Authors and contributors
Fabio Proietti

Feedback
Please direct any comments or suggestions about this document to
fabio.proietti AT istruzione DOT it

Publication date and  version

last modified 2012-10-12

bibliografia

http://it.wikipedia.org/wiki/Ascii

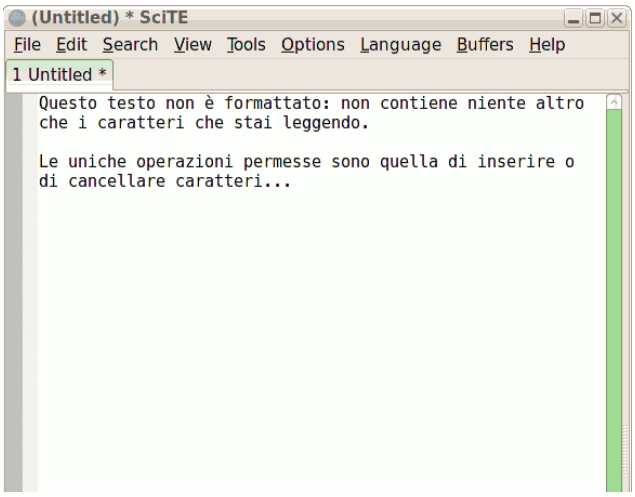
http://en.wikipedia.org/wiki/Western_Latin_character_sets_%28computing%29

## *I formati digitali dei dati*

La parola digitale deriva dall'inglese "digit" che significa "cifra numerica". Il primo esempio di comunicazione “digitale” è stato forse il codice Morse, risalente al 1840, che usava il punto, la linea e le pause per rappresentare le lettere e le parole. Oggi i computer rappresentano le lettere attraverso i numeri.
I dati vengono sempre salvati in modo numerico, ma possono essere disposti, raggruppati e organizzati in molti modi diversi a seconda delle necessità. Ogni organizzazione diversa è un diverso **formato**. Di solito ad ogni formato corrisponde una propria **estensione**, ma l'estensione fa parte del nome del file e il nome non garantisce sempre una corretta descrizione del contenuto del file (qualcuno potrebbe averlo rinominato).
Anche se oggi esistono file di diversi tipi (immagini, suoni, video, ecc.) è particolarmente importante provare ad approfondire il formato usato per i file di tipo **testuale**, sia perché sono semplici da studiare, ma anche perché sono molto usati.

### *Il testo non formattato*

Dopo aver scritto un documento al computer è possibile scegliere il formato in cui salvarlo, ad esempio, si potrebbe scegliere tra l'estensione .txt e l'estensione .doc.
La nostra scelta determinerà il modo in cui i dati verranno organizzati all'interno della memoria del computer. Per effettuare una scelta consapevole bisogna conoscere il significato delle **estensioni** dei file.
Ad esempio, è poco utile cambiare il colore del testo e poi salvarlo usando l'estensione **.txt**, un'estensione usata per il testo "non formattato" (plain text).
Un programma che permette di creare facilmente testo non formattato è il blocco note, o notepad.

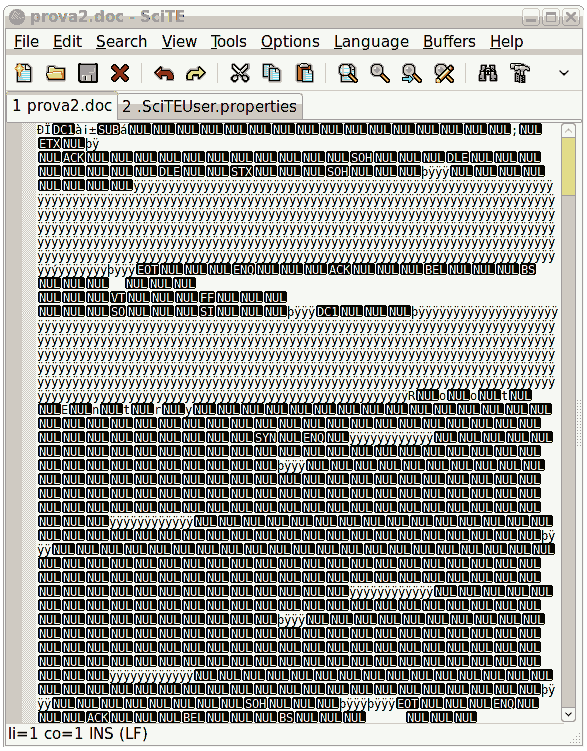


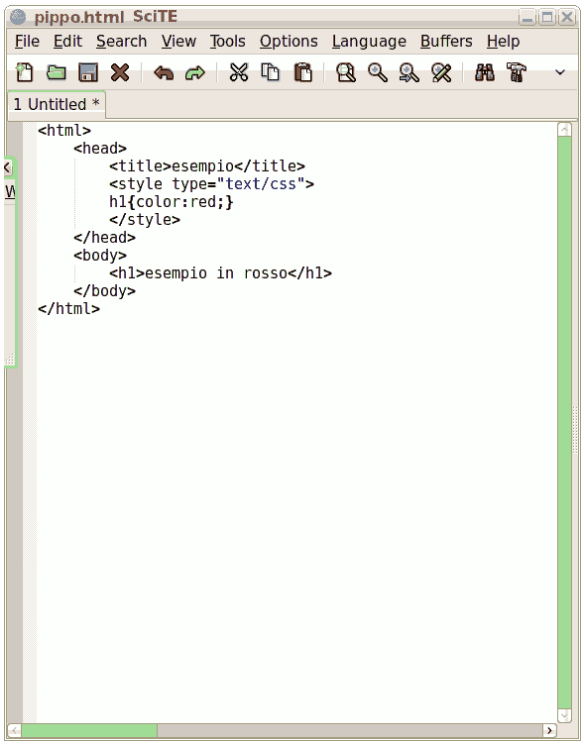
## *Il testo formattato*

Quando al testo viene applicato un particolare "stile grafico", si parla di "testo formattato" (rich text). Se vogliamo colorare o usare il grassetto sul testo dobbiamo usare un programma di scrittura che consente di formattare il testo.
Nel prossimo esempio si analizzeranno due diversi file, che hanno lo stesso contenuto: una semplice parola, colorata di **rosso**. Il primo, però, sarà con estensione **.doc**, mentre l'altro con estensione **.htm**.
Entrambe le estensioni consentono di ottenere un testo formattato.

Per aprire questi file useremo un semplice editor di testo (come il blocco note) Aprendo il file .doc vedremo una lunga lista di strani caratteri, come quella in basso a sinistra. È abbastanza difficile pensare che sia necessario tutti questi caratteri per poter memorizzare una sola parola colorata di rosso... Infatti, la stessa parola in rosso, in un file.htm, appare memorizzata in modo molto più semplice (a destra).

I caratteri strani sono in realtà Byte (numeri) che non dovrebbero essere visualizzati come caratteri in un editor, ma che ci permettono comunque di avere un'idea del contenuto del file.




In informatica la parola **formattare** ha diversi significati che possono essere confusi:

- formattare un testo vuol dire dare una forma estetica al testo
- formattare una unità di memoria vuol organizzare un'area di memoria in modo che vi possano essere memorizzati i dati.

## *Il formato dei dati*

Le domande che ci possiamo porre sono:

- Quale è il migliore formato per il testo formattato, ad esempio, tra .doc e .htm?
- Perché il formato .doc è così complicato rispetto al .htm, se entrambi riescono a produce la stessa cosa?

La semplicità del formato html consente facilmente di capire il suo contenuto anche usando un semplice editor come il notepad, e consente:

1. di **condividere** più facilmente i nostri file con gli altri;
2. di mantenere la **portabilità** dei dati, anche tra sistemi operativi molto diversi tra loro;
3. di **risparmiare** il costro della licenza di un programma che potrebbe essere indispensabile per leggere e/o modificare i file;
4. di poter **conservare e riutilizzare** per molto tempo importanti dati.

Infatti, se usiamo il formato .doc, nessuno ci garantisce che i computer del futuro saranno in grado di aprire un formato così "strano" come quello proposto da Microsoft Word. Questo dipende dalle scelte di Microsoft.
Da questo punto di vista non sembra una scelta saggia usare il formato .doc. Chi ci guadagna ad usare questo formato? Ovviamente, solo chi vende il programma... Chi lo usa quali vantaggi ha?

Il formato .doc è un formato "segreto" o meglio "proprietario", mentre il formato .htm è un formato "aperto". Chi usa un formato proprietario diventa **dipendente** dal proprietario di quel formato di dati. Un formato aperto invece è facile da comprendere anche senza l'aiuto di nessuno.
Siamo ancora abbastanza tranquilli, anche se il Ministero della Difesa italiano utilizza un formato di dati proprietario (cioè, segreto) di un'azienda americana?

Attività: Navigando su wikipedia.org è possibile leggere anche il codice sorgente delle pagine (aprendo un articolo e cliccando su modifica). In che formato vengono salvati i dati di wikipedia per consentire una collaborazione più facile tra gli utenti?

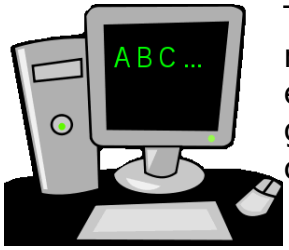
## *I numeri nel computer*


All'interno del computer i numeri interi (che possono misurare fino 4 Byte) possono essere memorizzati usando diversi metodi che si differenziano tra loro per l'ordine con cui si dispongono le cifre (i Byte): ad esempio, si può memorizzare e leggere le cifre del numero da sinistra verso destra o viceversa, cioè il numero 23 può essere rappresentato in memoria come 0023 oppure 2300, utilizzando, rispettivamente, le rappresentazioni **big-endian** o **little-endian**. Questa scelta dipende dall'architettura elettronica della CPU di un computer. Ad esempio le CPU **Intel** adottano la disposizione **big-endian** mentre le CPU **Motorola** quella **little-endian**.

Se un file viene condiviso tra due computer che usano una diversa rappresentazione dei numeri interi, possono verificarsi dei problemi nell'interpretazione di questi numeri. Per questa ragione, si può aggiungere nei file un codice BOM (Byte Order Mark) che specifica il tipo di disposizione delle cifre (big-endian o little-endian), altrimenti sarebbe impossibile comprendere la disposizione dei numeri. Al momento del salvataggio di un file contenente solo

numeri interi, dovrebbe essere possibile scegliere la convenzione da utilizzare.

## *I caratteri nel computer*




Tutto ciò che si trova nella memoria del computer è rappresentato come un numero, anche la musica, le immagini e il testo. In un file di testo, per trasformare le lettere e tutti gli altri simboli in numeri, si utilizza una tabella di conversione, che assomiglia molto alla tabella ASCII.

| simbolo | numero |
|---|---|
| A | $(65)_{10}$ |
| B | $(66)_{10}$ |
| C | $(67)_{10}$ |

Su ogni riga della tabella c'è un simbolo e il numero corrispondente. I numeri sono astrazioni, ma sulla carta possono essere rappresentati in diversi modi equivalenti tra loro: base 2, base 10 o base 16. Il suo valore (e il carattere a cui è associato) è unico.

Nel testo "formattato" il computer può rappresentare graficamente il carattere “A” usando diversi fonti tipografiche (in inglese, font), come Arial, Serif, ecc., ma questo solo se si lavora usando un testo formattato.

Se si usa il blocco note, il testo non è formattato e la rappresentazione grafica non ha nessuna importanza, quello che viene memorizzato è solo il valore numerico del simbolo “A” e questo non dipende dal font. Chi leggera quel valore numerico, per poterlo interpretare correttamente, dovrà usare la stessa codifica, ma questo non è garantito perché in diversi computer sono in uso diverse codifiche.

## *La codifica dei caratteri*

In passato, per memorizzare i simboli della tastiera, era adottata la tabella di codifica ASCII che conteneva solo 127 elementi (elementi dell'alfabeto inglese e altri simboli), per i quali era sufficiente usare un singolo Byte per ogni carattere. Grazie a questo non si poneva il problema della memorizzazione di numeri con più cifre (big-endian o little-endian).

Nelle seguenti tabelle sono elencati prima i caratteri non stampabili, come quello corrispondente al tasto "invio" (return) o "esc", e poi quelli stampabili.

| Dec | Abbrev. | CS | CEC | Descrizione |
|---|---|---|---|---|
| 0 | NUL | ^@ | \0 | Null character |
| 1 | SOH | ^A | | Start of Header |
| 2 | STX | ^B | | Start of Text |
| 3 | ETX | ^C | | End of Text |
| 4 | EOT | ^D | | End of Transmission |
| 5 | ENQ | ^E | | Enquiry |
| 6 | ACK | ^F | | Acknowledgment |
| 7 | BEL | ^G | \a | Bell |
| 8 | BS | ^H | \b | Backspace (in molti  comune che pda il codice Delete") |
| 9 | HT | ^I | \t | Horizontal Tab |
| 10 | LF | ^J | \n | Line feed |
| 11 | VT | ^K | \v | Vertical Tab |
| 12 | FF | ^L | \f | Form feed |
| 13 | CR | ^M | \r | Carriage return |
| 14 | SO | ^N | | Shift Out |
| 15 | SI | ^O | | Shift In |
| 16 | DLE | ^P | | Data Link Escape |
| 17 | DC1 | ^Q | | Device Control 1 (oft. XON) |
| 18 | DC2 | ^R | | Device Control 2 |
| 19 | DC3 | ^S | | Device Control 3 (oft. |
| 20 | DC4 | ^T | | Device Control 4 |
| 21 | NAK | ^U | | Negative dgement |
| 22 | SYN | ^V | | Synchronous Idle |
| 23 | ETB | ^W | | End of Trans. Block |
| 24 | CAN | ^X | | Cancel |
| 25 | EM | ^Y | | End of Medium |
| 26 | SUB | [^Z | | Substitute |
| 27 | ESC | ^[ | | Escape |
| 28 | FS | ^\ | | File Separator |
| 29 | GS | ^] | | Group separator |
| 30 | RS | ^^ | | Record Separator |
| 31 | US | ^_ | | Unit Separator |
| 127 | DEL | ^? | | Delete (in molti sistemi è comune che abbia un altro codice) |

| Dec | Glifo | | Dec | Glifo | | Dec | Glifo |
|---|---|---|---|---|---|---|---|
| 32 | Spazio | | 68 | D | | 105 | i |
| 33 | ! | | 69 | E | | 106 | j |
| 34 | " | | 70 | F | | 107 | k |
| 35 | # | | 71 | G | | 108 | l |
| 36 | $ | | 72 | H | | 109 | m |
| 37 | % | | 73 | I | | 110 | n |
| 38 | & | | 74 | J | | 111 | o |
| 39 | ' | | 75 | K | | 112 | p |
| 40 | ( | | 76 | L | | 113 | q |
| 41 | ) | | 77 | M | | 114 | r |
| 42 | * | | 78 | N | | 115 | s |
| 43 | + | | 79 | O | | 116 | t |
| 44 | , | | 80 | P | | 117 | u |
| 45 | - | | 81 | Q | | 118 | v |
| 46 | . | | 82 | R | | 119 | w |
| 47 | / | | 83 | S | | 120 | x |
| 48 | 0 | | 84 | T | | 121 | y |
| 49 | 1 | | 85 | U | | 122 | z |
| 50 | 2 | | 86 | V | | 123 | { |
| 51 | 3 | | 87 | W | | 124 | \| |
| 52 | 4 | | 88 | X | | 125 | } |
| 53 | 5 | | 89 | Y | | 126 | ~ |
| 54 | 6 | | 90 | Z | | 127 | DEL |
| 55 | 7 | | 91 | [ | | | |
| 56 | 8 | | 92 | \ | | | |
| 57 | 9 | | 93 | ] | | | |
| 58 | : | | 94 | ^ | | | |
| 59 | ; | | 95 | _ | | | |
| 60 | < | | 96 | ` | | | |
| 61 | = | | 97 | a | | | |
| 62 | > | | 98 | b | | | |
| 63 | ? | | 99 | c | | | |
| 64 | @ | | 100 | d | | | |
| 65 | A | | 101 | e | | | |
| 66 | B | | 102 | f | | | |

Oggi, per poter adottare anche lettere di alfabeti diversi da quello inglese, come quello italiano, francese, greco o cirillico, sono stati sviluppati nuove codifiche dei caratteri, come, ad esempio: ISO 8859, Windows-1252 e UTF-8.

Vediamo le differenze tra questi tre tipi:

- Lo standard internazionale ISO 8859 si divide in 16 gruppi di alfabeti. Ad esempio, l'alfabeto dell'europa occidentale si trova nella ISO 8859-1 (detta anche latin1) e nella ISO 8859-15 (che contiene anche il simbolo dell'euro).
- Microsoft, per i suoi sistemi operativi, per il gruppo di lingue dell'europa dell'ovest, ha creato la codifica chiamata Windows-1252 che è molto simile alla ISO 8859-1, ma rispetto alla quale possiede alcune piccole differenze. Questo causa, anche oggi, l'errata convinzione che Windows-1252 e ISO 8859-1 siano la stessa cosa, ma il simbolo euro e alcune virgolette non corrispondono. Oltre a questo problema, la codifica Windows-1252 viene chiamata nei programmi del sistema operativo Microsoft con il termine ANSI (American National Standards Institute), ma questa codifica non è mai stata uno standard ANSI.
- Universal Character Set (UCS) è un insieme di caratteri, che contiene circa 100000 simboli, definiti da un nome e un numero intero. Tale numero può essere usato per memorizzare un carattere nella memoria del computer. La più diffusa codifica usata dai computer per codificare nelle memorie dei calcolatori i caratteri UCS è l'UTF-8, ma essendo una codifica che utilizza fino a 4 Byte, si può distingure tra UTF-8 big-endian, little-endian, con o senza BOM.

Gli ultimi sistemi operativi Linux adottano la codifica UTF-8.

Gli ultimi sistemi operativi Microsoft adottano la codifica Windows-1252.

Gli ultimi sistemi operativi Apple adottano la codifica macintosh (Mac OS Roman).

La maggior parte di queste codifiche è compatibile con la codifica ASCII, nel senso che riescono a mostrare correttamente i vecchi file di testo. Grazie a questa compatibilità verso il passato, tutte le codifiche si possono sovrapporre per i primi 127 elementi e per lo stesso motivo la memoria occupata da un "vecchio" ASCII carattere è ancora un Byte.

Quando nei file di testo si usano anche altri caratteri, invece, diventa difficile condividerli tra i tre precedenti sistemi operativi. Come è possibile, allora, che questi tre sistemi navighino sulle pagine internet senza troppe difficoltà? La codifica usata dalle pagine web è quasi sempre specificata al loro interno e il browser deve solo adottare la codifica richiesta.

Ricordiamo, comunque, che sul web è raccomandato l'uso della codifica UTF-8, la stessa usata anche dal sistema operativo Linux.

attività: Aprire l'editor del testo (notepad di Windows) e digitare 123 sul tastierino numerico tenendo premuto il tasto ALT: si possono generare anche simboli che non sono presenti sulla nostra tastiera. Chi usa il sistema operativo GNU Linux, può usare un altro editor di testo come Gedit e, tenendo premuto CRTL + SHIFT, digitare 'u', seguita da un numero esadecimale (da 2 a 8 cifre) (come U+20AC).

Esistono programmi per generare immagini in ASCII Art

Quando si usa un file di testo formattato, non vi è più il problema della codifica, perché di solito questa viene specificata all'interno dello stesso file.