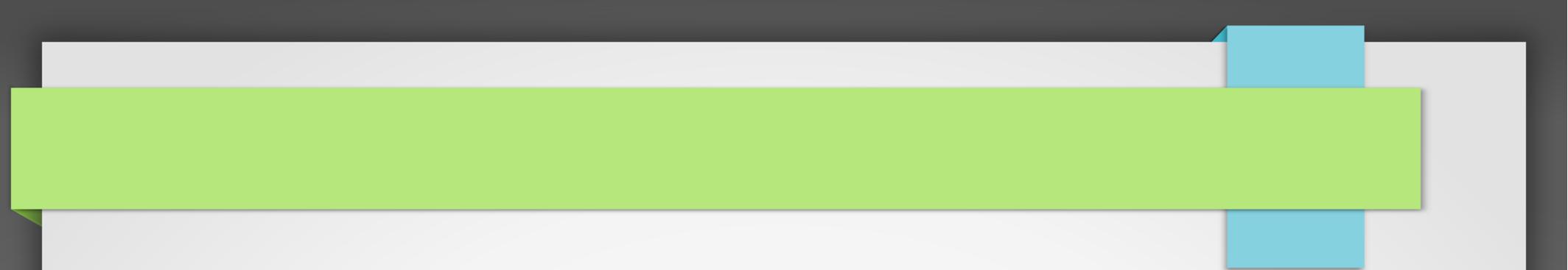


# Rappresentazione binaria e dati

Loriano Storchi

[loriano@storchi.org](mailto:loriano@storchi.org)

<http://www.storchi.org/>



# BREVE DIGRESSIONE

# Rappresentazione binaria dei numeri (breve digressione)

- In un sistema di numerazione posizionale data la base questa definisce direttamente il numero di simboli (cifre) che si usano per scrivere il numero.
  - Ad esempio nel sistema decimale usiamo 10 simboli (0,1,2,3,4,5,6,7,8,9)
- I moderni sistemi di numerazione sono posizionali, quindi il numero e' scritto specificando l'ordine delle cifre, ed ogni cifra assume un valore a seconda della sua posizione
  - Ad esempio  $423 = 4 * 10^2 + 2 * 10^1 + 3 * 10^0$

Se volete 4 centinaia 2 decine e 3 unita'

# Rappresentazione binaria dei numeri (breve digressione)

- In generale data una base  $b$  avro'  $b$  simboli (cifre) e quindi un numero intero  $N$  sara' scritto come:
  - Valore  $N = c_n * b^n + c_{n-1} * b^{n-1} + \dots + c_0 * b^0$
- Similmente se ho un numero  $N = 0.c_1c_2\dots c_n$ 
  - Valore  $N = c_1 * b^{-1} + c_2 * b^{-2} + \dots + c_n * b^{-n}$
- Consideriamo adesso il caso piu' semplice quello delle rappresentazione di numeri interi senza segno (i Naturali)
- Se uso un sistema di numerazione con base  $b$  con  $n$  cifre potro' rappresentare un massimo di  $b^n$  numeri diversi, quindi tutti i numeri da 0 fino a  $b^n - 1$
- Ad esempio in base 10 e' chiaro che usando due cifre posso rappresentare tutti i numeri da 0 a 99 quindi  $10^2 = 100$  numeri distinti

# La base binaria

- Usando una base 2, quindi solamente due simboli 0 ed 1, sempre rimanendo nell'ambito della rappresentazione di numeri interi positivi usando n cifre potro' rappresentare al massimo tutti i numeri compresi fra 0 e  $2^n - 1$ 
  - Ad esempio usando due cifre potro' rappresentare 4 numeri distinti:
    - $00 = 0 * 2^1 + 0 * 2^0 = 0$
    - $01 = 0 * 2^1 + 1 * 2^0 = 1$
    - $10 = 1 * 2^1 + 0 * 2^0 = 2$
    - $11 = 1 * 2^1 + 1 * 2^0 = 3$

# La base binaria

- Un byte (un boccone) rappresenta modernamente la sequenza di 8 bit ed è divenuto storicamente l'elemento base dell'indirizzabilità e quindi l'unità di misura base dell'informazione.
- 8 bit significa che con 1 byte posso rappresentare  $2^8 = 256$  valori differenti. Quindi nel caso di numeri interi unsigned i numeri da 0 a 255. Se uso un bit per indicare il segno ad esempio 0 positivo ed 1 negativo, posso rappresentare i numeri interi da -128 a 127 (da 10000000 a 01111111)
- Oppure con 8 bit posso rappresentare 256 differenti caratteri.

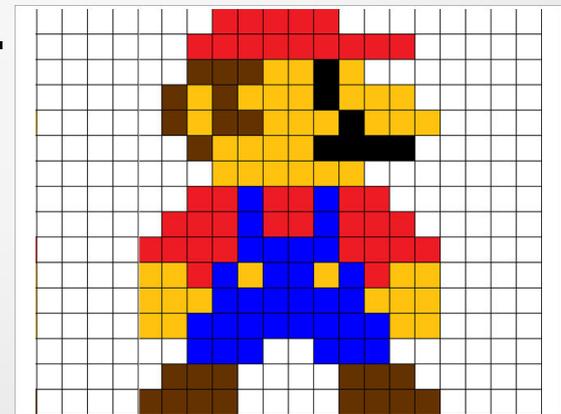
# ASCII

- Extended ASCII  
usa 8-bit
- ASCII originale  
US-ASCII 7-bit

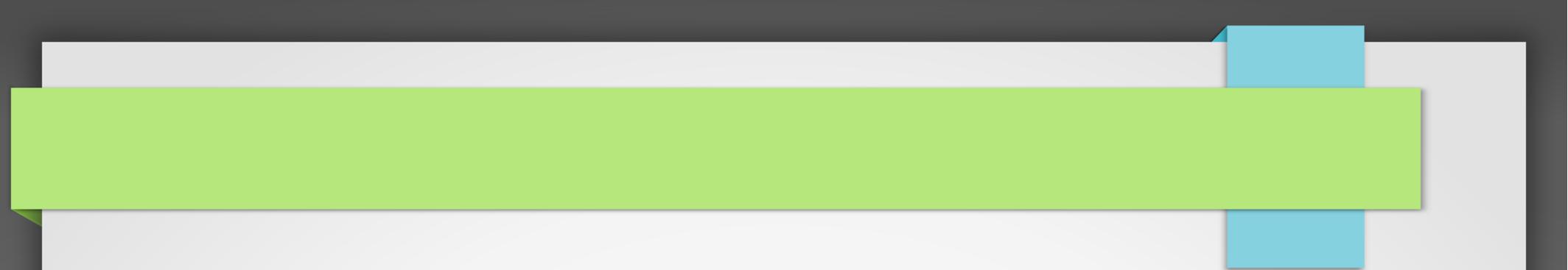
000	NUL	033	!	066	B	099	c	132	ä	165	ñ	198	ä	231	b
001	Start Of Header (SOH)	034	"	067	C	100	d	133	å	166	²	199	Å	232	þ
002	Start Of Text (STX)	035	#	068	D	101	e	134	å	167	³	200	Ā	233	Û
003	End Of Text (ETX)	036	\$	069	E	102	f	135	ç	168	¸	201	Ē	234	Ü
004	End Of Transmission (EOT)	037	%	070	F	103	g	136	è	169	©	202	Ĕ	235	Ý
005	Enquiry	038	&	071	G	104	h	137	é	170		203	Ė	236	ÿ
006	Acknowledge (ACK)	039		072	H	105	i	138	è	171	½	204	Ħ	237	Ÿ
007	Bell	040	(	073	I	106	j	139	ĩ	172	¾	205	=	238	˘
008	Backspace (BS)	041	)	074	J	107	k	140	ı	173	ı	206	Ī	239	˙
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	Ĳ	240	˚
010	Line Feed (LF)	043	+	076	L	109	m	142	Ā	175	»	208	Ĵ	241	¸
011	Vertical Tab	044	,	077	M	110	n	143	Ā	176	ˆ	209	Ķ	242	˘
012	Form Feed (FF)	045	-	078	N	111	o	144	Ē	177	˜	210	Ĭ	243	¼
013	Carriage Return (CR)	046	.	079	O	112	p	145	Ĕ	178		211	Ī	244	½
014	Shift Out	047	/	080	P	113	q	146	Ė	179		212	Ī	245	¾
015	Shift In	048	0	081	Q	114	r	147	ø	180	ı	213	Ĳ	246	˘
016	Dataline Escape (DLE)	049	1	082	R	115	s	148	ö	181	Ā	214	Ĳ	247	˙
017	DC 1 (XON)	050	2	083	S	116	t	149	ò	182	Ā	215	Ĳ	248	˚
018	DC 2	051	3	084	T	117	u	150	ú	183	Ā	216	Ĳ	249	˘
019	DC 3 (XOFF)	052	4	085	U	118	v	151	û	184	©	217	Ĳ	250	˙
020	DC 4	053	5	086	V	119	w	152	ÿ	185		218	Ĳ	251	˚
021	Negative Acknowledge (NAK)	054	6	087	W	120	x	153	ō	186	ı	219	█	252	˘
022	Synchronous Idle	055	7	088	X	121	y	154	Û	187	Ĳ	220	█	253	˚
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	Ĳ	221	ı	254	█
024	Cancel	057	9	090	Z	123	{	156	£	189	¢	222	ı	255	
025	End Of Medium	058	:	091	[	124		157	∅	190	¥	223	█		
026	Substitute	059	;	092	\	125	}	158	×	191	ı	224	ó		
027	Escape (ESC)	060	<	093	]	126	~	159	f	192	ı	225	ß		
028	File Separator	061	=	094	^	127 (DEL)	▯	160	á	193	ı	226	ö		
029	Group Separator	062	>	095	_	128	ç	161	ı	194	ı	227	ö		
030	Record Separator	063	?	096	`	129	ü	162	ó	195	ı	228	ø		
031	Unit Separator	064	@	097	a	130	é	163	ú	196	-	229	ó		
032	SPACE (SP)	065	A	098	b	131	ä	164	ñ	197	ı	230	ı		

# Codifica dell'informazione

- Una codifica e' una convenzione, come visto prima quindi la sequenza di bit 01001100 puo' rappresentare il carattere L (L maiuscolo) oppure se invece intendiamo un valore intero unsigned rappresenta il numero decimale 76.
- Ad esempio ogni immagine e' composta da pixel, se usassi 1 solo bit per ogni pixel potrei avere solo immagini in bianco e nero (1 pixel nero , 0 pixel bianco), se uso piu' bit per rappresentare ogni pixel posso invece avere range di grigi o colori. E da qui suoni, video ..







# I DATI LE VARIABILI

# Le variabili

- Un programma e' un set di istruzioni memorizzate nelle memoria di un computer. Queste istruzioni elaborano informazioni, dati e ne produce altra informazione.
- Visto che questa informazione durante l'esecuzione del programma varia, gli elementi del programma che contengono questi dati sono chiamati variabili
- In matematica ad esempio l'espressione  $y = m * x + q$  ogni lettera e' una variabile.
- Nei linguaggi di programmazione ogni variabile in pratica rappresenta una locazione nella memoria del computer dove sono memorizzati i dati
- La dichiarazione (allocazione , creazione) di una variabile rappresenta il momento in cui in memoria viene riservato lo spazio necessario a contenere quel dato

# Tipi di dato

- Vedi differenza fra linguaggi fortemente tipizzati e non, tipizzazione dinamica e statica
- I linguaggi di programmazione hanno dei tipi di dato nativi, come ad esempio gli integer, i floating-point, i boolean e i character. Diversi tipi diverse dimensioni e dunque diversi range

label	size (bytes)	smallest value	largest value
byte	1	-128	127
short	2	-32768	32767
int	4	-2147483648	2147483647
long	8	-9223372036854775808	9223372036854775807
char	2	0	65535

# Operazioni

- Posso ovviamente eseguire operazioni fra le variabili, esempio in C

```
int i = 5;
```

```
int j = 7;
```

```
int k;
```

← Dichiarazione

```
k = i + (j * 3) * 4;
```

A k viene assegnato il valore risultate dall'operazione di destra. In molti linguaggi C, C++ se scrivo  $k = 3/2$  il risultato sara' 1 non 1.5