

INFORMATICA

Intelligenza Artificiale & Data Analytics

Precorsi a.a. 2023/24

Docente: Gloria Pietropolli

3. APPROFONDIMENTO SUI FILE

virtual box: <https://tecadmin.net/how-to-install-virtualbox-on-macos/>
ubuntu + first vm
https://linuxhint.com/install_ubuntu_virtualbox_2004/#:~:text=First%2C%20open%20VirtualBox.&text=Now%2C%20type%20in%20a%20name,emory%20size%20for%20the%20VM.

FILE BINARI

I file binari sono una tipologia di file che:

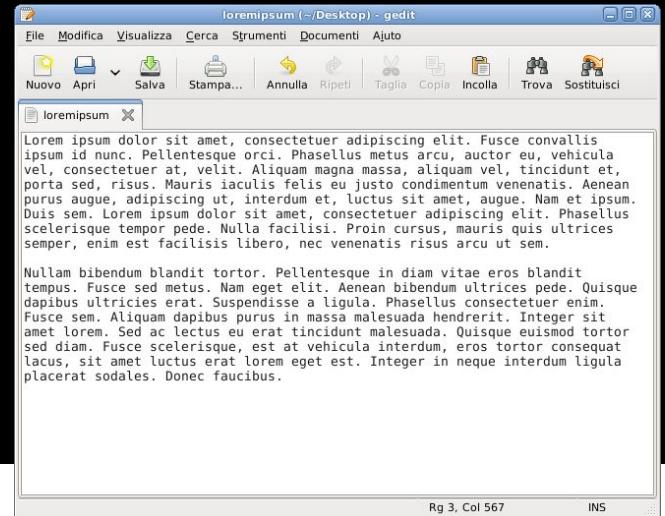
- 1) contengono dati in un formato non testuale
- 2) non sono leggibili da noi umani, contengono dati codificati non comprensibili
- 3) conservano info che richiedono rappresentazione dettagliata e specifica

A differenza dei file di testo, possono contenere dati complessi:

- **immagini** (come JPG o PNG)
- **audio** (come MP3 o WAV)
- **video** (come MP4 o AVI)
- **software eseguibili** (come EXE su Windows)

0000	FF	D8	FF	E1	1D	FE	45	78	69	66	00	00	49	49	2A	00
0010	08	00	00	00	09	00	0F	01	02	00	06	00	00	00	7A	00
0020	00	00	10	01	02	00	14	00	00	00	80	00	00	00	12	01
0030	03	00	01	00	00	00	01	00	00	00	1A	01	05	00	01	00
0040	00	00	A0	00	00	00	1B	01	05	00	01	00	00	00	A8	00
0050	00	00	28	01	03	00	01	00	00	00	02	00	00	00	32	01
0060	02	00	14	00	00	00	B0	00	00	00	13	02	03	00	01	00
0070	00	00	01	00	00	00	69	87	04	00	01	00	00	00	C4	00
0080	00	00	3A	06	00	00	43	61	6E	6F	6E	00	43	61	6E	6F
0090	6E	20	50	6F	77	65	72	53	68	6F	74	20	41	36	30	00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	B4	00	00	00
00B0	01	00	00	00	B4	00	00	00	01	00	00	00	32	30	30	34
00C0	3A	30	36	3A	32	35	20	31	32	3A	33	30	3A	32	35	00
00D0	1F	00	9A	82	05	00	01	00	00	00	86	03	00	00	9D	82
00E0	05	00	01	00	00	00	8E	03	00	00	00	90	07	00	04	00

FILE DI TESTO



Rg 3, Col 567

INS

Un file di testo contiene solo dati testuali (codifica binaria di caratteri comprensibili a un lettore umano) come lettere, numeri e segni di punteggiatura.

Possiamo rappresentare dati numerici?

I dati numerici possono essere rappresentati in un file di testo utilizzando caratteri numerici e simboli appropriati. Ad esempio, "42" rappresenta il numero 42.

Possiamo rappresentare dati testuali?

I dati testuali sono la forma nativa dei file di testo e possono rappresentare qualsiasi sequenza di caratteri comprensibili, come parole, frasi o documenti.

Possiamo rappresentare istruzioni?

Le istruzioni o comandi possono essere incluse in un file di testo per scopi di programmazione o scripting. Ad esempio, "stampare('Ciao, mondo!')" rappresenta un comando di stampa in un linguaggio di programmazione.

CODIFICHE PER LA RAPPRESENTAZIONE DI SIMBOLI TESTUALI IN BINARIO

Abbiamo visto la codifica ASCII

- Utilizza 7 bit per codificare 128 caratteri

Notate qualche criticità?

- Mancano numerosi **simboli di punteggiatura**
- Mancano **caratteri di alfabeti internazionali**

ASCII Code Chart															
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Negli anni sono nate nuove codifiche:
Unicode è oggi la più usata

UNICODE

Consorzio di aziende interessate a fornire i loro servizi in più lingue ed alfabeti utilizzando una codifica comune

Nato nel 1991, estende la codifica a 16 bit

Quanti byte?

Quanti simboli posso codificare così?

Initial repertoire covers these scripts: Arabic, Armenian, Bengali, Bopomofo, Cyrillic, Devanagari, Georgian, Greek and Coptic, Gujarati, Gurmukhi, Hangul, Hebrew, Hiragana, Kannada, Katakana, Lao, Latin, Malayalam, Oriya, Tamil, Telugu, Thai, and Tibetan.^[26]

Unicode 13.0 è stato esteso a 21 bit, coprendo un numero enorme di alfabeti da tutto il mondo

Tabelle dei caratteri:

https://en.wikipedia.org/wiki/Unicode#Code_planes_and_blocks

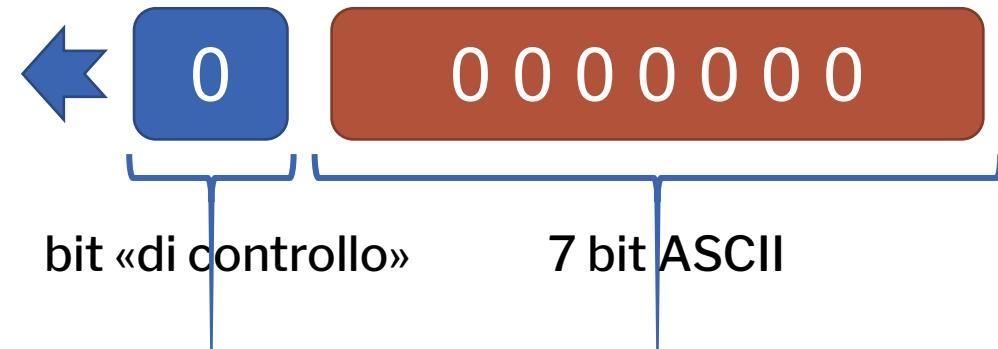
UTF-8

UTF-8 è uno stratagemma utilizzato per poter applicare la codifica Unicode con parsimonia

L'idea è quella che, per i primi 128 caratteri ASCII, possiamo evitare di usare più di 1 byte, alleggerendo la pesantezza dei file di testo

0 = carattere ASCII
→leggi i 7 bit successivi e converti

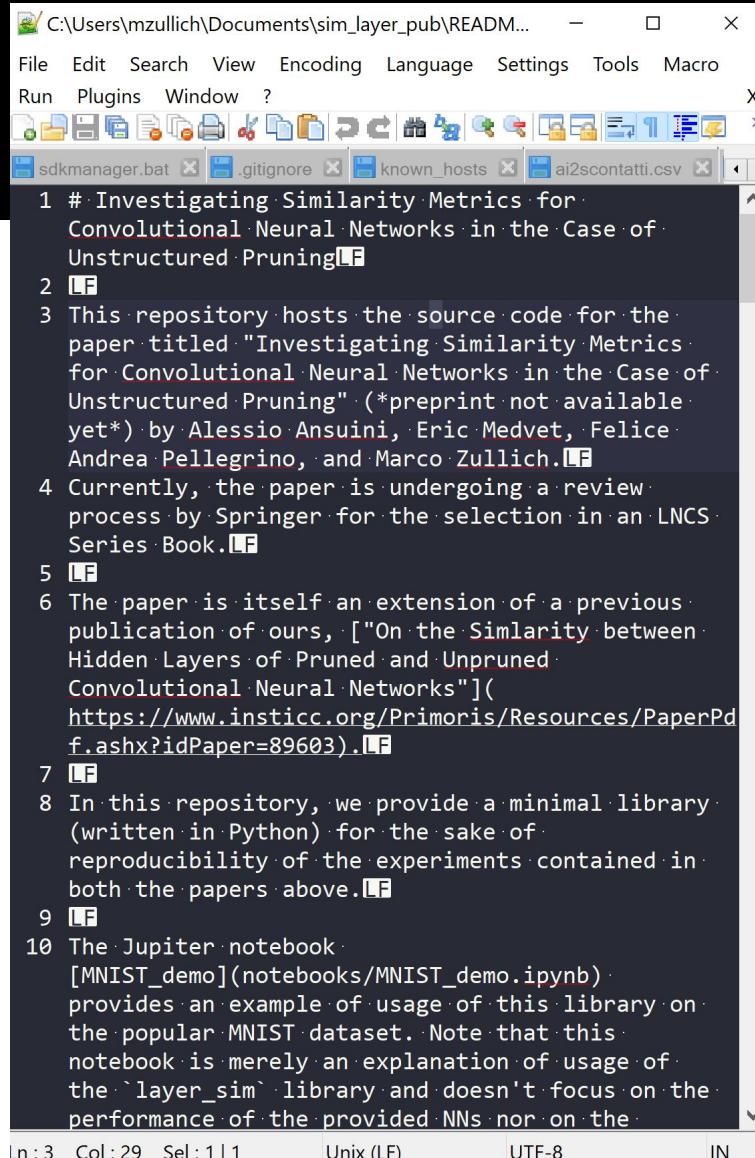
1 = carattere non-ASCII
→leggi i anche i byte successivi
Quanti? fino a 4; dipende dai bit di controllo dei byte successivi



VISUALIZZAZIONE DEI FILE TESTUALI

I file testuali possono essere visualizzati da appositi programmi, detti editor di testo (*text editor* - TE), che si occupano di:

- Convertire il file dal linguaggio binario al linguaggio umano, usando la codifica opportuna
- Visualizzare a schermo (o, eventualmente, stampare) il contenuto del file

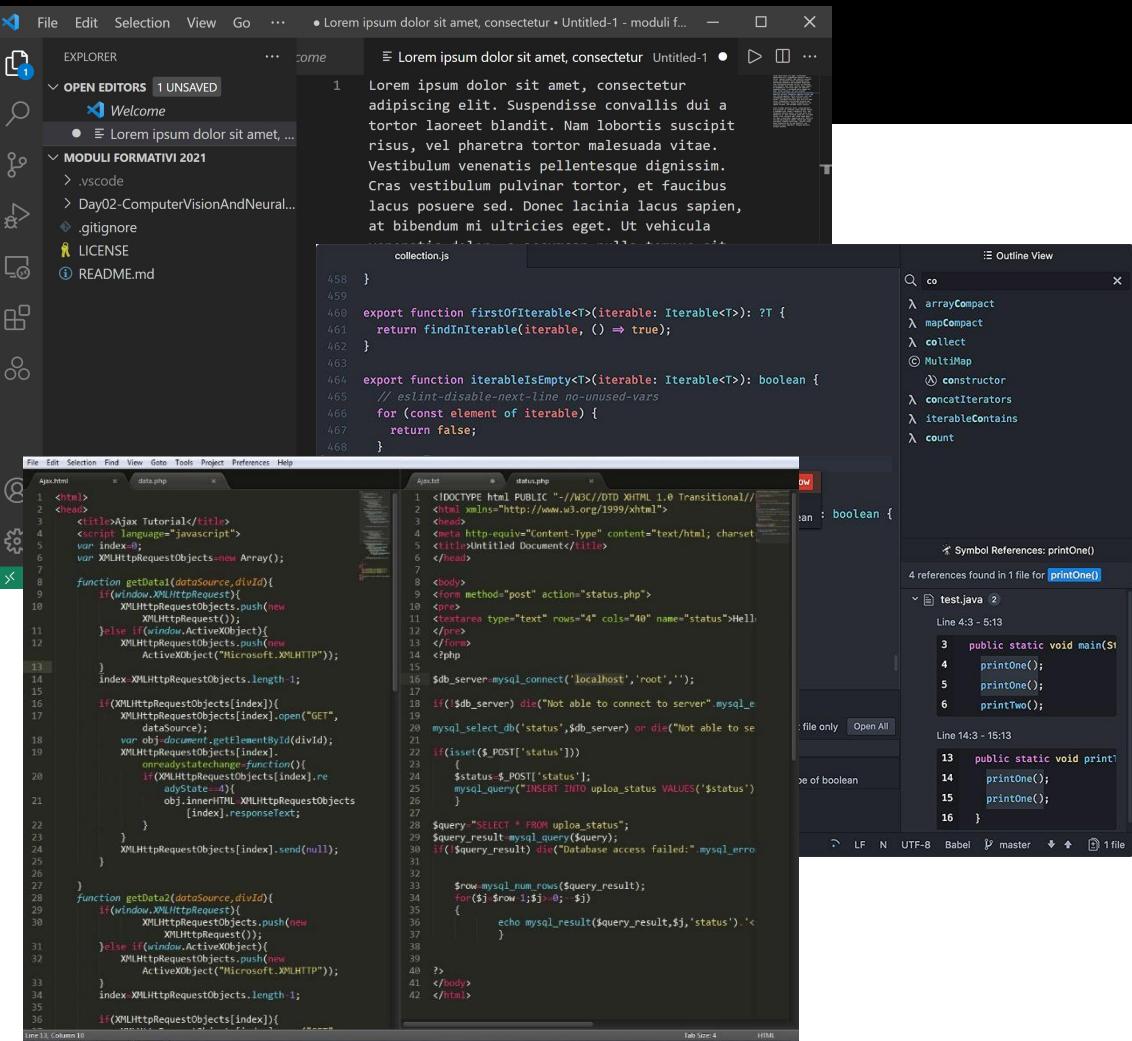
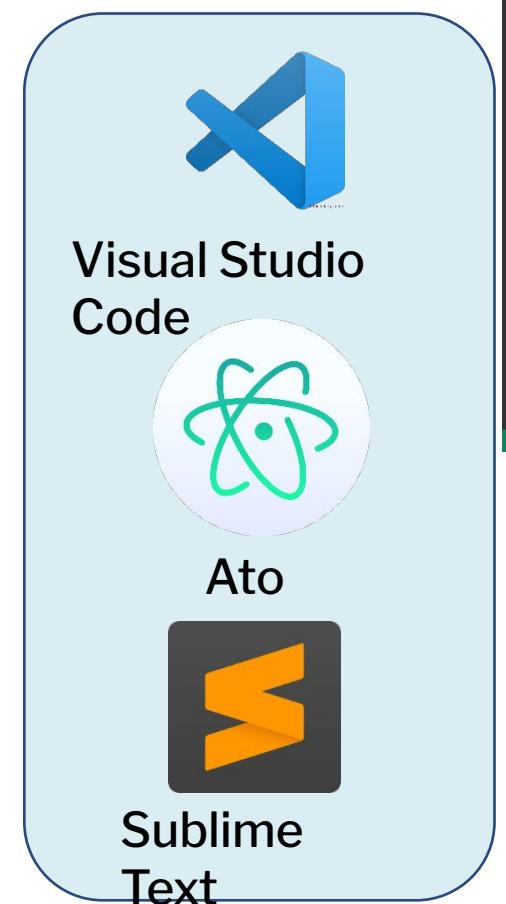


The screenshot shows a Windows-style text editor window titled 'C:\Users\mzullich\Documents\sim_layer_pub\README...'. The window contains the following text:

```
1 # Investigating Similarity Metrics for  
2 Convolutional Neural Networks in the Case of  
3 Unstructured Pruning  
4 This repository hosts the source code for the  
5 paper titled "Investigating Similarity Metrics  
6 for Convolutional Neural Networks in the Case of  
7 Unstructured Pruning" (*preprint not available  
8 yet*) by Alessio Ansuini, Eric Medvet, Felice  
9 Andrea Pellegrino, and Marco Zullich.  
10 Currently, the paper is undergoing a review  
11 process by Springer for the selection in an LNCS  
12 Series Book.  
13 The paper is itself an extension of a previous  
14 publication of ours, "[On the Similarity between  
15 Hidden Layers of Pruned and Unpruned  
16 Convolutional Neural Networks]"  
https://www.insticc.org/Primoris/Resources/PaperPdf.ashx?idPaper=89603.  
17 In this repository, we provide a minimal library  
18 (written in Python) for the sake of  
19 reproducibility of the experiments contained in  
20 both the papers above.  
21 The Jupiter notebook  
22 [MNIST_demo](notebooks/MNIST_demo.ipynb)  
23 provides an example of usage of this library on  
24 the popular MNIST dataset. Note that this  
25 notebook is merely an explanation of usage of  
26 the `layer_sim` library and doesn't focus on the  
27 performance of the provided NNs nor on the
```

Ln : 3 Col : 29 Sel : 1 | 1 Unix (LF) UTF-8 IN

TE CONSIGLIATI



The screenshot shows a Windows Notepad window with two tabs open. The top tab contains a .gitignore file with the following content:

```
1 # game archive
2 game.zip
3 LF
4 # Compiled Lua sources
5 luac.out
6 LF
7 # luarocks build files
8 *.src.rock
9 *.zip
10 *.tar.gz
11 LF
12 # Object files
13 *.o
14 *.osi
15 *.kolo
16 *.obj
17 *.elf
```

The bottom tab contains a Python script named architectures.py, which imports torch.nn and defines an MLP class. The script uses nn.Sequential to stack layers like nn.Flatten, nn.Linear(28*28, 16), nn.ReLU, nn.BatchNorm1d, nn.Linear(16, 32), nn.ReLU, and nn.Dropout(p=.2). It also includes nn.BatchNorm1d(num_features=32) and nn.Linear(32, 24) layers.

```
import torch.nn as nn
from torch import Tensor

class MLP(nn.Module):
    def __init__(self):
        super().__init__()
        self.layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(28*28, 16),
            nn.ReLU(),
            nn.Dropout(p=.2), # we add a dropout here. it's

            nn.BatchNorm1d(num_features=16),
            nn.Linear(16, 32),
            nn.ReLU(),
            nn.Dropout(p=.2), # we add a dropout here. it's

            nn.BatchNorm1d(num_features=32),
            nn.Linear(32, 24),
            nn.ReLU(),
        )

architectures.py [unix] (00:20 10/04/2021)
architectures.pyv [noeo1][unix] 63L 2170B
```

CARATTERI DI CONTROLLO: A CAPO

Zoomando nel file precedente, notiamo un particolare simbolo alla fine di ogni riga

In altri file, ne notiamo uno aggiuntivo

Questi caratteri indicano al TE di andare a capo.

Line Feed (LF) è il carattere che identifica l'«a capo» nei sistemi Unix (Linux, Mac). Vuol dire letteralmente «fornitura riga».

Carriage Return (CR) invece simboleggia il rientro del carrello di una macchina da scrivere.

```
# Investigating Similarity Metrics for  
Convolutional Neural Networks in the Case of  
Unstructured PruningLF
```

```
1 private int function(numero):CRLF  
2 ... global xyz = 55CRLF  
3 ... return 3 * xCRLF  
4 CRLF  
5 x = 0CRLF  
6 y = function(x)CRLF  
7 numero = 1CRLF  
8 print(xyz)CRLF  
9 CRLF  
10 CRLF  
11 CRLF
```



PROBLEMA: MODIFICHE MASSIVE AD UN FILE

Giovanni ha appena ricevuto un file di testo da Alessandra

Alessandra utilizza Ubuntu (Linux), mentre Giovanni utilizza Windows

Aprendo il file con un editor di testo, Giovanni nota un problema di visualizzazione: mancano tutti gli «a capo»

Investigating Similarity Metrics for Convolutional Neural Networks in the Case of Unstructured Pruning
This repository hosts the source code for the paper titled "Investigating Similarity Metrics for Convolutional Neural Networks in the Case of Unstructured Pruning" (*preprint not available yet*) by Alessio Ansuini, Eric Medvet, Felice Andrea Pellegrino, and Marco Zullich. Currently, the paper is undergoing a review process by Springer for the selection in an LNCS Series Book. The paper is itself an extension of a previous publication of ours, ["On the Similarity between Hidden Layers of Pruned and Unpruned Convolutional Neural

Che cosa è successo?

Come possiamo porvi rimedio?

[NOZIONI DI] ESPRESSIONI REGOLARI

La modifica “a mano” di problemi di formattazione/codifica file generalizzati può diventare un lavoro

- impegnativo
- soggetto ad errori

Strumenti che permettono di effettuare una sostituzione massiva in maniera automatica e precisa, in pochissimo tempo.

Le **espressioni regolari (regex)** sono degli strumenti che permettono di identificare **pattern (motivi) ripetuti** all’interno di stringhe di testo: una volta identificati questi pattern, possiamo proporre ad un programma come Notepad++ una stringa per sostituirli

PATTERN

L'idea base di un'espressione regolare è che vogliamo, all'interno di un file di testo, andare a trovare delle **sottostringhe** che corrispondono ad un specifico pattern

Un pattern è una generica combinazione di caratteri (anche uno solo!) da identificare *consecutivamente e nel medesimo ordine* all'interno della stringa.

stringa
abcdefghijklmnopqrstuvwxyz
ghijklmno
sottostringa

Nell'esempio qui sopra...

«ghijklmno» è un pattern che corrisponde alla stringa, in quanto le lettere nella stringa si trovano consecutivamente e nel medesimo ordine.

«abde»?

«onm»?

CARATTERI SPECIALI

Le regex ammettono anche **caratteri speciali** nei pattern.

Siccome questi caratteri, come `\n` o `\r`, non si possono trovare su una tastiera, si utilizzano delle combinazioni speciali di caratteri per esprimelerli.

Queste combinazioni speciali iniziano con il backslash «\», che viene anche chiamato **escape character** ad indicarne la sua specialità.

Possiamo esprimere LF con il carattere `\n`

Possiamo esprimere CR con il carattere `\r`

I caratteri `^` e `$` identificano rispettivamente l'inizio e la fine di una stringa

RISOLVERE IL PROBLEMA

Ora abbiamo gli strumenti per risolvere il problema di Giovanni.

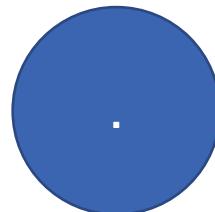
Pattern

Carattere/i in sostituzione

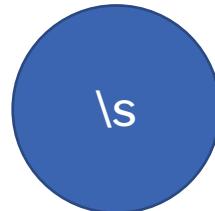
OPERIAMO LA SOLUZIONE
SU NOTEPAD++

METACARATTERI

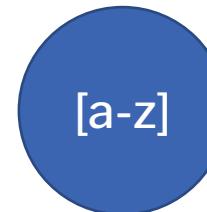
Inoltre, nelle regex possiamo usare anche i c.d. *metacaratteri*, ovvero gruppi di caratteri che possiamo utilizzare nei pattern per dare maggiore flessibilità alla nostra ricerca di pattern.



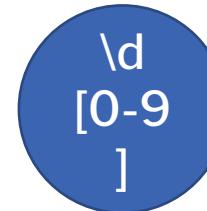
Tutti i caratteri (speciali, a capo, lettere/numeri...)



Tutti i caratteri di spaziatura (spazio, a capo...)



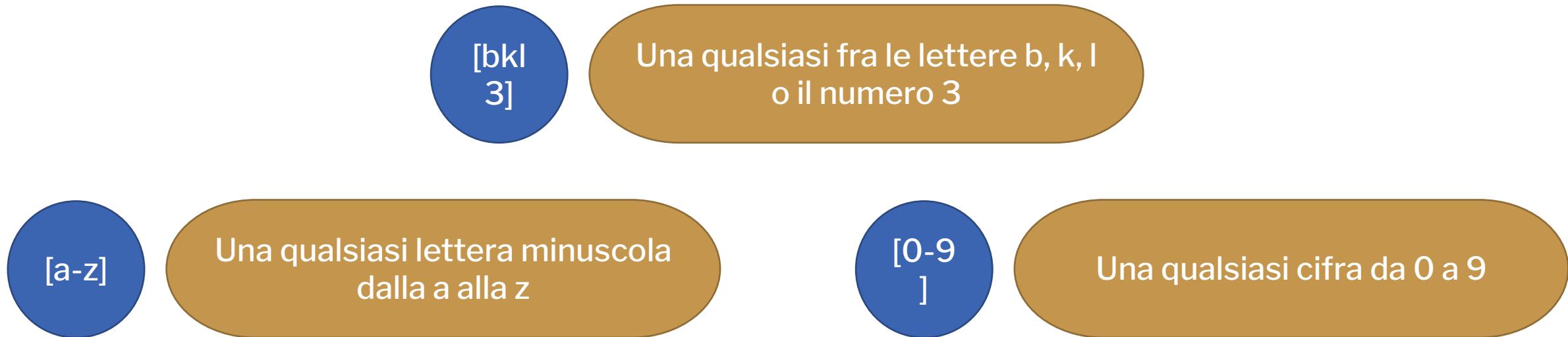
Tutte le lettere minuscole dalla a alla z



Tutte le cifre

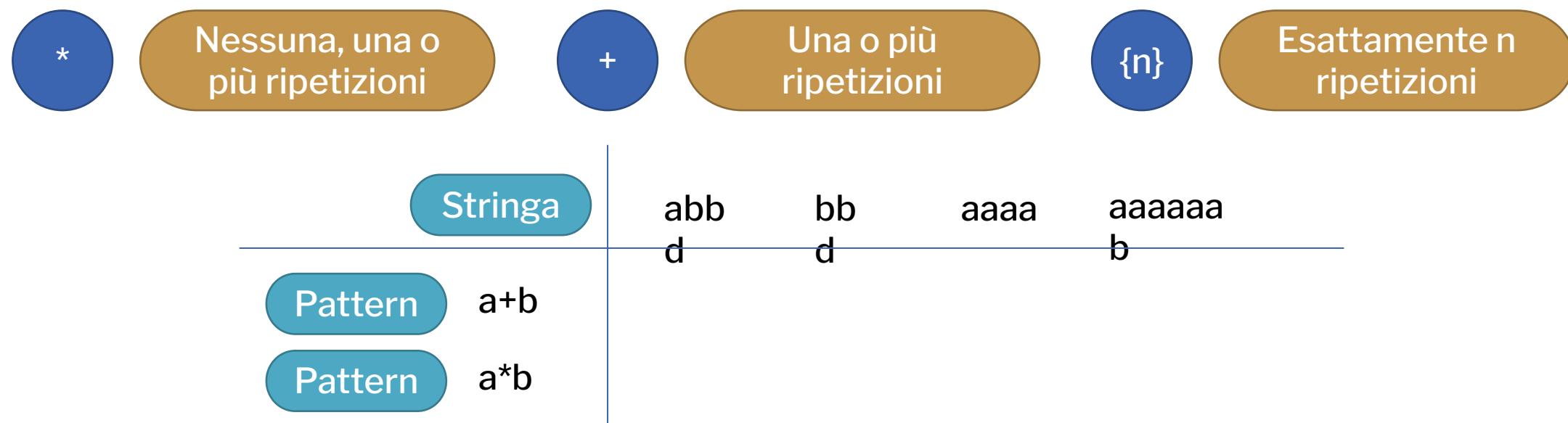
GRUPPI DI CARATTERI

Con le regex è anche possibile indicare specifici gruppi di caratteri usando le parentesi quadre:



QUANTIFICATORI

Possiamo anche utilizzare dei caratteri speciali per quantificare il numero di ripetizioni consecutive del pattern che andiamo a ricercare.



UN ESERCIZIO LEGGERMENTE PIÙ DIFFICILE

Nicoletta gestisce un forum con alcuni utenti iscritti.

Oggi deve fare una modifica al programma che gestisce il sito.

Ha appena scoperto che vi sono alcuni utenti il cui nome inizia per un numero e che possono dare problemi a detto programma.

Nicoletta deve quindi reperire tutti gli utenti i cui nomi iniziano con un numero affinché questo numero venga rimosso dal nome utente.

SixPackMuscularM
an

Interista05
1

1InformaticoBell
o

123456stell
a

Quale pattern posso utilizzare?

VISUALIZZARE ALTRI TIPI FILE CON UN TE

Alcuni TE, come Notepad++, supportano l'apertura di file di formato arbitrario.

Possiamo provare ad aprire:

- File binari
 - Immagini

Notiamo un fenomeno bizzarro. Qualcuno sa cosa sta succedendo?

RIASSUMENDO

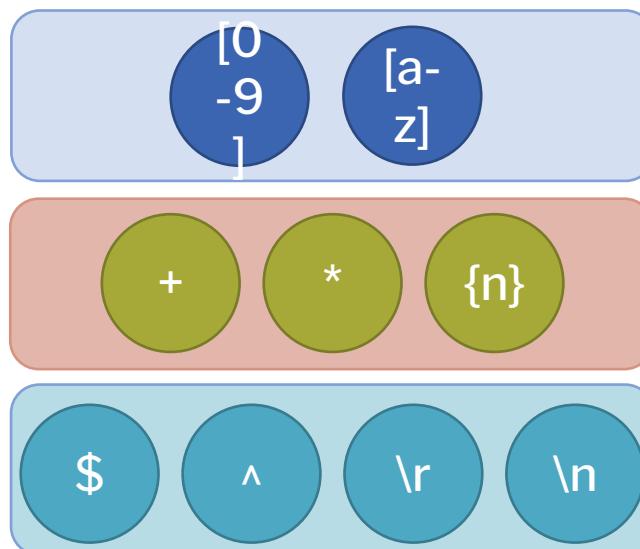
I file binari sono file contenenti **qualsiasi tipo di dato in formato binario**. Possono contenere anche istruzioni in linguaggio macchina.

I file di testo invece possono contenere solamente stringhe di testo codificate in linguaggio binario.

Esistono vari sistemi per codificare caratteri testuali in binario: noi abbiamo visto ASCII e Unicode (con codifica UTF-8). Entrambe le codifiche prevedono l'utilizzo di caratteri di controllo (che non hanno un corrispettivo simbolo visuale) ma che influenzano la maniera in cui un file di testo viene visualizzato da un editor di testo.

A volte è necessario operare modifiche massive ai file di testo: ci possono venire in aiuto le espressioni regolari, che ci permettono, tramite l'ausilio di caratteri speciali, gruppi e quantificatori, di identificare *pattern* ricorrenti all'interno di un testo, per poi possibilmente sostituire tali pattern con altri a nostro piacimento.

```
1 private int function(numero):\r\n2     global xyz = 55\r\n3     return 3 * x\r\n4 \r\n5 x = 0\r\n6 y = function(x)\r\n7 numero = 1\r\n8 print(xyz)\r\n9 \r\n10 \r\n11 
```



3.

APPROFONDIMENTO SUI FILE

Il formato *markdown* (.md)

OK I FILE DI TESTO, MA... LA FORMATTAZIONE?

I file di testo sono leggeri e permettono di trasportare un grosso numero di informazioni.

Tuttavia mancano numerose opzioni per la visualizzazione *differenziata* di particolari tipi di testo o dato:

- Titoli
- Formule matematiche
- Link
- Immagini

Definition

A function of a real variable $y = f(x)$ is differentiable at a point a of its domain, if its domain contains an open interval I containing a , and the limit

$$L = \lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h}$$

exists. This means that, for every positive real number ε (even very small), there exists a positive real number δ such that, for every h such that $|h| < \delta$ and $h \neq 0$ then $f(a + h)$ is defined, and

$$\left| L - \frac{f(a + h) - f(a)}{h} \right| < \varepsilon,$$

where the vertical bars denote the absolute value (see (ε, δ) -definition of limit).

If the function f is differentiable at a , that is if the limit L exists, then this limit is called the derivative of f at a , and denoted $f'(a)$ (read as "f prime of a ") or $\frac{df}{dx}(a)$ (read as "the derivative of f with respect to x at a ", " dy by dx at a ", or " dy over dx at a "); see § Notation (details), below.

IL FORMATO MARKDOWN

Il **markdown** è un formato di file che opera piccole modifiche ai file *plaintext*

aggiunge combinazioni di caratteri che vanno ad indicare che una specifica parte del testo è relativa ad un titolo o ad una formula matematica

Definition

A function of a real variable $y = f(x)$ is *differentiable* at a point a of its domain, if its domain contains an open interval I containing a , and the limit

$$L = \lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h}$$

exists. This means that, for every positive real number ε (even very small), there exists a positive real number δ such that, for every h such that $|h| < \delta$ and $h \neq 0$ then $f(a + h)$ is defined, and

$$\left| L - \frac{f(a + h) - f(a)}{h} \right| < \varepsilon,$$

where the vertical bars denote the absolute value (see (ε, δ) -definition of limit).

If the function f is differentiable at a , that is if the limit L exists, then this limit is called the *derivative* of f at a , and denoted $f'(a)$ (read as "f prime of a ") or $\frac{df}{dx}(a)$ (read as "the derivative of f with respect to x at a ", " dy by dx at a ", or " dy over dx at a "); see § Notation (details), below.

COME FUNZIONA?

Titolo

Aggiungo uno o più cancelletti/hashtag (#) all'inizio della riga.
Ogni cancelletto aggiuntivo indica un titolo meno *importante*

Italico /
Grassetto

Circondo le parole che voglio rendere italiche o grassette
rispettivamente con underscore (_) o due asterischi (**)

Link

Due parti: sito a cui linko e testo da visualizzare. Indico
prima il testo da visualizzare fra parentesi quadre e quindi il
sito fra parentesi tonde

Formula
matematica

Circondo il testo che voglio far diventare formula
matematica con uno o due dollari (\$)

Funzione
Funzione iniettiva

frase grassetta
frase italica

[facebook](www.facebook.com)

\$x+y=z\$
\$\$E=mc^2\$\$

AUTOREMMA FORMATTARE CORRETTAMENTE IL TESTO

Italico Funzione Titolo Link a wikipedia

Una funzione è una relazione tra due insiemi, chiamati dominio e codominio della funzione, che associa a ogni elemento del dominio uno e un solo elemento del codominio.

Sia X il dominio e Y il codominio, la relazione è indicata così: $f:X \rightarrow Y$.

**Formula
matematica**

NB: link alla pagina insiemi di wikipedia:
<https://it.wikipedia.org/wiki/Insieme>

Funzione

Una funzione è una relazione tra due insiemi, chiamati dominio e codominio della funzione, che associa a ogni elemento del dominio uno e un solo elemento del codominio.

Sia X il dominio e Y il codominio, la relazione è indicata così: $f:X \rightarrow Y$.

COME VISUALIZZARE IL RISULTATO FINALE?

Il file **markdown** è un file di testo contenente righe di codice
Il codice è composto dai testi del documento + i caratteri collegati con la formattazione e alcuni costrutti speciali

Un **compilatore** si occupa di convertire il file di testo in una sequenza di istruzioni che fa sì che il testo possa essere visualizzato, comprensivo di formattazione, sullo schermo (oppure stampato)

online: stackedit.io

offline: utilizzare software come VSCode + estensione Markdown.

RIASSUMENDO

Il problema con la visualizzazione dei file di testo è che non abbiamo controllo sulla formattazione delle varie parti di questo file una volta che esso verrà stampato o visualizzato a schermo.

Il markdown rappresenta uno dei vari metodi per poter aggiungere alcuni livelli di formattazione al testo.

Abbiamo controllo, ad esempio, su titoli/sottotitoli, formule matematiche, testo grassetto/italico, link, ecc.

La filosofia del markdown è quella di aggiungere il minor *overhead* possibile rispetto ad un regolare file di testo per poter indicare la formattazione.

3.

APPROFONDIMENTO SUI FILE

I file temporanei e nascosti, immagini, archivi

I FILE TEMPORANEI

L'OS può dover salvare informazioni temporanee -che però non possono stare in memoria- su disco

Questi files vengono rimossi ciclicamente dai computer, di solito in fase di chiusura del sistema (quando spengo il PC)

File Temporanei (FT)

- questo non è un formato file, ma uno stato di un determinato file che ne impone la rimozione dal sistema appena non più utile

I FT IN WINDOWS E UNIX

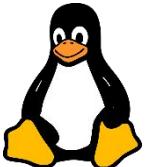
Sia su Windows, che su Linux, che su Mac, i file non hanno un *tag* che ne indica la temporaneità; tuttavia vengono salvati in specifiche cartelle che vengono ciclicamente ripulite da parte dell'OS.



Su Win, questa cartella risiede in C:\users\<current_user>\AppData\Local\Temp



Su MacOS, abbiamo sempre /tmp, più ~/Library/Caches

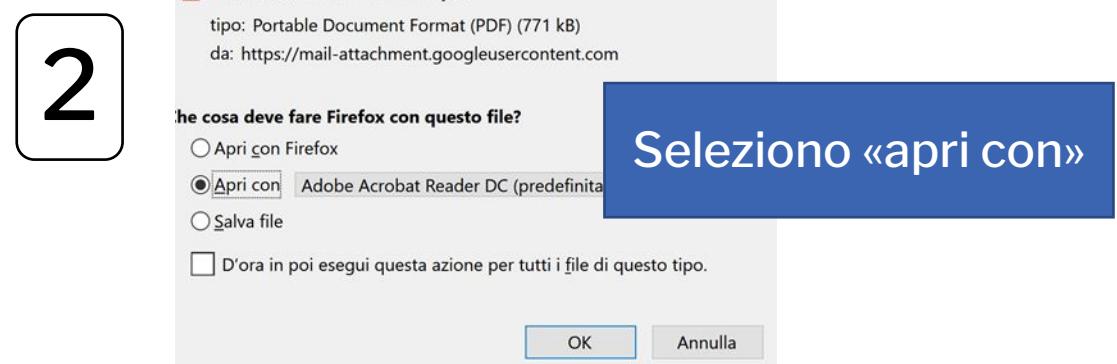


SALVATAGGIO DI UN FT

Uno dei metodi più veloci per andare a causare la creazione di un FT è tramite lo scaricamento di un file dal proprio browser. Proviamo a scaricare, ad esempio, un allegato di un'e-mail.



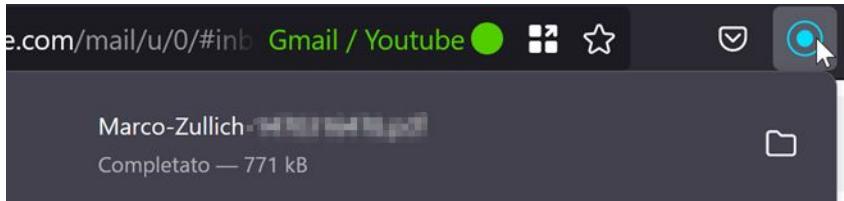
Selezione «scarica»



Selezione «apri con»

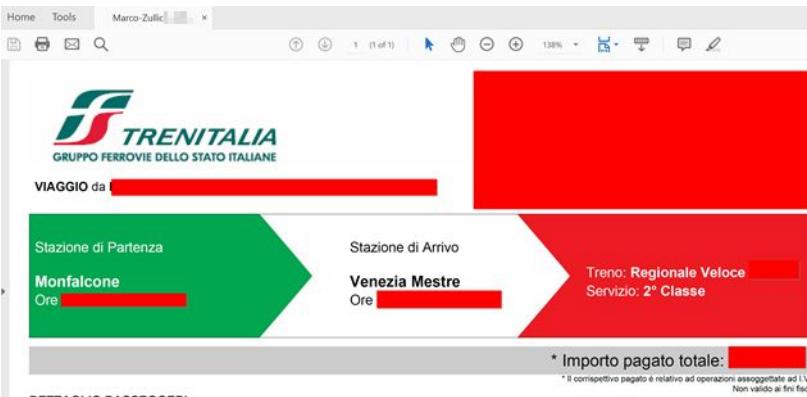
SALVATAGGIO DI UN FT

3



Verifico che il download del file sia completato

Il file viene aperto automaticamente nel lettore di PDF di sistema



Dal menu dei download posso comunque aprire la cartella di salvataggio del file



E vedo che il file in realtà si trova nella cartella dei FT

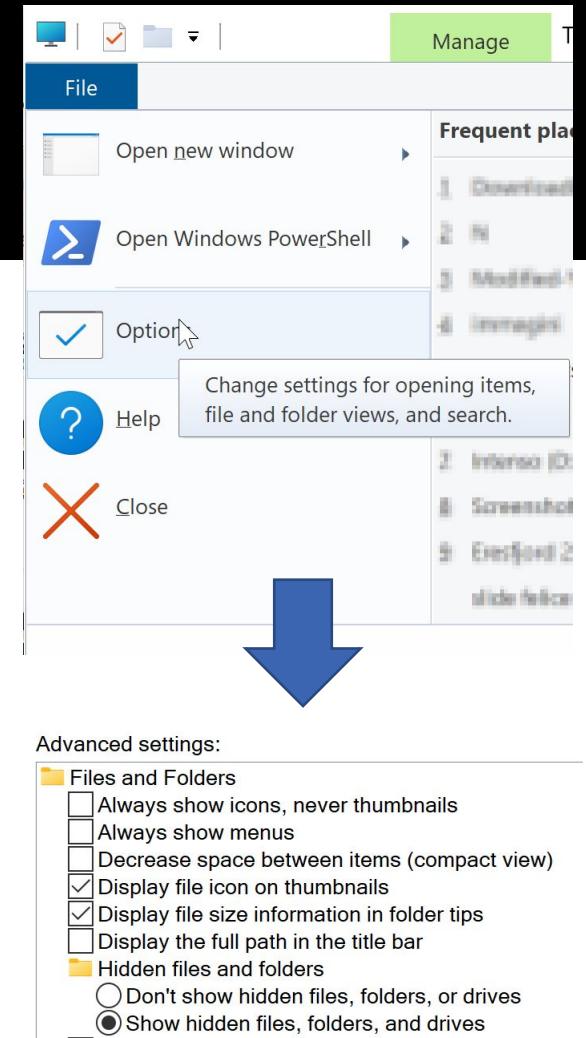
Name	Date modified	Type	Size
Marco-Zullich-[REDACTED].pdf	8/25/2021 11:57 P...	Adobe Acrobat D...	772 KB
mat-debug-744.log	8/18/2021 9:33 PM	Text Document	0 KB
mat-debug-2160.log	8/18/2021 7:27 PM	Text Document	0 KB
mat-debug-2424.log	8/20/2021 2:42 AM	Text Document	0 KB
mat-debug-2824.log	8/19/2021 11:57 P...	Text Document	0 KB
mat-debug-4588.log	8/19/2021 2:12 PM	Text Document	0 KB

FILE NASCOSTI

l'OS può anche creare dei file **non visibili**

es: file necessari per il corretto funzionamento di determinati programmi, ma un utente regolare del computer non ha interesse a vederli

I filesystem danno la possibilità di creare file e cartelle nascosti che si rendono visibili solo dopo aver selezionato una determinata opzione nel proprio OS



ALTRI TIPI FILE

Esistono numerosissime tipologie di file diverse dal binario e dal plaintext ,ne vedremo ancora due:

Immagini

Archivi

Ci sono ancora un numero enorme di formati che non andremo a vedere: PDF, documenti Word, fogli Excel...

IMMAGINI [DIGITALI]

per rappresentare un'immagine è necessario innanzitutto **discretizzarla**, per poterla caricare su disco o RAM

La discretizzazione consiste in una **suddivisione** dello spazio bidimensionale

Ogni quadratino viene rappresentato in memoria da un numero limitato (e possibilmente piccolo) di **bit**

Il numero di pixel sull'asse orizzontale e verticale definisce la **risoluzione** dell'immagine

Esempio: $3840 \times 2160 = 8.294.400$ (circa 8 megapixel)



SCALA DI GRIGIO

Immagine bianco e nero (scala grigio)

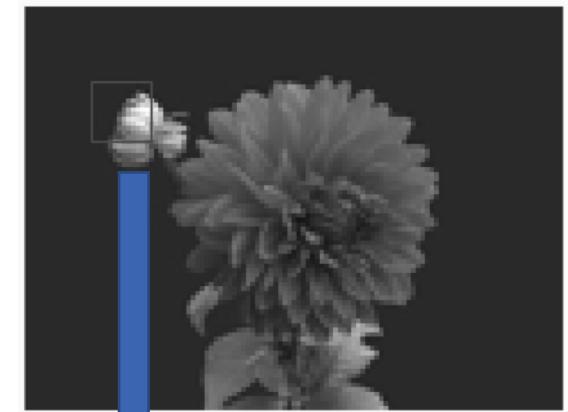
Associamo un numero da 0 a 255 ad ogni pixel.

Il numero rappresenta l'intensità del grigio del pixel

L'immagine viene quindi rappresentata da una **matrice** di numeri naturali con valore massimo 255

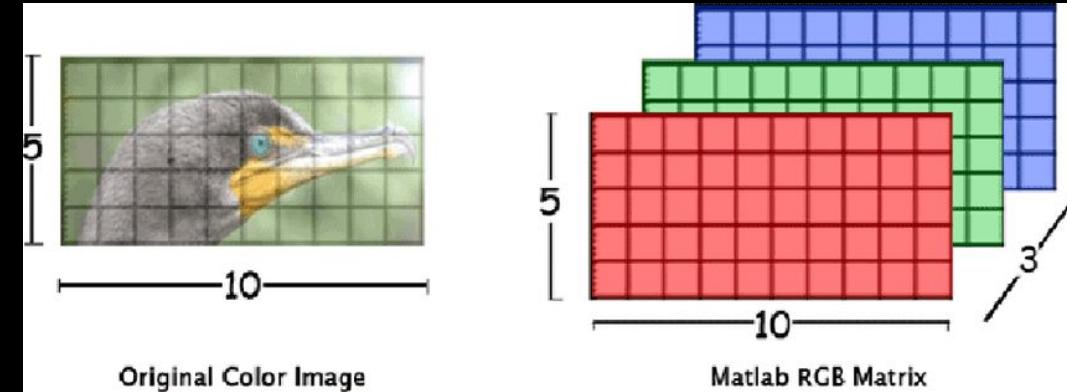
Vi suonano nuovi
questi valori?

= 1 Byte di informazione



25	25	25	25	25	25	25	25	25	25	25	26	27	25	24
25	25	25	25	25	25	25	25	25	25	26	24	22	23	25
25	25	25	25	25	25	25	25	25	24	24	24	25	26	25
25	25	25	25	25	25	25	25	25	24	22	20	39	111	190
25	25	25	25	25	26	24	31	132	195	178	189	227	182	28
25	25	25	25	25	23	27	114	172	169	211	230	217	232	
25	25	25	25	25	21	55	173	212	234	247	251	255	240	
25	25	25	24	25	26	111	193	203	218	229	240	248	228	
25	24	24	23	24	42	178	220	231	242	240	242	224	219	
25	24	24	26	23	48	156	192	209	221	244	247	219	186	
25	25	26	24	22	89	172	161	165	179	194	217	228	182	
25	25	25	26	22	101	142	149	142	155	163	168	186	171	
25	25	24	23	68	138	136	139	156	154	153	143	134	131	
25	24	23	34	136	194	175	153	141	150	162	146	109	80	

E I COLORI?



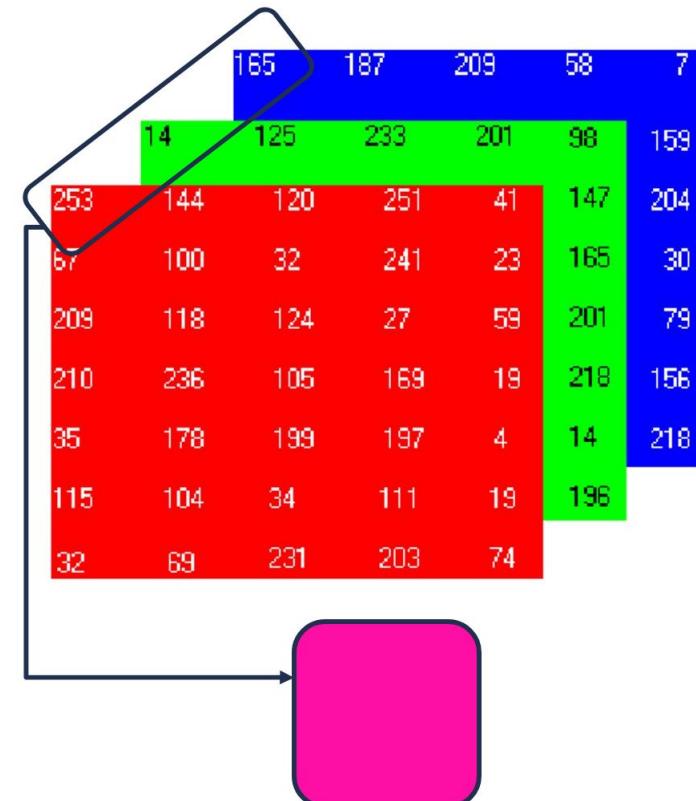
Non possiamo rappresentare un'immagine a colori come una matrice

Ordinando i colori in una retta il numero di byte necessari a rappresentare un'immagine crescerebbe di parecchio.

Si considerano i tre colori fondamentali **rosso**, il **verde** e il **blu**, e rappresentiamo l'intensità di ognuno di essi (codifica **RGB**)

Combinando questi tre colori secondo l'intensità indicata, riusciamo ad ottenere quasi tutti i colori dello spettro

L'immagine a colori viene quindi rappresentata come una collezione di tre matrici, una per ogni colore fondamentale



I FORMATI D'IMMAGINE

Rappresentazione molto pesante

Altri formati riescono a ridurre informazioni ridondanti (tramite *trasformata di Fourier o del Coseno*) e ad alleggerire la pesantezza del file

Strategia adottata dal formato **JPEG** - tra i più utilizzati per il salvataggio di un file immagine.

ARCHIVI

I file archivio nascono principalmente per:

Tenere all'interno di un singolo file un insieme di file mantenendo la struttura ad albero del filesystem di origine

Effettuare una compressione senza perdita del file/dei file contenuti all'interno dell'archivio

Da dove derivano queste necessità?

Inviare i file via email

Caricare i file all'interno di un sito

COMPRESsIONE

Possiamo pensare almeno a due metodi per provare a ridurre la dimensione di un file. Rimaniamo sempre sull'esempio dei file di testo.

Ammettiamo di aver scritto un libro di matematica in file markdown.

Il libro conterrà un numero molto elevato di parole ripetute, come *funzione, somma, derivata, limite...*

Posso decidere di rimpiazzare queste parole con dei simboli costruendo un dizionario che associa i termini ai simboli. In fase di decompressione, ri-sostituisco i simboli con le parole utilizzando il dizionario.

Ho un file di testo con codifica Unicode/UTF-8, ma, dei tanti caratteri che ho a disposizione, ne utilizzo pochissimi (lettere alfabeto, qualche punteggiatura, pochi numeri).

Posso quindi fare dei conti e notare che questi simboli sono solamente 63.

Di conseguenza, non ho bisogno nemmeno dei 7 bit ASCII per codificare il testo; necessito solo di un dizionario che mi dica a quale simbolo corrisponde una determinata sequenza di bit

Di quanti bit ho bisogno?

6: infatti $2^6=64$, rimarrà spazio per un carattere «inutilizzato»

I FORMATI DEGLI ARCHIVI

.zip

Il formato più utilizzato. Permette di comprimere file multipli (anche intere cartelle) con una compressione **senza perdite**.

In coda all'archivio viene aggiunto automaticamente un file di riepilogo che specifica la composizione (file e cartelle) dell'archivio.

Posso decomprimere l'archivio in maniera parziale (senza estrarre per forza tutti i file) oppure aggiungere altri file ad un archivio già esistente.

.rar

Sta cadendo un po' in disuso. A suo tempo (anni '00) era molto utilizzato per comprimere materiale piratato di grosse dimensioni. Più veloce di .zip. Non posso decomprimere in maniera parziale.

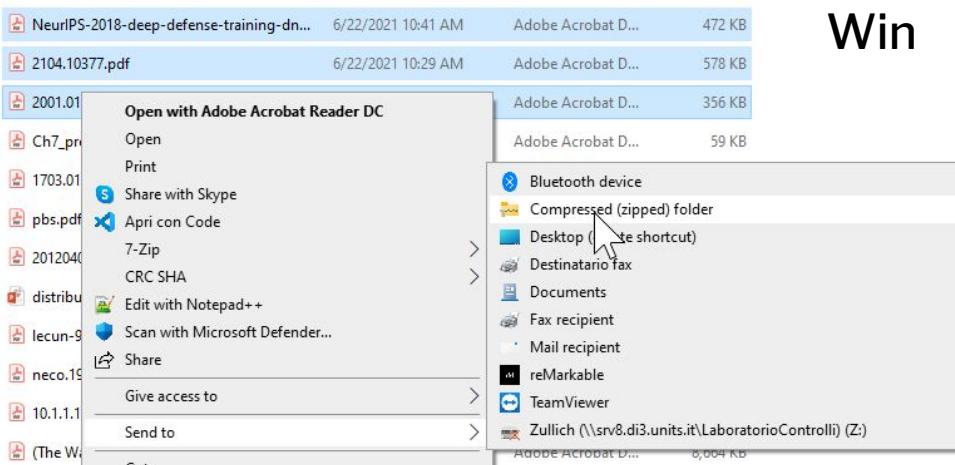
.7z

Open source. Comprime meglio di .rar, ma è leggermente più lento.

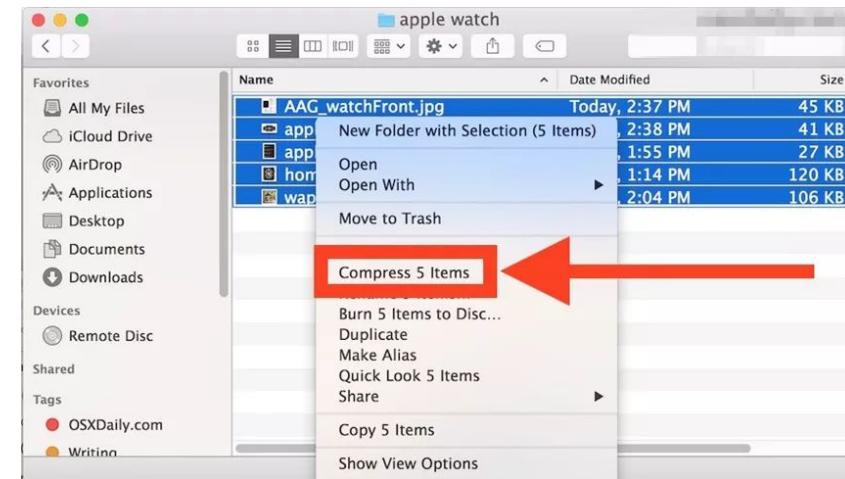
.tar + .gz

.tar è un formato archivi senza compressione (in pratica è come fosse una cartella). .gz può comprimere un file singolo. È più veloce di .7z, ma il risultato è più pesante.

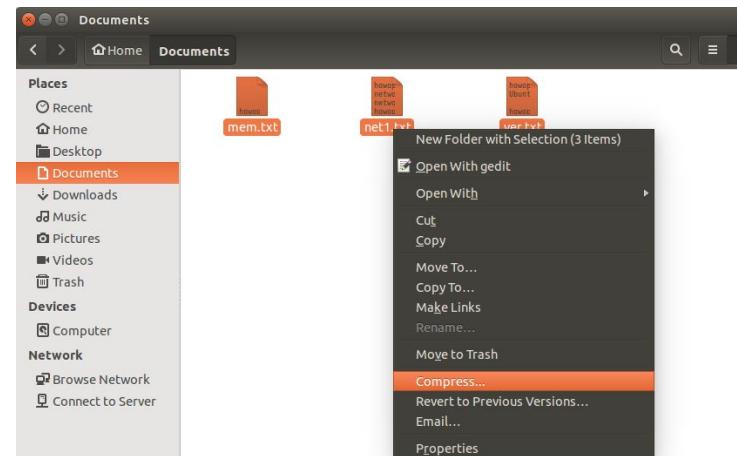
CREAZIONE ARCHIVI ZIP



Win



MacOS



Ubuntu-Gnome

GESTIONE DEGLI ARCHIVI TRAMITE 7-ZIP



7-Zip è un programma Windows per gestione archivi

Ne esistono versioni gratuite per Ubuntu (**p7zip GUI**) e Mac (**KEKA**)

7-Zip permette di aprire tutti i principali formati archivio

APERTURA ARCHIVIO E INTERFACCIA

Estrai file: decomprimi i file o cartelle selezionati e sposta in una cartella regolare del disco fisso mantenendo la stessa struttura ad albero dell'archivio

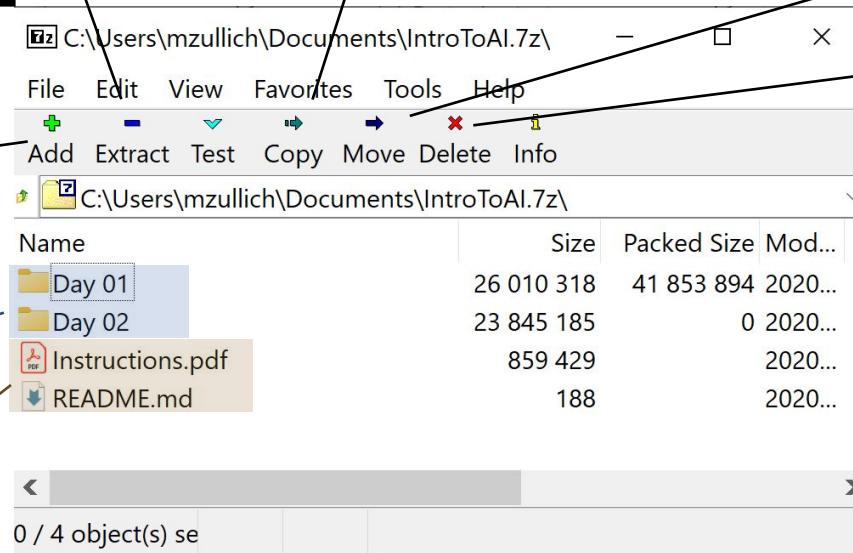
Crea una copia dei file su disco fisso (sostanzialmente uguale a estrai)

Muovi i file selezionati fuori dall'archivio nel disco fisso

Aggiungi file a archivio

Cartelle

File regolari



Rimuovi i file selezionati dall'archivio

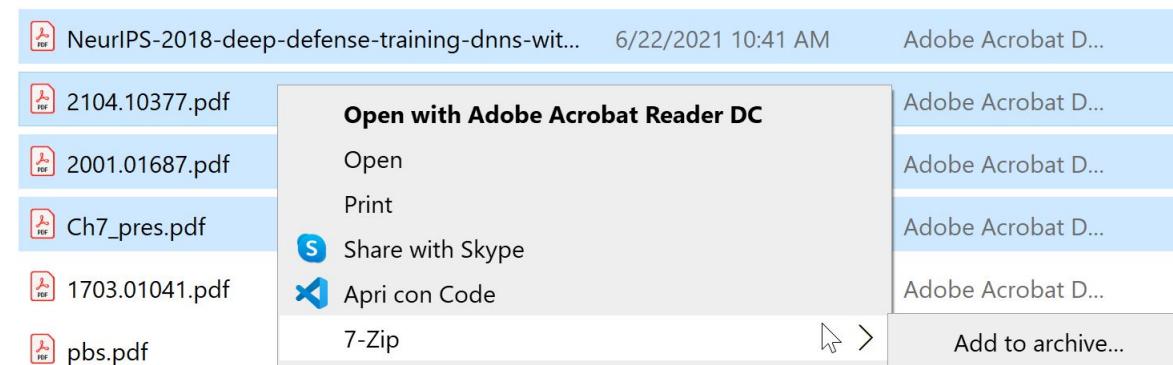
Facendo doppio clic su uno dei file indicati, quest'ultimo si apre nel suo lettore predefinito

CREAZIONE DI UN ARCHIVIO (I)

È semplice creare un archivio .7z usando 7-Zip.

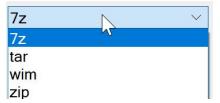
Ci posizioniamo su Explorer nella cartella contenente i file da comprimere. Possiamo fare selezioni multiple cliccando sui file desiderati mentre teniamo premuto il tasto Ctrl.

Clic dx → 7-zip → Add to archive...



CREAZIONE DI UN ARCHIVIO (II)

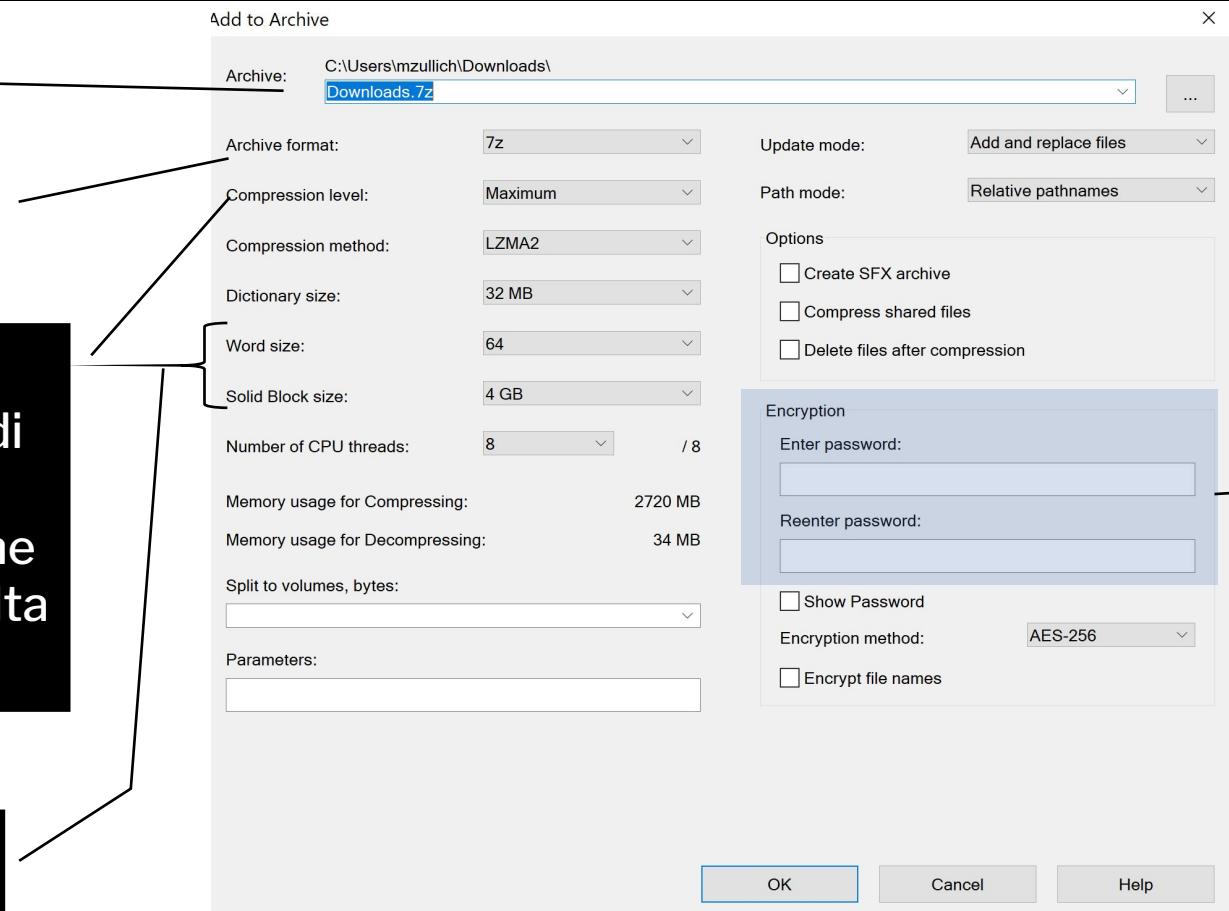
Nome archivio



Formato

Livello di compressione
Regola il compromesso fra velocità di [de]compressione e dimensioni archivio. Inoltre, livelli di compressione molto elevata possono richiedere molta memoria RAM

Parametri legati al dizionario



È inoltre possibile proteggere l'archivio con una password che verrà richiesta per accedere ad ognuno dei file in esso contenuti

RIASSUMENDO

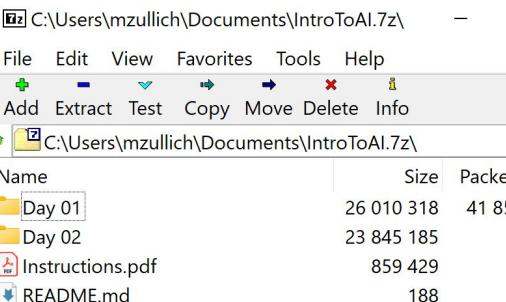
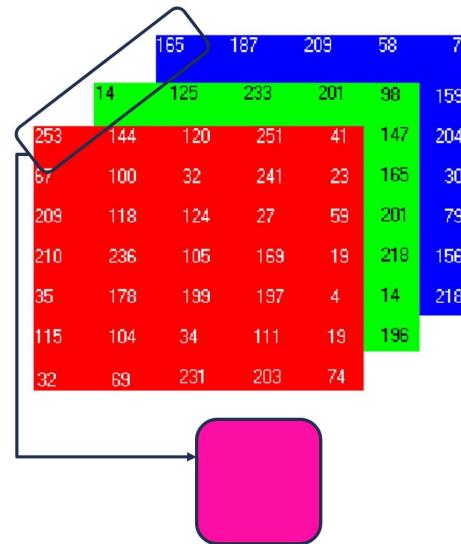
I **file temporanei**

- non necessitano di essere salvati sul disco fisso
- devono essere rimossi al riavvio dell'OS
- Vengono cartelle che l'OS ripulisce ciclicamente

Le **immagini** vengono rappresentate come griglie di pixel. Ad ogni pixel corrispondono valori (da 0 a 255) di **intensità dei colori rosso, verde, blu**.

Gli **archivi** contengono insiemi di file organizzati *ad albero* e compressi per risparmiare spazio su disco. La compressione avviene tramite l'utilizzo di **dizionari** per individuare le *parole* più frequenti.

Esistono vari formati di archivio, di cui forse **.zip** e **.7z** sono i più comuni (il secondo è più efficiente del primo), anche se su Linux si utilizza ancora molto **.tar + .gz**.



LA PROSSIMA VOLTA

Seguiremo il processo di installazione di Ubuntu su Windows tramite una macchina virtuale

Inizieremo poi ad orientarci in Ubuntu esplorandone velocemente le funzionalità

Infine, introdurremo l'interfaccia a linea di comando di Unix, necessaria per utilizzare Ubuntu a piene capacità