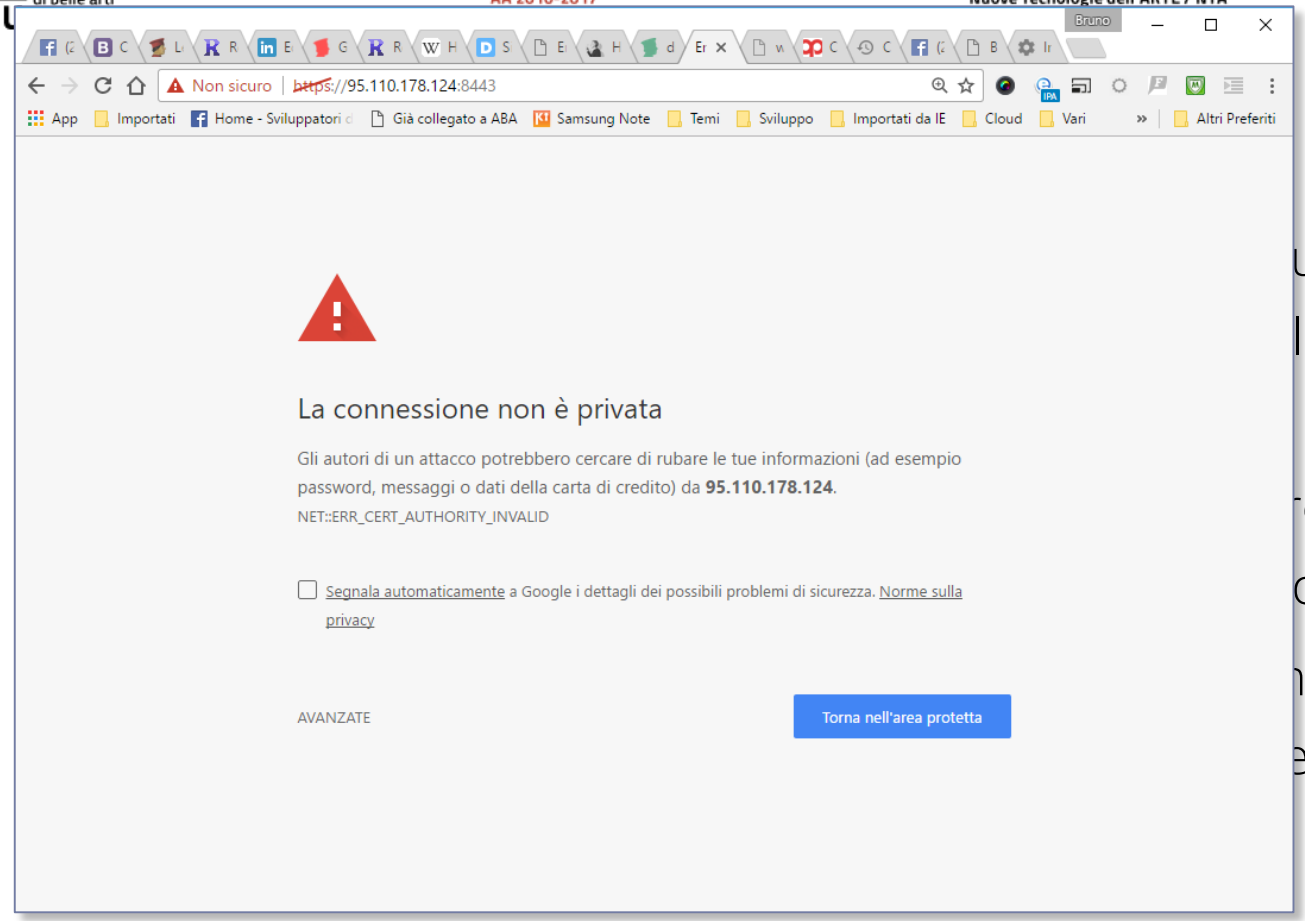
- Un **Web Server** è un programma (in esecuzione su un computer remoto o sul computer locale) in grado di gestire le richieste di trasferimento di pagine web ad un client, tipicamente un **web browser**. Il Web Server gestisce due flussi di informazioni:
  - le richieste che arrivano dai client (che vengono analizzate ed elaborate)
  - le risposte inviate ai client (che sono il risultato di queste elaborazioni)
- Per comunicare con i client un web server utilizza il protocollo HTTP (per default sulla porta 80) o il suo corrispondente sicuro HTTPS (sulla porta 443).

HTTP e HTTPS

- In informatica un protocollo di comunicazione è un insieme di regole che stabiliscono come si attua il trasferimento di dati tra entità.
- **HTTP** (HyperText Transfer Protocol) è il protocollo usato come principale sistema per la trasmissione d'informazioni sul web. È ottimizzato per il trasferimento delle pagine web.
- **HTTPS** (HyperText Transfer Protocol Secure) è l'implementazione di HTTP che prevede il trasferimento di dati in forma criptata, impedendo a terzi in ascolto di leggere i dati trasmessi.

HTTP e HTTPS

- Quando si collegano ad un sito con protocollo **HTTPS** tutti i maggiori browser richiedono al web server una certificazione sull'identità (e quindi anche sulla proprietà) del sito.
- Questo perché si presuppone che i dati trasferiti con protocollo sicuro sia sensibili (ad esempio accesso ad un conto corrente online).
- Se la certificazione non viene fornita (esistono organismi terzi che si occupano di verificare, su richiesta dei proprietari, l'autenticità dei siti web) il browser avvisa l'utente.



- utti i maggiori
  l'identità (e
- rotocollo sicuro
  online).
- ni terzi che si
  enticità dei siti

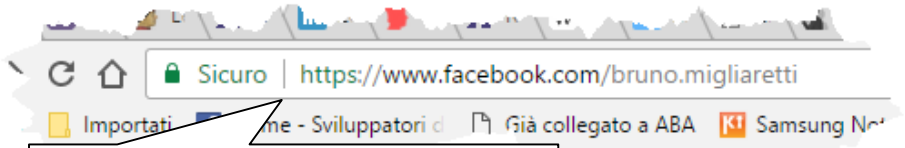
HTTP e HTTPS

- Il passo successivo dipende dal browser:
  - Opera blocca il sito
  - Mozilla Firefox consente di aggiungere un eccezione al controllo di sicurezza
  - Chrome, Microsoft Internet Explorer e Microsoft Edge consentono di bypassare l'avvertimento
- In linea di massima se l'utente è certo dell'identità del server (perché, ad esempio, ne è lui il proprietario) può bypassare l'avvertimento e collegarsi ugualmente.

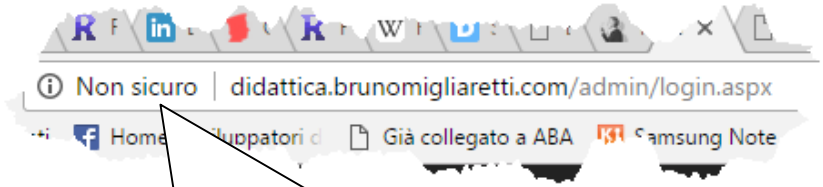
HTTP e HTTPS

- In generale il protocollo HTTP e considerato poco sicuro e tutti i siti che trattano dati sensibili (transizione economiche o privacy) sono passati negli ultimi anni a HTTPS.
- Chrome a partire da quest'anno ha iniziato a segnalare come **non sicuri** i siti che usano **http**, cominciando da quelli che richiedono password, ma annunciando che questa segnalazioni verrà estesa a tutti i siti che **non usano https**.

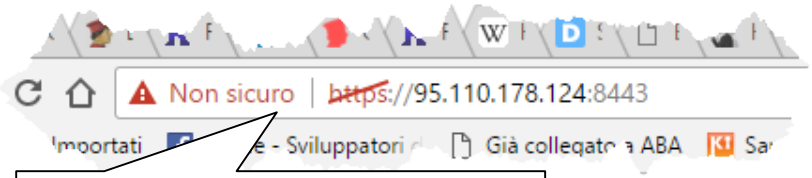
CHROME E LA SICUREZZA



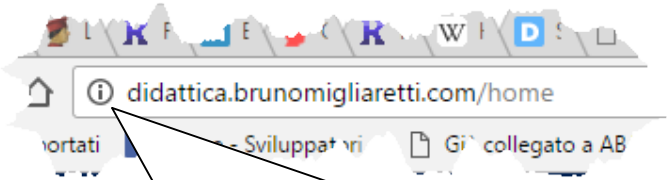
Il sito implementa **https** e la sua autenticità è certificata



Il sito non implementa **https** e richiede una password



Il sito implementa **https**, ma la sua autenticità non è certificata



Il sito non implementa **https.** Cliccando sull'icona si tra le informazioni sul sito l'utente legge l'invito a non inserire dati sensibili in quanto la connessione non è protetta

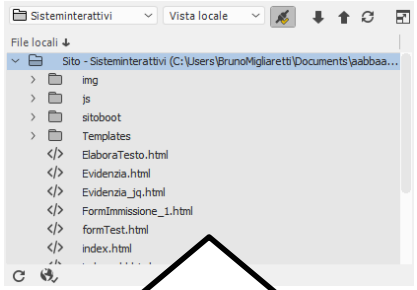
BROWSER

- Un Web browser è un **HTTP/HTTPS** client, cioè un programma, dotato di interfaccia grafica, che:
  - interagisce con un server web, richiedendone i servizi (per es. pagine Web)
  - riceve i dati dal server e li ricompone
  - visualizza le pagine Web (ipertesti), mostrandone il contenuto e interpretando correttamente i linguaggi che vengono utilizzati per descriverne i contenuti (HTML, CSS, Javascript)

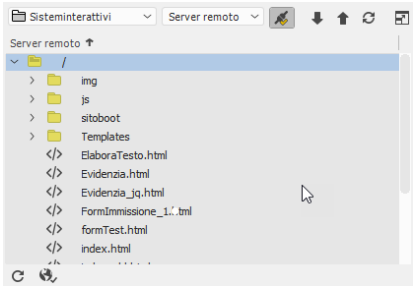
PAGINE STATICHE E DINAMICHE

- Quando **ci connettiamo ad una risorsa in rete, identificata da un URL:**
  - Nel caso più semplice l'indirizzo di una pagina scritta in HTML il cui **contenuto è fisso (STATICA);**
  - In altri casi, l'URL può contenere l'indirizzo di una **pagina "dinamica" (per esempio** scritta in ASP, PHP, o JSP) il cui **contenuto viene generato (selezionato, composto) al momento della richiesta;**

SITO SEMPLICE



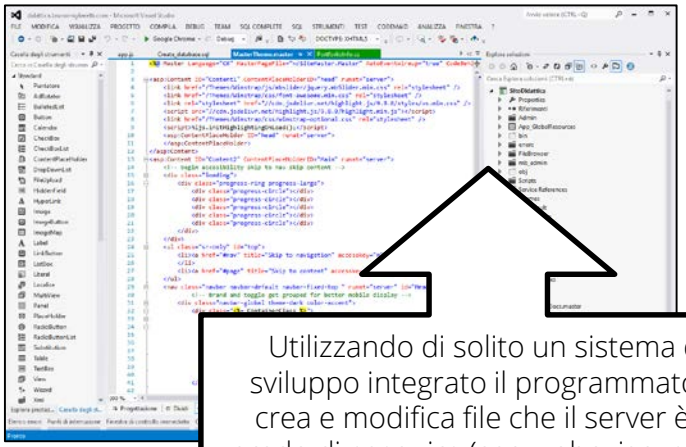
Sincronizzazione (**FTP**)



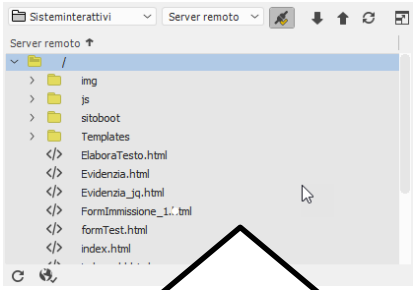
L'amministratore con un semplice editor di testi crea e modifica le pagina HTML. I fogli di stile CSS e i file Javascipt e li salva nel disco locale.

L'utente con un browser richiede le pagine web e le consulta.

APPLICAZIONE WEB

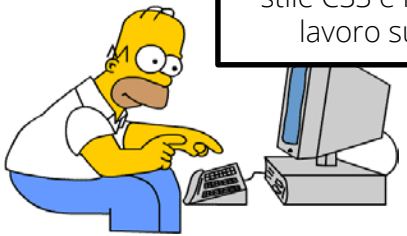


Sincronizzazione

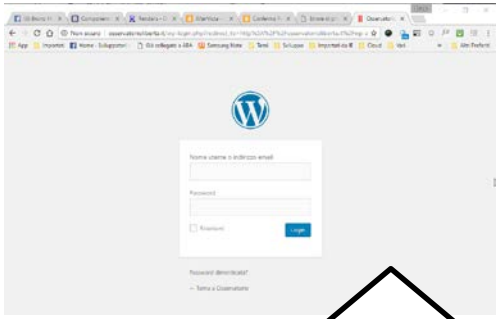


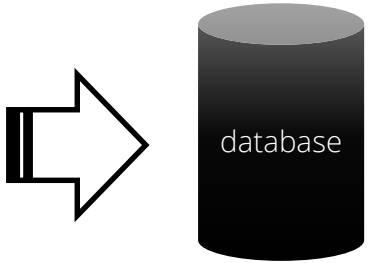
Utilizzando di solito un sistema di sviluppo integrato il programmatore crea e modifica file che il server è in grado di eseguire (aspx, php, jsp, ecc. a secondo dell'ambiente scelto) , i fogli di stile CSS e i file Javascipt e testa il suo lavoro su un server Web Locale

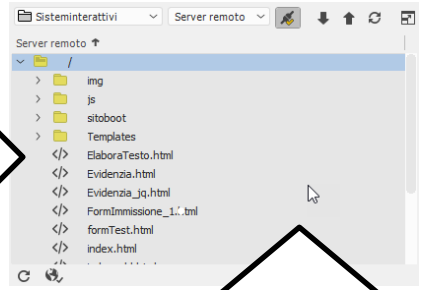
L'utente con un browser richiede le pagine web. Il server elabora le richieste e risponde restituendo pagine scritte in HTML.



CONTENT MANAGEMENT SYSTEM







L'amministratore ha accesso ad una applicazione web con la quale può modificare il database in cui sono memorizzati i contenuti del sito.

L'utente con un browser richiede le pagine web. Il server recupera i contenuti nel database e risponde restituendo pagine scritte in HTML.

LE APPLICAZIONI WEB

- Il passaggio di semplici documenti HTML tra il server e il client non permette lo sviluppo di applicazioni web complesse che coinvolgano una fase di elaborazione oltre che di passaggio di dati.
- Per questo motivo sono state sviluppate tecnologie che permettano una maggiore interazione dell'utente con il server web e una capacità di elaborazione sia del server che del client web.

COME FUNZIONA INTERNET

1. Le informazioni che ci arrivano da **Internet** ci arrivano in formato **pagina web**

COME FUNZIONA INTERNET

2. Come nella comunicazione tradizionale le informazioni sono organizzate semanticamente in **indici**, **titoli**, **sottotitoli**, **paragrafi**, **figure**, ecc., in modo che siano intellegibili.

COME FUNZIONA INTERNET

3. Questa formattazione semantica viene fatta utilizzando **HTML**, un linguaggio che contiene istruzioni che servono, appunto, ad organizzare i contenuti di una pagina web.

COME FUNZIONA INTERNET

4. La pagina web per essere fruita deve essere presentata in maniera piacevole e leggibile su vari tipi di schermo e di dispositivo. A tal fine è stato costruite un altro linguaggio che ha il compito di definire le regole di visualizzazione della varie parti di una pagina: **CSS**

COME FUNZIONA INTERNET

5. Infine le pagine web sono interattive:
   - Interattività ipertestuale immediata (HTML)
   - Interattività lato server (Invio richieste e ricevo risposte e risultati di elaborazioni)
   - Interattività lato client: utilizzo di programmi più o meno complessi scritti in **javascript** per rendere più semplice e produttiva l'attività di navigazione dell'utente.

HTML

HTML

- **HTML** è l'acronimo di **H**yper**T**ext **M**arkup **L**anguage ("Linguaggio a marcatori per gli Ipertesti").
- Non è un linguaggio di programmazione non ha, cioè, meccanismi che consentono di prendere delle decisioni ("in questa situazione fai questo, in quest'altra fai quest'altro"), e non è in grado di compiere delle iterazioni ("ripeti questa cosa, finché non succede questo"), né ha altri costrutti propri della programmazione.
- Si tratta invece di un linguaggio di contrassegno (o 'di marcatura'), che permette di articolare gli elementi di una pagina in blocchi le cui caratteristiche vengono definite attraverso degli appositi marcatori, detti "**tag**".

A COSA SERVE

- Comunica al browser di quali risorse ha bisogno per comporre la pagina web
- Definisce e articola il contenuto della pagina:
  - Struttura semantica della pagina
  - Elementi di formattazione
  - Componenti multimediali
  - Componenti funzionali

COME LO FA: I TAG

- I **tag** vanno inseriti tra parentesi uncinate: **<TAG>**
- La chiusura del tag viene indicata con una barra: **</TAG>**
- Il contenuto che il tag modifica va inserito tra l'apertura e la chiusura del tag medesimo:

  Questa ****parola**** è in grassetto.

- che nel rendering verrà reso:

  Questa **parola** è in grassetto.

- Alcuni tag non hanno (o possono non avere) contenuto (**empty tag**) . Ad esempio l'interruzione di linea la indico così:

  **
**

GLI ELEMENTI

- Ogni tag definisce un **elemento** (element) del documento HTML
- Esistono vari tipi di elementi:
  - Elementi visibili che definiscono componenti semantiche del testo (titoli, intestazioni, piè di pagina, paragrafi, ecc.)
  - Elementi visibili che definiscono componenti che consentono l'interazione con l'utente (link, campi che consentono l'immissione di testo, bottoni, ecc.)
  - Elementi visibili che servono a formattare parti di testo (grassetto, corsivo, ecc.)
  - Elementi visibili e non visibili che consentono l'inserimento di risorse multimediali (immagini, audio, video, ecc.)
  - Elementi non visibili che servono a caricare risorse funzionali necessarie alla pagina web (script, fogli di stile)
  - Elementi non visibili che racchiudono codice eseguibile o istruzioni di stile.

GLI ATTRIBUTI

- Le caratteristiche di un tag vengono determinate dagli attributi del tag. Ogni tag ha per i suoi attributi dei valori predefiniti che io posso modificare:

```
<tag attributo_1="valore1" attributo_2="valore2">contenuto</tag>
```

- Alcuni attributi sono generali, comuni a tutti i tag (id, class, ecc.), altri sono specifici:

```
<img width="20" height="20" src="miaImmagine.gif" alt="alt“ />
```

- Una caratteristica importante del codice HTML è che i tag possono essere annidati l'uno dentro l'altro.
- È quindi opportuno usare l'indentazione. Grazie ad essa il codice HTML risulta più leggibile.

HTML 5

STRUTTURA DELLA PAGINA

```
<!DOCTYPE html>
<html lang="it">
<head>
     <script src="mioscripr.js"></script>
     <link rel="stylesheet" href="styles.css">
     <meta charset="utf-8">
     <meta name="google" value="notranslate">
</head>
<body>
     <h1>Benvenuto!</h1>
     <p>Questo &egrave; il mondo di HMTL!</p>
</body>
</html>
```

PROLOGO

```
<!DOCTYPE html>
```

- In HTML 5 il prologo è la semplice dichiarazione del tipo di documento come **HTML**

ELEMENTO RADICE (ROOT)

```
<html lang="it">
</html>
```

- <HTML> è obbligatorio
- L'elemento <html> può assumere questi attributi:
  - **dir** Determina la direzione del testo
  - **lang** Specifica il linguaggio di base dell'elemento quando è interpretato come HTML

HEAD

```
<head>
        <script src="mioscripr.js"></script>
        <link rel="stylesheet" href="styles.css">
        <meta charset="utf-8">
        <meta name="google" value="notranslate">
</head>
```

- La sezione <head> contiene informazioni che non vengono direttamente visualizzate nella pagina:
  - <link> Contiene informazioni su documenti esterni collegati: fogli di stile, favicon, ecc.
  - <meta> Specifica informazioni di vario tipo sul documento.
  - <noscript> Usato per visualizzazioni alternative nei browser che non supportano gli script.
  - <object> Racchiude un oggetto.
  - <script> Contiene script di programmazione o carica uno script di programmazione esterno .
  - <style> Definisce le regole di formattazione per il documento corrente
  - <title> Specifica il titolo del documento che compare nella barra del titolo del browser

BODY

```
<body>
      <h1>Benvenuto!</h1>
      <p>Questo &egrave; il mondo di HMTL!</p>
</body>
```

- Il corpo del documento è la sezione in cui si sviluppa il contenuto. È racchiusa, come in HTML, tra i tag <body>...</body>.
- Gli elementi che possono comparire all'interno del corpo sono in genere suddivisi in due categorie:
  - **elementi blocco** ed gli elementi blocco sono quelli che definiscono la struttura del documento. Possono contenere altri elementi blocco, elementi inline o testo. Quando sono inseriti danno origine ad una nuova riga nel flusso del documento.
  - **elementi inline**: quando sono inseriti non danno origine a una nuova riga e possono contenere solo dati (essenzialmente testo) o altri elementi inline.

ELEMENTI DI BASE

TITOLI (HEADINGS)

- Gli heading (titoli) sono elementi blocco contrassegna dai tag da **<h1>** a **<h6>** e possono essere utilizzati per strutturare gerarchicamente un documento.
- Per default i browser rendono i titoli in grassetto con corpi crescenti da **<h6>** (più piccolo del testo normale) a **<h1>**.

```
<h1>Questo è un titolo di primo livello</h1>
<h2>Questo è un titolo di secondo livello</h2>
```

DIV

- L'elemento ** ** è un generico elemento blocco che racchiude una porzione di documento
- Non è prevista alcuna formattazione di default

```
<div>
        Testo o qualsiasi altro elemento
</div>
```

P

- L'elemento **<p>** è un elemento blocco che racchiude una paragrafo
- Normalmente i browser rendono l'elemento **<p>** assegnandogli un margine inferiore e la dimensione di carattere di default.
- La specifica prevede che l'elemento **<p>** non contenga elementi blocco

```
<p>
        Testo e elementi inline o inline-block
</p>
```

UL

- L'elemento **<ul>** rappresenta una lista non ordinata.
- Gli elementi della lista sono rappresentati da elementi **<li>**
- Per default una lista **<ul>** viene resa come lista puntata
- All'interno di un elemento **<li>** può essere inserito un elemento **<ul>** o **<ol>** per creare liste annidate a più livelli

```
<ul>
     <li>Elemento lista</li>
     <li>Elemento Lista
          <ul>
               <li> Elemeto Lista di secondo livello</li>
          </ul>
     </li>
     <li>Elemento lista</li>
     <li>Elemento lista</li>
</ul>
```

OL

- L'elemento **<ol>** rappresenta una lista ordinata.
- Gli elementi della lista sono rappresentati da elementi **<li>**
- Per default una lista **<ol>** viene resa come lista numerata
- All'interno di un elemento **<li>** può essere inserito un elemento **<ul>** o **<ol>** per creare liste annidate a più livelli

```
<ol>
     <li>Elemento lista</li>
     <li>Elemento Lista
          <ol>
               <li> Elemeto Lista di secondo livello</li>
          </ol>
     </li>
     <li>Elemento lista</li>
     <li>Elemento lista</li>
</ol>
```

ELEMENTI SEMANTICI

ELEMENTI SEMANTICI

- Gli elementi semantici sono quelli che ci aiutano a dare una struttura logica ad un documento
- Normalmente sono elementi di tipo blocco
- Nella maggior parte dei casi i browser non prevendono alcuna formattazione di default e vengono resi sullo schermo come testo normale

HEADER

- La specifica impone che questo elemento debba delimitare l'intestazione di una sezione del documento.
- Le intestazioni possono essere molteplici all'interno di un documento. Si pensi ad esempio blog: nell'home page ci sarà un'intestazione contenente il nome del blog, un menù di navigazione e qualche altra informazione. Sotto l'intestazione potrebbe esserci una sezione contenente gli articoli recenti, ognuna di queste dovrà avere anch'essa un'intestazione che potrebbe l'autore e la data di pubblicazione.
- Non è prevista lacuna formattazione di default

```
<header class="main-header">
   <div class="logo">Logo</div>
   <nav>
      <!-- Menu del sito -->
   </nav>
</header>
```

NAV

- L'elemento <nav> specifica una porzione di documento destinato alla navigazione per esempio la lista di un menu.

```
<nav>
  <ul class="navbar">
    <li><a href="#titoli">Titoli</a> </li>
    <li><a href="#paragrafo">Paragrafo</a> </li>
    <li> <a href="#liste">Liste</a> </li>
  </ul>
</nav>
```

SECTION

- L'elemento **<section>** rappresenta una sezione generica di un documento o applicazione.
- Individua un raggruppamento tematico di contenuti, ed in genere contiene un titolo introduttivo ad esempio le sezioni di una homepage o i capitoli di un libro.
- Se la sezione racchiusa è una notizia l'elemento più appropriato per tale impiego è il tag **<article>.**

```
<section>
        Testo o qualsiasi altro elemento
</section>
```

ARTICLE

- L'elemento **<article>** dev'essere utilizzato quando s'inseriscono informazioni indipendenti dal resto del documento o della sezione in cui è racchiuso.
- Può trattarsi ad esempio di un articolo di giornale o un post in un blog o un qualsiasi altro elemento.

```
<article>
  <header>
    <h1>Titolo dell'articolo</h1>
  </header>
  <!-- Contenuto dell'articolo -->
  <p>.....</p>
  <p>......</p>
</article>
```

ASIDE

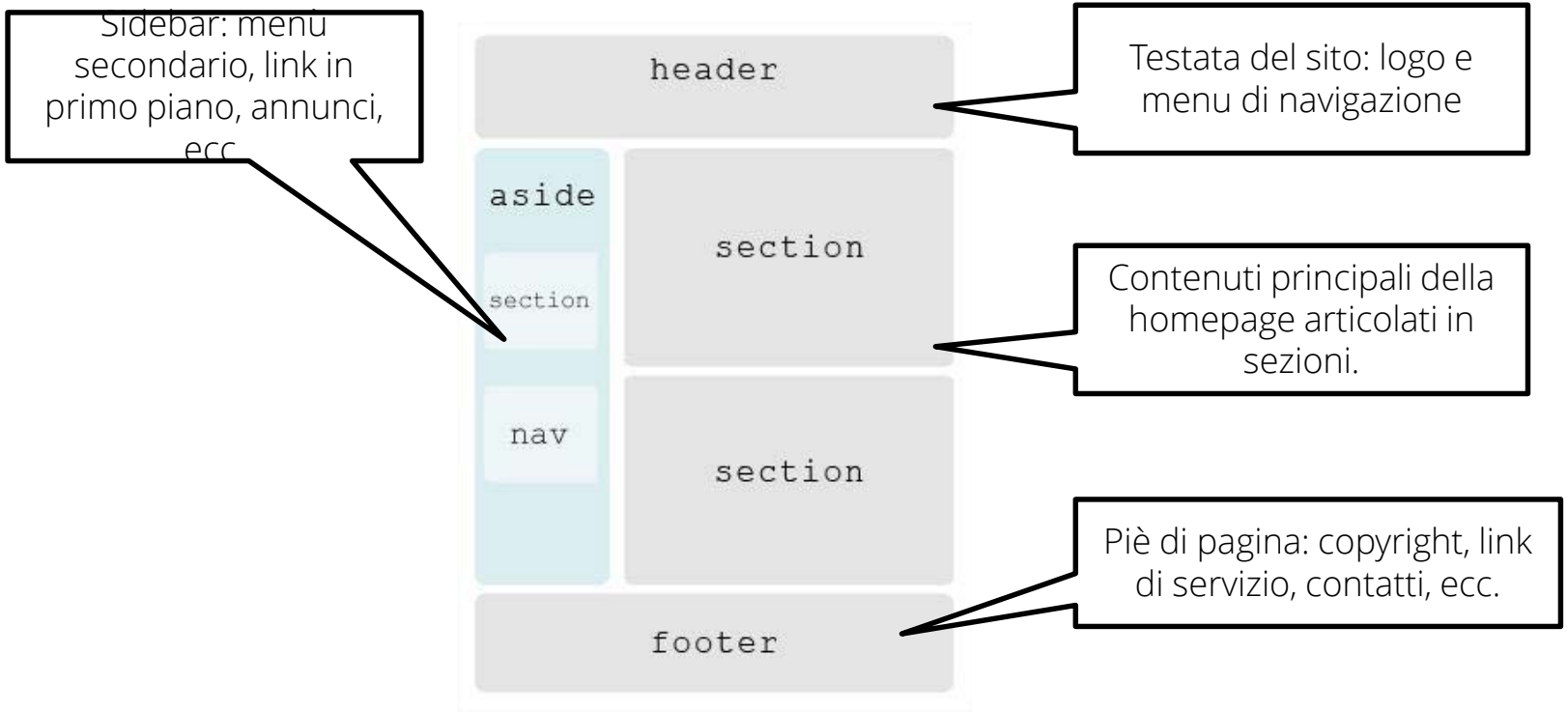
- L'elemento **<aside>** viene utilizzato per definire una porzione di codice correlata al contenuto ma separata come per esempio una sidebar.
- Le informazioni che racchiude, se rimosse, non devono incidere negativamente sulla completezza dell'informazione contenuta nell'elemento a cui è associato.
- Può essere utilizzato ad esempio per racchiudere elementi **<nav>** aggiuntivi, banner pubblicitari o note.

FOOTER

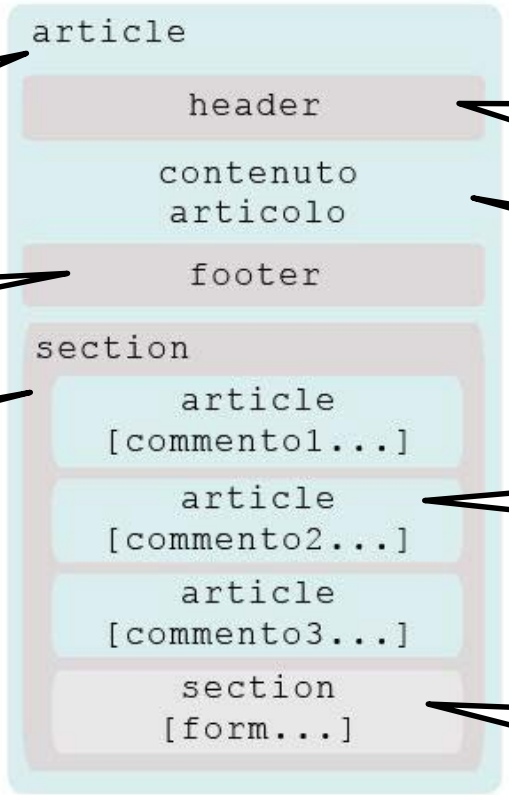
- L'elemento **<footer>** definisce una porzione di documento che rappresenta il piede della pagina o di una parte del documento.
- Non è prevista lacuna formattazione di default
- Analogamente alle intestazioni anche i piè di pagina possono essere molteplici all'interno di una pagina web.

```
<footer>
  &copy; 2017 Mia Azienda - Tutti i diritti riservati
</footer>
```

HOME PAGE GENERICA



POST IN UN BLOG



Il tag **<article>** ingloba l'intero blocco.

Testata dell'articolo con titolo e autore

Contenuto dell'articolo fatto di paragarfi, immagini, tabelle, ecc.

Piede dell'articolo con data e ora di pubblicazione

Commenti all'articolo

Sezione commenti

Modulo per l'inserimento di un commento

```
<article>
  <header>
    <h1>Titolo</h1>
    <div class="autore>">Mario Rossi</div>
  </header>
  <p>....</p>
  <!-- Contenuto articolo -->
  <footer>
    <time datetime="2017-03-01T14:00" pubdate >01/03/2017 14:00</time>
  </footer>
  <section class="commenti">
    <article>
      <!-- Autore commento 1, avatar, testo commento -->
    </article>
    <article>
      <!-- Autore commento 2, avatar, testo commento -->
    </article>
    <section class="aggiungi-commento">
      <form method="post">
        <label>Aggiungi commento:</label>
        <textarea rows="5" maxlength="500"></textarea>
        <button type="submit">Invia</button>
      </form>
    </section>
  </section>
</article>
```

FIGURE

- Il tag **<figure>** specifica un contenuto indipendente dal testo, come illustrazioni, diagrammi, foto, esempi di codice, ecc
- Mentre il contenuto dell'elemento **<figure>** è in relazione al flusso principale, la sua posizione è indipendente dal flusso principale, e se rimosso non dovrebbe influenzare il flusso del documento.
- Di norma l'elemento **<figure>** contiene un elemento **<figcaption>** che arricchisce l'elemento contenuto in **<figure>** con una descrizione

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">
  <figcaption>Una didascalia</figcaption>
</figure>
```

TIME

- L'elemento **<time>** rappresenta una data.
- È normalmente utilizzato all'interno di un elemento **<article>**
- L'attributo **datetime** indica la data in forma "iso" (**anno-mese-giorno ora:minuti:secondi**) in modo che programmi esterni che consultano l'articolo possano determinarla in modo certo
- L'attributo **pubdate** indica, se presente, che la data si riferisce alla pubblicazione dell'articolo in cui **<time>** è inserito altrimenti si presumerà che la data sia la data di pubblicazione della pagina.

```
<time datetime="2010-04-21 11:45" pubdate ="pubdate" >
      21 Pubblicato il 21 febbraio 2017 alle ore 11:45
</time>
```