

Prologo + Parte I

Informazione digitale (lezione 1)

Che lingua parla un computer?

Informatica B
Università di Torino, Dipartimento di Psicologia
Antonio Lieto

Antonio Lieto

Ricercatore di Informatica(INF/01) presso il
Dipartimento di Informatica

Ricevimento: al Dip. di Informatica in via Pessinetto,
I piano (davanti Osp. Amedeo di Savoia, autobus: 3, 9, 59);

<https://www.antoniolieto.net>

appuntamento via e-mail

lieto@di.unito.it

<http://www.antonioletto.net/teach.html>

→ ↺ ⓘ Not Secure | www.antonioletto.net/teach.html



[back to the home page](#)

teaching activities

Academic Year 2018/2019

Professor for the Course on [Design and Evaluation of Cognitive Artificial Systems](#) at the Doctoral School in Computer Science, Department of Computer Science, University of Turin. This doctoral course introduces the main modelling methods, paradigms and technologies involved in the design of cognitive artificial systems, with a particular emphasis on Cognitive Architecture (CA).

Professore aggregato for the Course [Informatica B \(Information Science B\)](#) (6 ECFU) at the Department of Psychology, University of Turin. ([slides](#)) My teaching duties include 36 hours of frontal lessons where I teach a basic course in Computer Science (digital representation, computer architecture, operative systems, introduction to databases, networks, internet protocols, web technologies).

Professore aggregato for the Course on [Web Technologies](#) (6 ECFU) at the SAA School of Business Administration, Department of Management, University of Turin. My teaching duties include 42 hours of frontal lessons where I teach the basics of internet protocols, web languages (ex. html, xml, javascript, php) and human-computer interaction/cognitive ergonomics.

Academic Year 2017/2018

Professore aggregato for the Course [Informatica A \(Information Science A\)](#) and [Informatica B \(Information Science B\)](#) (6 ECFU) at the Department of Psychology, University of Turin. ([slides](#)) My teaching duties include 72 hours of frontal lessons where I teach a basic course in Computer Science (digital representation, computer architecture, operative systems, introduction to databases, networks, internet protocols, web technologies).

Professore a Contratto (Adjunct Professor) for the Course on [Web Technologies](#) (6 ECFU) at the SAA School of Business Administration, Department of Management, University of Turin. My teaching duties include 42 hours of frontal lessons where I teach the basics of internet protocols, web languages (ex. html, xml, javascript, php) and human-computer interaction/cognitive ergonomics.

Academic Year 2016/2017

Professore a Contratto (Adjunct Professor) for the Course on [Web Technologies](#) (6 ECFU) at the SAA School of Business Administration, Department of Management, University of Turin. My teaching duties include 42 hours of frontal lessons where I teach the basics of internet protocols, web languages (ex. html, xml, javascript, php) and human-computer interaction/cognitive ergonomics. **Students/Evaluation** of the course and of the teaching activity via Edumeter ([PDF](#)).

Professore a Contratto (Adjunct Professor) for the introductory course of "[Computer Science for Humanities](#)" (Humanities Computing) (9 ECFU) at the University of Genova (Italy). This was a course to the main concepts of computer science for Humanities faculties. The final part of the course was specifically devoted to a technological introduction of the field of "Digital Humanities" (Digital semantic technologies that can be useful for humanists).

My teaching duties were of 54 hours teaching based on frontal lessons and lab activities. "[Slides Corso](#)" (su AulaWeb). Statistics of the **Students/Evaluation** of the course and of the teaching activity ([PDF 2](#)).

Back to the [home page](#)

Teaching activities

Slides Corso Informatica B (Unito, Dipartimento di Psicologia)

A questo link (http://www.antoniolieto.net/tesi_infpsych_unito) potete consultare una pagina con diverse proposte di tesi.

Disponibile un pagina ([questa](#)) che genera automaticamente alcuni esercizi.

[Prologo \(Info su Programma e Mod. Esame\)](#) (PDF)

[Slides lezione 1](#) (PDF)

[Slides lezioni 2-4](#) (PDF)

[Slides su Software e Sistemi Operativi](#) (PDF)

[Slides su Reti di Calcolatori e Internet](#) (PDF)

[Slides Basi di Dati, DBMS, SQL](#) (PDF)

[Slides Intro Fogli di calcolo](#) (PDF), [Esercizio Serie di Fibonacci \(+ Eq. Retta\)](#), File [Temperature.txt](#)

[Slides Intro Web](#) (PDF)

[Slides Linguaggi Web](#) (PDF)

[Web 2.0, Analisi delle Interfacce](#) (PDF)

[Slides Scienze Cognitive Computazionali e IA](#) (cenni storici, **parte facoltativa**, PDF)

[Slides su Interfacce e Tecnologie Persuasive](#) (**parte facoltativa**, PDF)

Cenni di storia dell'informatica - Corso di "Informatica A e B" (Unito).

Organizzazione

- Lezione:
 - lunedì 12:00-14:00 (qui)
 - martedì 15.00-17.00 (aula

VARle: mail (1)

- Io rispondo alle mail

Buongiorno,
ho cercato le slides/le prove
d'esame, ma sul sito **non ci
sono**

Buongiorno,
sono dell'altro corso, ma
l'indirizzo del docente sul
sito **non c'è**

VARle: mail (2)

- Io rispondo alle mail

Buongiorno,
volevo sapere quando
iniziano le lezioni del **corso
di informatica**.

Buongiorno,
volevo sapere dove posso
trovare il materiale didattico
del **corso di informatica**.

Buongiorno,
volevo sapere quali sono i
libri di testo del suo **corso di
informatica**.

VARle: esami (1)

- Ho studiato tantissimo, è la quarantasettesima volta che provo...speriamo!!!!

Parte 2.B (22 punti; sufficienza: 12 punti)

testo

3. [14 punti] Dato il seguente frammento di schema relazionale,

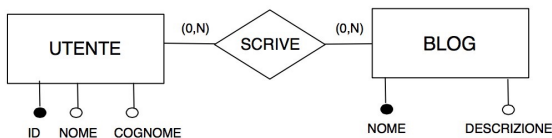
GIOCATORE(matricola, cognome, nome, ruolo)

TESSERATO(matricolagiocatore, nomesquadra)

- (i) [3 punti] indicare le chiavi primarie delle due tabelle sottolineando gli attributi appropriati (*SUGGERIMENTO: un giocatore può essere tesserato per una sola squadra*)
- (ii) [2 punti] indicare eventuali vincoli di integrità referenziale
- (iii) [9 punti] esprimere in SQL la seguente interrogazione (**non tralasciare la join**):

"Estrarre nome e cognome dei calciatori di ruolo PORTIERE che giocano nell'Udinese"

4. [8 punti] Costruire uno schema relazionale di una base di dati in modo che traduca il seguente schema entità-associazione (indicare anche gli eventuali vincoli di integrità referenziale).



Parte 2.B (22 punti; sufficienza: 12 punti)

consegnato

3. [14 punti] Dato il seguente frammento di schema relazionale,

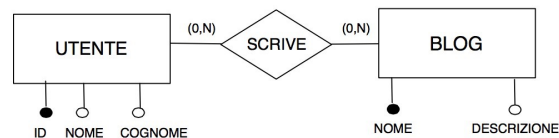
GIOCATORE(matricola, cognome, nome, ruolo)

TESSERATO(matricolagiocatore, nomesquadra)

- (i) [3 punti] indicare le chiavi primarie delle due tabelle sottolineando gli attributi appropriati (*SUGGERIMENTO: un giocatore può essere tesserato per una sola squadra*)
- (ii) [2 punti] indicare eventuali vincoli di integrità referenziale
- (iii) [9 punti] esprimere in SQL la seguente interrogazione (**non tralasciare la join**):

"Estrarre nome e cognome dei calciatori di ruolo PORTIERE che giocano nell'Udinese"

4. [8 punti] Costruire uno schema relazionale di una base di dati in modo che traduca il seguente schema entità-associazione (indicare anche gli eventuali vincoli di integrità referenziale).



VARLe: esami (2)

- Mi sono registrato all'esame, poi deve essere successo qualcosa...
- Ci deve essere un problema, perché le tasse le ha pagate mio zio e l'edumeter lo faccio proprio stasera...non può proprio fare niente?

Elenco Studenti Iscritti all'Appello								
#	▲ Data Iscr.	▲ Matricola	▲ Cognome e Nome	▲ Cod. AD	Anno Freq.	CFU	Esito	
1	18/09/2015			LIN0068 ⓘ	2014/2015	9	Assente	⊙
2	22/09/2015			LIN0068 ⓘ	2013/2014	9	Assente	⊙
3	15/09/2015			LIN0068 ⓘ	2014/2015	9	25	⊙
4	20/09/2015			LIN0068 ⓘ	2014/2015	9	Insufficiente	⊙
5	15/09/2015			LIN0068 ⓘ	2014/2015	9	Insufficiente	⊙
6	22/09/2015			LIN0068 ⓘ	2014/2015	9	30	⊙
7	18/09/2015			LIN0068 ⓘ	2012/2013	9	Insufficiente	⊙
8	19/09/2015			LIN0068 ⓘ	2014/2015	9	22	⊙
9	23/09/2015			LIN0068 ⓘ	2013/2014	9	21	⊙
10	18/09/2015			LIN0068 ⓘ	2013/2014	9	26	⊙
11	22/09/2015			LIN0068 ⓘ	2014/2015	9	27	⊙
12	23/09/2015			LIN0068 ⓘ	2012/2013	9	27	⊙

VARLe: esami (3)

Parte 2.B (22 punti; sufficienza: 12 punti)

consegnato

3. [14 punti] Dato il seguente frammento di schema relazionale,

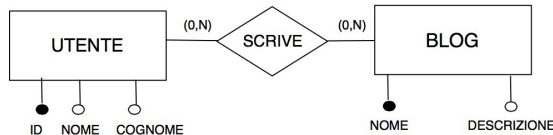
GIOCATORE(matricola, cognome, nome, ruolo)

TESSERATO(matricolagiocatore, nomesquadra)

- (i) [3 punti] indicare le chiavi primarie delle due tabelle sottolineando gli attributi appropriati (*SUGGERIMENTO: un giocatore può essere tesserato per una sola squadra*)
- (ii) [2 punti] indicare eventuali vincoli di integrità referenziale
- (iii) [9 punti] esprimere in SQL la seguente interrogazione (**non tralasciare la join**):

"Estrarre nome e cognome dei calciatori di ruolo PORTIERE che giocano nell'Udinese"

4. [8 punti] Costruire uno schema relazionale di una base di dati in modo che traduca il seguente schema entità-associazione (indicare anche gli eventuali vincoli di integrità referenziale).



- Non ho avuto tempo di studiare però:
- mi devo assolutamente laureare domani
- mi scade il permesso di soggiorno
- domani devo partire per gli USA e starò via 55 anni
-
- non può proprio fare niente?

Programma del corso

- Introduzione
- L'informazione digitale
 - Impariamo “la lingua” dei computer
- Architettura degli elaboratori
 - Come è fatto un computer
- Il sistema operativo e i programmi
 - Ciò che ci permette di usare un computer
- Reti di elaboratori
 - Comunicare grazie al computer
- Principali applicativi di supporto alla produttività personale
 - Alcune compiti comuni che il computer ci aiuta a svolgere:
Esempio: scrivere una relazione, fare elaborazioni/calcoli su dati organizzati in formato tabellare, generare grafici per visualizzare in modo efficace i risultati di elaborazioni su dati.

Obiettivi:

- 1 - Conoscere i fondamenti su cui si basa il mondo digitale (binario, digitalizzazione di immagini, suoni, calcoli, occupazione, conversione).
- 2 – Conosce l'hardware per capire, almeno a grandi linee, su cosa stiamo lavorando e cosa ci servirebbe per lavorare meglio.
- 3 – Conoscere la teoria dei sistemi operativi come cuore alla base di ogni sistema.
- 4 – Conoscere i principali sistemi operativi

5 – Conoscere i principi di funzionamento e interazione in reti locali e Internet

6 – Capire il concetto di algoritmo e come utilizzare gli algoritmi per risolvere i problemi

7 - Imparare ad utilizzare le principali tecnologie web (intro ad HTML e CMS)

- N.B.: nel corso delle lezioni saranno previste ore dedicate ad attività di laboratorio (per esercizi)
- Le esercitazioni riguarderanno fogli di calcolo e le basi del linguaggio HTML

Esami

- Modalità dell'esame: prova **scritta**
- Prova scritta:
 - Validità del esito dello scritto: **nella stessa sessione**
 - **Es.** *Sessione estiva*: giugno/luglio 2019
 - Al termine della sessione l'esito dello scritto scade

Frequently Asked Questions

- “Sono computer-fobico: come posso fare?”

...

- “È necessario frequentare? Il programma è diverso?”
 - Stesso programma per frequentanti e non.
Frequenza vivamente consigliata:
 - tutto il programma viene svolto a lezione
 - l’esperienza insegna che gli studenti che hanno frequentato studiano più facilmente e conseguono voti più alti

Come studiare per questo corso

- Non sottovalutare il corso
- Non basta leggere il materiale didattico
- Non basta imparare a memoria il materiale didattico
- Non basta “sapere ripetere la lezione”
le domande di esame possono richiedere rielaborazione e ragionamento critico
- Bisogna fare propri i concetti:
 - Capirli
 - Fare esercizi
- Non è possibile passare lo scritto “tirando a caso”
- Non sostenere l'esame “tanto per provare”

Informatica (1)

- Oggi tendiamo a chiamare informatica tutto ciò che riguarda i computer... Ma è davvero così?

Dal *Dizionario Garzanti* :

Informatica

Etimologia: Dal fr. *informatique*, tratto da *infor(mation)* (*autom*)*atique* 'informazione automatica'

Definizione: scienza che studia l'informazione e, più specificamente, l'elaborazione dei dati e il loro trattamento automatico mediante elaboratori elettronici.

Informatica (2)

UTENTE

Sa usare un computer



INFORMATICO

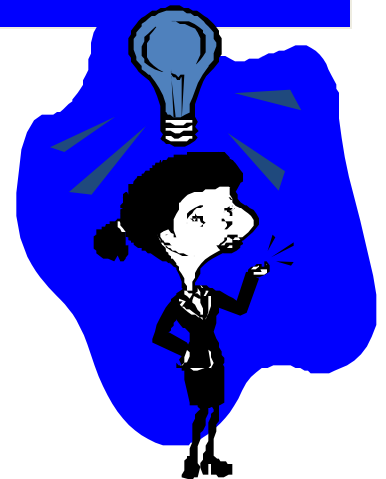
Opera sul computer per mettere gli utenti in condizioni di usarlo al meglio



SCIENZIATO

DELL'INFORMAZIONE

Studia modi per allargare il campo di utilizzo dei computer o per migliorare le funzionalità esistenti



Informatica (3)

- Lo scopo di questo corso: approfondire nostra conoscenza dei computer e delle reti dei computer
- Quindi, diventare *utenti informati*: lavorare con il computer in un modo più *consapevole* ed *efficiente*

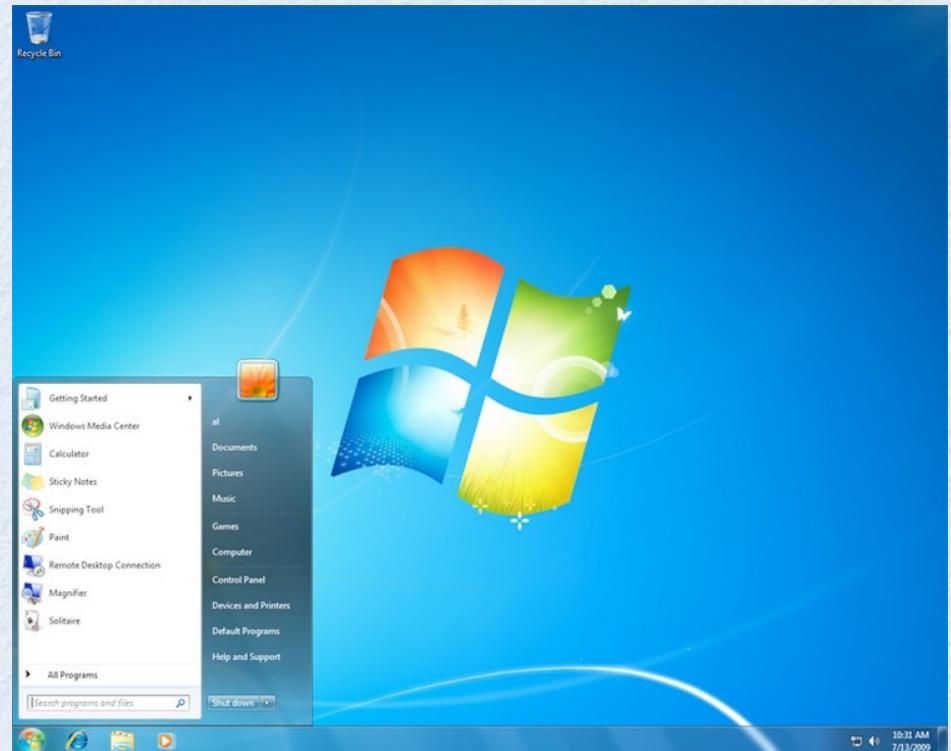
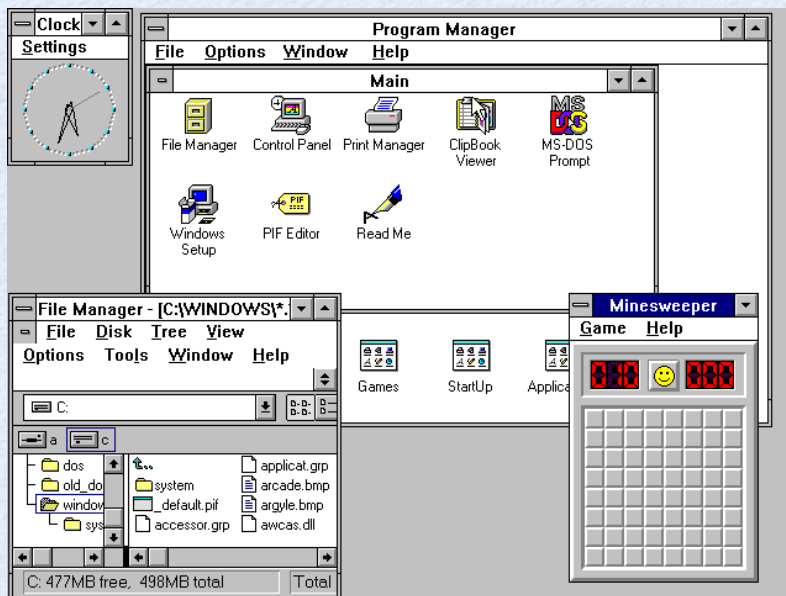


A cosa serve questo corso?

Lo sviluppo tecnologico rende presto obsolete le conoscenze tecniche molto dettagliate...

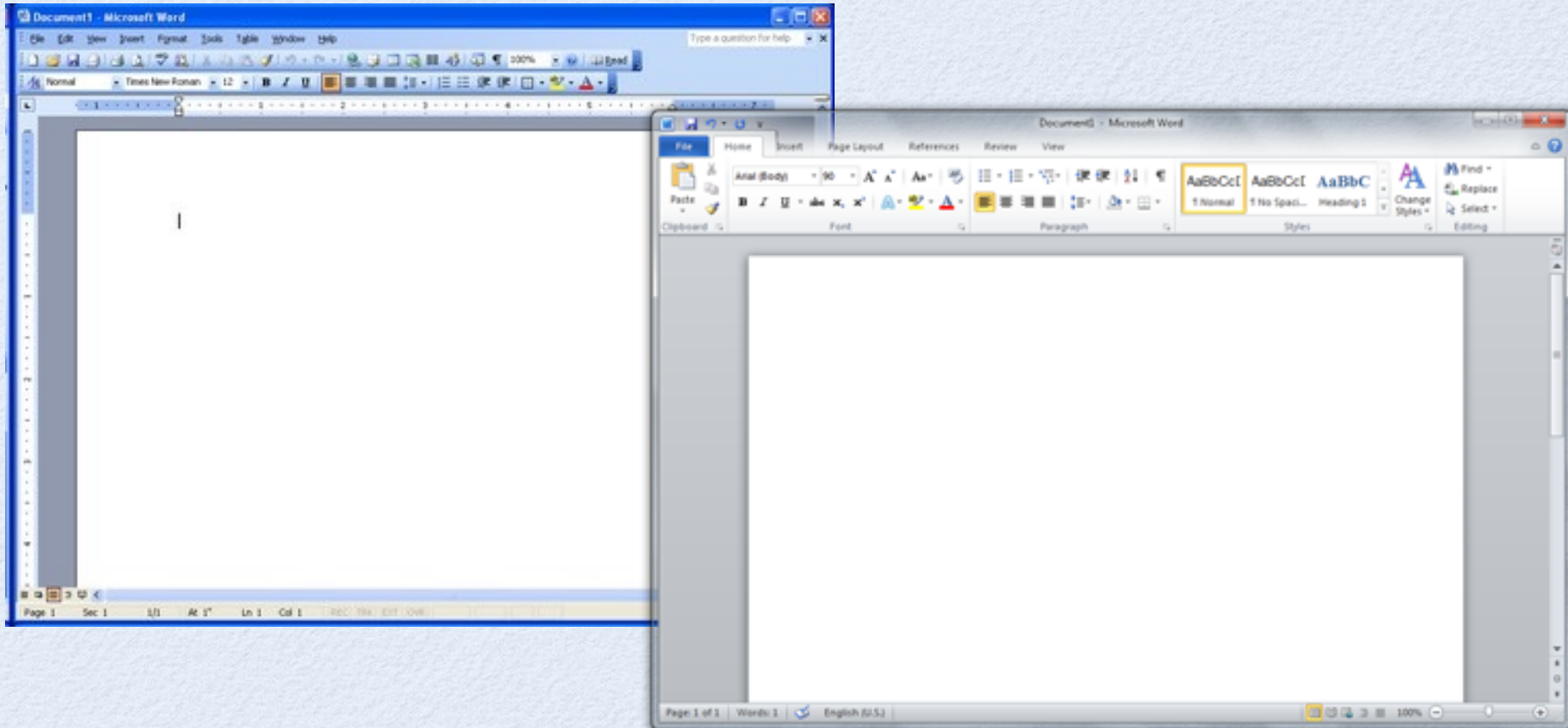
A cosa serve questo corso?

Es.: evoluzione dell'interfaccia grafica di Windows...



A cosa serve questo corso?

... evoluzione dell'interfaccia grafica di Office...



A cosa serve questo corso?

- ... i concetti di base dell'informatica non variano e costituiscono una base culturale importante per un utente
- Se non si conoscono gli strumenti tecnologici, non se ne possono sfruttare le potenzialità (qual è il software/il computer più adatto a risolvere un problema?)
- Il computer è versatile: non si possono prevedere gli usi futuri

Breve Storia del calcolo automatico

Il calcolo automatico

-  Effettuare calcoli a mente o “a mano” è un’attività noiosa e incline all’errore
-  Per secoli l’uomo ha cercato di porre rimedio a questo fatto “automatizzando” il processo di calcolo

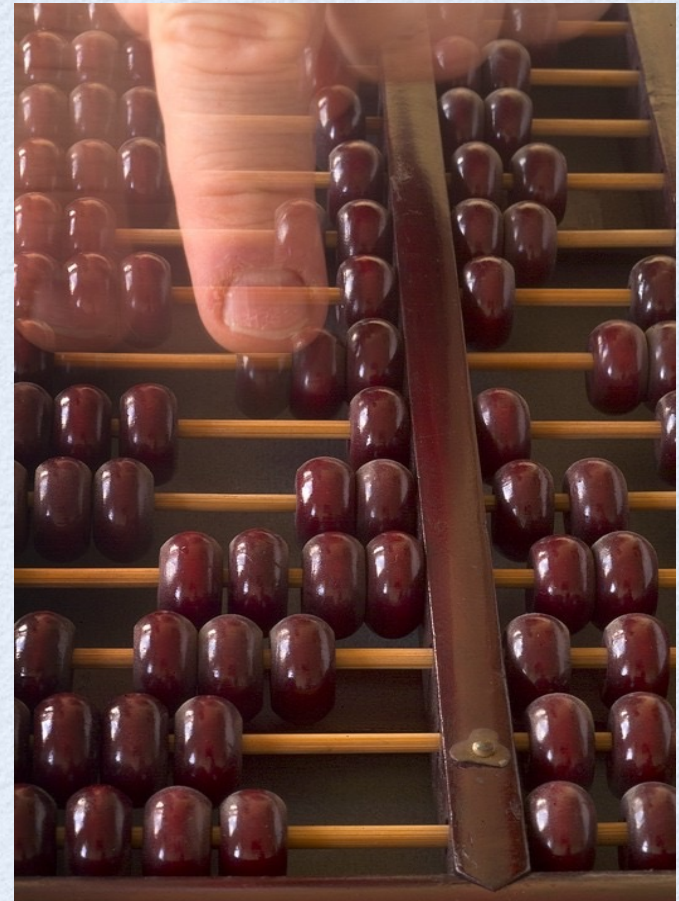
Il calcolo automatico

- 📌 Esempio: ordinamento di 4000 cognomi
- 📌 è necessaria la macchina?
 - 📌 no, l'essere umano è in grado di individuare ed applicare un *algoritmo* per risolvere il problema, ma...
 - non ha voglia di farlo
 - 📌 impiega troppo tempo
 - 📌 rischia di commettere errori
 - 📌

L'àbaco

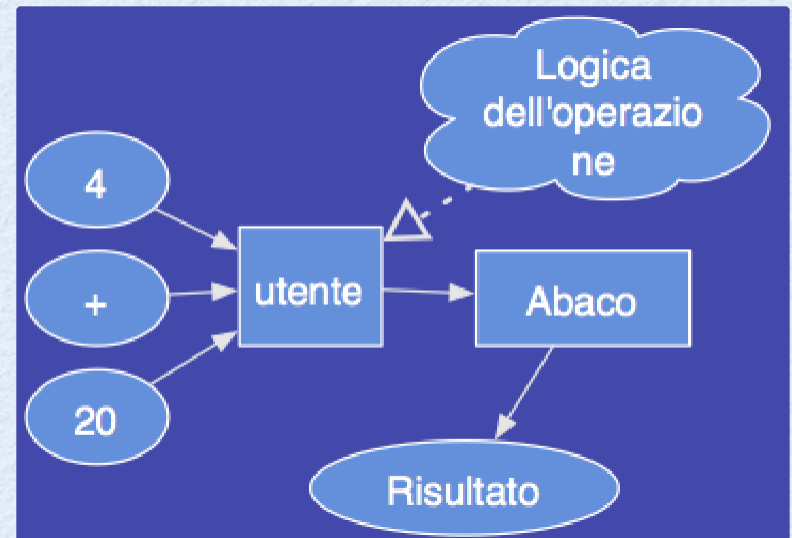
📌 È la prima “macchina di calcolo” nota

📌 I primi abachi risalgono al V millennio a.C.



L'àbaco

- 📌 L'àbaco non è molto diverso da un foglio di carta
- 📌 La logica e la correttezza dell'operazione dipendono unicamente dall'utente



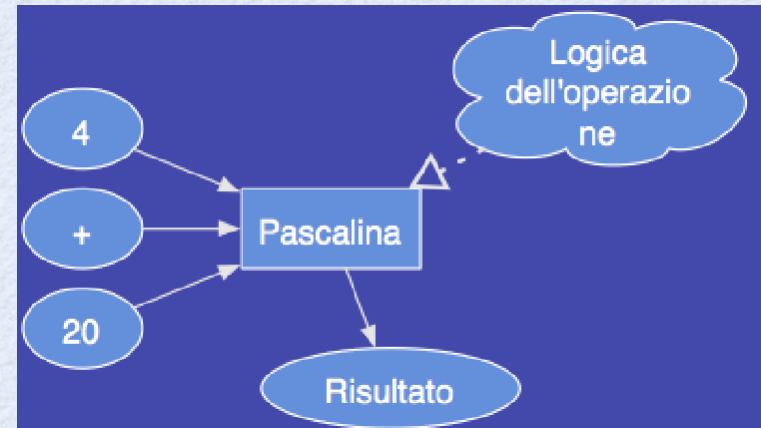
La Pascalina

Bisogna aspettare fino al XVII secolo d.C. per avere la prima vera innovazione rispetto all'àbaco



La Pascalina

 La pascalina è il primo grande passo avanti: la logica dell'operazione è controllata dalla macchina



Problema

-  La pascalina permetteva di effettuare “solo” addizioni e sottrazioni
-  Moltiplicazioni e divisioni potevano essere calcolate mediante ripetizioni di addizioni e sottrazioni, ma era di nuovo l’utente a dovere controllare il processo

Problema (2)

-  Si può pensare di affrontare il problema modificando la macchina in modo da introdurre la divisione e la moltiplicazione
-  Nuovi problemi: radice quadrata? logaritmo? ...
-  In effetti non si sta risolvendo il problema, lo si “rimanda”

Problema (3)

 Il problema vero è che la logica che governa le operazioni è “cablata” nella macchina calcolatrice

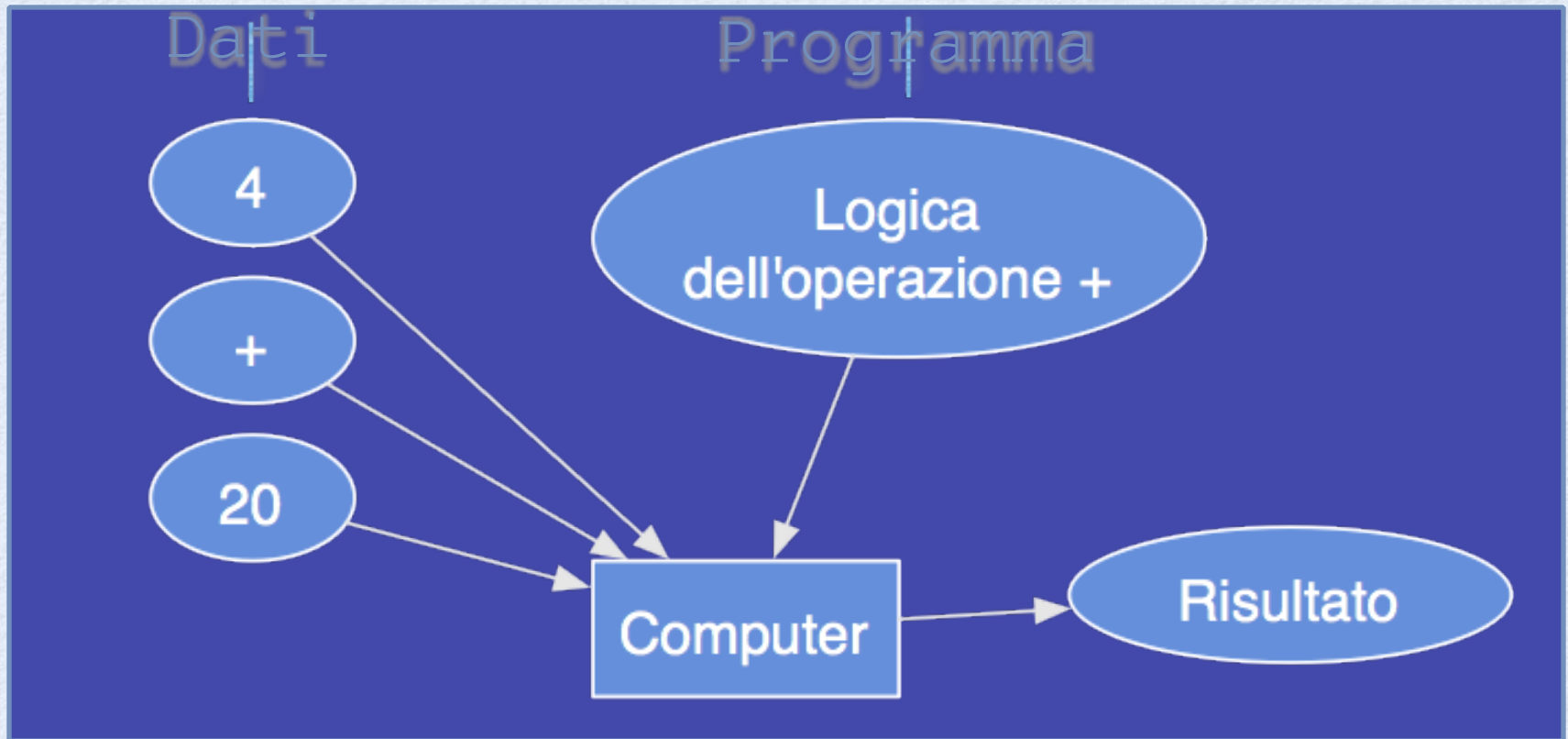
 La soluzione è di trattare tale logica come parte dell'input della macchina

La “Macchina Analitica”

Introdotta da Charles Babbage intorno al 1840, è il primo esempio di macchina di calcolo “programmabile”



Programma come dato



Da Babbage ai giorni nostri

Babbage era molto avanti rispetto ai suoi tempi:

- la tecnologia non era ancora sviluppata a sufficienza (la macchina analitica avrebbe dovuto funzionare a vapore!)
- l'esigenza di automatizzare il calcolo non era forte

Per questi motivi le sue idee non ebbero il successo che avrebbero meritato e furono dimenticate per quasi un secolo

Da Babbage ai giorni nostri

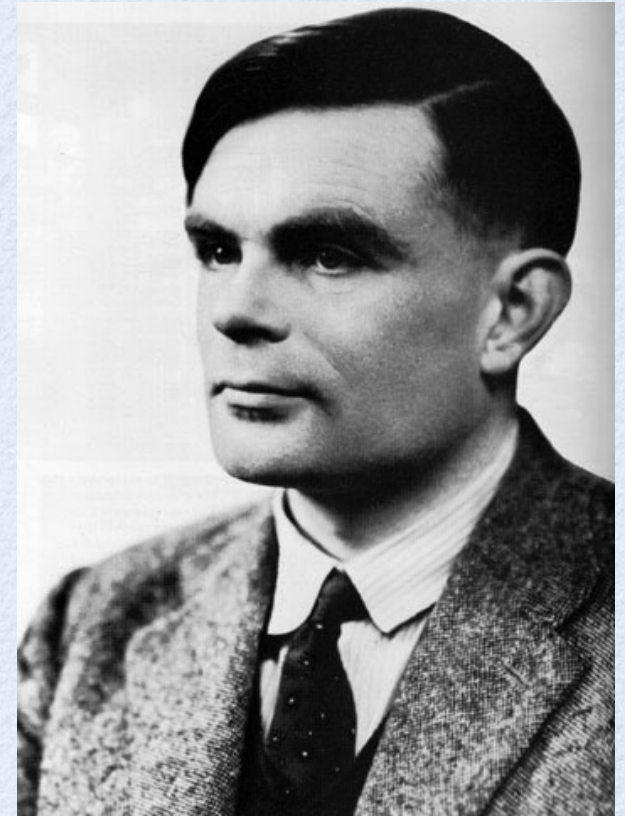
Le idee di Babbage vennero “riscoperte” nella prima metà del ‘900 da **Alan Turing** e da **John von Neumann**

Alan Turing

- 📌 Uno dei padri dell'Informatica

- 📌 Moltissime sue idee/
congetture sono attuali anche
oggi

- 📌 Introduce la “macchina di
Turing”



La macchina di Turing

- 🔊 È un modello astratto di calcolatore:
è un “esperimento mentale”, non una macchina effettivamente costruibile
- 🔊 Pensata per studiare i limiti di ciò che è calcolabile automaticamente
- 🔊 È universale perché può calcolare tutto ciò che è calcolabile in modo automatico
- 🔊 Alla base del suo funzionamento c'è la stessa idea alla base della macchina analitica

John von Neumann

- Descrive un calcolatore effettivamente costruibile
- È ancora oggi il modello su cui si basano i computer moderni



Di cosa parliamo quando parliamo di Informatica

📌 Informatica in inglese:

📌 Information and Communication Technology:

Informatica come tecnologia

risvolti fisico-matematici e ingegneristici del calcolo automatico; teorie e tecniche di comunicazione

📌 Computer Science:

Informatica come scienza

basata su modellizzazione, formalizzazione e verifica sperimentale (teoria degli automi, logica formale, teorie del linguaggio)

“Computer Science is no more about computers than astronomy is about telescopes.” (E. Dijkstra)



Temi dell'Informatica

- Calcolabilità
- Complessità
- Teoria dei linguaggi formali
- Algoritmi e strutture dati
- Linguaggi di programmazione
- Compilatori e interpreti
- Architettura dei calcolatori
- Sistemi operativi
- Ingegneria del software
- Reti di calcolatori
- Interazione uomo-macchina
- Sicurezza
- Intelligenza artificiale
- Basi di dati
- ...

Temi dell'Informatica

• Ciascuno di questi temi è un “edificio” più o meno complesso; ad es.

Intelligenza artificiale:

• Risoluzione dei problemi

• Rappresentazione della conoscenza

• Meccanismi di ragionamento

• Apprendimento automatico

• Pianificazione, scheduling, diagnosi,

• configurazione, soddisfacimento di vincoli,...

Introduzione all'Informatica

 Informatica: infor(mazione) (auto)matica
insieme dei processi e delle tecnologie che
consentono

 la creazione,

 la raccolta,

 l'elaborazione,

 l'immagazzinamento,

 la diffusione

 dell'informazione

Introduzione all'Informatica

📌 Le tecnologie informatiche hanno tre funzioni principali:

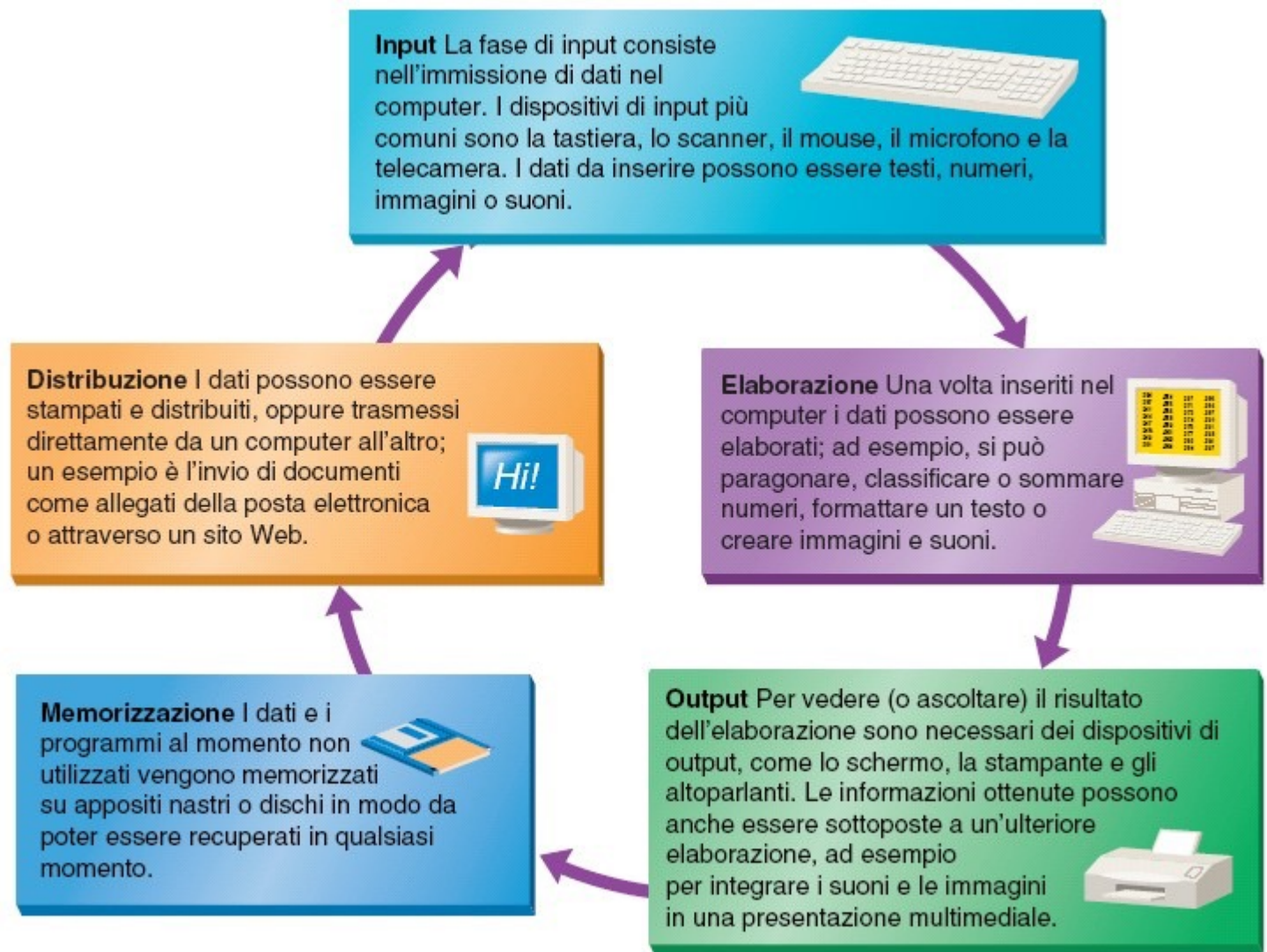
📌 elaborare dati per ottenere informazioni significative

📌 mantenere le informazioni elaborate per utilizzarle in combinazione con altre come dati di un nuovo processo di elaborazione

📌 organizzare le informazioni in una nuova forma per renderle più comprensibili, più interessanti o più utili

📌 N.B.: Il calcolatore non ‘inventa’ nuove informazioni

Il ciclo di elaborazione



Dati e informazioni



Terminologia

Computer

 elaboratore: dispositivo in grado di eseguire automaticamente una serie di istruzioni

elettronico: usa componenti elettronici

 digitale: elabora informazioni binarie (0 e 1)



Hardware

 componente fisica del sistema informatico

 computer, stampanti, monitor, tastiere, mouse, modem,
 cavi telefonici

Terminologia

• Algoritmo (aritmetica elementare formulata dal matematico persiano medioevale al-Khuwarizmi)

• insieme delle operazioni necessarie a risolvere un problema (es. ricetta, istruzioni di montaggio LEGO®)

• Esistono vari algoritmi per risolvere lo stesso problema

• Un programma in genere realizza numerosi algoritmi

Terminologia

Software

 insieme dei programmi che fanno funzionare l'hardware

Due categorie principali di software

 sistemi operativi: permettono all'elaboratore di svolgere le proprie funzioni (software di base)

 programmi applicativi: svolgono i compiti più vari, in funzione delle esigenze dell'utenza

Software

- Svariate decine di sistemi operativi

- MS-DOS, Windows 2000/XP/Vista/7/8

- MacOS X

- Unix (BSD, Linux, Solaris, ...)

Centinaia di migliaia di programmi applicativi, ad esempio:

- browser, videoscrittura, fogli elettronici

- gestione di basi di dati

- grafica avanzata

- controllo di centraline

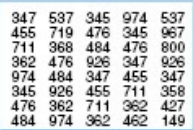
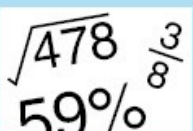
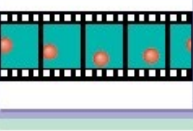
-

Dati

- Materia prima del trattamento dell'informazione
- Hardware e software hanno lo scopo di elaborare e presentare dati
- Un computer può elaborare dati di ogni genere
 - dati semplici: lettere e numeri
 - dati complessi: suoni, immagini, filmati, grafici

Tipi di Dato

Evoluzione
temporale
(storica) dei dati
elaborabili

Tipo di dati	Descrizione
	I dati numerici sono stati i primi a essere elaborati dai computer, prima per le organizzazioni militari e poi per le grandi aziende; ancora oggi, il trattamento dei dati nella gestione degli inventari e dei libri paga e nella registrazione delle vendite avviene più o meno allo stesso modo.
	Le parole possono essere elaborate in promemoria, lettere, relazioni, articoli, libri ecc. Inoltre, grazie a un processo noto come desktop publishing, è possibile dare al testo un formato tipografico.
	L'elaborazione di numeri può consistere in operazioni relativamente semplici, come la stima delle spese annuali per l'università, o in operazioni più complesse, come la redazione del bilancio dello stato.
	I grafici servono per illustrare e rendere più comprensibile il significato di una tabella di dati numerici.
	Le fotografie e altri elementi grafici possono essere memorizzati, elaborati e inseriti in documenti o presentazioni multimediali.
	Personaggi e oggetti animati si muovono sullo schermo per divertire e informare l'utente.
	I suoni, come la musica, la voce e gli effetti sonori possono essere memorizzati, elaborati e riascoltati.
	I video, come interviste e film, vengono memorizzati come dati in modo da poter essere modificati o rivisti in qualsiasi momento.

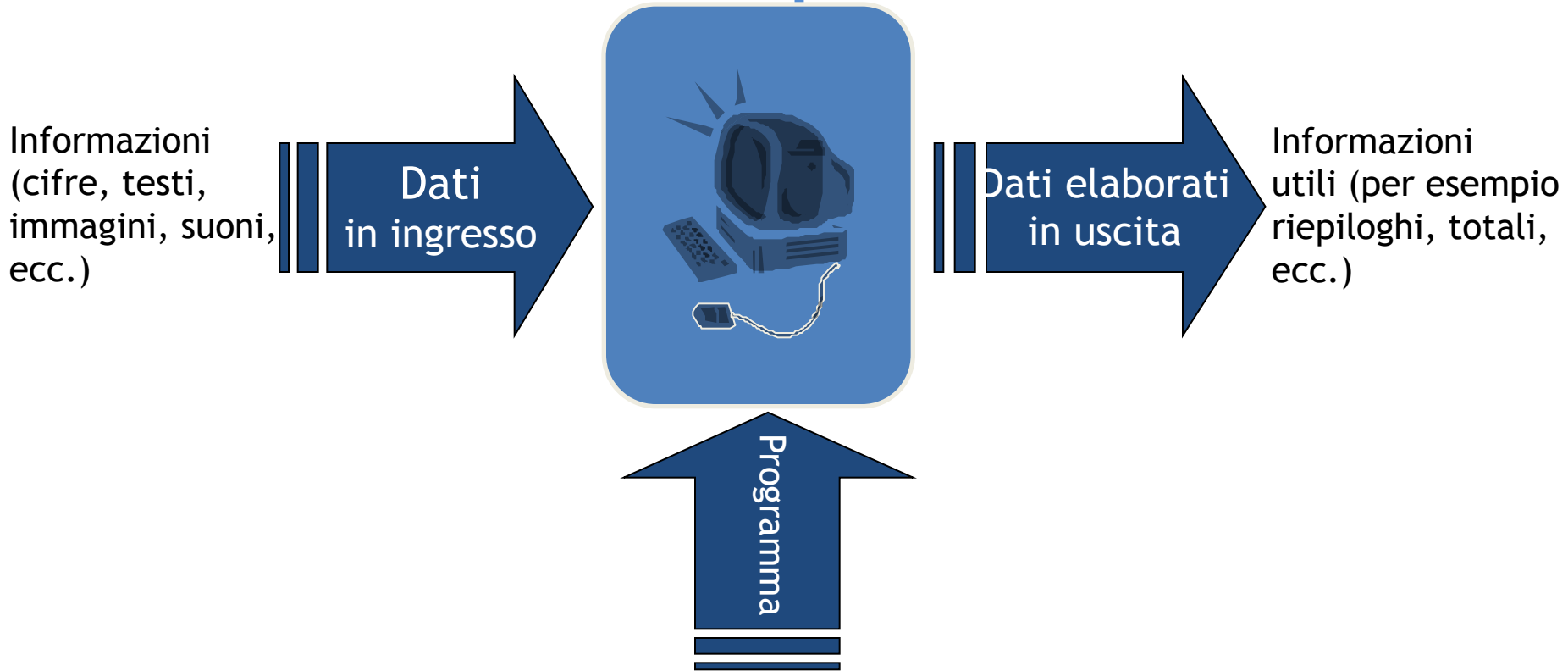
Dove utilizziamo i computer

- Nei grandi sistemi informativi informatici
 - Prenotazioni aeree
 - Previsioni del tempo
 - Casse automatiche (Bancomat, Carte di credito)
 - Stampati pubblicitari (lettere personalizzate)
 - ...

Dove utilizziamo i computer

- Nei piccoli sistemi elettronici
 - Orologi
 - Centraline auto (ABS, ASR, ESP, airbag, ...)
 - Ricevitori digitali (es. di segnali da satellite)
 - Videogiochi (home e portatili)
 - Telefoni cellulari (e carte SIM)
 - Schede telefoniche
 - ...

Il computer



- *Computer*: macchina programmabile, multiuso
- Accetta informazioni e le elabora o manipola creando informazioni utili
- *Programma*, una sequenza di istruzioni che descrive come il computer elabora o manipola informazioni

Computer

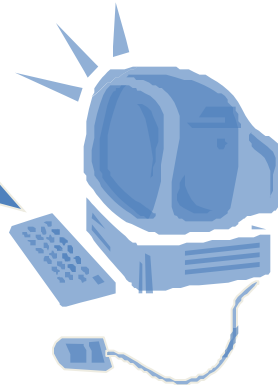
Sottende l'idea di “**calcolo**”. Vedremo tali aspetti in maggiore dettaglio in fasi successive (introducendo il concetto di “**algoritmo**”).

- Computer come una macchina in grado di elaborare dei dati a partire da programmi che descrivono l'elaborazione da compiere.

HARDWARE (HW)

La macchina. Tutto ciò che in un computer si può toccare (“prendere a calci”).

HARDWARE = corpo



SOFTWARE (SW)

I programmi che il computer esegue. Tutto ciò contro cui si solo imprecare.

SOFTWARE = mente

- **HARDWARE + SOFTWARE** definiscono le capacità di un computer, cosa è in grado di fare.

L'uso dei computer

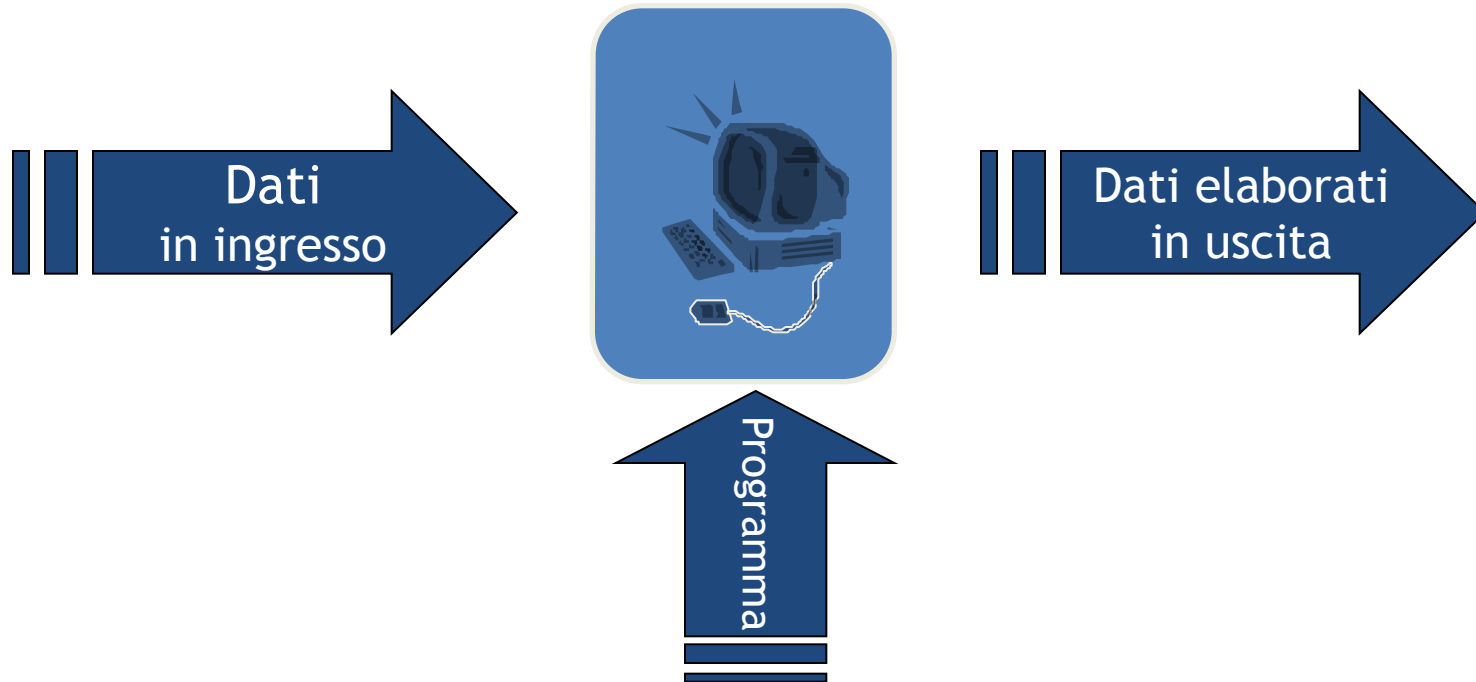
- **A casa:**
 - Elettrodomestici
 - Immagini digitali nei televisori, suono digitale nei compact disk
- **Mezzi di trasporto**
- **Sistemi informativi (basi di dati):** supermercati, biblioteche, sul personale di un'organizzazione ecc.
- **Fabbriche:** dispositivi automatici (robot) guidati dai computer
- **Ospedale:** analisi, chirurgia
- **Ufficio:** video-scrittura, contabilità
- **Comunicazione di massa:** editoria elettronica, gli effetti speciali in cinema e TV
- Ecc. ...



Sondaggio

- Sondaggio...
 - Corso di informatica precedente? ECDL?
 - Quanti non usano regolarmente e-mail ed Internet per comunicare?
 - Quanti non usano i social network?
 - Quanti usano regolarmente un programma di elaborazione testi quale ad es. Microsoft Word / OpenOffice Writer?
 - Quanti conoscono ed usano un foglio elettronico quale ad es. Microsoft Excel / OpenOffice Calc?
 - Quanti conoscono ed usano un programma per la creazione delle presentazioni ad es. Microsoft PowerPoint / OpenOffice Impress?
 - Quanti di voi sanno creare e gestire una pagina web?

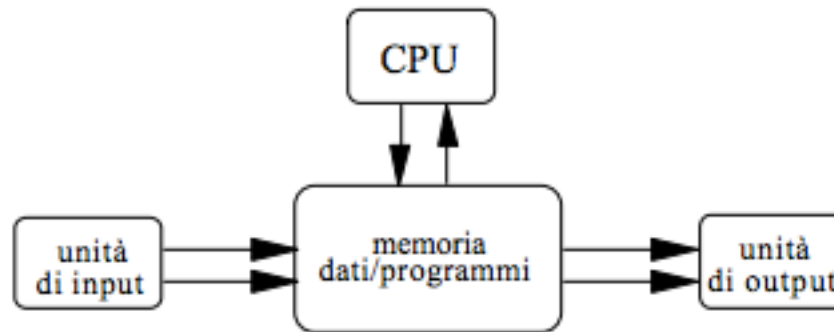
L'attività di un computer



- Un computer è in grado di elaborare dati esclusivamente **rappresentati con (alcuni) numeri interi**.
 - È una limitazione imposta dal suo hardware.
 - Ma è una vera limitazione?
 - Come scopriremo in questa prima parte del corso, molte cose possono essere rappresentate come numeri interi.
 - Il limite non è sul **tipo** di informazione, ma possono esserci delle differenze relative alla **qualità**.

Architettura di Von Neumann

- https://it.wikipedia.org/wiki/John_von_Neumann



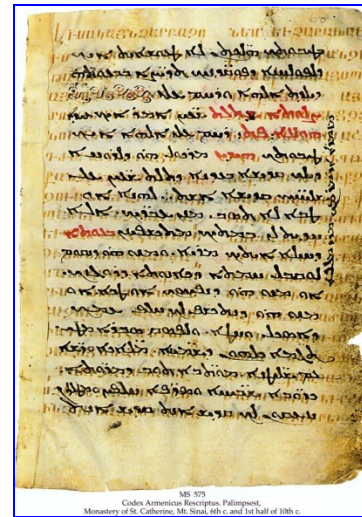
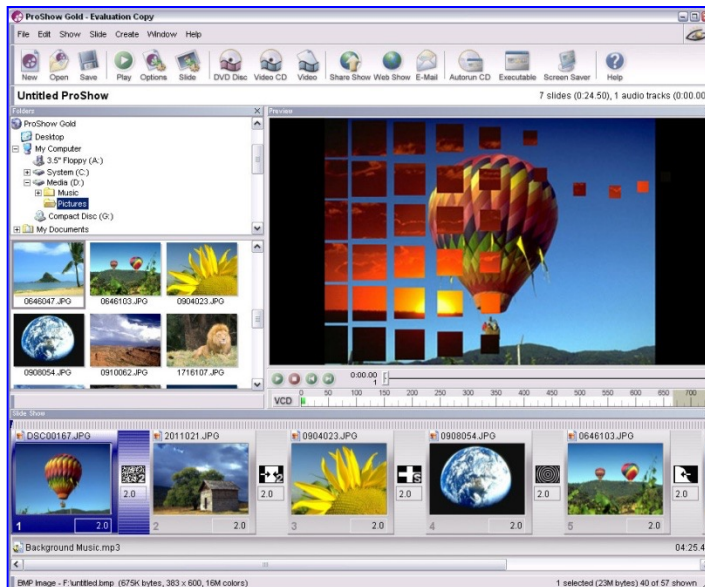
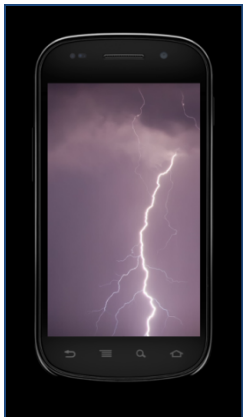
Un dispositivo di **input** e un dispositivo di **output** permettono di **accedere** dall'esterno alla **memoria** del calcolatore, consentendo, rispettivamente, di inserirvi e di estrarne dei dati.

Le **informazioni** contenute in **memoria** vengono **elaborate** da una singola **unità di calcolo** (detta **CPU** - Central Processing Unit), che opera sequenzialmente su di essi.

La caratteristica più importante della **macchina di von Neumann** è costituita dal fatto che sia dati che programmi vengono trattati in modo sostanzialmente omogeneo, ed immagazzinati nella stessa unità di memoria.

Tipi di informazione

- Esistono vari tipi di informazione, di natura e forma diversa, così come rappresentazioni diverse della stessa informazione.
 - La scelta della rappresentazione è in genere vincolata al tipo di utilizzo ed al tipo di operazioni che devono essere fatte sulle informazione stesse.



Tipi di informazione

- Il computer memorizza ed elabora informazioni che devono essere rappresentate in una forma gestibile → **rappresentazione digitale**

attinente ai **numeri, numerico**; in particolare si dice di apparecchiatura che trasforma informazioni (testi, immagini, suoni ecc.) in sequenze numeriche e le elabora o le gestisce in questo formato;

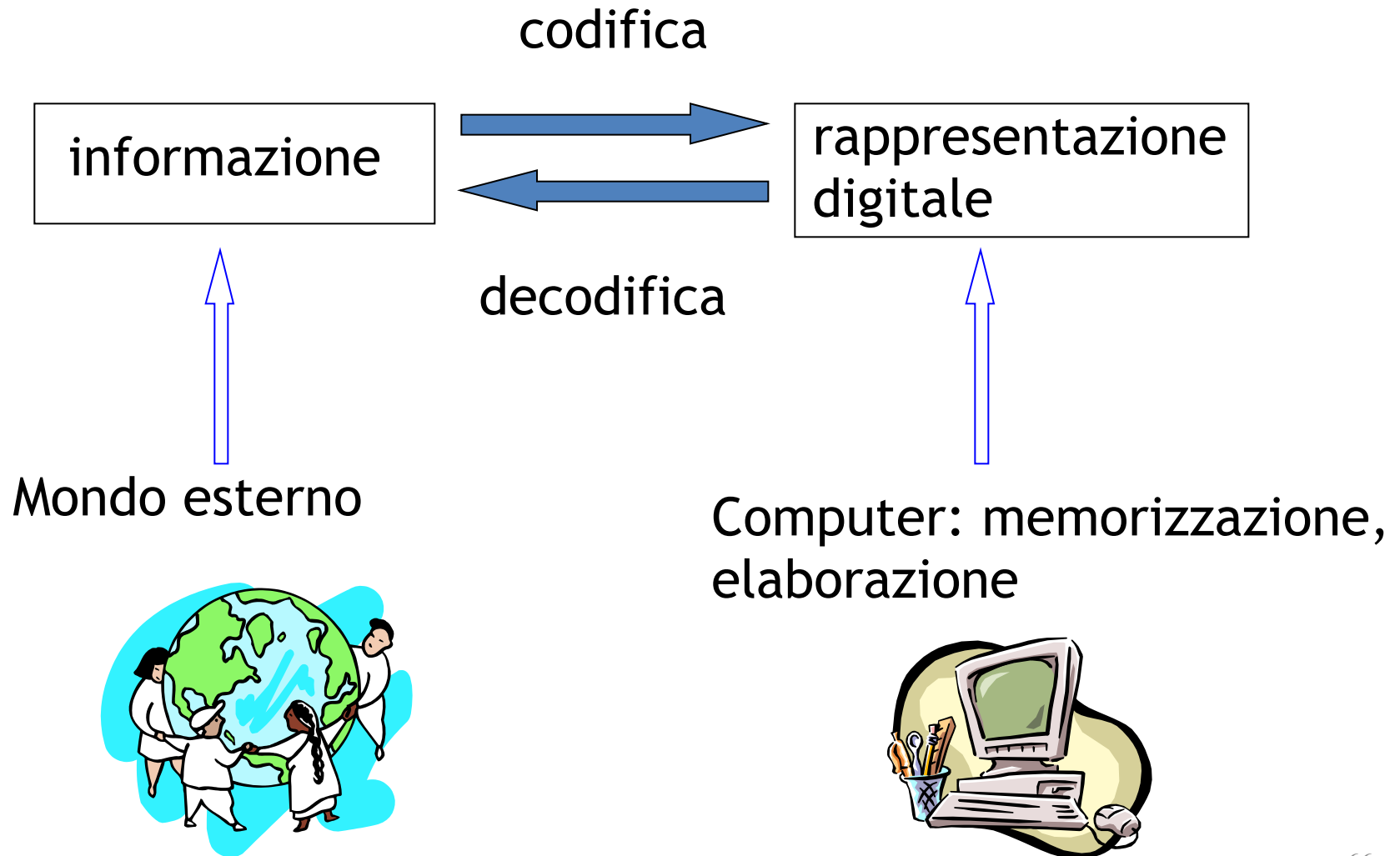
■ Etimologia ← dall'ingl. digital, deriv. di digit 'cifra, unità numerica'.

Il “**Digitale**” implica una **discretizzazione** di elementi continui (**analogici**). *Trasformazione analogico/digitale*

Tipi di informazione

- Il computer memorizza ed elabora informazioni che devono essere rappresentate in una forma gestibile → **rappresentazione digitale**
 - Oggi “rappresentazione digitale” significa: rappresentare l’informazione per mezzo di bit
 - Cifre: **0,1**
 - rappresentazione **binaria**.
 - Più in generale “rappresentazione digitale” significa: rappresentare l’informazione per mezzo di (un insieme finito di) simboli.

Tipi di informazione



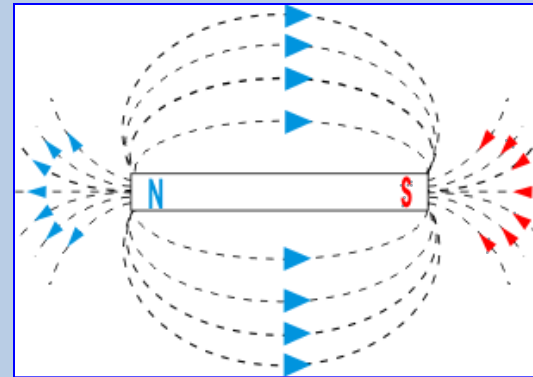
Rappresentazione digitale = rappresentazione binaria

- L'entità minima di informazione che possiamo trovare all'interno di un elaboratore prende il nome di **bit**.
 - Binary digit - cifra binaria.
 - Un bit può assumere due valori.
 - Rappresentazione **binaria**
 - Solo due simboli (0 e 1).
-
- Perché usiamo la rappresentazione binaria?
 - Perché le informazioni rappresentate nel sistema binario possono essere elaborate secondo *vari criteri* e con *vari strumenti* ...

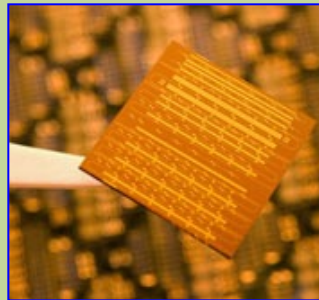
Perché la rappresentazione binaria?

- I due simboli (0 e 1) possono essere rappresentate da:

Due stati di polarizzazione di una sostanza magnetizzabile



Due stati di carica elettrica di una sostanza

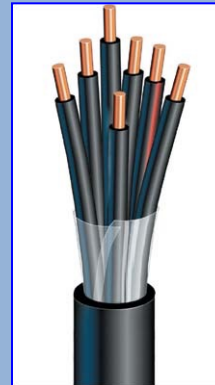


© 2000 Roger Edwards

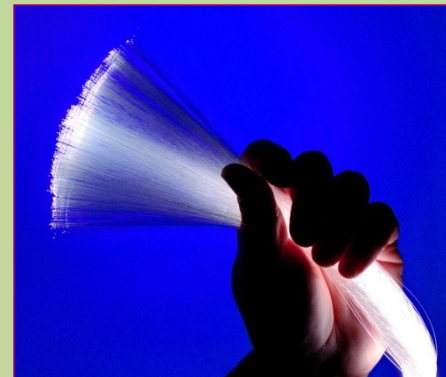
Perché la rappresentazione binaria?

- I due simboli (0 e 1) possono essere rappresentate da:

Al passaggio/non
passaggio di corrente
attraverso un cavo
conduttore



Al passaggio/non
passaggio di luce
attraverso un cavo ottico



Rappresentazione binaria

- In generale: la rappresentazione binaria è basata sulla *presenza* (1) o *assenza* (0) di un fenomeno fisico (in un certo luogo, a un certo punto di tempo).
- Discreta: non esiste alcuna gradazione di valore.
 - Il fenomeno è presente oppure assente, con nessuna via di mezzo.
 - Differenza con ciò che è “analogico” (continuo). La distinzione tra analogico e digitale è cruciale.
- La **logica** è la base del funzionamento del computer:
 - “Associando il valore *vero* con la presenza di un fenomeno e il valore *falso* con la sua assenza, possiamo implementare il mondo logico per mezzo del mondo fisico.”

Codifica dell'informazione

Con 1 bit rappresentiamo solo 2 diverse informazioni:

1 è Vero, 0 è Falso

1 è On, 0 è Off

...

Come possiamo rappresentare più informazioni??

Codifica dell'informazione

- Per poter rappresentare un numero maggiore di informazione si usano *sequenze* di bit.
- Per esempio, per rappresentare quattro informazioni diverse possiamo utilizzare due bit che ci permettono di ottenere quattro configurazioni distinte

00 01 10 11

Il processo secondo cui si fa corrispondere ad un'informazione una sequenze di bit prende il nome **codifica dell'informazione**.

Codifica binaria

- Esempio: un esame può avere quattro possibili esiti: *ottimo*, *discreto*, *sufficiente*, *insufficiente*.
- Codifico (due bit):
 - *ottimo* con 00
 - *discreto* con 01
 - *sufficiente* con 10
 - *insufficiente* con 11

Codifica binaria

- Esempio: otto colori: *nero*, *rosso*, *blu*, *giallo*, *verde*, *viola*, *grigio*, *arancione*.
- Codifico (tre bit):
 - *nero* con 000
 - *rosso* con 001
 - *blu* con 010
 - *giallo* con 011
 - *verde* con 100
 - *viola* con 101
 - *grigio* con 110
 - *arancione* con 111

Codifica binaria

- Con 2 bit si codificano 4 informazioni (2^2).
- Con 3 bit si codificano 8 informazioni (2^3).
- ...

- Con N bit si possono codificare 2^N informazioni differenti.

Codifica binaria

- Se il problema è quello di dover rappresentare M informazioni differenti si deve selezionare il numero di N bit in modo tale che

$$2^N \geq M$$

- Esempio: per rappresentare **40 informazioni** differenti quanti bit bisogna utilizzare?

Codifica binaria

- Se il problema è quello di dover rappresentare M informazioni differenti si deve selezionare il numero di N bit in modo tale che

$$2^N \geq M$$

- Esempio: per rappresentare **40 informazioni** differenti quanti bit bisogna utilizzare?

$$M = 40$$

$$2^N \geq 40$$

Codifica binaria

- Se il problema è quello di dover rappresentare M informazioni differenti si deve selezionare il numero di N bit in modo tale che

$$2^N \geq M$$

- Esempio: per rappresentare 40 informazioni differenti devo utilizzare 6 bit perché

$$2^6 = 64$$

- 5 bit non sono sufficienti perché $2^5 = 32$.

Codifica binaria

- Esiste una particolare aggregazione di bit che è costituita da 8 bit ($2^8 = 256$ informazioni) e prende il nome di byte.

Il byte e' usato come unita' di misura per indicare le dimensioni della memoria, le dimensioni del disco, la potenza di un elaboratore.

Usando sequenze di byte (e quindi aggregati di 8 bit di bit) si possono rappresentare caratteri, numeri immagini, suoni...

Codifica binaria

- Esiste una particolare aggregazione di bit che è costituita da 8 bit ($2^8 = 256$ informazioni) e prende il nome di byte.
- Di solito si usano i multipli del byte:

Kilo	KB	2^{10} (~ un migliaio, 1024 byte)
Mega	MB	2^{20} (~ un milione, 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo, 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi, 1GB x 1024)

Codifica binaria

- Di solito si usano i **multipli del byte**

Kilo	KB	2^{10} (~ un migliaio , 1024 byte)
Mega	MB	2^{20} (~ un milione , 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo , 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi , 1GB x 1024)

- **1KB = 1024 byte = ? bit**

Codifica binaria

- Di solito si usano i **multipli del byte**

Kilo	KB	2^{10} (~ un migliaio , 1024 byte)
Mega	MB	2^{20} (~ un milione , 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo , 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi , 1GB x 1024)

- **$1\text{KB} = 1024 \text{ byte} = (1024 \times 8) \text{ bit} = 8192 \text{ bit}$**

Codifica binaria

- Di solito si usano i **multipli del byte**

Kilo	KB	2^{10} (~ un migliaio , 1024 byte)
Mega	MB	2^{20} (~ un milione , 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo , 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi , 1GB x 1024)

- $1\text{KB} = 1024 \text{ byte} = (1024 \times 8) \text{ bit} = 8192 \text{ bit}$
- Tutte le potenze in tabella si riferiscono ai byte!; 2^{10} byte...

Codifica binaria

- Di solito si usano i **multipli del byte**

Kilo	KB	2^{10} (~ un migliaio, 1024 byte)
Mega	MB	2^{20} (~ un milione, 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo, 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi, 1GB x 1024)

- **$1\text{MB} = 1024 \text{ KB} = ? \text{ byte}$**

Codifica binaria

- Di solito si usano i multipli del byte

Kilo	KB	2^{10} (~ un migliaio, 1024 byte)
Mega	MB	2^{20} (~ un milione, 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo, 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi, 1GB x 1024)

- $1\text{MB} = 1024 \text{ KB} = (1024 \times 1024) \text{ byte}$
- $\text{byte} = 1.048.576; ? \text{ Bit}$

Codifica binaria

- Di solito si usano i **multipli del byte**

Kilo	KB	2^{10} (~ un migliaio , 1024 byte)
Mega	MB	2^{20} (~ un milione , 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo , 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi , 1GB x 1024)

- $1\text{MB} = 1024 \text{ KB} = 1024 \times 1024 \text{ byte}$
- $\text{byte} = 1048576; (1048576 \times 8) \text{ Bit} = \text{circa } 8 \text{ milioni}$

Codifica binaria

- Di solito si usano i multipli del byte

Kilo	KB	2^{10} (~ un migliaio, 1024 byte)
Mega	MB	2^{20} (~ un milione, 1KB x 1024)
Giga	GB	2^{30} (~ un miliardo, 1MB x 1024)
Tera	TB	2^{40} (~ mille miliardi, 1GB x 1024)

- $1\text{GB} = 1024\text{ MB} = 1024 \times (1024 \times 1024)\text{ byte}$
- $1.073.741.824\text{ byte};$
- $= ?\text{ Bit} = 1073741824 \times 8 = \text{circa } 8\text{ miliardi di bit}$

Codifica dei caratteri

- Alfabeto latino
 - Lettere maiuscole e minuscole
 - Cifre numeriche (0, 1, 2, ..., 9)
 - Simboli di punteggiatura (, . ; : ! “ ? ...)
 - Segni matematici (+, -, {, [, >, ...)
 - Caratteri nazionali (à, è, ì, ò, ù, ç, ñ, ö, ...)può essere codificato usando un byte (220 caratteri circa).
- Il metodo di codifica più diffuso tra i produttori di hardware e di software prende il nome di **ASCII** (American Standard Code for Information Interchange).

Riepilogo

distinzione tra **hardware** e **software**; **computer** come **calcolatore** ed **elaboratore di informazioni**

Architettura di Von Neumann: dati e programmi entrambi in memoria;
dispositivi di input-output

l'informazione per essere elaborata al computer? Deve essere “codificata”

Rappresentazione digitale = rappresentazione discreta tramite numero finito di simboli

Distinzione analogico/digitale

oggi vuol dire **rappresentazione binaria (1 0)**

L'entità minima di informazione è il bit

Codifica dei caratteri (ASCII)

- Tabella parziale:

ASCII	Simbolo
00000000	NUL (spazio bianco)
...	...
00111110	>
00111111	?
01000000	@
01000001	A
01000010	B
01000011	C
...	...

Codifica delle parole

- Parole sono sequenze di caratteri.
- Codifica della parole cane:

01100011	01100001	01101110	01100101
c	a	n	e

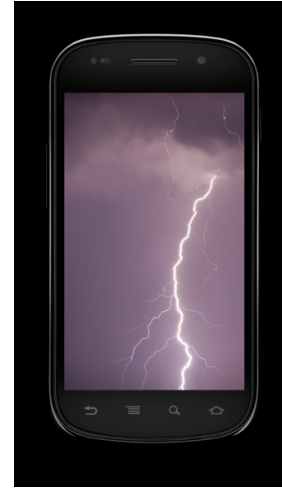
- Il problema inverso: data una sequenza di bit, il testo che essa codifica può essere ottenuto nel modo seguente:
 1. Si divide la sequenza in gruppi di otto bit (byte).
 2. Si determina il carattere corrispondente ad ogni byte.

Codifica dei caratteri

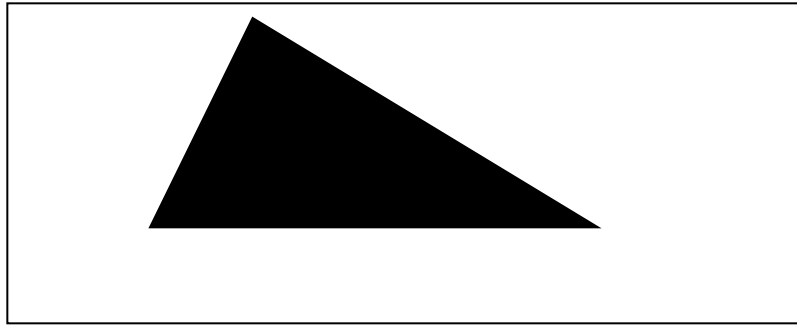
- Abbiamo considerato il codice ASCII:
 - ASCII: 8 bit per carattere.
 - ASCII base: usa solo 7 degli 8 bit, non codifica ad es. i caratteri nazionali (Console et al.).
 - ASCII esteso: usa tutti gli 8 bit.
- Altri codici (lunghezza variabile):
 - UTF-8: 8 bit per i caratteri di ASCII esteso, 32 bit per altri caratteri (ampio uso nel WWW).
 - UTF-16: 16 bit per alcuni caratteri, 32 bit per altri (usato, ad esempio, per testo in Microsoft Windows).

Codifica delle immagini

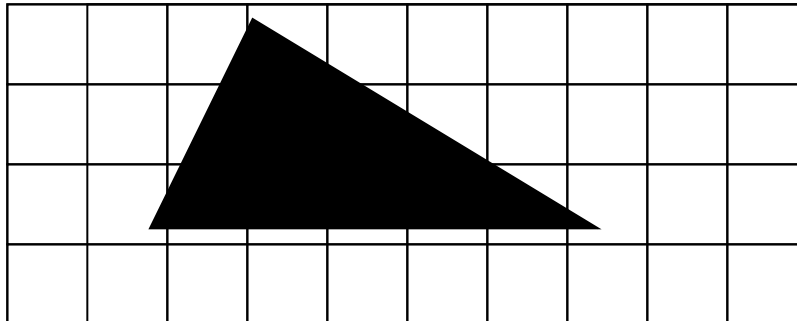
- Sistemi di supporto alla progettazione CAD/CAM
- Campo medico (per esempio, i sistemi di TAC o di ecografia)
- Telecamere e videoregistratori professionali
- Effetti speciali in film, televisione
- Telefoni cellulari
- Ecc.



Codifica delle immagini



Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante.

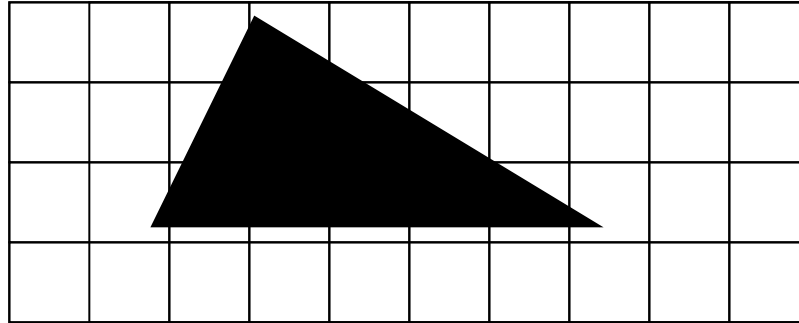


Risoluzione
10 X 4

Codifica delle immagini

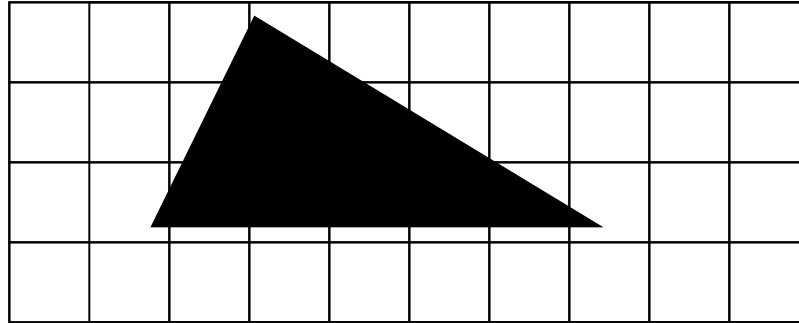
- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (**picture element**) e può essere codificato in binario secondo la seguente convenzione:
 - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il bianco è predominante.
 - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il nero è predominante.

Codifica delle immagini



???

Codifica delle immagini



0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

Codifica delle immagini

0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0



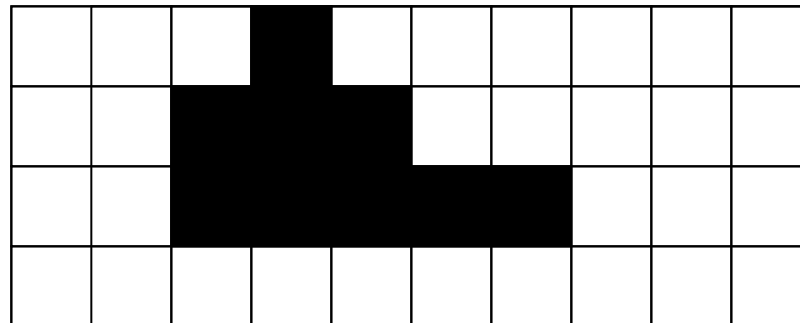
Poiché una sequenza di bit è lineare, è necessario definire convenzioni per ordinare la griglia dei pixel in una sequenza. Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra

0000000000 0011111000 0011100000 0001000000

Codifica delle immagini

Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria

Se riconvertiamo la sequenza di stringhe
0000000000 0011111000 0011100000 0001000000
in immagine otteniamo:

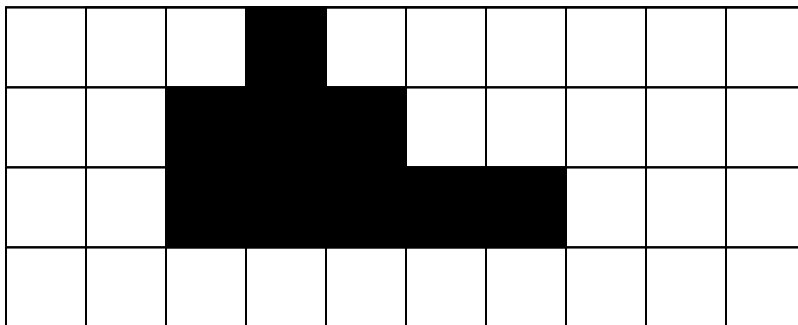


Riconversione

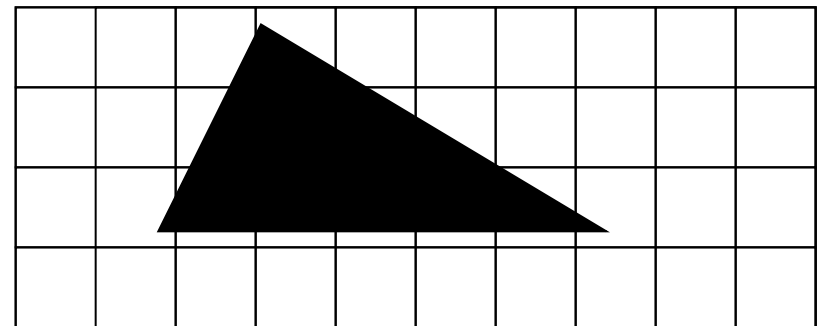
Codifica delle immagini

Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria

Se riconvertiamo la sequenza di stringhe
0000000000 0011111000 0011100000 0001000000
in immagine otteniamo:



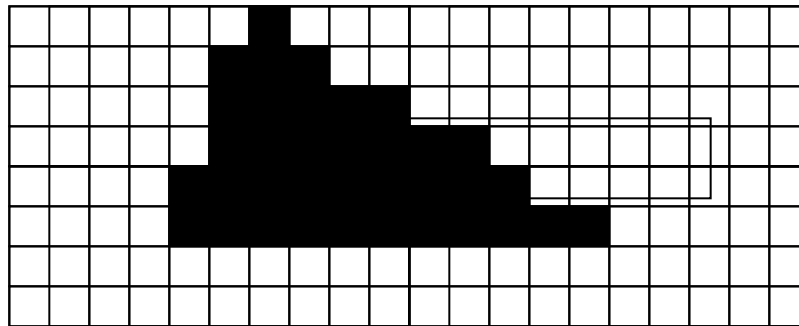
Riconversione



Originale

Codifica delle immagini

La rappresentazione sarà più fedele
all'aumentare
del numero di pixel, ossia al diminuire delle
dimensioni dei quadratini della griglia in cui è
suddivisa l'immagine.



Risoluzione
20 x 8

Codifica delle immagini

Quindi: le immagini sono rappresentate con un certo livello di approssimazione, o meglio, di risoluzione, ossia il numero di pixel usati per riprodurre l'immagine:

- 640 x 480 pixel; 800 x 600 pixel
- 1024 x 768 pixel; 1280 x 1024 pixel
- 1400 x 1280 pixel

Codifica delle immagini

- Assegnando un bit ad ogni pixel è possibile codificare solo immagini in bianco e nero.
- Per codificare le immagini con **diversi livelli di grigio** oppure **a colori** si usa la stessa tecnica: per ogni pixel viene assegnata una *sequenza* di bit.

Codifica delle immagini (grigio e colore)

- Per memorizzare un pixel non è più sufficiente un solo bit (di spazio di memoria).
 - Per esempio, se utilizziamo quattro bit possiamo rappresentare $2^4 = 16$ livelli di grigio o 16 colori diversi.
 - Mentre con otto bit ne possiamo distinguere $2^8 = 256$, ecc.

Attenzione : fare sempre attenzione nel calcolo
- a bit e byte 256 colori = 1 byte per pixel

- 1 pixel a 2 colori 1 bit
- 1 pixel a 256 colori 1 byte ($1 * 8$ bit)
- 1 pixel a 65535 colori 2 byte ($2 * 8$ bit)
- 1 pixel a 16 Mil. di colori 3 byte ($3 * 8$ bit)

Codifica delle immagini

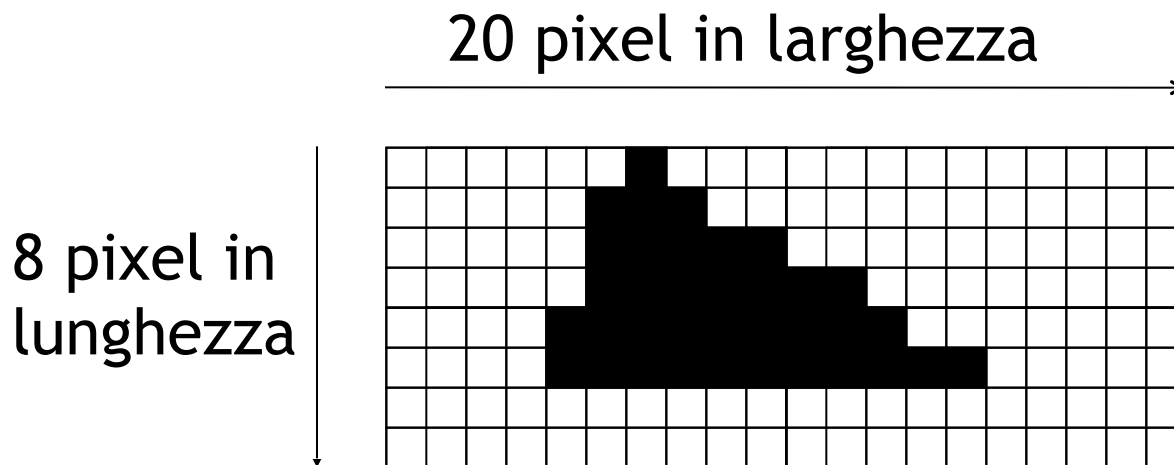
- Complessivamente per essere riprodotta (per esempio, sullo schermo), la rappresentazione binaria di un'immagine deve anche avere informazioni su:
 - Numero di righe dell'immagine.
 - Numero di colonne dell'immagine.
 - Numero di colori usati.

Qualità delle immagini digitali

- Due parametri di qualità: **risoluzione** e **profondità del colore**
 - La **risoluzione** indica la precisione con cui viene effettuata la suddivisione di un'immagine in pixel.
 - La risoluzione si misura dunque in **pixel**.
 - La **profondità del colore** è il numero di bit o byte utilizzati per rappresentare ciascun pixel.
 - La profondità del colore si misura quindi in **bit o byte**.
 - Indica (ma non è uguale a) il numero di colori diversi che possono essere rappresentati.
- La **dimensione** dell'immagine è il numero di **bit o byte** che servono per memorizzarla.
 - **Dimensione** = **risoluzione** x **profondità del colore**.
 - **Maggiore la qualità, maggiore la dimensione!**

Dimensione di un'immagine

- Esempio 1: sto usando **160 pixel** (20x8).

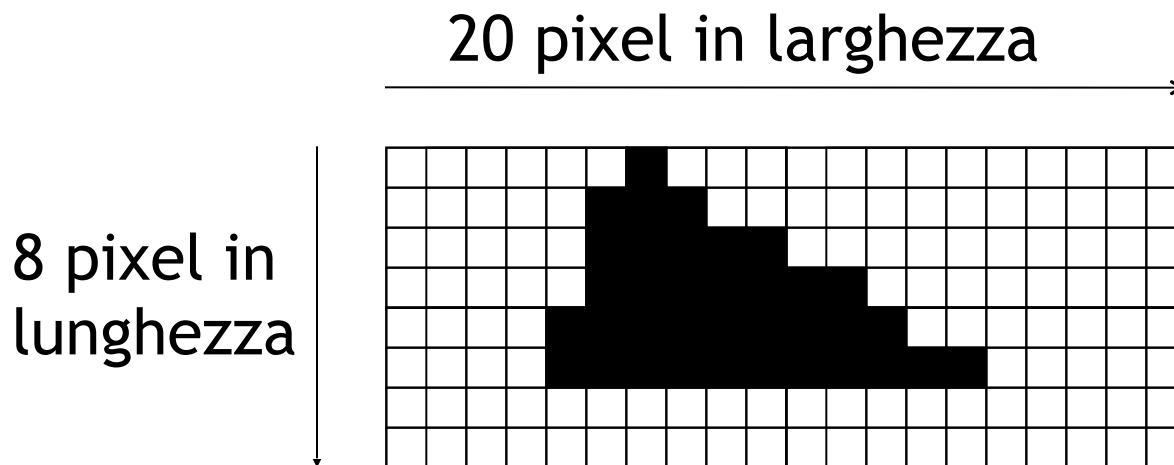


Sto inoltre usando 2 colori = **1 bit** per pixel.

- Quanti bit occupa l'immagine?
- Quanti byte?

Dimensione di un'immagine

- Esempio 1: sto usando **160 pixel** (20x8).



Sto inoltre usando 2 colori = **1 bit** per pixel.

- Quanti bit occupa l'immagine?
- Quanti byte?

L'immagine occupa dunque $160 \times 1 = \mathbf{160 \text{ bit}}$.
In byte: $160 \text{ bit} = 160/8 \text{ byte} = \mathbf{20 \text{ byte}}$.

Dimensione di un'immagine

- **Esempio 2:** 640 pixel in larghezza, 480 pixel in lunghezza, 256 colori.
- Per distinguere 256 colori quanti bit sono necessari?
- Quale è la risoluzione totale?
- Quanti byte occupa l'immagine? Quanti bit?

Dimensione di un'immagine

- **Esempio 2:** 640 pixel in larghezza, 480 pixel in lunghezza, 256 colori.
- Per distinguere 256 colori quanti colori sono necessari?
- Sono necessari **otto bit** per la codifica di ciascun pixel.
- **Quale è la risoluzione totale?**
- **$640 \times 480 = 307200$**
- **Quanti byte occupa l'immagine? Quanti bit?**
- **$307.200 \times 8 = 2.457.600$ bit; 307.200 byte (circa 307,2 KB)**

Esercizio 3

- Abbiamo un'immagine 640 X 480 pixel con 65.535 colori

Quanti bit occorrono per 65.535 colori?

- 1) Calcolate la risoluzione totale (base x altezza)
- 2) Moltiplicate la risoluzione totale x il numero di bit
- 3) Trasformate in byte

Soluzione Esercizio 3

- Abbiamo un'immagine 640 X 480 pixel con 65.535 colori

Quanti bit occorrono per 65.535 colori?

16

1) Calcolate la risoluzione totale (base x altezza):

307.200

2) Moltiplicate la risoluzione totale x il numero di bit

$307.200 * 16 =$ **4.915.200**

3) Trasformate in byte: **614.400 byte** (614,4 KB)

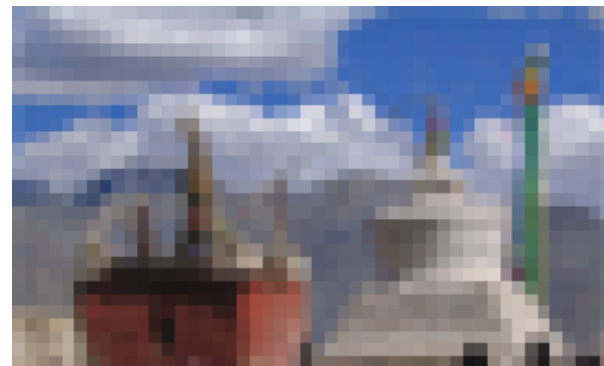
Qualità e dimensione



318x234 pixel
3 B per pixel (16M colori)
 $318 \times 234 \times 3 = 223236 \text{ B} \approx 218 \text{ KB}$



80x59 pixel
3 B per pixel (16M colori)
 $80 \times 59 \times 3 = 14160 \text{ B} \approx 14 \text{ KB}$



48x35 pixel
3 B per pixel (16M colori)
 $48 \times 35 \times 3 = 5040 \text{ B} \approx 5 \text{ KB}$



318x234 pixel
3 B per pixel (16M colori)
 $318 \times 234 \times 3 = 223236 \text{ B} \approx 218 \text{ KB}$



318x234 pixel
4 bit per pixel (16 colori)
 $318 \times 234 \times 4 = 297648 \text{ bit} \approx 36 \text{ KB}$



318x234 pixel
3 bit per pixel (8 colori)
 $318 \times 234 \times 3 = 223236 \text{ bit} \approx 27 \text{ KB}$

Compressione delle immagini

- Le immagini codificate pixel per pixel sono dette immagini in grafica **bitmap** (mappa di bit) o **raster**.
 - Le immagini bitmap occupano parecchio spazio.
- Esistono delle **tecniche di compressione** che permettono di ridurre le dimensioni.
 - Sono essenzialmente modi “più intelligenti” di memorizzare le immagini, invece che elencare semplicemente i pixel che le compongono.
 - A volte comportano una riduzione della qualità dell’immagine (che normalmente risulta impercettibile per l’occhio umano).
- Esempi di formati diffusi per le immagini:
 - **GIF, PNG e JPEG.**

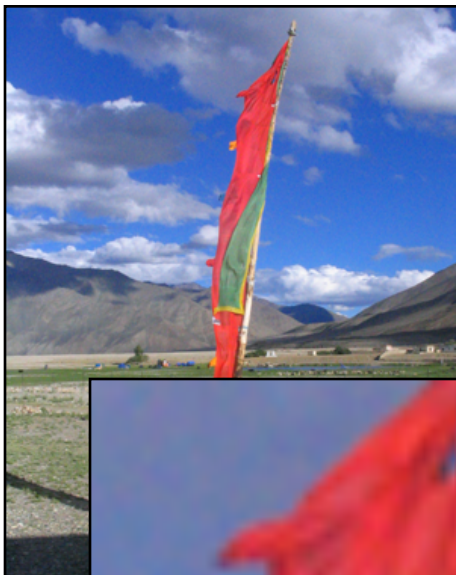
Compressione delle immagini

- Distinzione tra compressione **lossless** e compressione **lossy**.
 - **Lossless**: compressione senza perdita di informazioni.
 - Dalla versione compressa, si può ricostruire perfettamente la versione non-compressa.
 - Per esempio, GIF, PNG (ma dipende dal numero di colori: c'è più informazione su slide 46).
 - **Lossy**: compressione con perdita di informazioni
 - Dalla versione compressa, non è possibile recuperare la versione originale.
 - Per esempio, JPEG.

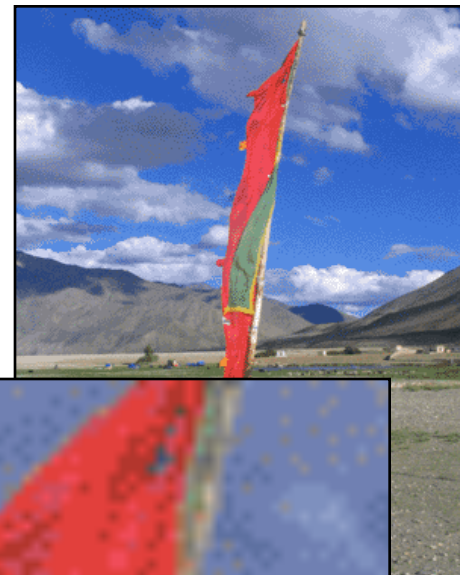
Compressione GIF

- **GIF = Graphic Interchange Format**
 - Applicabile quando i colori sono “pochi” (max 8 bit = 256 colori).
 - Vantaggi:
 - Se l’immagine ha pochi colori non viene “deteriorata” (compressione lossless).
 - Svantaggi:
 - Se l’immagine ha molti colori (es. una foto) bisogna ridurli, perdendo in qualità.
- Pertanto questo formato è preferibile per immagini dalle linee nette e con poche sfumature.

Compressione GIF



352 KB



47 KB



Compressione JPEG

- **JPEG = Joint Photographic Expert Group**
 - Studiata appositamente per le fotografie.
 - Si basa sul principio di rinunciare ad una parte dell'informazione presente nell'immagine (compressione lossy) quando quell'informazione non verrebbe comunque percepita dall'occhio umano.
 - Ovviamente se si esagera con la riduzione dell'informazione la perdita di qualità diventa percepibile.
 - Le idee su cui è basata (in breve):
 - L'occhio umano è più sensibile alle **variazioni di luminosità che di tonalità**: meglio dunque sacrificare le seconde.
 - Tali variazioni di luminosità sono ben percepite su aree ampie, **ma non su aree piccole** (un singolo pixel molto più luminoso in mezzo ad altri più scuri non viene notato).
- Il formato JPEG è adatto ad **immagini di tipo fotografico**, dove la perdita di qualità si nota di meno, o in generale ad immagini dove un limite sul numero di colori produrrebbe differenze troppo significative.

Compressione JPEG

352 KB



38 KB



23 KB



PNG

PNG (Portable Network Graphics)

- ◆ Consente di esportare in 3 formati:
8-24-32 bit
- ◆ Lossy
- ◆ Migliore resa del JPEG
- ◆ Ora è pienamente supportate dai browser

Grafica bitmap vs. grafica vettoriale

- Un **oggetto bitmap/raster** è memorizzato semplicemente come una griglia di pixel a ciascuno dei quali è associato un colore.
 - Una volta disegnata una linea, essa non è più una “linea” ma solo un insieme di pixel sullo schermo.
 - La qualità dipende da tanti fattori:
 - Numero di bit utilizzati per ciascun pixel
 - Caratteristiche dell’immagine.
 - Algoritmo di compressione usato.
 - Fattore di ingrandimento sul video.
- L’ingrandimento fa perdere risoluzione

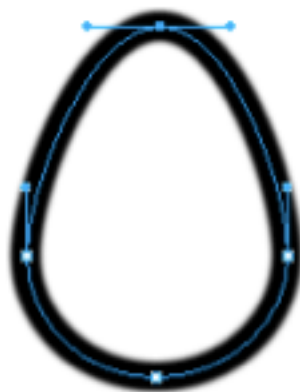


Grafica bitmap vs. grafica vettoriale

- Un **oggetto vettoriale** è costituito da una sequenza di segmenti e un insieme di attributi.
 - Segmenti:
 - Possono essere dritti o curvi, e possono essere uniti a formare una linea spezzata, aperta o chiusa.
 - Vengono memorizzati registrando le coordinate delle estremità di ciascun segmento.
 - Attributi:
 - Ad esempio il colore e lo spessore della linea, il riempimento se si tratta di una figura chiusa (ad es. un rettangolo), etc.
 - Gli attributi sono memorizzati **separatamente** dalle coordinate: è possibile modificare le coordinate lasciando inalterati gli attributi, oppure modificare gli attributi senza cambiare la forma dell'oggetto.
 - La qualità dell'immagine è indipendente dal fattore di ingrandimento.

Oggetto Vettoriale

- sequenza di segmenti dritti o curvi, uniti a formare una linea aperta o chiusa
- unità di memorizzazione base: segmento
- come viene memorizzata l'informazione sui segmenti? coordinate delle estremità + attributi (colore, spessore del tratto...)
- memorizzazione separata -> manipolazione basata sul tipo di memorizzazione: modifico le coordinate senza cambiare gli attributi o viceversa



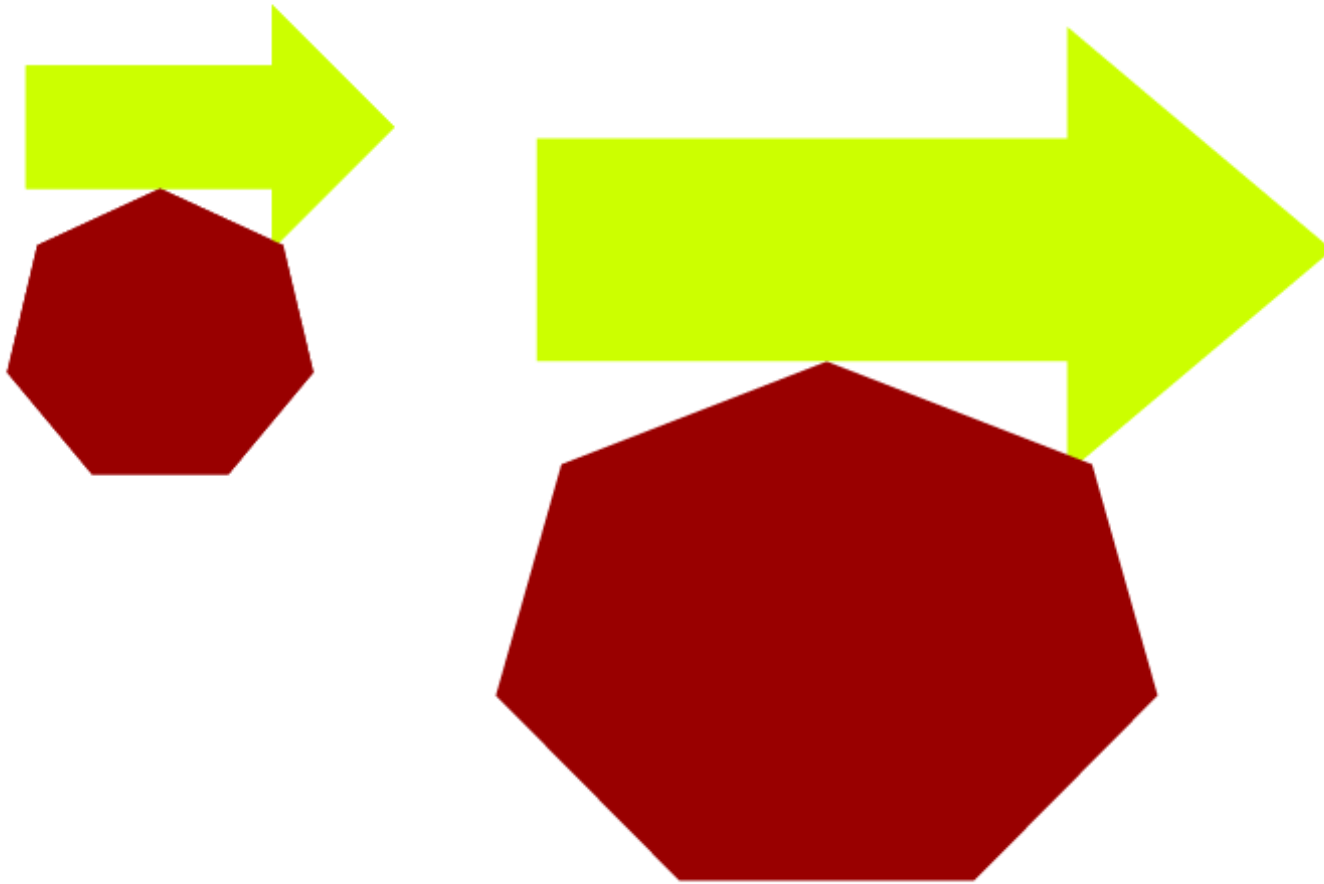
Oggetto Bitmap

- griglia di pixel
- unità di memorizzazione base: pixel
- come viene memorizzata l'informazione?



- Per ogni pixel viene memorizzata l'informazione sul colore
- manipolazione basata sul tipo di memorizzazione, ossia ossia sul concetto di pixel: coloro diversamente i pixel; tipica manipolazione: applicazione di filtri

Grafica bitmap vs. grafica vettoriale



L'ingrandimento non fa perdere la risoluzione

Le immagini vettoriali

Le immagini vengono costruite a partire dalla descrizione degli elementi che le compongono mediante un linguaggio testuale

- Ad es, SVG (Standard Vector Graphics)

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

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

My first SVG



Codifica di immagini in movimento

- Un filmato è una **sequenza di immagini statiche** (dette *fotogrammi* o *frame*).
- Per codificare un filmato si **digitalizzano i suoi fotogrammi**.
- Tanto maggiore è il numero di fotogrammi tanto migliore apparirà la qualità del movimento.
- I filmati in digitale possono essere molto pesanti.

Compressione di immagini in movimento

- Si possono comprimere le immagini con le tecniche viste prima.
- Si memorizza il primo fotogramma e nei successivi si memorizzano le differenze rispetto a quello iniziale.
- Dopo un certo numero di fotogrammi si memorizza un nuovo fotogramma in modo completo.
- Esempi di formati per il video: AVI, MOV.
- Compressione: MPEG (Moving Picture Expert Group), differenza tra fotogrammi.
- CODEC (Compress or DECompress): software in grado di codificare e decodificare un flusso di dati, come una sequenza video.

Codifica dei suoni

- Cosa è un suono ?
- Dal punto di vista fisico un suono è un'alterazione della pressione dell'aria che, quando rilevata dall'orecchio umano, viene trasformata in un particolare stimolo al cervello
- La durata, l'intensità e la variazione nel tempo della pressione dell'aria sono le quantità fisiche che rendono un suono diverso da ogni altro

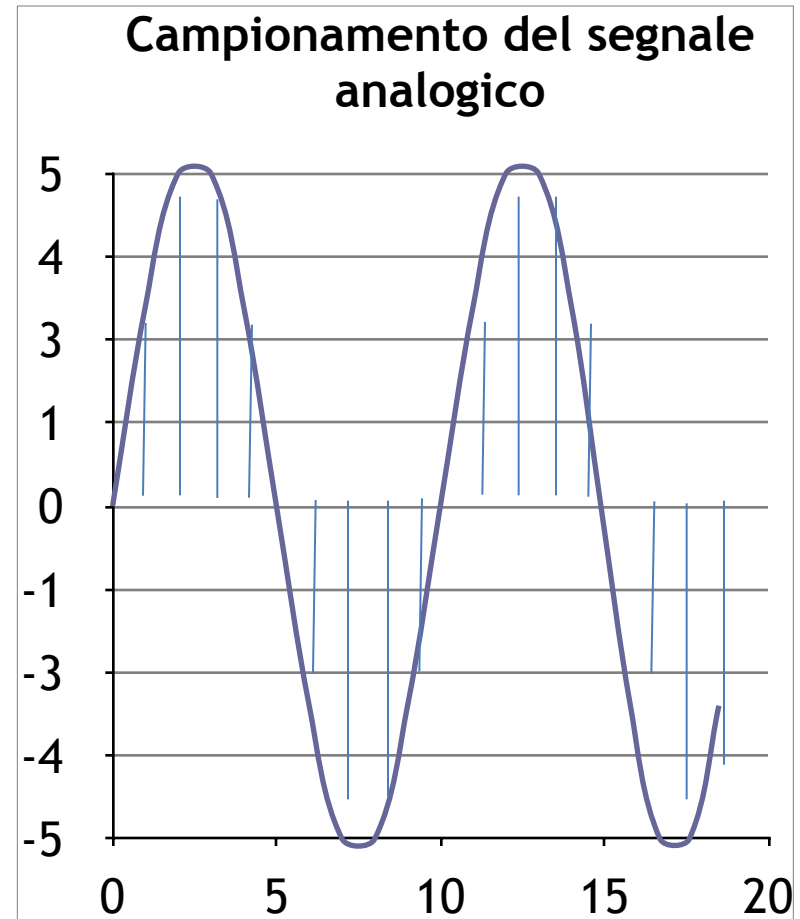
Codifica dei suoni

- Un suono può essere rappresentato mediante un'onda che descrive la variazione della pressione dell'aria nel tempo -> **onda sonora**
- Sull'asse delle ascisse viene rappresentato il **tempo** e sull'asse delle ordinate viene rappresentata la **variazione di pressione** corrispondente al suono stesso.
- Un esempio di **rappresentazione analogica** : l'onda è continua; i valori dell'onda corrispondono a numeri reali.



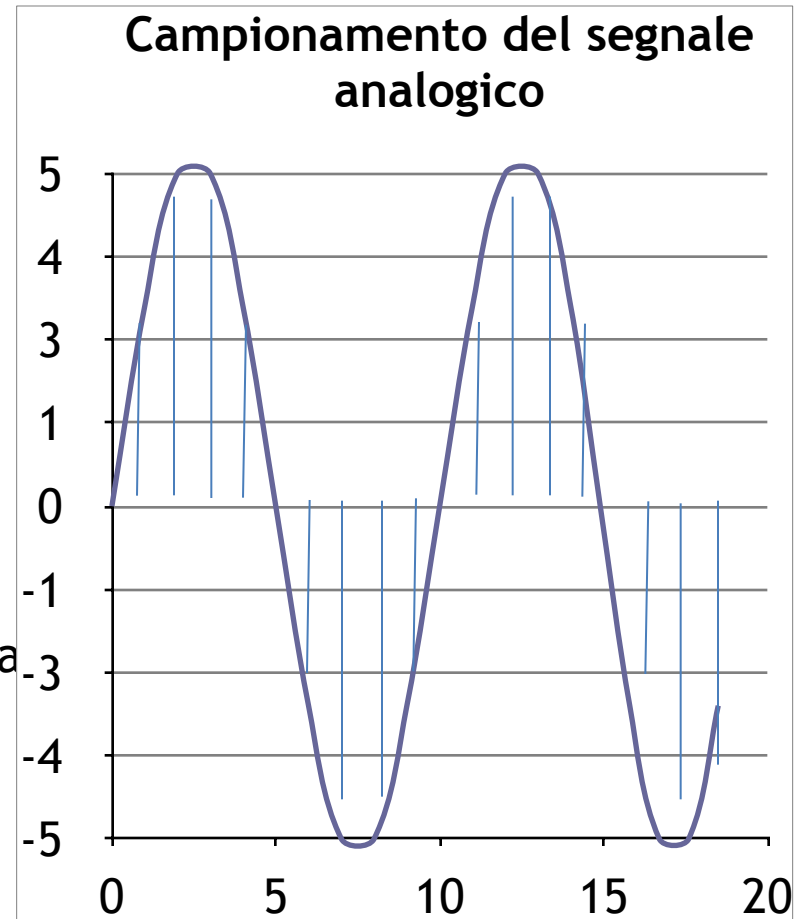
Codifica dei suoni

- Come nel caso delle immagini avevamo bisogno di una tecnica di **discretizzazione**, analogamente nel caso dei suoni abbiamo bisogno di una tecnica che ci permetta di passare dalla rappresentazione fisica continua dell'onda sonora a una rappresentazione digitale discreta
- Intuitivamente: non memorizziamo l'informazione completa sull'onda -> come nel caso delle immagini si **deve perdere in qualità dell'informazione per avere una rappresentazione digitale**



Codifica dei suoni

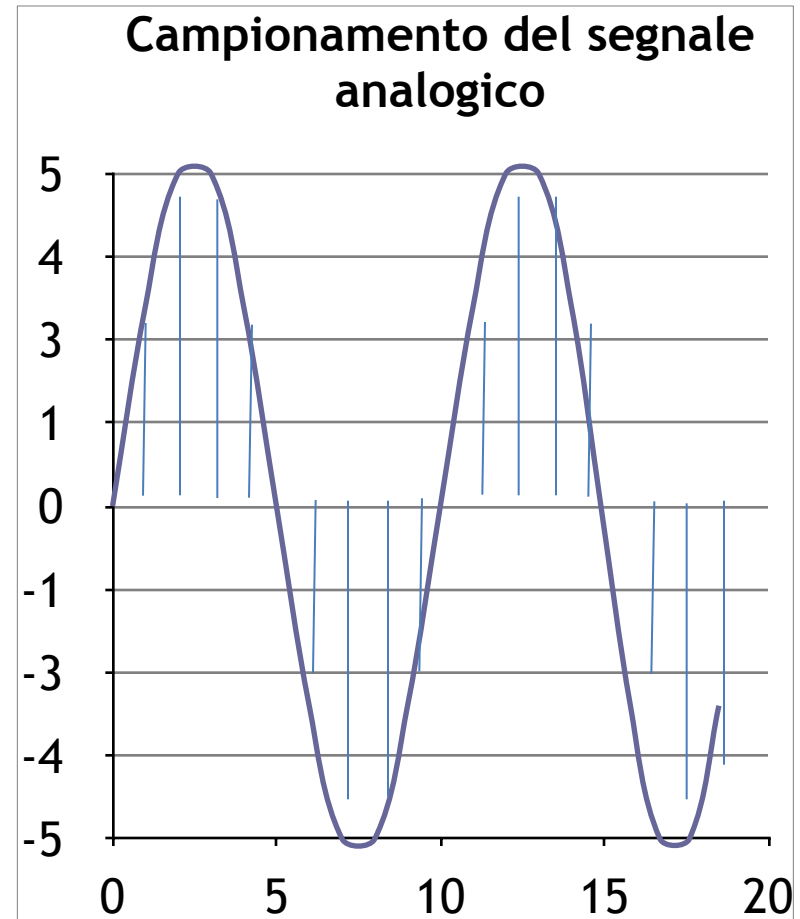
- La conversione di un segnale continuo in una successione di numeri viene eseguita con due successive operazioni elementari:
- 1. **Campionamenti sull'onda:** si preleva una successione di campioni a intervalli costanti di tempo (in altre parole: si misura il valore dell'onda a intervalli di tempo costante)
- 2. **Quantizzazione e codifica:** ogni campione viene quantizzato ossia convertito in un numero; la sequenza dei valori numerici ottenuta dai campioni può essere facilmente codificata formato digitale->



Associo una configurazione di bit a ogni campione

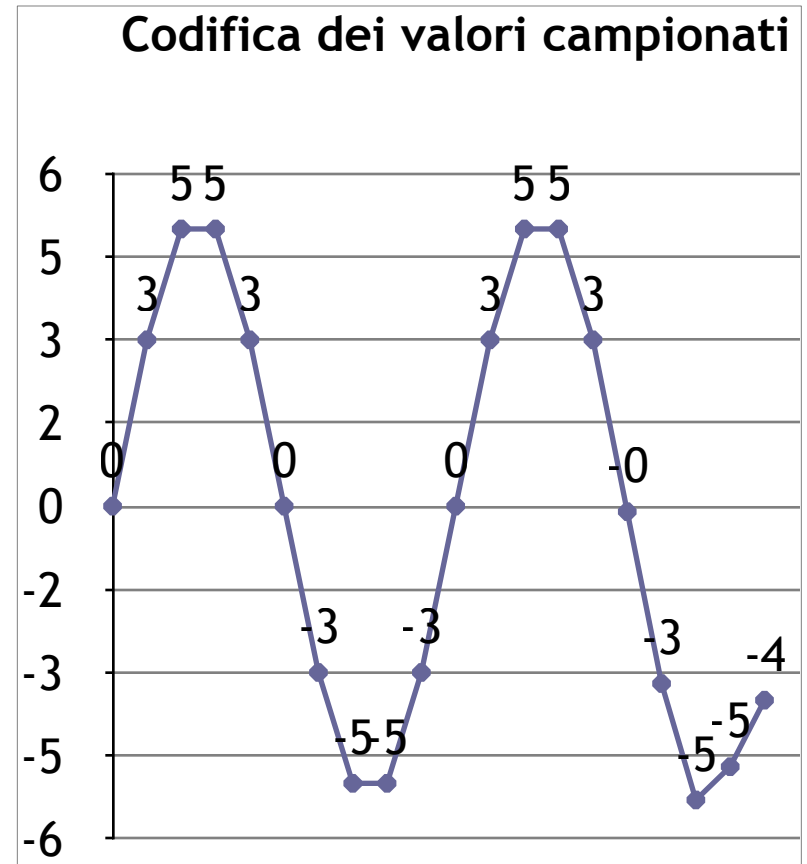
Codifica dei suoni

- Il segnale analogico deve essere convertito in rappresentazione digitale (**digitalizzato**).
 - Quanto più frequentemente il valore di intensità dell'onda viene campionato, tanto più precisa sarà la sua rappresentazione.
 - Il numero di campioni raccolti per ogni secondo definisce la **frequenza di campionamento**. Il numero di campioni raccolti per ogni secondo si misura in Hertz (Hz)
- Es. CD Musicale.
- 44000 campionamenti al secondo,
- Standard: 16 bit per campione



Codifica dei suoni

- La sequenza dei valori numerici ottenuti dai campioni può essere facilmente codificata con sequenze di bit (la **codifica dei numeri** è il prossimo argomento di questa lezione).
- La rappresentazione è tanto più precisa quanto maggiore è il numero di bit utilizzati per codificare l'informazione estratta in fase di campionamento.



Codifica dei suoni

- Codifiche standard
 - MP3 , WAV (MS-Windows), AIFF (Audio Interchange File Format, Apple), MIDI.
- MP3
 - Variante MPEG per suoni.
 - Lossy.
 - Grande diffusione, molto efficiente (fattore di compressione circa 5:1 - 10:1, circa 1-2 MB ogni minuto).
- MIDI: codifica le note e gli strumenti che devono eseguirle
 - Efficiente, ma solo musica, non voce.

Osservazione

- Per calcolare lo spazio occupato da un file di testo, da un'immagine, da un file audio, la tecnica è sempre la stessa:
- Si trova lo 'spazio' (bit) occupato dalla codifica di ogni unità elementare costituente il file (un carattere per il testo, un pixel per l'immagine, un campione per il file audio)
- Si trova il numero di unità elementari che costituiscono il file (il numero di caratteri per il testo, il numero di pixel per l'immagine -sfruttando la risoluzione, il numero di campioni per il file audio - sfruttando la frequenza di campionamento)
- Si moltiplicano queste due quantità

Cosa memorizzo in 1MB?

8 milioni di bit
(~1MB)

Un milione di caratteri
70 caratteri per riga
40 righe per pagina
400 pagine circa



Libro giallo



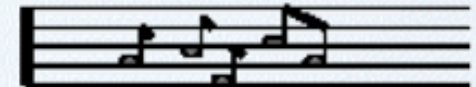
1.000 x 1.000 pixel
256 livelli di grigio



Immagine in bianco
e nero ad alta risoluzione



125 sec. di voce o 5.6 sec.
di musica ad alta fedeltà



1/30 di secondo di filmato
in bianco e nero
ad alta risoluzione



Codifica dei numeri

- Il codice ASCII consente di codificare le cifre decimali da “0” a “9” fornendo in questo modo una rappresentazione dei numeri.
- Per esempio: il numero 324 potrebbe essere rappresentato dalla sequenza di byte:

00110011 00110010 00110100
3 2 4

- Quale è il problema di questa rappresentazione ?

Codifica dei numeri

00110011	00110010	00110100
3	2	4

- Questa rappresentazione non è efficiente e soprattutto *non è adatta per eseguire le operazioni aritmetiche sui numeri* (sono solo simboli allineati).
- Sono stati pertanto studiati **codici alternativi per rappresentare i numeri in modo efficiente** ed eseguire le usuali **operazioni aritmetiche**.

Codifica dei numeri

(il sistema decimale)

- La rappresentazione dei numeri con il sistema decimale può essere utilizzata come punto di partenza per definire un metodo di codifica dei numeri all'interno degli elaboratori.
- **Notazione posizionale**: ogni cifra del numero assume un certo valore in funzione della sua **posizione**
 - Esempio: la sequenza di cifre 324 viene notazione compatta, sta per:
 - 3 centinaia + 2 decine + 4 unità
 - $324 = 3 \times 100 + 2 \times 10 + 4 \times 1$
 - $324 = 3 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$ (notazione esplicita)

Codifica dei numeri

- La numerazione decimale quindi utilizza **una notazione posizionale basata sul numero 10**.
- La notazione posizionale può essere utilizzata in qualunque altro sistema di numerazione.
- Per ogni sistema di numerazione si usa un numero di cifre uguale alla base.
 - **Base:** il numero di cifre disponibile nel sistema.
 - In base 10 (il sistema di numerazione *decimale*), i numeri vengono codificati utilizzando le dieci cifre 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
 - In base 2, (il sistema di numerazione *binario*) i numeri vengono codificati utilizzando le due cifre 0 e 1.

Riepilogo

Codifica in termini binari di diversi input:

- Testo (ASCII, UTF-8...)
- Immagini (Pixel, Risoluzione, Profondità)
- Filmati (riduzione al problema “immagini statiche”)
- Suoni (passaggio da onda sonora analogica a digitale tramite campionamento e quantizzazione)
- Numeri (da binario a decimale)

- Il sistema di numerazione binario (base 2) utilizza una notazione posizionale basata su 2 cifre (0 e 1) e sulle potenze di 2
- Il numero 1001 può essere rappresentato esplicitamente come
$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = ?$$

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$$

Conversione base 2 $\rightarrow$ base 10

- Esempio: la sequenza “1011” denota il numero


$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11 \text{ (in base 10)}$$

- Esempio: la sequenza “10011” denota il numero


$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19 \text{ (in base 10)}$$

- Per evitare ambiguità si usa la notazione

$$1011_2 = 11_{10}, 10011_2 = 19_{10}$$

Conversione base 10 $\rightarrow$ base 2

Per convertire un numero in base 2 si devono trovare i resti delle divisioni successive del numero per la base 2

Esempio: 157_{10}

157 / 2 resto 1		10011101
78 / 2 resto 0		
39 / 2 resto 1		
19 / 2 resto 1		
9 / 2 resto 1		
4 / 2 resto 0		
2 / 2 resto 0		
1 / 2 resto 1		

Conversione base 10 $\rightarrow$ base 2

- Consideriamo ad esempio il numerale 13 in base 10 e cerchiamo la rappresentazione corrispondente in base 2.

Conversione base 10 $\rightarrow$ base 2

13/2	=	6	resto	1	
6/2	=	3	resto	0	
3/2	=	1	resto	1	
1/2	=	0	resto	1	

13 espresso in base 2
1101

Conversione base 10 $\rightarrow$ base 2

- Consideriamo ad esempio il numerale 42 in base 10 e cerchiamo la rappresentazione corrispondente in base 2.

Conversione base 10 $\rightarrow$ base 2

42/2	=	21	resto	0
21/2	=	10	resto	1
10/2	=	5	resto	0
5/2	=	2	resto	1
2/2	=	1	resto	0
1/2	=	0	resto	1



42 espresso in base 2
101010

Esercizi: conversione base 2 -> 10

- 111011
- 101111101
- 110000
- 110010101
- 101010101010
- 101000000001

Suggerimenti:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Ancora: $2^9 = 512$. $2^{10} = 1024$. $2^{11} = 2048$

Soluzioni

- $111011 =$
- $101111101 = 381$
- $110000\ 48 =$
- $110010101 = 405$
- $101010101010 = 2730$
- $101000000001 = 2561$

Esercizi: conversione base 10-> 2

128

1234

5654

1233

234

511

Soluzioni

128	10000000
1234	10011010010
5654	1011000010110
1233	10011010001
234	11101010
511	111111111

Es. Operazioni

- Addizione Binaria

$0 + 0$	$=$	0	con riporto	0
$0 + 1$	$=$	1	con riporto	0
$1 + 0$	$=$	1	con riporto	0
$1 + 1$	$=$	0	con riporto	1

Addizione binaria

$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$	$\begin{array}{r} 101 + \\ 11 = \\ \hline 1000 \end{array}$	$\begin{array}{r} 10110101 + \\ 1000110 = \\ \hline 11111011 \end{array}$
--	---	---

$0 + 0 = 0$	con riporto	0
$0 + 1 = 1$	con riporto	0
$1 + 0 = 1$	con riporto	0
$1 + 1 = 0$	con riporto	1

Addizione binaria

$$\begin{array}{rcl} 10111001 & + & (185) \\ 10000100 & = & (132) \\ \hline 10011101 & & (317) \end{array}$$

Notazione esadecimale

- Vale la pena menzionare un altro tipo di notazione: la notazione **esadecimale**, ossia in base 16.
 - Il principio è lo stesso della notazione decimale (base 10) e binaria (base 2).
 - Si usano 16 cifre: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

– Il numero **B7FC** può essere rappresentato esplicitamente come

$$(11) \times 16^3 + 7 \times 16^2 + (15) \times 16^1 + (12) \times 16^0 = 47100$$

A	B	C	D	E	F
↓	↓	↓	↓	↓	↓
10	11	12	13	14	15

Notazione esadecimale

- Vale la pena menzionare un altro tipo di notazione: la notazione **esadecimale**, ossia in base 16.
 - Il principio è lo stesso della notazione decimale (base 10) e binaria (base 2).
 - Si usano 16 cifre: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- **Esempi:**
 - $A3_{16} = 10 \times 16^1 + 3 \times 16^0 = 160 + 3 = 163_{10}$
 - $101_{16} = 1 \times 16^2 + 0 \times 16^1 + 1 \times 16^0 = 256 + 0 + 1 = 257_{10}$
 - $FAB_{16} = 15 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 = 3840 + 160 + 11 = 4011_{10}$
- Usata spesso in ambito informatico perché:
 - **Più compatta** della notazione binaria (e anche di quella decimale!)

A93E ₁₆ (43326 ₁₀)	
$A_{16} = 10_{10} = 1010_2$	$9_{16} = 9_{10} = 1001_2$
$3_{16} = 3_{10} = 0011_2$	$E_{16} = 14_{10} = 1110_2$
1010100100111110 ₂	

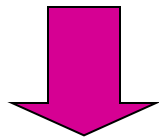
Specifica dei Colori nel Web

- Nelle pagine Web i colori vengono codificati utilizzando il sistema esadecimale.
- Perché? La rappresentazione è più concisa, esprimo tutto con due caratteri alfanumerici per **colore primario** (vedi slide seguente)

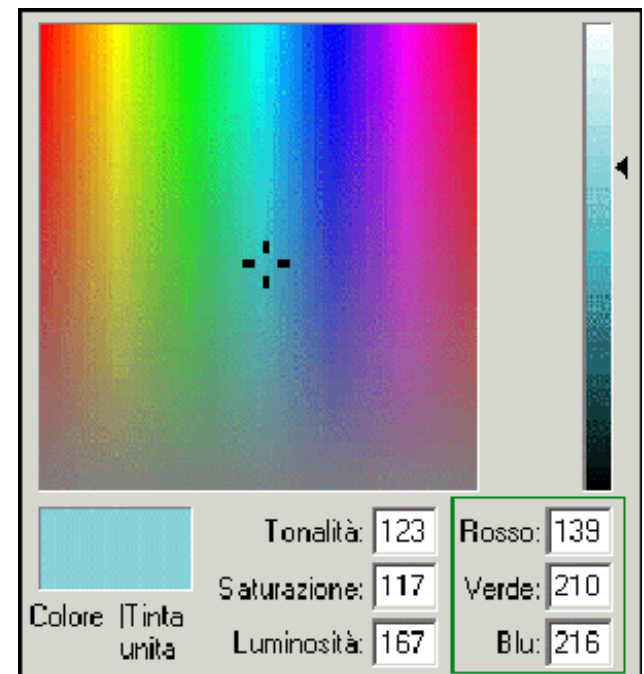
$$139_{10} = 8B_{16}$$

$$210_{10} = D2_{16}$$

$$216_{10} = D8_{16}$$

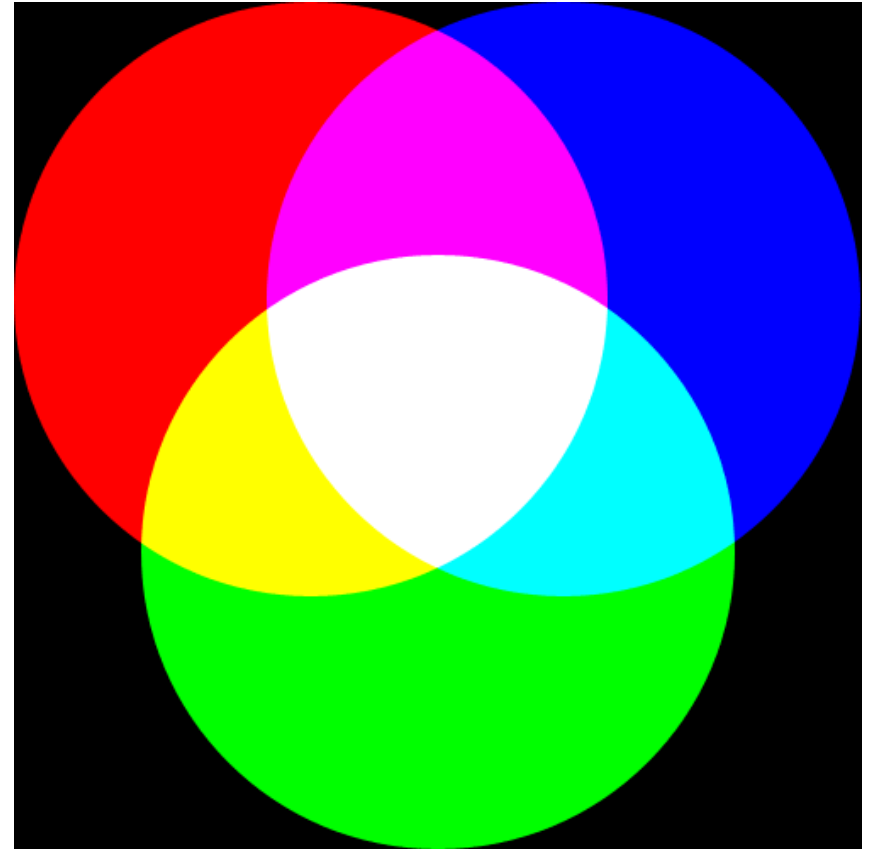


Codifica del colore:
8B D2 D8



L'uso del colore

- Il colore può essere generato componendo 3 colori: red, green, blue (codifica RGB).
 - secondo questo modello qualsiasi colore può essere rappresentato componendo Rosso Verde e Blu (colori primari)
 - idea base: invece che di partire da tanti colori e di rappresentarne diverse sfumature, possiamo rappresentare molte sfumature/gradazioni a partire dai 3 colori primari

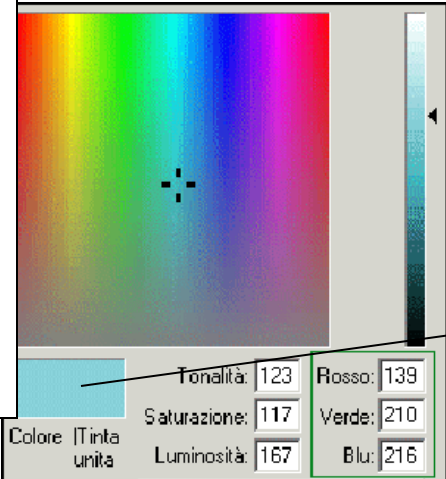


Tutti i possibili colori sono ottenuti dalla combinazione di diverse gradazioni dei 3 colori primari

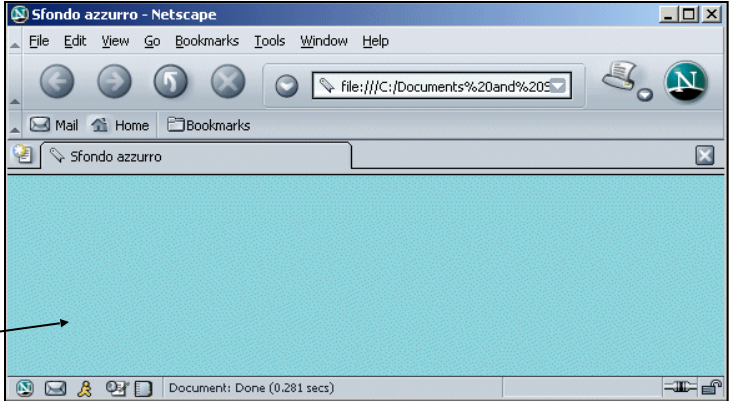
Specifica dei Colori nel Web

- In particolare quando si vuole specificare un colore è sempre possibile specificarlo specificando la sua codifica esadecimale RGB.

```
1
2 <html>
3 <head>
4 <title>Sfondo azzurro</title>
5 </head>
6 <body bgcolor="#8BD2D8">
7 </body>
8 </html>
9
```



Color selection tool showing a rainbow gradient and a color picker. The selected color is a light blue. The RGB values are: Rosso: 139, Verde: 210, Blu: 216. The hex code is #8BD2D8.



Screenshot of a Netscape browser window titled "Sfondo azzurro - Netscape". The browser shows a light blue background, which is the result of the HTML code. The address bar shows the file path: file:///C:/Documents%20and%20... The status bar at the bottom indicates "Document: Done (0.281 secs)".

Ognuno dei 16.777.216 colori dello spettro RGB e' associato a un codice esadecimale che ci permette di identificarlo

Specifica dei Colori nel Web

- Alcune combinazioni da ricordare:

#00 00 00 = nero

#FF FF FF = bianco

#FF 00 00 = rosso puro

#00 FF 00 = verde puro

#00 00 FF = blu puro

#FF FF 00 = giallo

#FF 00 FF = viola



N.B.

- Non vediamo (interi con segno, overflow, numeri a virgola mobile)