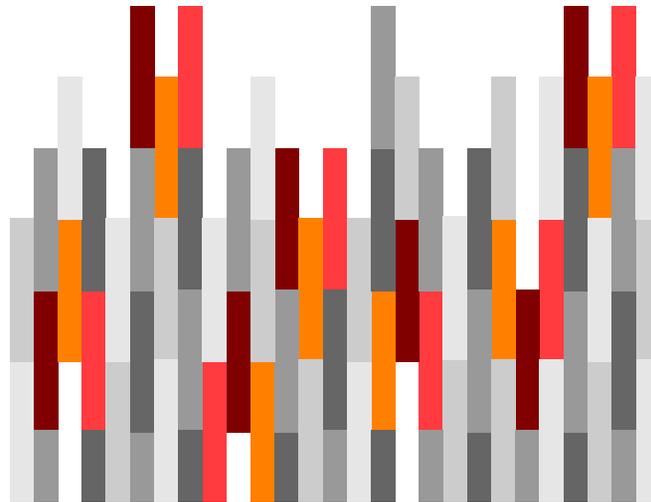


# Fluency



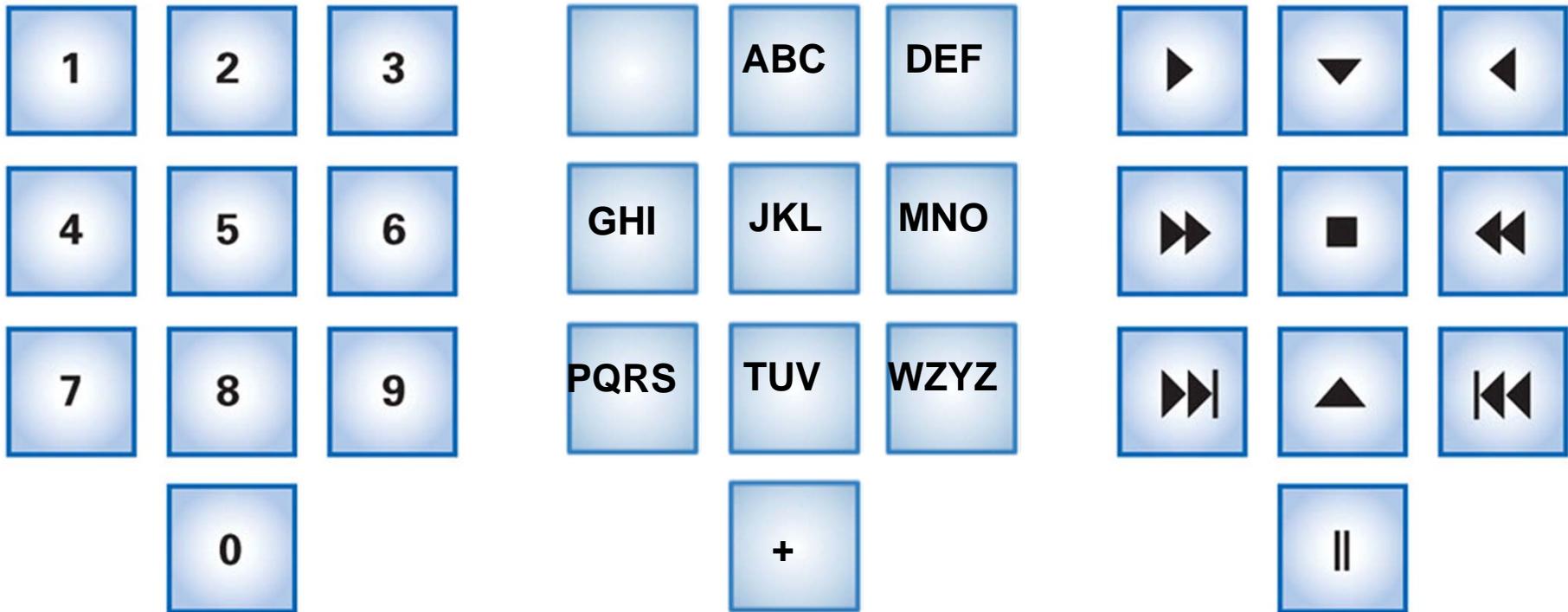
## La rappresentazione dell'informazione

Capitolo 5

# Quanti sono?

- Nel mondo ci sono **10** tipi di persone
  - quelli che capiscono il *codice binario* e quelli che non lo capiscono.

# Simboli tasti del telefonino



# Digitalizzare l'informazione

- Digitalizzare
  - rappresentare l'informazione **per mezzo di numeri**
- Discretizzazione
  - limitazioni dei **numeri disponibili**

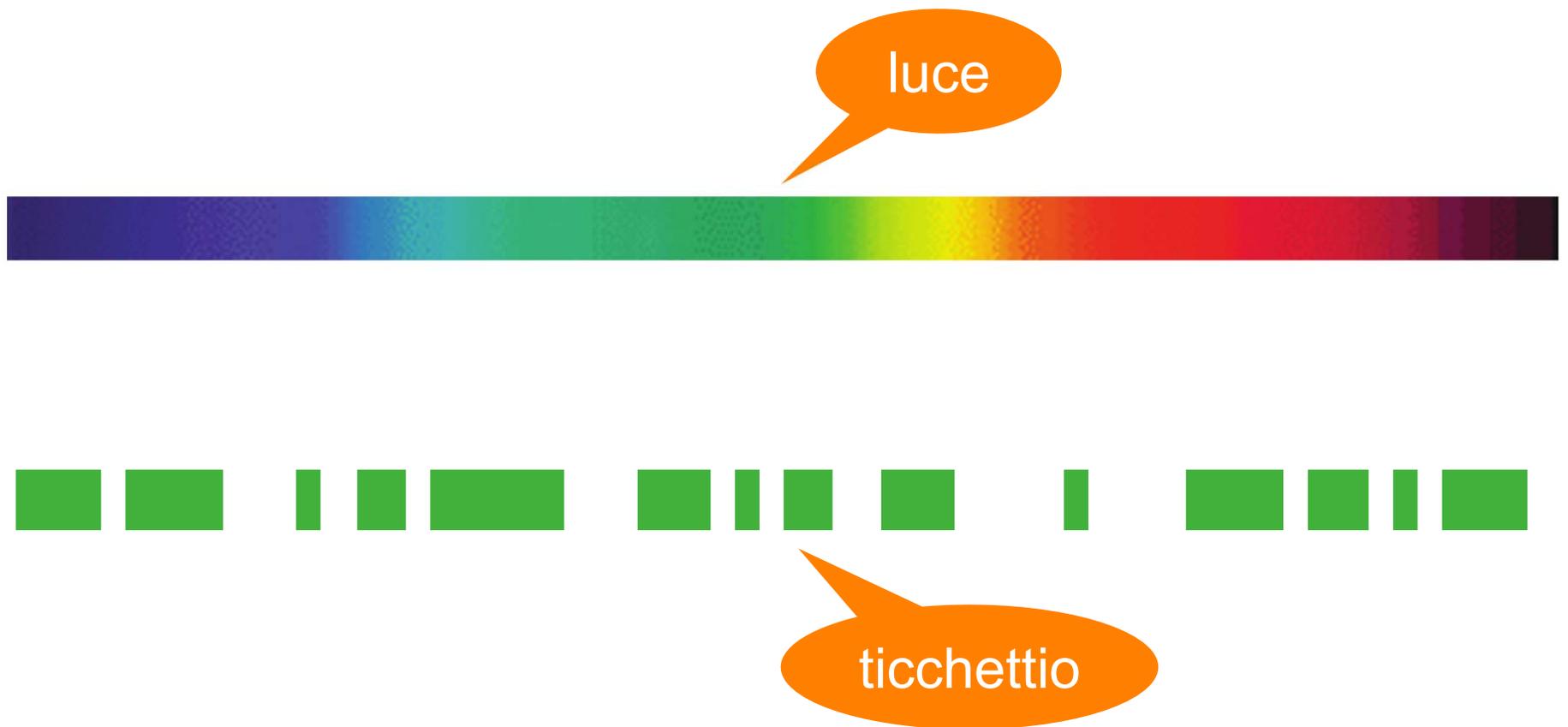
# Pro della digitalizzazione

- **Ordinare gli elementi**
  - definisce un ordinamento dei simboli
    - p.e.: dal “più piccolo” al “più grande”
- Scelta dei **simboli**
  - possono essere codificati con gli stessi numeri?

# Altre rappresentazioni

- Ogni insieme di simboli potrebbe rappresentare
  - una lettera,
  - un comando del player
  - ...

# Continuo vs discreto



# Rappresentazione binaria

- **Due soli simboli**
- Nel mondo fisico
  - la più semplice è la ***presenza*** o ***assenza*** di un fenomeno
- Nel mondo logico
  - i concetti di ***vero*** e ***falso***

# Implementare la logica

- Il *vero*
  - rappresentato con la *presenza* di un fenomeno fisico
- Il *falso*
  - rappresentato con la *assenza* dello stesso fenomeno

# Rappresentazione PandA

- PandA (*Presence and Absence*)
- È discreta
  - il fenomeno o
    - è presente
    - oppure non lo è (vero o falso)
  - non esiste alcuna gradazione di valori

# Un sistema binario

- Sistema binario
  - le due configurazioni di PandA
- L'unità base di PandA è il **bit**
  - abbreviazione di *binary digit*

# Alcuni PandA

Presenza	Assenza
Vero	Falso
1	0
Acceso	Spento
+	-
Si	No
Favorevole	Contrario
Yang	Yin
Lisa	Bart
...	...

# La memoria di un computer

- È organizzata come una **sequenza di bit**
  - ogni posizione registra la presenza o l'assenza
  - “leggibile” successivamente

# Analogia

- Un sentiero di blocchi di cemento
  - su ogni blocco **può esserci** un sasso
- Se su un blocco
  - **c'è un sasso**: rappresenta 1
  - **non c'è un sasso**: rappresenta 0

# Esempio



10100010

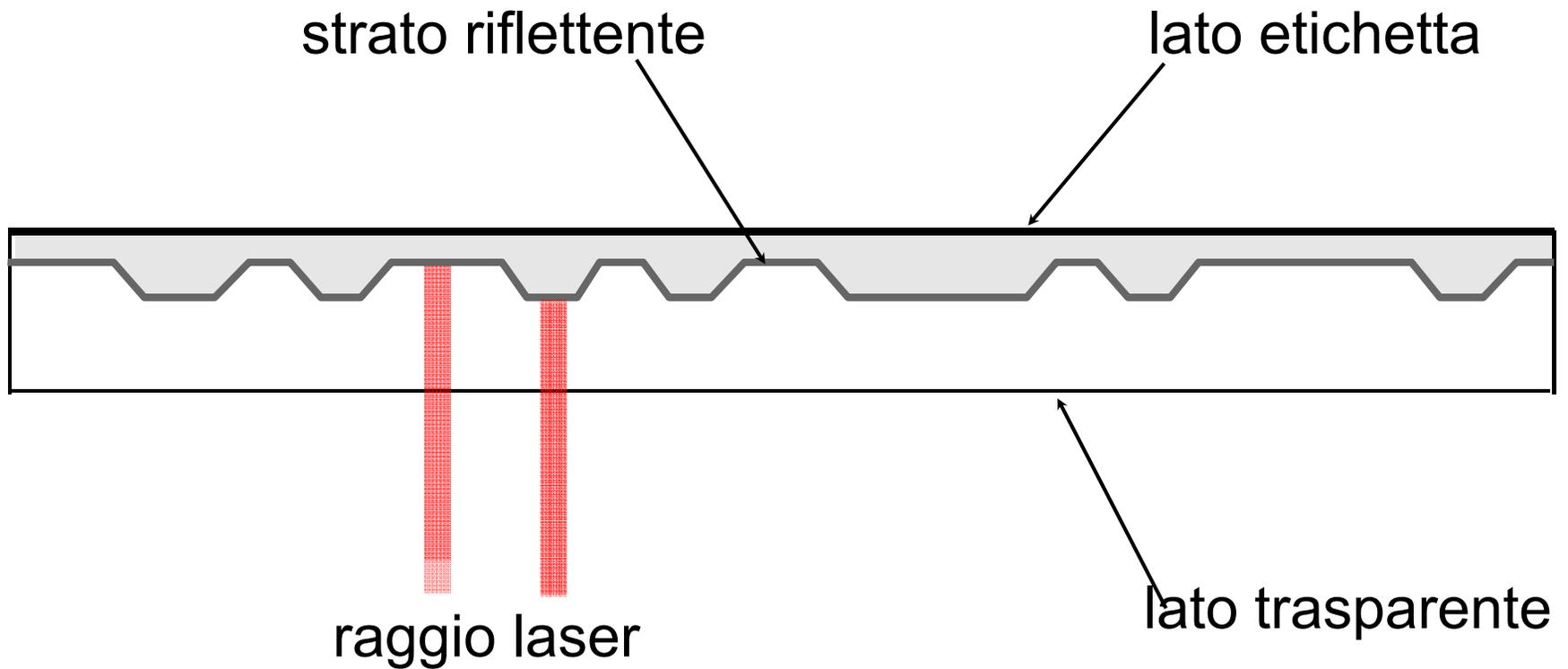
# Possibili codifiche

Symbol	Binary	Physical Bits	Hex	Symbol	Binary	Physical Bits	Hex
AAAA	0000		0	PAAA	1000		8
AAAP	0001		1	PAAP	1001		9
AAPA	0010		2	PAPA	1010		A
AAPP	0011		3	PAPP	1011		B
APAA	0100		4	PPAA	1100		C
APAP	0101		5	PPAP	1101		D
APPA	0110		6	PPPA	1110		E
APPP	0111		7	PPPP	1111		F

# Altre codifiche PandA

- **Sassi bianchi o neri** su ogni blocco
  - pietre su tutti i blocco, ma
  - *neri* per uno stato e *bianche* per l'altro
- Più pietre di due colori per blocco
  - la preponderanza di nere significa 0
  - la preponderanza di bianche significa 1

# Codifica del CD-ROM



# Combinare più bit

- Una *sequenza* di bit
  - rappresenta abbastanza simboli per codificare dati complessi
- PandA
  - 2 valori, con sequenze di lunghezza  $n$
  - possiamo creare  $2^n$  simboli

# Quanti simboli con $n$ bit

$n$	$2^n$	simboli
1	$2^1$	2
2	$2^2$	4
3	$2^3$	8
4	$2^4$	16
5	$2^5$	32
6	$2^6$	64
7	$2^7$	128
8	$2^8$	256
9	$2^9$	512
10	$2^{10}$	1024

# Esadecimale

- Cifre in base 16
- Perché usare le cifre *esadecimali*?
  - per noi umani scrivere una sequenza di bit è:
    - lungo,
    - noioso,
    - è facile commettere errori

# Esadecimale

- Possono essere rappresentate dai 16 simboli delle sequenze di 4 bit
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

# Binario → esadecimale

0000 = 0

0001 = 1

0010 = 2

0011 = 3

0100 = 4

0101 = 5

0110 = 6

0111 = 7

1000 = 8

1001 = 9

1010 = A

1011 = B

1100 = C

1101 = D

1110 = E

1111 = F

# AB è un numero?

- Se sì, per quali basi?

# Conversioni

- Binario → esadecimale
- Esadecimale → binario

# Binario → esadecimale

$$\boxed{10011} = 13$$

$$\boxed{1010111110} = 2BE$$

# Esadecimale → binario

1F = 

1	1111
---	------

D72 = 

1101	0111	0010
------	------	------

# Rappresentazione binaria

- Per la codifica di *numeri* e di *caratteri*
- Anche per *digitalizzare* i suoni, i video e altro

# Digitalizzare il testo

- Quali caratteri codificare?
  - più piccola la lista dei simboli
    - minori i bit necessari
  - minore il numero di bit
    - meno caratteri codificabili

# I caratteri da codificare

- 95 caratteri (lingua inglese)
  - 26 lettere minuscole e 26 maiuscole.
  - 10 cifre numeriche sono necessari 7 bit  $2^7=128$
  - 10 segni aritmetici,
  - 20 segni di interpunzione (spazi inclusi)
  - 3 caratteri non stampabili (a capo, tabulazione, ...)

# Codifica ASCII

- *American Standard Code for Information Interchange*
  - rappresentazione a 7 bit
- Limiti
  - non basta per rappresentare i caratteri dei linguaggi diversi dall'inglese
    - lingue latine, nord europee, ...

# Extended ASCII

- Estensione di ASCII a 8 bit (256 simboli)
  - la prima metà è l'ASCII originale
    - con 0 alla **MSD** di ogni gruppo di bit
- Vantaggi
  - codifica quasi tutti i linguaggi occidentali
  - include molti altri simboli utili

# Tabella extended ASCII

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>0</sub>	S <sub>1</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>Y</sub>	E <sub>Z</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	P <sub>T</sub>
1000	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	I <sub>N</sub>	N <sub>L</sub>	S <sub>S</sub>	E <sub>S</sub>	H <sub>S</sub>	H <sub>J</sub>	Y <sub>S</sub>	P <sub>D</sub>	P <sub>V</sub>	R <sub>I</sub>	S <sub>2</sub>	S <sub>3</sub>
1001	D <sub>C</sub>	P <sub>1</sub>	P <sub>2</sub>	S <sub>E</sub>	C <sub>C</sub>	M <sub>M</sub>	S <sub>P</sub>	E <sub>P</sub>	O <sub>S</sub>	O <sub>O</sub>	O <sub>A</sub>	C <sub>S</sub>	S <sub>T</sub>	O <sub>S</sub>	P <sub>M</sub>	A <sub>P</sub>
1010	A <sub>0</sub>	i	¢	£		¥		§	¨	©	♀	«	¬	-	®	—
1011	°	±	²	³	´	µ	¶	·	¸	¹	♂	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

# Codifica UNICODE

- Utilizza 32 bit
- Rappresenta anche i caratteri di alfabeti non europei
  - p.e.: asiatici, arabi, ebraici, cirillici, ...
- I primi 256 caratteri sono quelli di extended ASCII

# Codifica numero telefonico

- Il numero

Codifica ogni cifra in un byte  
ASCII

- come se  
computa

a di un

# Codifica ridondante

- Il codice per telecomunicazioni
  - lettere intelligibili anche in presenza di rumore
  - necessariamente inefficiente
- Es.: alfabeto NATO

# Codifica NATO

A Alpha	H Hotel	O Oscar	V Victor
B Bravo	I India	P Papa	W Whiskey
C Charlie	J Juliet	Q Quebec	X X-ray
D Delta	K Kilo	R Romeo	Y Yankee
E Echo	L Lima	S Sierra	Z Zulu
F Foxtrot	M Mike	T Tango	
G Golf	N November	U Uniform	

# Perché BYTE con la Y?

- Quantità di memoria intermedia
  - *bit* (bocconcino, assaggio)
  - *word* (parola) in generale lunga 32 bit
- Il termine bite (boccone) è un gruppo di 8 bit,
  - per evitare confusioni si decise di cambiare la *i* con la *y*

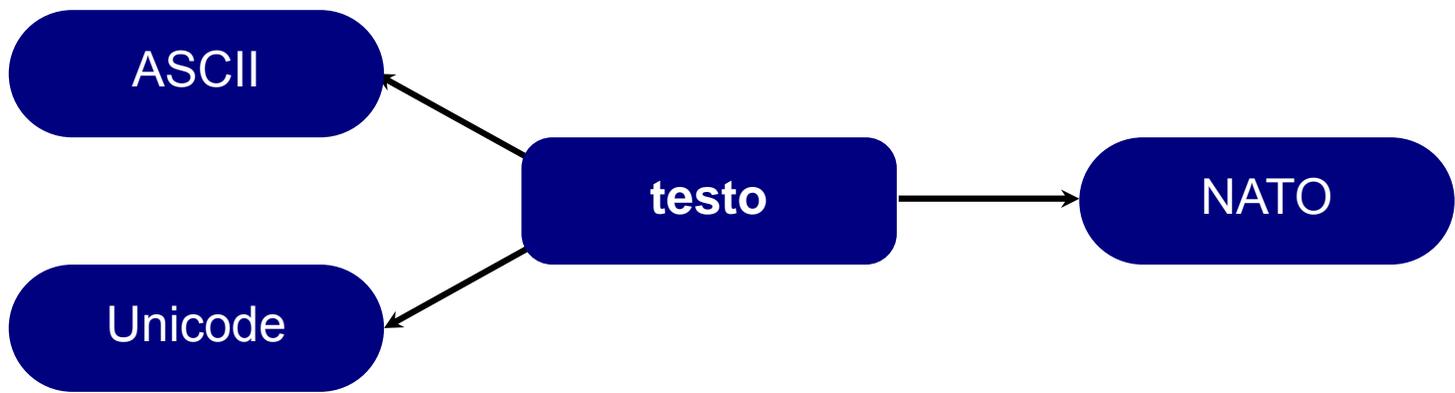
# Dato vs informazione

- I dati sono valori
  - non hanno significato ulteriore
- L'Informazione ha un significato

# Dato vs informazione

- 21
  - piacevole temperatura primaverile (in °Celsius)
  - ottima temperatura per sciare (in °Fahrenheit)
  - voto non troppo alto
  - piano intermedio di un grattacielo
  - anni che designavano i maggiorenni

**digitalizzazione  
discretizzazione**



**dato vs  
informazione**

