



UNIVERSITÀ
DI PARMA

rappresentazione di informazioni multimediali



alberto ferrari - fondamenti di informatica

caratteri e testo

- necessaria **convenzione** per codifica numerica (**binaria**) dei caratteri
- codifica **ASCII** (American Standard Code for Information Interchange)
 - inizialmente 7 bit $\Rightarrow 2^7 = 128$ caratteri
- caratteri **alfanumerici**: *lettere maiuscole, minuscole, numeri, spazio*
- simboli e **punteggiatura**: @, #, ...
- caratteri di **controllo (non tutti visualizzabili)**: TAB, LF, CR, BELL ecc.
- interruzione di riga (a capo)
 - Unix: **LF** (Line Feed, 0A)
 - Multics, Unix etc., Mac OS X, BeOS, Amiga, RISC OS
 - Vecchi Apple: **CR** (Carriage Return, 0D)
 - Commodore, Apple II family, Mac OS up to version 9
 - Windows: **CR+LF** (0D+0A)
 - Most early OSes, DOS, OS/2, Windows, Symbian

| Byte | Cod. | Char | Byte | Cod. | Char | Byte | Cod. | Char | Byte | Cod. | Char |
|----------|------|------------------|----------|------|------|----------|------|------|----------|------|------|
| 00000000 | 0 | Null | 00100000 | 32 | Spc | 01000000 | 64 | @ | 01100000 | 96 | ` |
| 00000001 | 1 | Start of heading | 00100001 | 33 | ! | 01000001 | 65 | A | 01100001 | 97 | a |
| 00000010 | 2 | Start of text | 00100010 | 34 | " | 01000010 | 66 | B | 01100010 | 98 | b |
| 00000011 | 3 | End of text | 00100011 | 35 | # | 01000011 | 67 | C | 01100011 | 99 | c |
| 00000100 | 4 | End of transmit | 00100100 | 36 | \$ | 01000100 | 68 | D | 01100100 | 100 | d |
| 00000101 | 5 | Enquiry | 00100101 | 37 | % | 01000101 | 69 | E | 01100101 | 101 | e |
| 00000110 | 6 | Acknowledge | 00100110 | 38 | & | 01000110 | 70 | F | 01100110 | 102 | f |
| 00000111 | 7 | Audible bell | 00100111 | 39 | , | 01000111 | 71 | G | 01100111 | 103 | g |
| 00001000 | 8 | Backspace | 00101000 | 40 | (| 01001000 | 72 | H | 01101000 | 104 | h |
| 00001001 | 9 | Horizontal tab | 00101001 | 41 |) | 01001001 | 73 | I | 01101001 | 105 | i |
| 00001010 | 10 | Line feed | 00101010 | 42 | * | 01001010 | 74 | J | 01101010 | 106 | j |
| 00001011 | 11 | Vertical tab | 00101011 | 43 | + | 01001011 | 75 | K | 01101011 | 107 | k |
| 00001100 | 12 | Form Feed | 00101100 | 44 | , | 01001100 | 76 | L | 01101100 | 108 | l |
| 00001101 | 13 | Carriage return | 00101101 | 45 | - | 01001101 | 77 | M | 01101101 | 109 | m |
| 00001110 | 14 | Shift out | 00101110 | 46 | . | 01001110 | 78 | N | 01101110 | 110 | n |
| 00001111 | 15 | Shift in | 00101111 | 47 | / | 01001111 | 79 | O | 01101111 | 111 | o |
| 00010000 | 16 | Data link escape | 00110000 | 48 | 0 | 01010000 | 80 | P | 01110000 | 112 | p |
| 00010001 | 17 | Device control 1 | 00110001 | 49 | 1 | 01010001 | 81 | Q | 01110001 | 113 | q |
| 00010010 | 18 | Device control 2 | 00110010 | 50 | 2 | 01010010 | 82 | R | 01110010 | 114 | r |
| 00010011 | 19 | Device control 3 | 00110011 | 51 | 3 | 01010011 | 83 | S | 01110011 | 115 | s |
| 00010100 | 20 | Device control 4 | 00110100 | 52 | 4 | 01010100 | 84 | T | 01110100 | 116 | t |
| 00010101 | 21 | Neg. acknowledge | 00110101 | 53 | 5 | 01010101 | 85 | U | 01110101 | 117 | u |
| 00010110 | 22 | Synchronous idle | 00110110 | 54 | 6 | 01010110 | 86 | V | 01110110 | 118 | v |
| 00010111 | 23 | End trans. block | 00110111 | 55 | 7 | 01010111 | 87 | W | 01110111 | 119 | w |
| 00011000 | 24 | Cancel | 00111000 | 56 | 8 | 01011000 | 88 | X | 01111000 | 120 | x |
| 00011001 | 25 | End of medium | 00111001 | 57 | 9 | 01011001 | 89 | Y | 01111001 | 121 | y |
| 00011010 | 26 | Substitution | 00111010 | 58 | : | 01011010 | 90 | Z | 01111010 | 122 | z |
| 00011011 | 27 | Escape | 00111011 | 59 | ; | 01011011 | 91 | [| 01111011 | 123 | { |
| 00011100 | 28 | File separator | 00111100 | 60 | < | 01011100 | 92 | \ | 01111100 | 124 | |
| 00011101 | 29 | Group separator | 00111101 | 61 | = | 01011101 | 93 |] | 01111101 | 125 | } |
| 00011110 | 30 | Record Separator | 00111110 | 62 | > | 01011110 | 94 | ^ | 01111110 | 126 | ~ |
| 00011111 | 31 | Unit separator | 00111111 | 63 | ? | 01011111 | 95 | _ | 01111111 | 127 | Del |

tabella ascii estesa

- caratteri accentati + caratteri per grafici
- code Page 437 per PC (DOS) in Nord America
- possibile mischiare testo in inglese e francese (anche se in Francia CP850); ma non assieme greco (CP737), russo ecc.
 - ISO 8859, estensioni standard per ASCII ad 8 bit
 - ISO 8859-1 (o Latin1): Lingue dell'Europa Occidentale
 - ISO 8859-2: Lingue dell'Europa Orientale
 - ISO 8859-5: Alfabeto cirillico
 - ISO 8859-15: Latin1 con simbolo euro (€)

<https://www.ascii-codes.com/>



unicode

- **unicode** associa un preciso **code-point (32 bit)** a ciascun simbolo
 - possibile rappresentare miliardi di simboli
 - primi 256 code-point = Latin1
- attualmente più di 30 sistemi di scrittura
 - rappresentazione di geroglifici e caratteri cuneiformi
 - da emoticon :-) a emoji 😊
 - proposta per Klingon (da Star Trek) ... *rifiutata* ☹

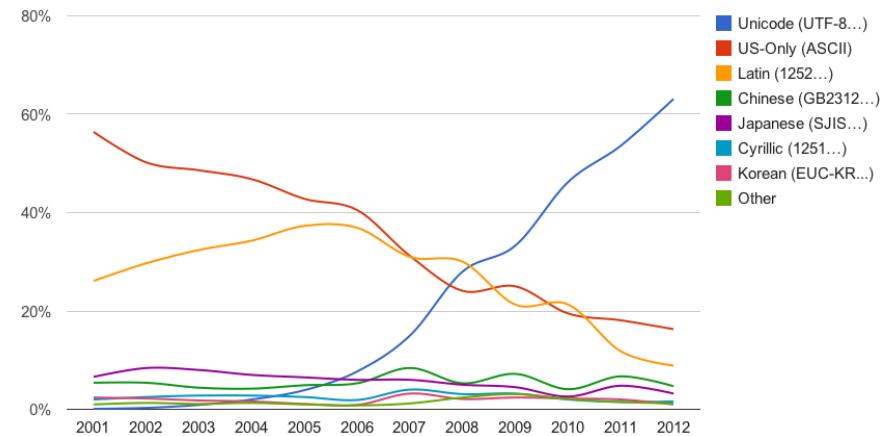


<https://unicode-table.com>



Unicode Transformation Format

- **codifica** di un code-point in una *sequenza di bit*
- servono uno o più code-unit
 - UTF-32 – code-unit di 32-bit, 1 carattere → 1 code unit
 - UTF-16 – code-unit di 16-bit, 1 carattere → 1 - 2 code unit
 - UTF-8 – code-unit di 8-bit, 1 carattere → 1 - 4 code unit
 - massima compatibilità con ASCII



UTF-8

- se bit più alto a 0, nel code-point
 - simbolo ASCII su 7 bit, invariato
- altrimenti, codifica su ***n code-unit*** (byte)
 - primo byte inizia con ***n bit a 1, poi uno a 0***
 - byte seguenti cominciano tutti con ***10***
 - bit di payload / lunghezza codifica: 7/8, 11/16, 16/24, 21/32

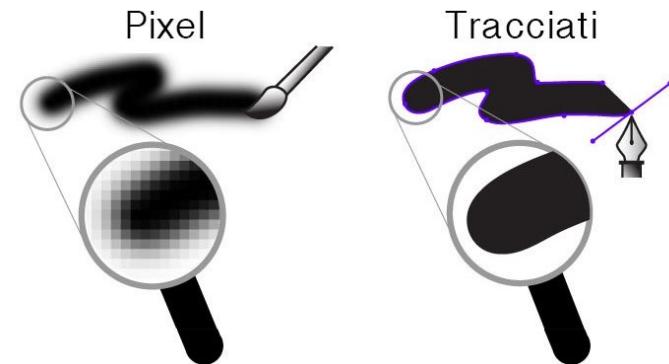
| Character | Binary code point | Binary UTF-8 | Hexadecimal UTF-8 |
|-----------|----------------------------|-------------------------------------|-------------------|
| \$ U+0024 | 010 0100 | 00100100 | 24 |
| ¢ U+00A2 | 000 1010 0010 | 11000010 10100010 | C2 A2 |
| € U+20AC | 0010 0000 1010 1100 | 11100010 10000010 10101100 | E2 82 AC |
| ø U+10348 | 0 0001 0000 0011 0100 1000 | 11110000 10010000 10001101 10001000 | F0 90 8D 88 |

rappresentazione di informazioni multimediali
immagini digitali



immagini digitali

- **digitalizzazione**: procedimento per convertire un'immagine in una sequenza binaria
- tipologie di immagini digitali
 - **raster** ⇒ immagine suddivisa in una griglia di punti (**pixel**)
 - **vettoriali** ⇒ insieme di primitive geometriche
 - linee, poligoni



palette

- rappresentazione digitale di una immagine
- la prima operazione è quella di definire una rappresentazione digitale per ogni **colore**
- stabilito il **numero di bit (profondità)** da utilizzare si definisce l'insieme dei colori (tavolozza, **palette**) che saranno utilizzati per rappresentare l'immagine

| colore | codice binario | valore decimale |
|--------------|----------------|-----------------|
| | 0000 | 0 |
| yellow | 0001 | 1 |
| light green | 0010 | 2 |
| light blue | 0011 | 3 |
| pink | 0100 | 4 |
| light purple | 0101 | 5 |
| orange | 0110 | 6 |
| purple | 0111 | 7 |
| teal | 1000 | 8 |
| light green | 1001 | 9 |
| purple | 1010 | 10 |
| orange | 1011 | 11 |
| green | 1100 | 12 |
| gray | 1101 | 13 |
| | 1110 | 14 |
| black | 1111 | 15 |

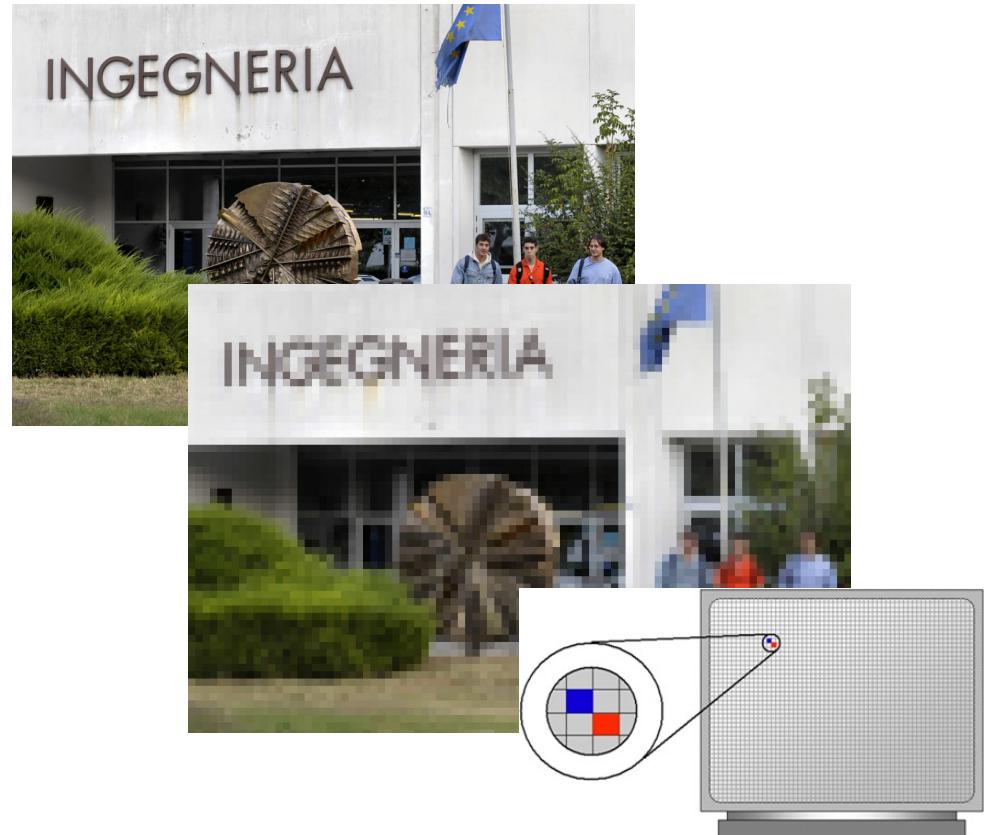
modelli di colore

- occhio sensibile a *variazioni luminosità*
 - *fotoricettori*: 6 mln di *coni*, 120 mln di *bastoncelli*
- **RGB**: rosso, verde, blu
 - 8 bit: 3 bit × R e G, 2 × B
 - 24 bit: 8 bit × R, G e B
 - 32 bit: canale alpha grado trasparenza/opacità
- **YUV**: luminosità, crominanza di R e B
 - sistema PAL, JPEG, MPEG
 - TV a colori (compatibilità B&W)
- **HSL (Hue Saturation Brightness)**: tonalità, saturazione e luminosità



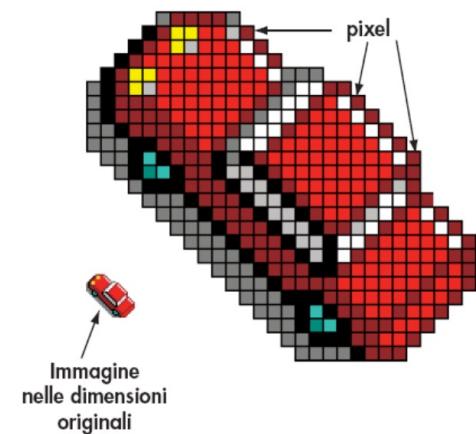
pixel

- immagine suddivisa in piccoli rettangoli
 - *elementi di base dell'immagine digitale*
 - **pixel** (*picture element*)
- per ogni pixel individuare un **colore** dominante
- l'immagine diventa una sorta di mosaico (*i tasselli del mosaico sono i pixel*)
- la tavolozza fornisce la sequenza di bit associata ad ogni pixel
- l'insieme di tutte le sequenze è la rappresentazione digitale dell'immagine



pixel

- il pixel è di un singolo colore
- il pixel non ha dimensione metrica
 - DPI (Dot Per Inch) (Punti per pollice)
 - DPI esprime la quantità di punti stampati o visualizzati su una linea lunga un pollice
- l'occhio umano non è in grado di percepire la suddivisione in pixel
 - su un monitor a 72 DPI
 - (le immagini con queste caratteristiche sono valide per il web)
 - su una stampa a 300 DPI
 - (600 DPI alta qualità)



immagini – approssimazione e risoluzione

- aumentare il numero di pixel (*e ridurre quindi la loro dimensione*) migliora la **definizione** dell'immagine
- i monitor dei computer usano lo stesso procedimento per visualizzare le immagini
- la dimensione ridotta dei pixel e il numero elevato di colori fanno apparire al nostro occhio le immagini come se fossero formate da **linee continue** e infinite **sfumature di colore**
- **risoluzione** dell'immagine
 - **numero dei pixel**
(righe x colonne)
 - **profondità** di colore
(dimensione palette)



immagini - memoria

- il numero di bit necessario per rappresentare un'immagine è elevato
- es. risoluzione di 1920 x 1080 pixel e 24 bit colore:
 - risulta “scomposta” in $1920 \times 1080 \cong 2$ milioni pixel
 - per pixel colore a 24 bit (3 byte) $\cong 6$ Megabyte
- **compressione**
 - per limitare l'occupazione di memoria si ricorre a rappresentazioni compresse
 - alcune tecniche di compressione mantengono inalterata la qualità dell'immagine, eliminando soltanto le informazioni ridondanti
 - altre riducono il numero di byte complessivi ma comportano anche perdita di qualità

immagini raster - formati

- il formato delle immagini identifica il ***tipo*** di rappresentazione digitale
- **BMP**: immagine (normalmente) non compressa
- **TIFF, PNG**: **comprimono** l'immagine, per ridurne l'occupazione, senza deteriorarla (compressione ***lossless***)
- **JPEG**: comprime (molto di più), ma deteriora l'immagine (compressione ***lossy***)

BMP

FILE INFO HEADER (14)

2 Tipo file (= "BM")
4 Dim. file (in byte)
4 Riservato
4 Offset immagine (in byte)

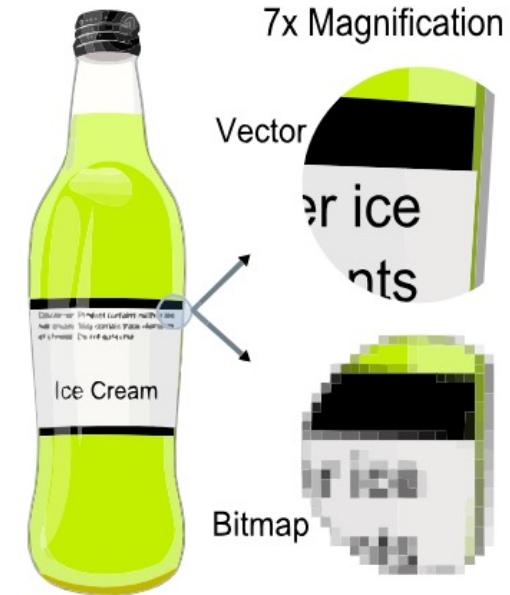
BITMAP INFO HEADER (40)

4 Dimensione struttura
4+4 Larghezza e altezza immagine
2 Piani (non usato)
2 # bit per pixel
4+4 Compressione e dim. img (0 senza compressione)
4+4 Risoluzione orizz. e vert. (pixel per metro)
4+4 # colori in palette e # colori importanti
Palette (RGBQUAD)
4 Blue, Green, Red, Riservato



grafica vettoriale

- ***immagine***: insieme di primitive geometriche
 - linee, poligoni..., colori, sfumature...
 - per ogni elemento vengono definite le coordinate dei punti di applicazione
-  qualità, a varie risoluzioni
-  compressione dati
-  gestione modifiche
-  non intuitiva
-  possibilmente onerosa



immagini vettoriali

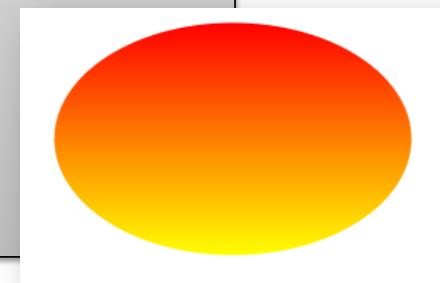
- **applicazioni**
 - editoria (DTP), video-editing, architettura,
 - grafica 3D (CAD)
 - font vettoriali (*caratteri scalabili in dimensione senza perdere definizione*)
- **formati**
 - PS (PostScript), PDF (Portable Document Format), WMF (Windows MetaFile)
 - DXF (AutoCAD), CDR (CorelDraw), SWF (Flash)
 - SVG (Scalable Vector Graphics, utilizzato nel web)

esempio file SVG in HTML

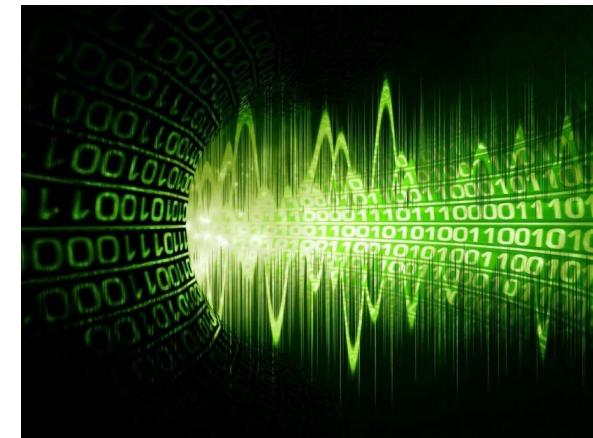
```
<!DOCTYPE html>
<html>
<body>

<svg height="150" width="400">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```



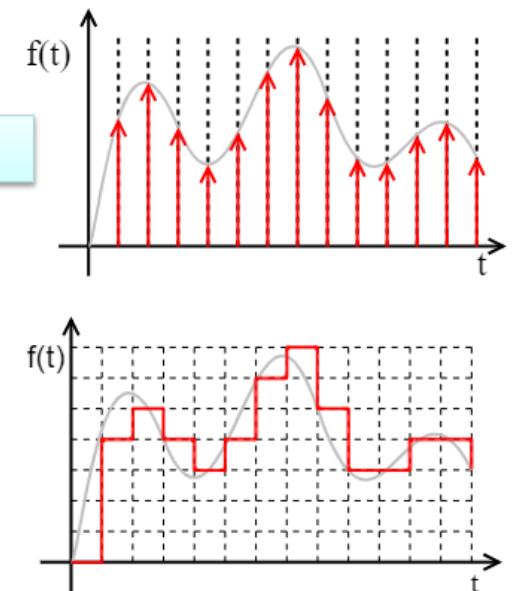
rappresentazione di informazioni multimediali
audio digitale



- **suono**
 - onde longitudinali, di **compressione** e **rarefazione** dell'aria
 - grandezza analogica → **discretizzazione**
 - **campionamento** (*sampling*) nel tempo
 - **quantizzazione** (*quantizing*) nelle ampiezze
 - qualità *CD*
 - 44 kHz, 16bit
 - spettro udibile: 20-20k Hz
 - qualità voce
 - mono, 8 kHz, 8 bit

$$\text{Hz} [=] \frac{1}{\text{s}}$$

bit



analogia immagini - suoni

- **analogia** fra il procedimento di digitalizzazione delle immagini e quello dei suoni:
 - scala dei valori sonori \Leftrightarrow tavolozza colori
 - frequenza di campionamento \Leftrightarrow numero pixel
- la scala dell'intensità sonora (numero di “*suoni differenti*”) e la frequenza di campionamento determinano la **qualità** del suono



suoni digitali

- come nel caso delle immagini la rappresentazione digitale dei suoni comporta un **elevato numero di byte**
- per **60 secondi** di audio
 - con rappresentazione a **8 bit** dell'intensità sonora e un campionamento a **8000 Hertz** sono necessari circa **660 Kbyte** (*qualità telefonica*)
 - con rappresentazione a **16 bit** dell'intensità sonora e un campionamento a **44 000 Hertz** i byte sono necessari circa **5 Mbyte** (*10Mb stereo*)

mp3

- analogamente alle immagini vengono usate **rappresentazioni compresse**
- la più nota è **MP3** (*Moving Picture Export Group Layer 3*)
 - l'**orecchio** umano è in grado di percepire solo suoni che stanno all'interno di un certo intervallo di frequenze
 - i suoni a frequenze superiori (**ultrasuoni**) o inferiori (**infrasuoni**) vengono eliminati dalla rappresentazione
 - questo, associato ad **altri procedimenti di compressione** permette di ridurre fino a oltre **12 volte** la quantità di dati digitali nella rappresentazione del suono senza un'apparente perdita di qualità



midi

- analogamente con quanto visto per le immagini vettoriali, nel caso di suoni prodotti da strumenti musicali, è possibile rappresentare, al posto del suono, la sequenza di **azioni** necessarie per **generarlo**
- si parla in questo caso di **suono sintetizzato**
- un esempio di questo tipo sono i suoni **MIDI** (*Musical Instrument Digital Interface*) nei quali vengono registrati gli eventi che generano un certo suono

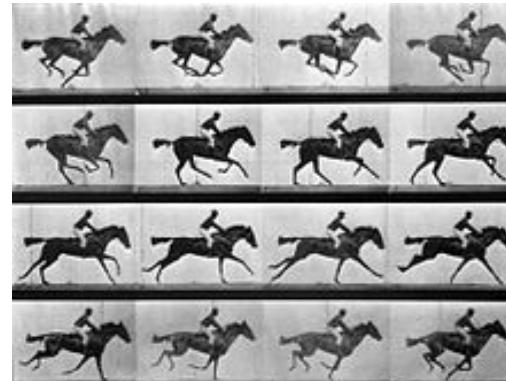


rappresentazione di informazioni multimediali

filmato digitale

filmati

- prendiamo come modello una pellicola cinematografica:
 - una sequenza di immagini statiche (fotogrammi)
 - una o più bande per il sonoro
- l'occhio umano non riesce a percepire come distinte due immagini separate da meno di un trentesimo di secondo



codifica filmati

- ogni singolo fotogramma viene digitalizzato utilizzando i procedimenti visti per la rappresentazione delle immagini
- la colonna sonora subisce lo stesso processo di conversione che abbiamo incontrato trattando i suoni digitali

- **problema** legato all'occupazione di memoria
 - (*soprattutto per trasmissione*)
- procedimenti di **compressione** per ridurre la dimensione
 - spesso solo una parte dell'immagine varia da un fotogramma al successivo
 - rappresentazione del fotogramma di partenza e poi solo della parte che in ogni fotogramma è differente dal precedente
- **fattori** che determinano la quantità di memoria:
 - **lunghezza** della sequenza
 - dimensione in **pixel**
 - numero di **colori**
 - numero di fotogrammi al secondo (**frame rate**)
 - qualità del **sonoro**



rappresentazione di informazioni multimediali

documenti strutturati

documenti strutturati

- **struttura logica**
 - determina il *ruolo* della varie parti del testo
 - *titoli, testo, note, etc.*
- **struttura grafica**
 - assegna una *resa grafica* ai ruoli
 - determina la resa grafica del documento nel suo complesso
 - “stampa” in modo diverso ciò che ha ruolo diverso
- **word processing**
 - non tanto scrivere, ma ingegnerizzare informazione



WYSIWYG

- *What You See Is What You Get*
 - focus su **grafica**, si perde di vista la struttura logica
 - grafica: non con i comandi grafici...
 - ma definendo gli stili delle varie parti di doc, come ruoli logici
 - es. stili di Word/Writer: “Titolo”, “Nota in Calce”, “Intestazione”
- non nomi grafici, ma logici
- in alternativa: editing basato su **comandi** o su **tag**

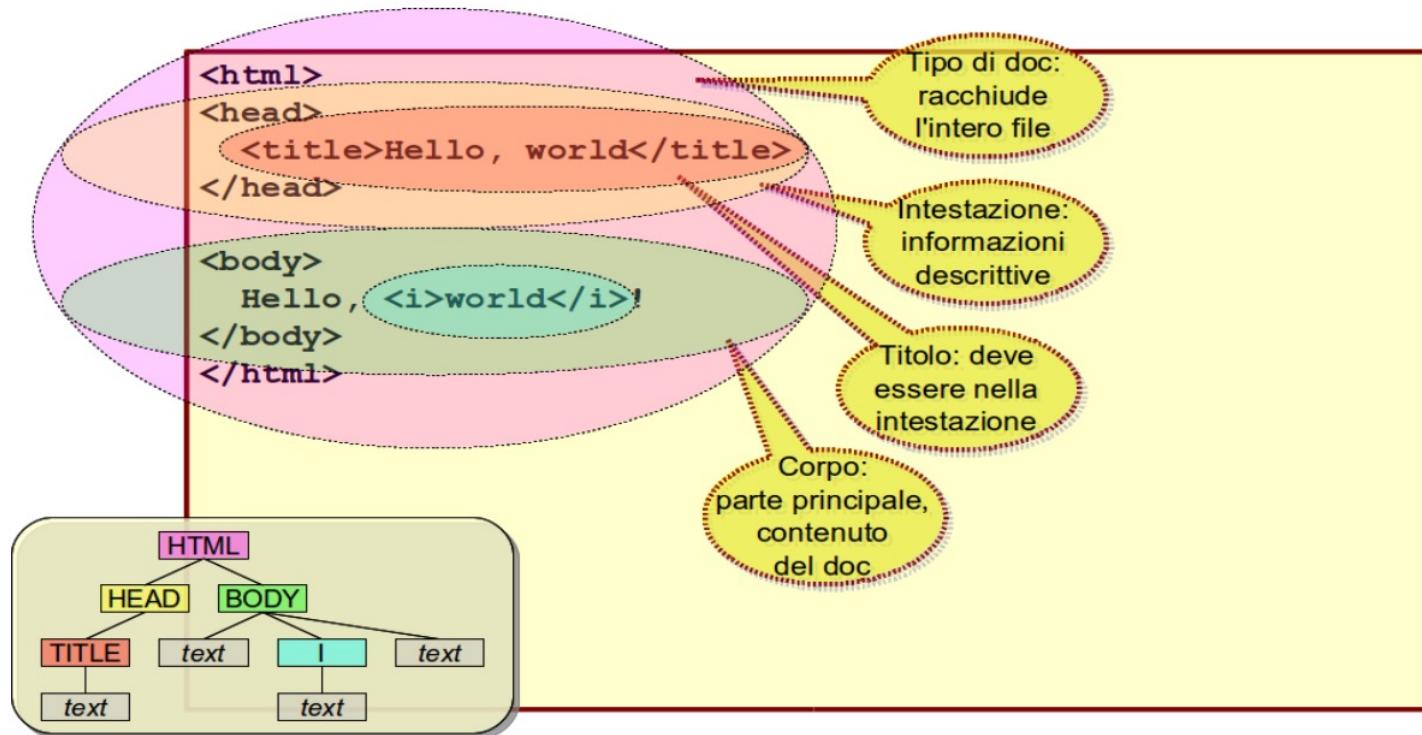


LATEX

HyperText Markup Language

- documenti strutturati
 - standard **W3C**: <http://www.w3.org/html/>
- HTML dichiara tipi di elementi
 - paragrafi, titoli, liste, collegamenti ipertestuali, elementi multimediali ecc.
- tipo di elemento descritto da tre parti
 - tag di **apertura**, **contenuto**, tag di **chiusura**
 - Bla bla, **in grassetto.**, normale.
 - molti tag permettono la definizione di attributi
 - <a href="<http://www.unipr.it/>">UniPR
- ***id*** e ***class***: attributi generici per assegnare ***ruoli logici***

struttura pagina HTML



tag di formattazione testo

```
<h1>Il titolo più grande</h1> ...
<h6>Il titolo più piccolo</h6>

<p>Questo è un paragrafo.<br />A-capo ma stesso paragrafo.</p>

<ul>
  <li>Primo elemento di una lista non ordinata.</li>
  <li>Secondo elemento, <b>in grassetto</b>.</li>
</ul>

<div class="remark" id="r01">
  Struttura generica di livello blocco, con un elemento generico
  <span class="techy">inline</span>.
</div>
```

- nuovi elementi di struttura di Html 5
 - header, main, nav, aside, footer
 - article, section, details, summary
 - menu, menuitem, figure, figcaption
- altri nuovi elementi
 - video, audio, canvas, embed
 - mark, time
 - output, progress, meter, datalist



Uniform Resource Locator

- *URL* è un *riferimento* per una *risorsa*
- il nome della risorsa dipende interamente dal protocollo
 - Per HTTP include:
 - nome dell'*host* su cui risiede la risorsa
 - numero di *porta* cui collegarsi (default = 80)
 - *percorso* della risorsa sulla macchina
 - stringa di *query* (dopo ?)
 - *frammento*: id di un elemento all'interno della risorsa (dopo #)

