

Informatica Grafica

Corso di Laurea in Ingegneria Edile – Architettura

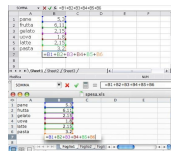
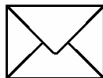
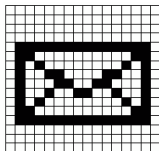
## **Elaborazione di documenti elettronici**

Paolo Torroni




Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)  
Università degli Studi di Bologna

Anno Accademico 2009/2010

# Elaborazione di documenti elettronici



## ► Elaborazione di documenti elettronici

- Che tipi di dati sono trattati dai computer? Come vengono rappresentati?
- Che strumenti usare per generare documenti elettronici?
- Notazione binaria, decimale, esadecimale
- Formati dei documenti elettronici. Codifica dei caratteri; font, caratteri e alfabeti. ASCII. Linguaggi di markup e HTML.
- Codifica di immagini e fonti tipografiche
- Come usare un word processor ( OpenOffice.org)
- Come usare un foglio elettronico ( OpenOffice.org)
- Come progettare un sito Web ( KomoZer)

# Documenti elettronici

- ▶ Programmi per i computer:
  - ▶ Software di base
    - ▶ **Sistema Operativo**
    - ▶ Driver di periferiche
    - ▶ Programmi che offrono servizi (*server* HTTP, HTML, etc.)
    - ▶ ...
  - ▶ Software applicativo
    - ▶ **Browser** e intrattenimento (player, videogiochi, ...)
    - ▶ Applicazioni gestionali e scientifiche
    - ▶ Antivirus
    - ▶ ...
    - ▶ **Applicazioni di produttività individuale**, per creare, visualizzare, trasmettere, stampare documenti elettronici

# Applicazioni di produttività individuale

- ▶ In questo corso:
  - ▶ Word processor
  - ▶ Spreadsheet
  - ▶ Computer Aided Design
  - ▶ Strumenti di fotoritocco
  - ▶ Web (HTML) editors
  - ▶ Strumenti per la creazione/gestione di piccoli database
- ▶ Fasi nella vita di un documento elettronico:

<b>Progetto e Generazione</b>	<b>Elaborazione</b>	<b>Fasi finali</b>
Progetto della struttura  Acquisizione dei contenuti	Aggiornamento della struttura Aggiornamento dei contenuti (aggiunte, correzioni, etc.) Rielaborazione dei contenuti (ordinamento, statistiche, etc.)	Visualizzazione dei contenuti Condivisione (stampa, trasmissione, conversione, etc.) Cancellazione

## Parte I

### Rappresentazione dell'informazione

HYUNIX 202A  
HY5DV641622AT-33  
KOREA

C243

C242

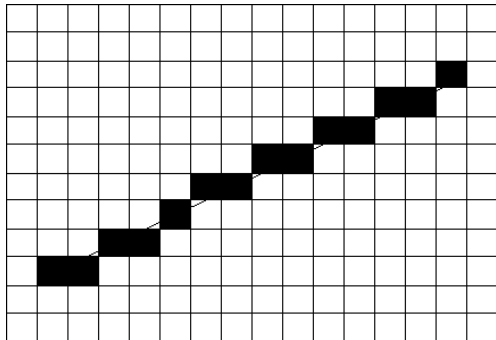
C241

1001011100101010001  
01010010010010101000  
01000101010010001010  
0010101001000101010  
0010110010110100100  
0110001011010100001  
001011100101010001  
0100100100101010100  
000101010010001010  
10101001000101010  
0110010110101010

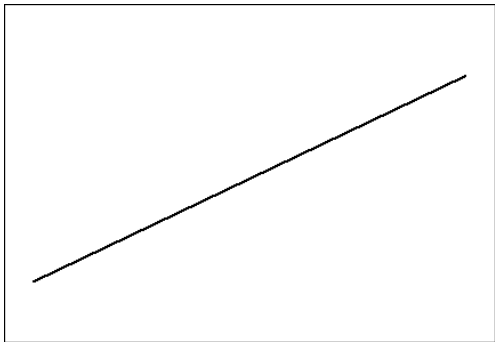




Che cosa rappresenta?



Che cosa rappresenta?











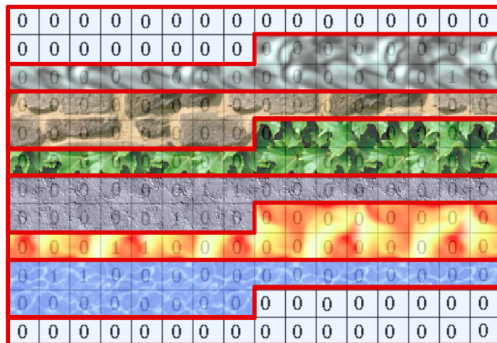


Che cosa rappresenta?

0	0	0	0	ت	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	ت	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	ت	0	0	0	1	0	0
0	0	0	0	پ	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	پ	0	0	0	0	0	0	1	ریا	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	ریا	0	0	0	0	0	0
0	0	0	0	ل	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	ل	1	0	0	0	0	0	0	د	0	0	0	0	0	0
0	0	0	1	ل	1	0	0	0	0	0	0	د	0	0	0	0	0	0
0	1	1	0	و	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	و	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	و	0	0	0	0	0	0	0	ت	0	0	0	0	0	0
0	0	0	0	و	0	0	0	0	0	0	0	ت	0	0	0	0	0	0



Che cosa rappresenta?





# Rappresentazione dei documenti elettronici

- ▶ **Formato binario:** 2 valori
- ▶ Cosa bisogna codificare?
  - ▶ **Dati**
    - ▶ Numeri
    - ▶ Caratteri
    - ▶ Elementi grafici
  - ▶ **Mark-up**
    - ▶ Struttura del documenti: sezioni, paragrafi, etc.
    - ▶ Formule (fogli elettronici)
    - ▶ Sequenze di immagini, etc.
  - ▶ **Sistema documentario**
    - ▶ Relazioni tra vari documenti elettronici
- ▶ Formato dei documenti digitali:
  - ▶ adeguato al **dominio**
  - ▶ il più possibile **indipendente dalle applicazioni**

# Numeri: sistemi di numerazione posizionali

- ▶ Sistema decimale: si basa su un *alfabeto* di 10 simboli (**cifre**)
  - ▶ 0, 1, 2, ..., 9
- ▶ Per rappresentare un numero si usa una sequenza (*stringa*) di cifre decimali
  - ▶ 0 → zero
  - ▶ 12 → dodici
  - ▶ 459 → quattrocentocinquantanove
  - ▶ etc.
- ▶ Meccanismo per associare una stringa a un valore: **codifica**
- ▶ Sistema di **numerazione posizionale**
  - ▶ I simboli hanno un **peso** che dipende dalla loro posizione nella stringa
  - ▶ Esempio: 134
    - ▶ 1 ha peso  $1 \times 100$
    - ▶ 3 ha peso  $3 \times 10$
    - ▶ 4 ha peso  $4 \times 1$
    - ▶ 134 vale  $100 + 30 + 4$

## Sistema decimale e sistema binario

- ▶ Il sistema decimale è anche detto **in base dieci**
- ▶ In generale, data una sequenza di  $n$  **cifre decimali**

$$c_n c_{n-1} \dots c_2 c_1 c_0$$

il **valore** di questa stringa è pari a:

$$c_n \times 10^n + c_{n-1} \times 10^{n-1} + \dots + c_2 \times 10^2 + c_1 \times 10^1 + c_0 \times 10^0$$

- ▶ Computer: sistema binario (**in base due**)
- ▶ La cifra è detta **Bit = Binary digit** e può valere 0 o 1
  - ▶ Data una sequenza di  $n$  **cifre binarie**

$$c_n c_{n-1} \dots c_2 c_1 c_0$$

il **valore** di questa stringa è pari a:

$$c_n \times 2^n + c_{n-1} \times 2^{n-1} + \dots + c_2 \times 2^2 + c_1 \times 2^1 + c_0 \times 2^0$$

# Codifica esadecimale

- ▶ Per convenienza, i bit sono raggruppati in gruppi di 8 (**byte**)
- ▶ Altro sistema di numerazione spesso utilizzato: base 16
- ▶ Si basa su un alfabeto di 16 cifre
  - ▶ 0, 1, 2, ..., 9, A, B, ..., F
  - ▶ le cifre da A a F hanno valore da 10 a 15
- ▶ 16 valori diversi: possibili combinazioni di 4 cifre binarie

<b>Codifica binaria (4 bit)</b>	<b>↔</b>	<b>Cifra esadecimale</b>	<b>↔</b>	<b>Valore (decimale)</b>
0000	↔	0	↔	0
0001	↔	1	↔	1
0010	↔	2	↔	2
0011	↔	3	↔	3
...	↔	...	↔	...
1110	↔	E	↔	14
1111	↔	F	↔	15

## Codifica esadecimale

- ▶ La codifica esadecimale consente di indicare il contenuto di un byte in modo compatto
- ▶ Esempio: 

0101 1101
-----------

 ↔ 5D

Codifica binaria (4 bit) ↔	Cifra esadecimale ↔	Valore (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

# Valori rappresentabili

- ▶ Universo dei valori rappresentabili: **dominio**.
- ▶ In un sistema in base  $B$ , la larghezza del dominio dipende da:
  - ▶ la base (es:  $B = 2$ ,  $B = 10$ ,  $B = 16$ )
  - ▶ Il numero  $n$  di cifre a disposizione
  - ▶  $\Rightarrow$  dominio di  $B^n$  valori rappresentabili
  - ▶ Es: con 3 cifre in base 10
    - ▶  $10^3 = 1000$  valori distinti (da 0 a 999)
- ▶ Con la codifica binaria, ragiono in termini di bit/byte:
  - ▶ con 1 bit: 2 valori distinti
  - ▶ con 1 byte:  $2^8 = 256$  valori distinti
  - ▶ con 4 byte:  $2^{4 \times 8} = 2^{32} = 4 \times 2^{30} = 4$  miliardi di valori distinti
- ▶ Approssimazioni di uso comune:
  - ▶  $2^{10} \sim 10^3 \rightarrow$  **K** (Kilo-), migliaia
  - ▶  $2^{20} \sim 10^6 \rightarrow$  **M** (Mega-), milioni
  - ▶  $2^{30} \sim 10^9 \rightarrow$  **G** (Giga-), miliardi
  - ▶  $2^{40} \sim 10^{12} \rightarrow$  **T** (Tera-), migliaia di miliardi

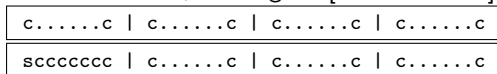


# Numeri senza segno, con segno, e decimali

- ▶ Rappresentazione dei numeri per un calcolatore
  - ▶ La memoria del computer non è infinita (limite fisico)
  - ▶ Per rappresentare ciascun numero: numero fisso di byte
  - ▶ Massimo intero rappresentabile: dipende dalla scelta fatta

## ▶ Numeri con segno?

- ▶ Metà dominio per i negativi, metà per i positivi e lo zero
- ▶ Esempio: 32 bit, senza segno:  $[0..2^{32} - 1]$  (0..4G)  
32 bit, con segno:  $[-2^{31}..2^{31} - 1]$  (-2G..2G)



## ▶ Numeri decimali?

- ▶ **Segno, mantissa ed esponente:**  $\pm 1.f \times 2^e$ 
  - ▶  $\pm$ : segno
  - ▶  $f$ : mantissa (precisione)
  - ▶  $e$ : esponente (massimo e minimo non nullo)

# Precisione singola e doppia

$$\pm 1.f \times 2^e$$

sffffffff		f...f		f...f		e...e
-----------	--	-------	--	-------	--	-------

sffffffff		f...f		f...f		f...f		f...f		f...f		fffff eee		e...e
-----------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-----------	--	-------

## ▶ Precisione singola:

- ▶ mantissa 23 bit ( $\sim 6$  cifre decimali significative)
- ▶ esponente 8 bit ( $\pm \sim 10^{-44} \dots \sim 10^{38}$ )

## ▶ Precisione doppia:

- ▶ mantissa 52 bit ( $\sim 15$  cifre decimali significative)
- ▶ esponente 11 bit ( $\pm \sim 10^{-323} \dots \sim 10^{308}$ )

## ▶ Combinazioni di bit riservate

- ▶  $-\infty$ ,  $+\infty$ , NaN

## ▶ Numeri non rappresentabili: positive/negative overflow/underflow

## ▶ Precisione limitata alle somme di potenze (negative) di 2!

*IEEE Standard 754 Floating Point Numbers*

# Caratteri

Tack	Obrigado	Vielen Dank
Merci	ありがとうございます	
Bedankt	Takk	感謝您
谢谢	Terima Kasih	Grazie
	ขอบคุณ	
Киитос	Спасибо	Thank You
	Tak	
Teşekkür Ederiz	감사합니다	
	Gracias	
Dziękujemy	Σας ευχαριστούμε	

# Caratteri

- ▶ **Carattere:** qualunque segno grafico utilizzato in tipografia per rappresentare le lettere, i segni di interpunzione, le cifre, etc.
- ▶ Tre diversi aspetti da considerare:
  - ▶ **Natura:** che cosa rappresenta.
    - ▶ 2 caratteri aventi la stessa natura:  $\sigma$  /  $\varsigma$
    - ▶ 2 caratteri aventi diversa natura:  $\mathfrak{a}$  /  $\mathring{\mathfrak{a}}$
    - ▶ variazioni della stessa lettera:  $\mathfrak{a}$  /  $\grave{\mathfrak{a}}$
  - ▶ **Glifo:** la forma ottenuta in stampa
    - ▶ diversi glifi, stessa natura:  $\mathfrak{a}$ ,  $\mathfrak{a}$ ,  $\mathfrak{a}$ ,  $\mathfrak{a}$ ,  $\mathfrak{a}$
  - ▶ **Codifica:** scelta della sequenza di simboli binari corrispondenti
    - ▶ es:  $\mathfrak{a} \rightarrow 1100001$

# Codifica dei caratteri

- ▶ Come codificare i caratteri?
- ▶ Problema semplice da risolvere con un alfabeto:
  - ▶ contare quanti lettere e simboli si vogliono rappresentare
    - ▶ es: A, B, ..., Z, a, b, ..., z, 0, 1, ..., 9, !, , #, €, (, ), ?, :, ...  $\Rightarrow$  100 simboli
  - ▶ prendere la minima potenza di due che ne contiene il numero
    - ▶  $100 \Rightarrow 2^7 = 128$
  - ▶ l'esponente del 2 identifica il numero di bit necessario
    - ▶  $128 \Rightarrow 7$
  - ▶ stabilire una corrispondenza tra stringhe di (7) bit e caratteri
- ▶ Problemi: **armonizzazione** tra codifiche di alfabeti diversi e tra architetture diverse
- ▶ Soluzione: definizione di **standard**

# ASCII

- ▶ **American Standard Code for Information Interchange:**
  - ▶ usa 7 bit (128 caratteri)
  - ▶ alcune combinazioni sono usate per *caratteri speciali* (a-capo, tabulazione, beep, ...)
  - ▶ orientato alla lingua inglese (alfabeto latino, senza accenti)
- ▶ Però:
  - ▶ molte altre lingue con diversi alfabeti
  - ▶ sforzi simili all'ASCII (arabo, greco, cirillico, cinese, etc.)
  - ▶ produttori di hardware: codifiche a 8 bit (256 caratteri, di cui solo i primi 128 standard ASCII)
- ▶ Problemi:
  - ▶ Interoperabilità nel passaggio di documenti da un sistema operativo a un altro
  - ▶ Sbilanciamento verso la lingua inglese
  - ▶ Documenti scritti in due lingue?

# ISO Latin e Unicode

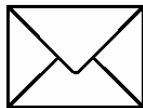
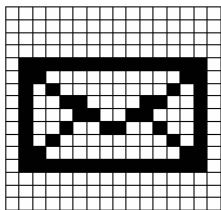
- ▶ **ISO Latin** (ISO Latin-1, ISO Latin-9)
  - ▶ codifica a 8 bit
  - ▶ standardizzazione della codifica degli alfabeti europei
  - ▶ compatibilità con ASCII
- ▶ **Unicode: Unified Character Set**
  - ▶ UCS-2: codifica a 16 bit
    - ▶ di norma: il primo byte identifica l'alfabeto
    - ▶ Es: 00 . . . . 0 corrisponde a ISO Latin-1
  - ▶ UCS-4: codifica a 32 bit
    - ▶ usata solo per alfabeti con moltissimi caratteri diversi: cinese letterario, alfabeti antichi (geroglifici, cuneiforme, etc.)
- ▶ **Unicode: Unified Transformation Format (UTF-8)**
  - ▶ Usa pochi byte per i caratteri più frequenti:
    - ▶ 8 bit per ASCII
    - ▶ 16 bit per ISO Latin-1 e per alfabeti non latini più comuni
    - ▶ 24/32 bit per alfabeti orientali e storici

## Elementi grafici





# Modalità di resa grafica digitale



- ▶ **Bitmap**, o **raster**, orientata al **pixel**.
  - ▶ **Griglie** di punti e colori.
  - ▶ Immagini catturate dalla realtà (fotografie: difficilmente riconducibili a elementi matematici. Contorni? Sfumature)
  - ▶ Si prestano a visualizzazione a video.
- ▶ **Vettoriale**, orientata alle **linee**, **curve**, **aree**.
  - ▶ **Istruzioni** necessarie a creare elementi grafici.
  - ▶ Immagini di sintesi.
  - ▶ Maggiore definizione (in assenza di sfumature elaborate)
  - ▶ Si prestano a ridimensionamento e resa di font.

# Bitmap

- ▶ Memorizzata come griglia di  $n \times m$  punti (**dot** o **pixel**, su schermo)
- ▶ **Qualità** dipende da:
  - ▶ Risoluzione spaziale ( $n, m$ ): **dot per inch (dpi)**
  - ▶ Risoluzione cromatica: **profondità di colore**
- ▶ **Resa** dipende anche dal **mezzo di output** utilizzato
- ▶ Aumento di qualità  $\Rightarrow$  aumento di **dimensioni**
- ▶ Tre risoluzioni cromatiche:
  - ▶ **bicromia** (1 bit per dot)
  - ▶ **palette** (1 byte, 256 colori)
  - ▶ **true color** (3 byte, 16ML colori)

# True color

- ▶ **Tricromia:** sistema colorimetrico a 3 valori (byte)
- ▶ Colori rappresentabili >> colori distinguibili dall'occhio umano
- ▶ 3 modalità
  - ▶ **Spazi colorimetrici additivi:** colori aggiunti a partire dal nero
    - ▶ **RGB** (Red, Green, Blue)
    - ▶ Usato dai monitor (luce trasmessa)
    - ▶ (0,0,0) indica il nero, (1,1,1) indica il bianco
    - ▶ Codifica esadecimale: #000000–#FFFFFF
    - ▶ Esempi: **#FF0000 rosso**, **#7F0000 rosso scuro**, **#0000FF blu**, **#FF00FF viola**, **#7F007F viola scuro**, etc.
  - ▶ **Spazi colorimetrici sottrattivi:** colori sottratti dal bianco
    - ▶ **CMYK** (Cyan, Magenta, Yellow, black)
    - ▶ Usato dalle stampanti (luce sottratta)
  - ▶ **Tonalità, luminosità e saturazione**
    - ▶ **HSB** (Hue, Saturation, Brightness)
    - ▶ Le tonalità (Hue) sono mescolate al bianco e/o nero
    - ▶ Hue: nome di un colore puro, da 0 a 360. (0: rosso, 60: giallo, 120: verde, 180: ciano, 240: blu, 300: magenta)
    - ▶ Saturation/Brightness: presenza di bianco/nero, da 0 a 100%

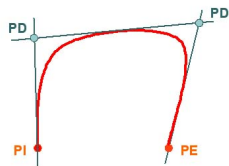
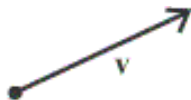
# Formati bitmap

- ▶ Compressione per ridurre le dimensioni delle immagini
- ▶ Due modalità
  - ▶ senza perdita di qualità (**lossless**)
  - ▶ con perdita di qualità (**lossy**)
- ▶ Formati principali
  - ▶ Lossless:
    - ▶ **BMP**, nessuna compressione: direttamente visualizzabile su schermo
    - ▶ **GIF** (Graphic Interchange Format) e **PNG** (Portable Network Graphics), basati su **palette**
    - ▶ GIF usato anche per semplici animazioni
    - ▶ **TIFF** (Tagged Image File Format), **contenitore** per vari formati
    - ▶ usato anche per fax, scansioni, etc., anche su più pagine
  - ▶ Lossy:
    - ▶ **JPEG** (Joint Photographic Expert Group), usa true color
    - ▶ Algoritmo che elimina le sfumature difficilmente percettibili.
    - ▶ Possibilità di scegliere il livello di compressione.

# Immagini vettoriali



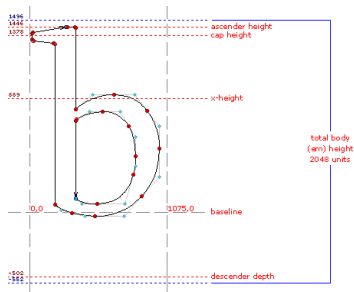
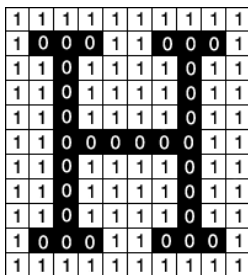
- ▶ Descrizione matematiche di elementi grafici
- ▶ **Vettori** usati per definire gli elementi di base
  - ▶ Punto di partenza, lunghezza, direzione
- ▶ **Curve di Bézier**, per rappresentare oggetti complessi e curvilinei
  - ▶ Punti di inizio e fine della curvatura (PI, PE)
  - ▶ Punti di direzione (PD)



# Fonti tipografiche

- ▶ **Fonte tipografica digitale:** assortimento completo di caratteri in una certa dimensione e stile
- ▶ **Glifo:** rappresentazione grafica dell'aspetto di un carattere
- ▶ Caratteristiche stilistiche di una fonte:
  - ▶ Presenza di **grazie** o **serif** (piedini terminali di abbellimento)
  - ▶ **Proporzionalità della larghezza** dei glifi
    - ▶ Serif, proporzionale:  
Times Roman
    - ▶ Sans serif, proporzionale:  
Helvetica
    - ▶ Serif, non proporzionale:  
Courier

# Tipi di fonti tipografiche



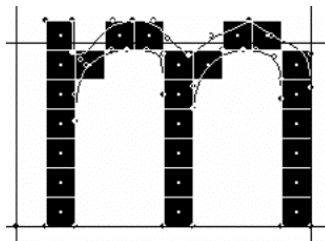
## ► Bitmap

- Glifi rappresentati come **immagini** separate di caratteri.
- Dimensione precisa (e.g. 12 punti). Scarsa qualità.
- Non più usate.

## ► Vettoriali (scalabili)

- Glifi rappresentati come **insiemi di formule**.
- Rendering grafico attraverso **rasterizzazione**.

## Rasterization e anti-aliasing



- ▶ **Rasterization:** Scelta dei pixel che devono essere accesi per riprodurre un glifo di una fonte vettoriale.
- ▶ **Anti-aliasing:** tecnica per ridurre l'effetto della visualizzazione "scalettata" delle immagini
  - ▶ Introduzione di tonalità di colore (grigio)



## Parte II

### Linguaggi di markup e HTML

## Il markup

- ▶ Markup: qualunque informazione si aggiunga ad un contenuto per renderlo più comprensibile o utilizzabile:
  - ▶ nel testo: punteggiatura, spazi, neretto, corsivo, etc.
  - ▶ divisione in pagine, organizzazione in righe, uso dei margini, numerazione delle pagine, etc.
- ▶ Preparare un documento con un programma applicativo  $\Rightarrow$  aggiungere markup al contenuto
- ▶ Ciascun programma applicativo propone un **formato dati** per indicare il markup e associarlo al contenuto
- ▶ Esempi di linguaggi di markup:
  - ▶ **HTML**, XML (W3C)
  - ▶ Formati proprietari:
    - ▶ Word, Excel, Power Point (Microsoft, Office), RTF
    - ▶ Pages, Numbers, Keynote (Apple, iWork)
    - ▶ StarOffice, OpenOffice (SUN)
    - ▶ WordPerfect (Corel), etc.
  - ▶ Linguaggi di programmazione: PostScript, PDF (Adobe),  $\LaTeX$

# Caratteristiche dei linguaggi di markup a confronto

- ▶ Leggibilità
  - ▶ **Formato binario**: uso di rappresentazioni interne non testuali, per codifica diretta dei dati
  - ▶ **Formato leggibile**: uso di caratteri speciali: <...>, {...}, ...
- ▶ Tipo di informazioni
  - ▶ **Presentazionale**: contiene istruzioni di presentazione tipografica (orientato alla visualizzazione)
  - ▶ **Descrittivo**: descrive il ruolo dei vari elementi del documento (uso di fogli di stile)
- ▶ Proprietà del formato dati
  - ▶ **Proprietario**: le caratteristiche del linguaggio sono controllate da un'impresa commerciale
  - ▶ **Non proprietario**: organizzazione senza fini di lucro o consorzio (apertura, pubblicazione, interoperabilità)

## Definire un linguaggio di markup

- ▶ Esistono molti linguaggi di mark-up, progettati con caratteristiche diverse a seconda dell'applicazione
- ▶ Per **descrivere** in modo rigoroso un linguaggio, bisogna usare un **meta-linguaggio**
- ▶ **XML** è un meta-linguaggio di markup molto usato
- ▶ **Document Type Definition (DTD)**
  - ▶ documento scritto in un meta-linguaggio, es: XML
  - ▶ elenca gli elementi di markup e le regole di strutturazione che descrivono le caratteristiche di un nuovo linguaggio di markup
- ▶ DTD XML usati per definire versioni e varianti di HTML
  - ▶ HTML 4.01 Strict
  - ▶ HTML 4.01 Transitional
  - ▶ XHTML 1.0 Strict, etc.
- ▶ <http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

# Tag in HTML

- ▶ 4 categorie di tag
  - ▶ **strutturali** (titoli, sottotitoli, elenchi, tabelle, etc.)
  - ▶ **presentazionali** (grassetto, corsivo, etc.)
  - ▶ **ipertestuali** (link)
  - ▶ **multimediali** (immagini e altri oggetti multimediali)
- ▶ La sintassi usa le parentesi uncinate:
  - ▶ *titolo*: `<h1> ... </h1>`
  - ▶ *elenchi numerati*: (ordered list & list items)  
`<ol>`  
    `<li> ... </li>`  
    `...`  
    `<li> ... </li>`  
`</ol>`
  - ▶ *grassetto*: `<b> ... </b>`
  - ▶ *URI*: `<a src="..."> ... </a>`
  - ▶ *immagine*: ` ... </img>`

# Fogli di stile

- ▶ Per definire la presentazione (grafica o attraverso media alternativi) si possono usare **fogli di stile**.
- ▶ In HTML sono realizzati secondo uno standard che si chiama **Cascading Style Sheet (CSS)**.
- ▶ 2 possibili modi di specificare un CSS:

- ▶ **Dentro l'HTML**, mediante il tag `<style>` nell'header;

```
<style>
  h1    { font-family: Arial; font-size: 120%; color:#666699; }
  ol    { font-family: Arial; color: #6666FF; }
  p     { font-family: Arial; font-size: 100%; color:#333333; }
</style>
```

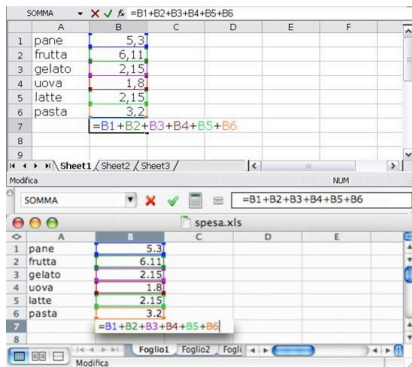
- ▶ **Come file esterno**, inserendo un `<link>` nell'header.

```
<link media="screen" href="sample-style.css" rel="stylesheet">
```

## Parte III

### Fogli elettronici (spreadsheet)

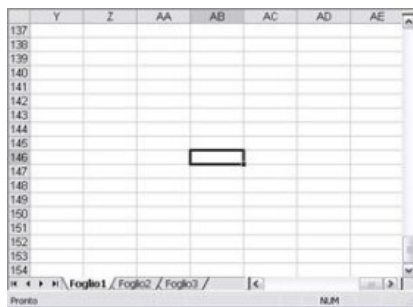
# Cos'è un foglio elettronico?



- ▶ Astrazione di un foglio di carta a quadretti per svolgere operazioni su dati
  - ▶ Raccolta
  - ▶ Presentazione (incolonnamento, etc.) e reporting
  - ▶ **Elaborazione** (operazioni aritmetiche, statistiche, etc.)
- ▶ Foglio "attivo"



## Organizzazione dei dati e riferimenti



- ▶ Dati contenuti in **celle**
- ▶ Riferimento al **contenuto** di celle: notazione **ColonnaRiga**
  - ▶ **Riga: numero**
  - ▶ **Colonna: lettere**
    - ▶ Singola cella: **AB146**, **AB\$146**, **\$AB146**, **\$AB\$146**
    - ▶ **Gruppi di celle: AB146:AB154**, **AB146:AE154**
- ▶ Formule: notazione **=...**
- ▶ Molte **funzioni** (somma, media, etc.) e **grafici** predefiniti

## Parte IV

### Esercitazioni in laboratorio

# Strumenti per la produzione di documenti elettronici

Sessioni pratiche (*hands-on*) introduttive all'uso degli strumenti.

1. **Word processor** di OpenOffice.org
  - ▶ Giovedì 8 ottobre 2009, 15.30-17.00, Lab4
  - ▶ Modulo A
2. **Fogli elettronici** con OpenOffice.org
  - ▶ Giovedì 15 ottobre 2009, 15.30-17.00, Lab4
  - ▶ Modulo B
3. Progettazione di **siti Web** con CompoZer
  - ▶ Giovedì 22 ottobre 2009, 15.30-17.00, Lab4
  - ▶ Capitolo 2.6 e Modulo F



Handouts and all other material for **Informatica Grafica per Ingegneria Edile-Architettura**, Università di Bologna - A.A. 2009/2010 by Paolo Torroni is licensed under a **Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License**.

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/>

Based on a work at University of Bologna, Italy. <http://www.unibo.it/>

Paolo Torroni's Web site: <http://lia.deis.unibo.it/~pt/>

Composed using the **L<sup>A</sup>T<sub>E</sub>X Beamer Class**, <http://latex-beamer.sourceforge.net/>