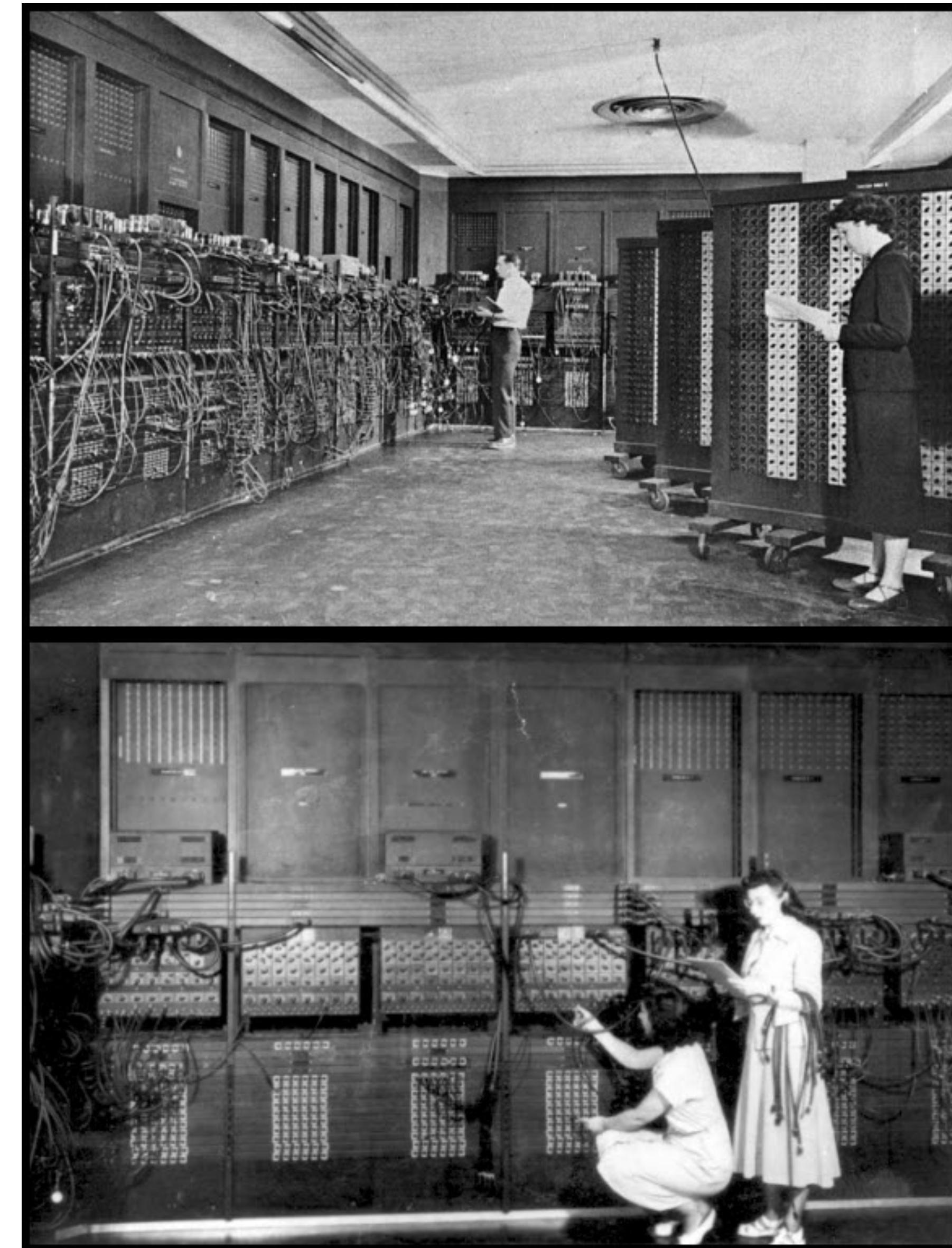
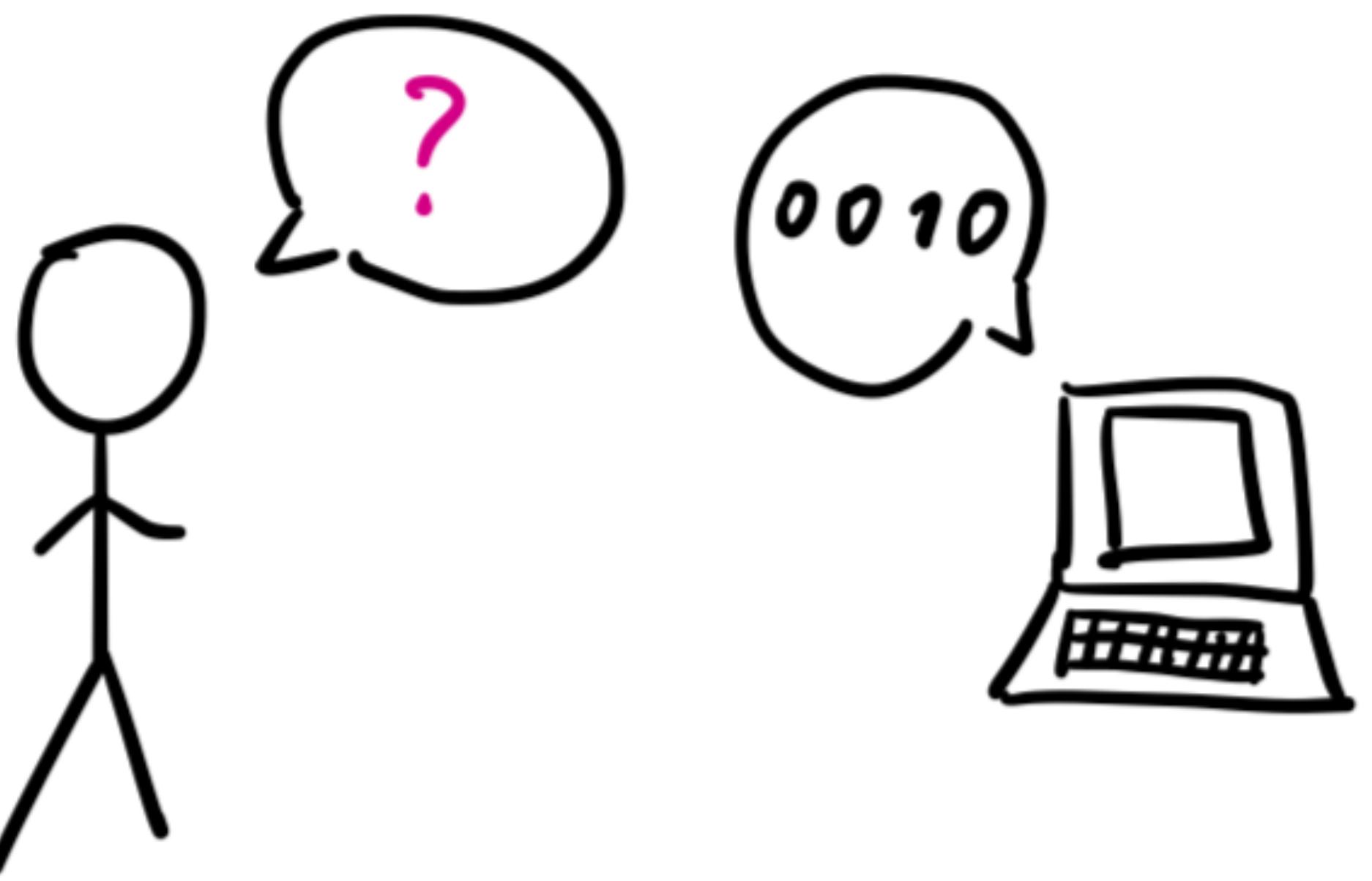
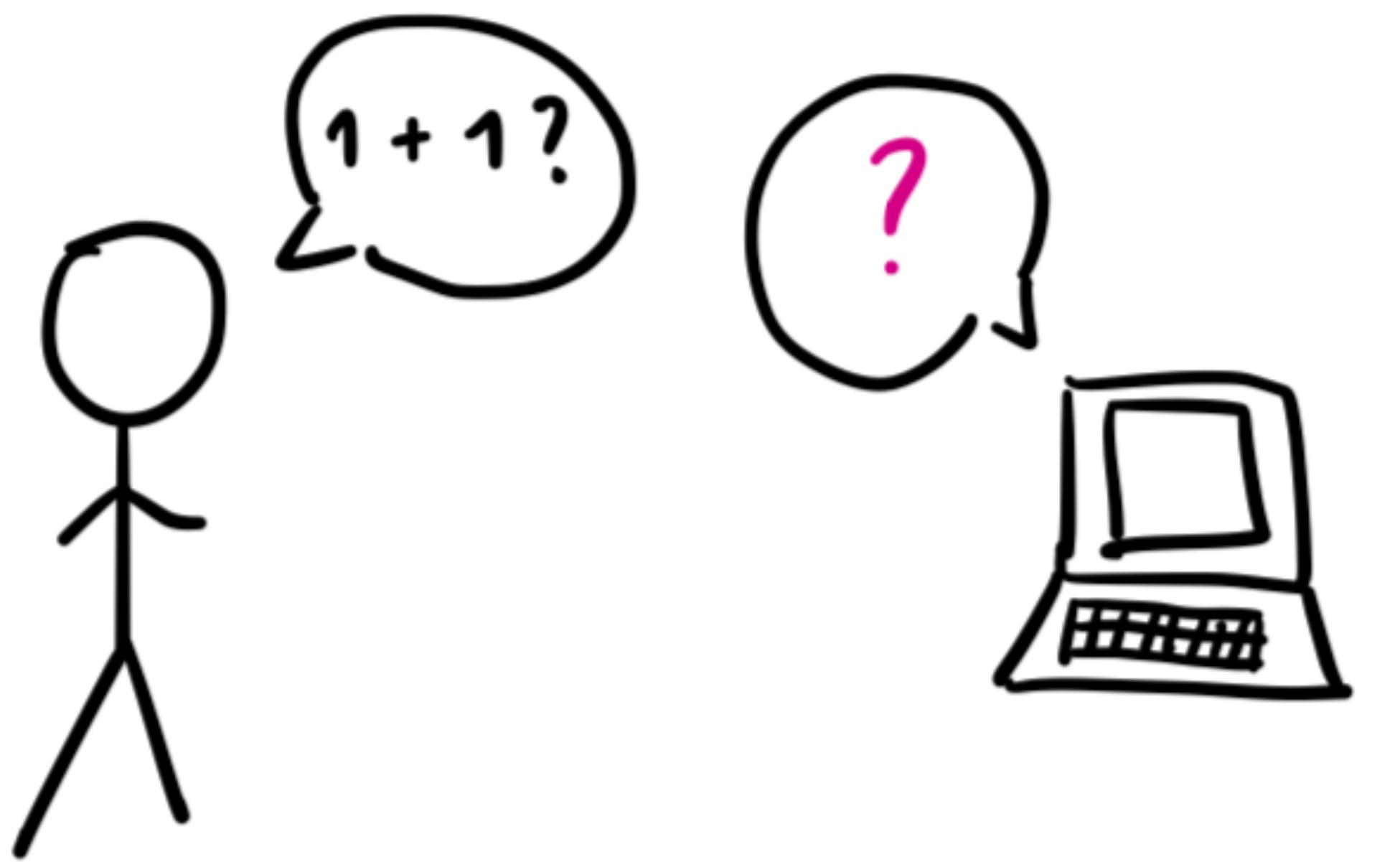


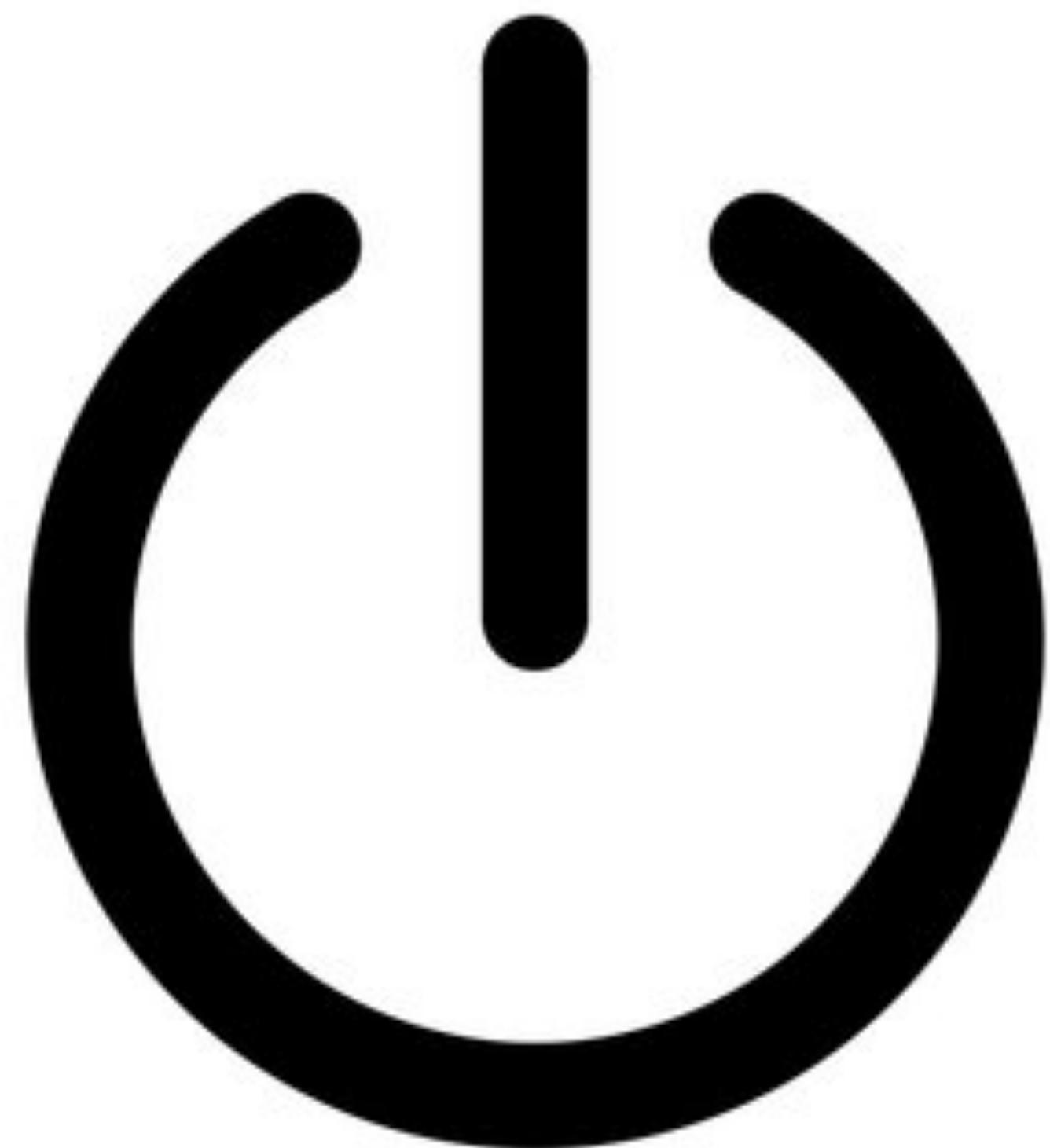
# Introduzione alla programmazione

E non solo



**Sopra:** il computer ENIAC (Electronic Numerical Integrator and Computer). **Sotto:** A scuola di programmazione? Fonte: <https://www.pingdom.com/blog/retro-delight-gallery-of-early-computers-1940s-1960s/>





# ALGORITMO

VS

# PROGRAMMA

Inizio

Inizializza la variabile 'risultato' a 1.

Leggi il numero di input 'n'

Ripeti i seguenti comandi fino a quando 'n'  
diventa Ø:

Moltiplica 'risultato' per 'n'

Decrementa 'n' di 1

Stampa il valore di 'risultato'

Stop

```
def fattoriale(n):  
    risultato = 1  
    while n > 0:  
        risultato = risultato * n  
        n = n - 1  
    print(risultato)
```



Loghi dei più noti linguaggi di programmazione. Fonte: <https://abrudz.github.io/logos/>

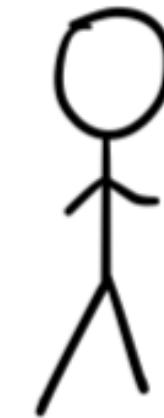
ESPRESSIVITÀ





LINGUAGGIO  
MACCHINA

LINGUAGGIO  
DI ALTO  
LIVELLO



TRADETTO IN

/ CON \

COMPIATORE      INTERPRETE



LINGUAGGIO

(di alto livello)

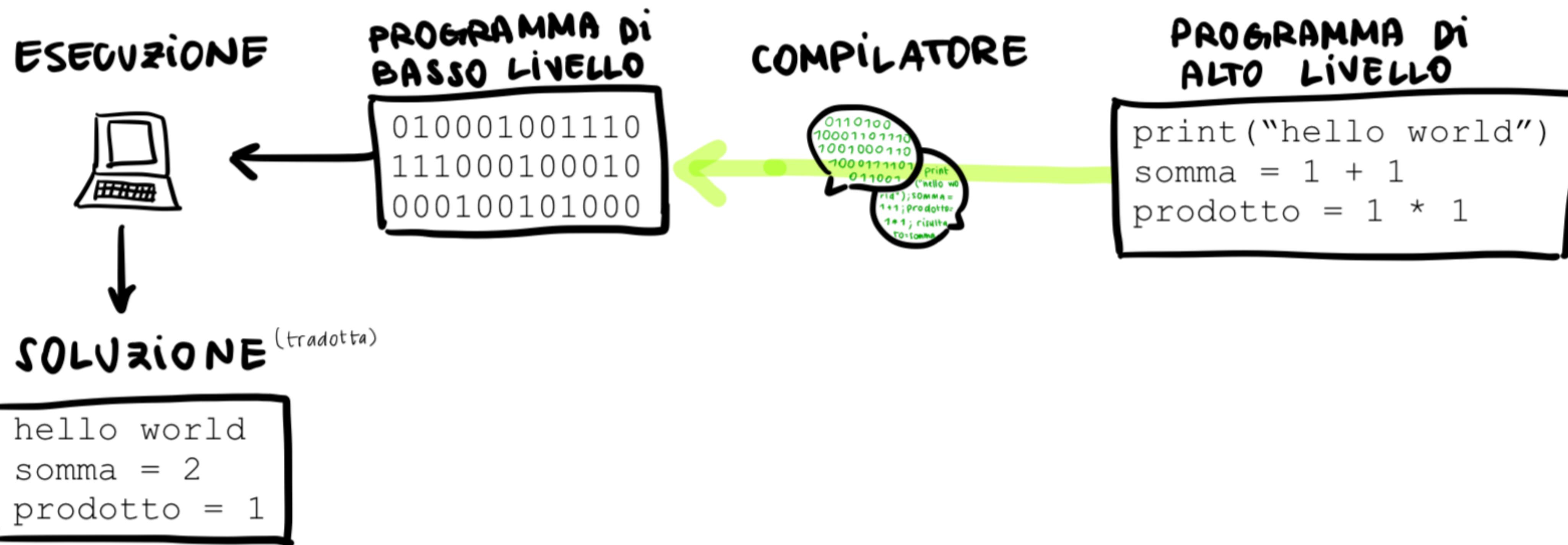
COMPILATI

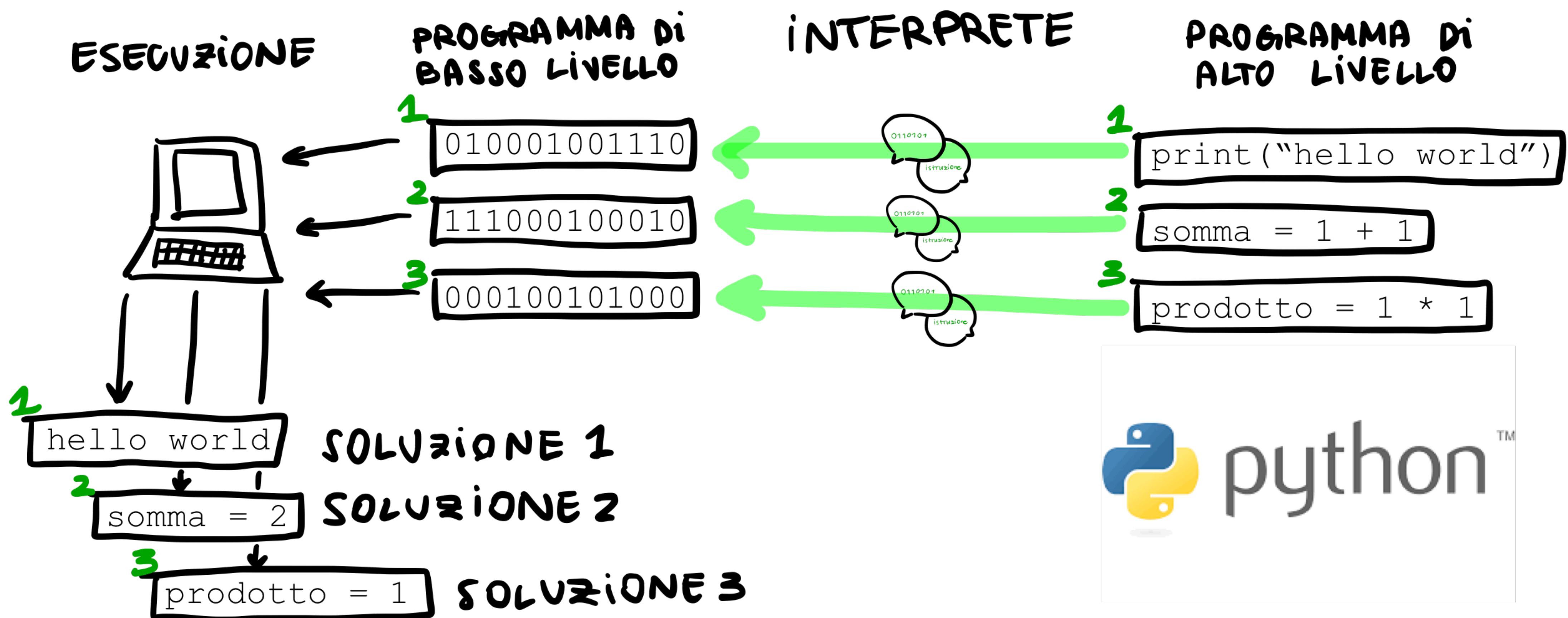


LINGUAGGIO

(di alto livello)

INTERPRETATO

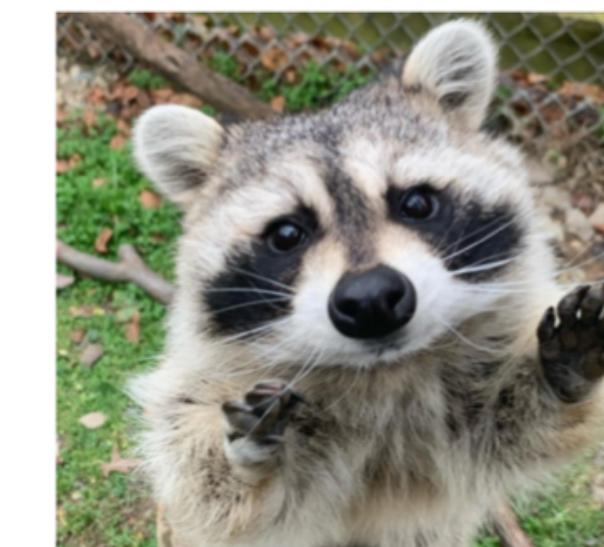
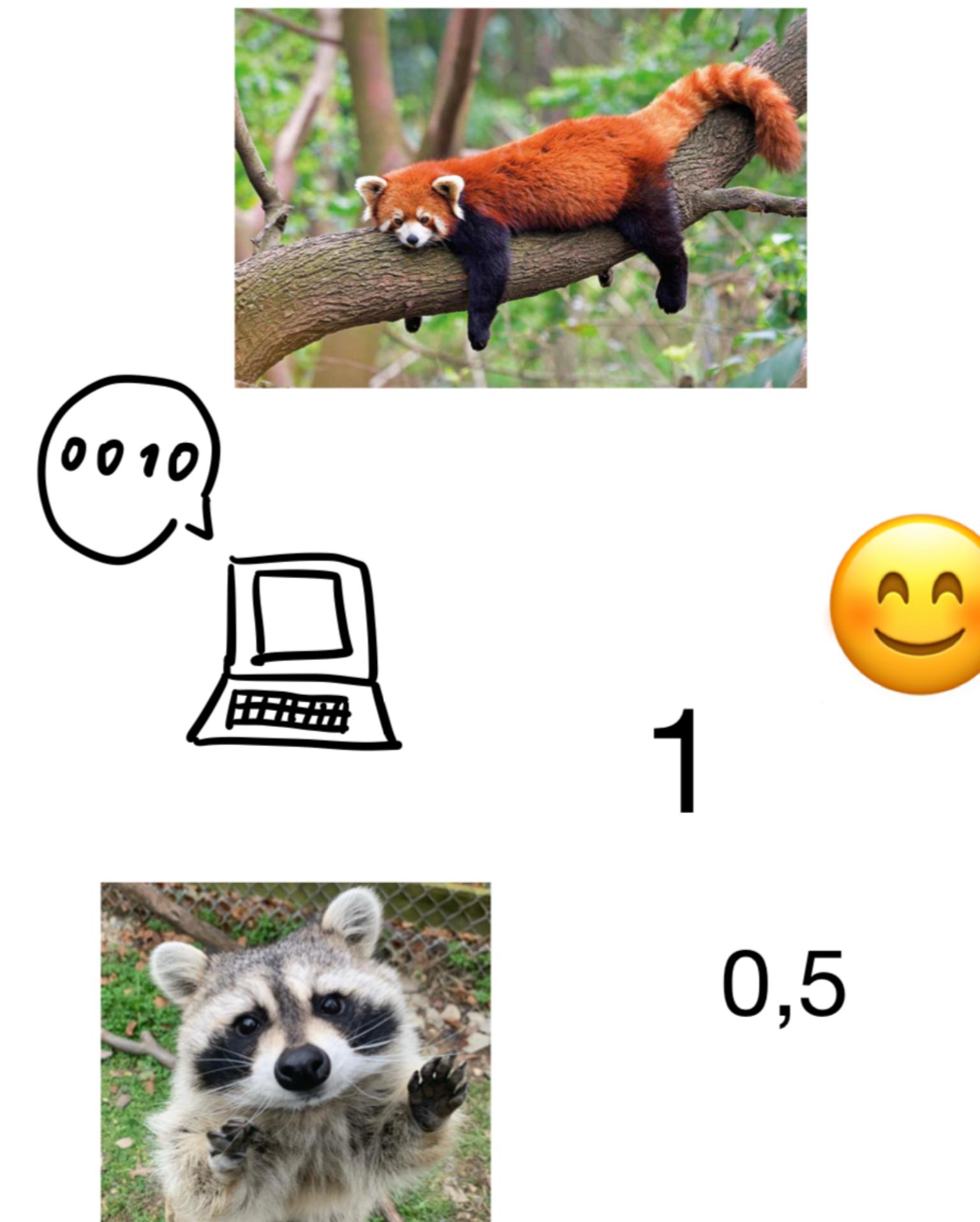




Hello  
world!



98

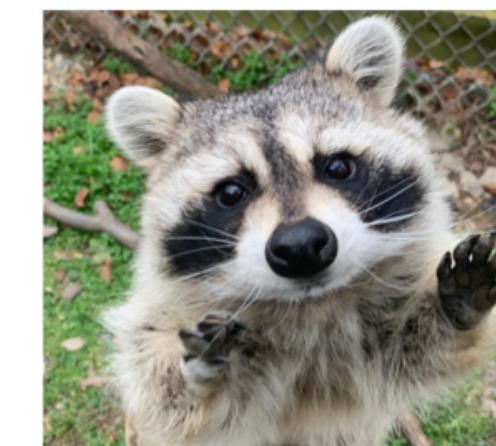


Hello  
world! → 01001000 01100101 01101100 01101100  
01101111 00100000 01110111 01101111  
01110010 01101100 01100100 00100001

1 → 00000000 00000000 00000000 00000001

0,5 → 00111111 00000000 00000000 00000000

-4 → 11111111 11111111 11111111 11111100



→ 11111111 11011000 11111111 11100000  
00000000 00010000 01001010 01000110  
01001001 01000110 00000000 00000001  
00000001 00000001 00000000 01100000  
00000000 01100000 00000000 00000000  
11111111 11100001 01010010 11010000  
01000101 01111000 01101001 01100110 ...



TRA D U Z I O N E  
i N C O D I C E

128525

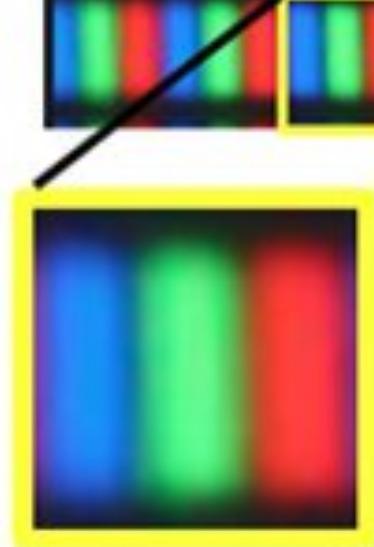
TRA D U Z I O N E  
i N B I N A R I O

00110001 00110010  
00111000 00110101  
00110010 00110101

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	:	;	<	=	>	?
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
@	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	[	\	]	^	-
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
'	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>	<b>j</b>	<b>k</b>	<b>l</b>	<b>m</b>	<b>n</b>	<b>o</b>
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
<b>p</b>	<b>q</b>	<b>r</b>	<b>s</b>	<b>t</b>	<b>u</b>	<b>v</b>	<b>w</b>	<b>x</b>	<b>y</b>	<b>z</b>	{		}	~	<b>DEL</b>

Tabella di codifica ASCII. Fonte: [link](#).

Close up of screen image

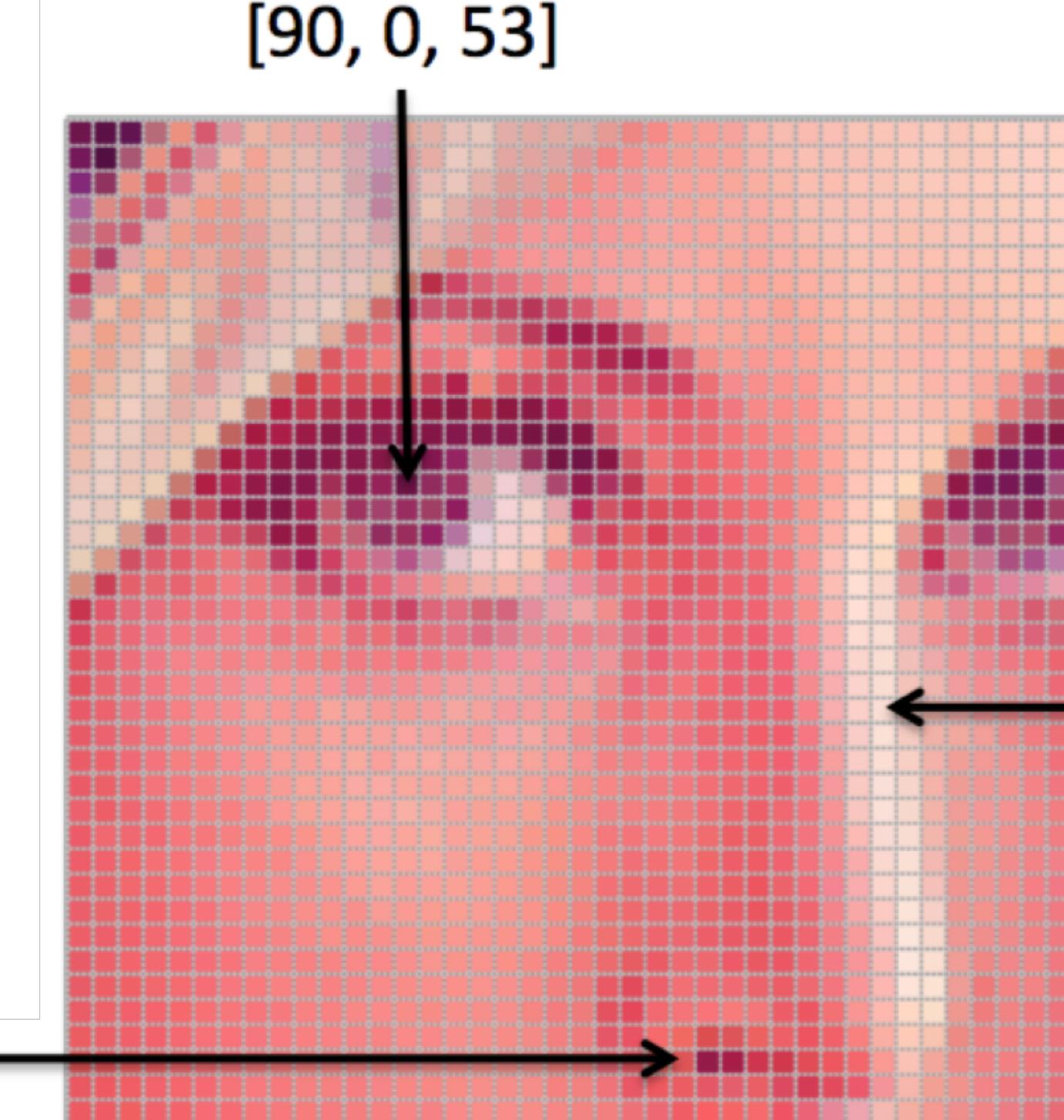


A pixel  
with one blue,  
one green and  
one red element

[90, 0, 53]

[249, 215, 203]

[213, 60, 67]



Alcuni valori in RGB di un'immagine a schermo. Fonte: link.

Ingrandimento di un'immagine a schermo. Fonte: link.

BASE 10 :  $1246_{10}$

$$\begin{array}{cccc} 1 & 2 & 4 & 6 \\ * & * & * & * \\ 10^3 & 10^2 & 10^1 & 10^0 \\ \parallel & \parallel & \parallel & \parallel \\ 1000 + 200 + 40 + 6 = 1246_{10} \end{array}$$

BASE 2 :  $1011_2$

$$\begin{array}{cccc} 1 & 0 & 1 & 1 \\ * & * & * & * \\ 2^3 & 2^2 & 2^1 & 2^0 \\ \parallel & \parallel & \parallel & \parallel \\ 8 + 0 + 2 + 1 = 11_{10} \end{array}$$

$$\begin{array}{r|l} 11_0 : 2 = & \text{Resto} \\ 5 : 2 = & 1 \\ 2 : 2 = & 1 \\ 1 : 2 = & 0 \\ 0 & 1 \end{array}$$

$\Rightarrow 1011_2$

Hello  
world! →

01001000 01100101 01101100 01101100  
01101111 00100000 01110111 01101111  
01110010 01101100 01100100 00100001

1 → 00000000 00000000 00000000 00000001

0,5 → 00111111 00000000 00000000 00000000

-4 → 11111111 11111111 11111111 11111100



→ 11111111 11011000 11111111 11100000  
00000000 00010000 01001010 01000110  
01001001 01000110 00000000 00000001  
00000001 00000001 00000000 01100000  
00000000 01100000 00000000 00000000  
11111111 11100001 01010010 11010000  
01000101 01111000 01101001 01100110 ...

00000000 00000000 00000000 00000001  
BYTE BIT

## Tipi di dato

Ogni linguaggio di programmazione offre al programmatore dei tipi di dato predefiniti. Concentriamoci sui tipi predefiniti offerti da Python (vedremo solo ciò che ci serve).

- Tipi numerici
  - `int` : 0, 3, -5, 399, 37555, ...
  - `float` : 0.5, 4.6782, 2993.45, ...
- Tipo testuali
  - `str` : "Hello world!"
- Tipo booleano
  - `bool` : True ( $\rightarrow$  1), False ( $\rightarrow$  0)
- Tipo vuoto
  - `NoneType` : None
- Tipi sequenziali, insiemistici, chiave-valore: li affronteremo la prossima volta!
  - `list` : [1, 3, "pizza", None, 5.7, False]
  - `tuple` : (1, 3, "pizza", None, 5.7, False)
  - `set` : {1, 3, "pizza", None, 5.7, False}
  - `dict` : {1: "ciao", "due" : None}

## Operatori per tipi numerici

Operatore	Descrizione	Esempio
<code>+</code>	Addizione	<code>12 + 10 -&gt; 22</code>
<code>-</code>	Sottrazione	<code>12 - 10 -&gt; 12</code>
<code>*</code>	Moltiplicazione	<code>12 * 10 -&gt; 120</code>
<code>/</code>	Divisione	<code>12 / 10 -&gt; 1,2</code>
<code>//</code>	Divisione intera	<code>12 // 10 -&gt; 1</code>
<code>%</code>	Modulo (resto della divisione)	<code>12 % 10 -&gt; 2</code>
<code>==</code>	Uguaglianza	<code>1 == 1 -&gt; True</code>
<code>!=</code>	Disuguaglianza	<code>1 != 1 -&gt; False</code>

## Operatori per tipi testuali

Operatore	Descrizione	Esempio
+	Concatenazione	"hello" + "world" -> "helloworld"
*	Ripetizione	"hello" * 2 -> "hellohello"
==	Uguaglianza	"hello" == "hello" -> True
!=	Disuguaglianza	"hello" != "hello" -> False
in	Appartenenza	"hello" in "hello world" -> True
not in	Non appartenenza	"hello" not in "hello world" -> False

## Operatori booleani

Operatore	Descrizione	Esempio
and	True se entrambi gli operandi sono True	True and True -> True True and False -> False
or	True se almeno uno degli operandi è True	True and False -> True False and False -> False
not	True se l'operando è False False se l'operando è True	not False -> True not True -> False

## Windows

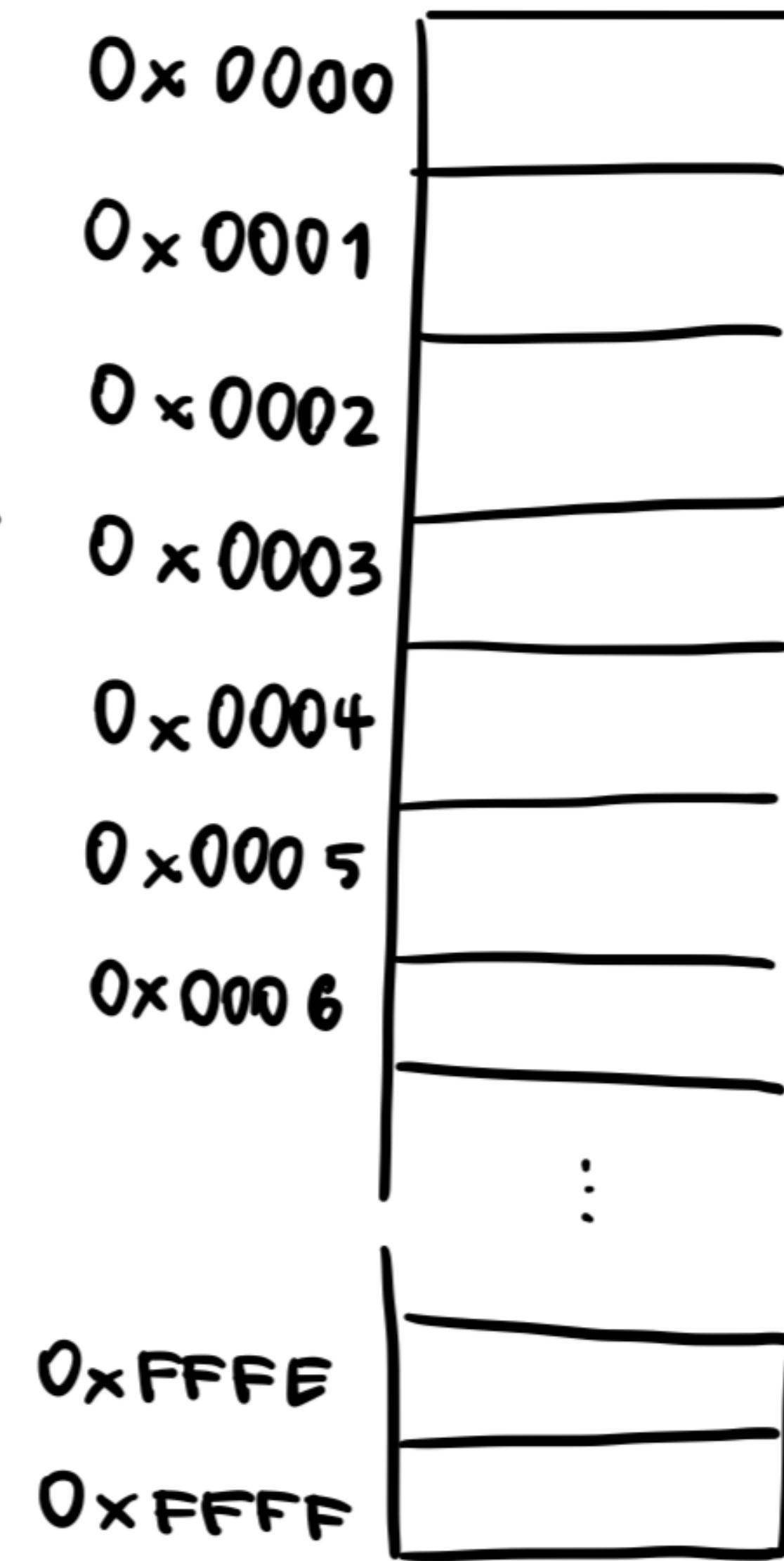
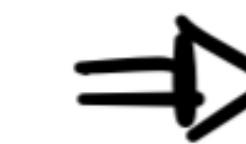
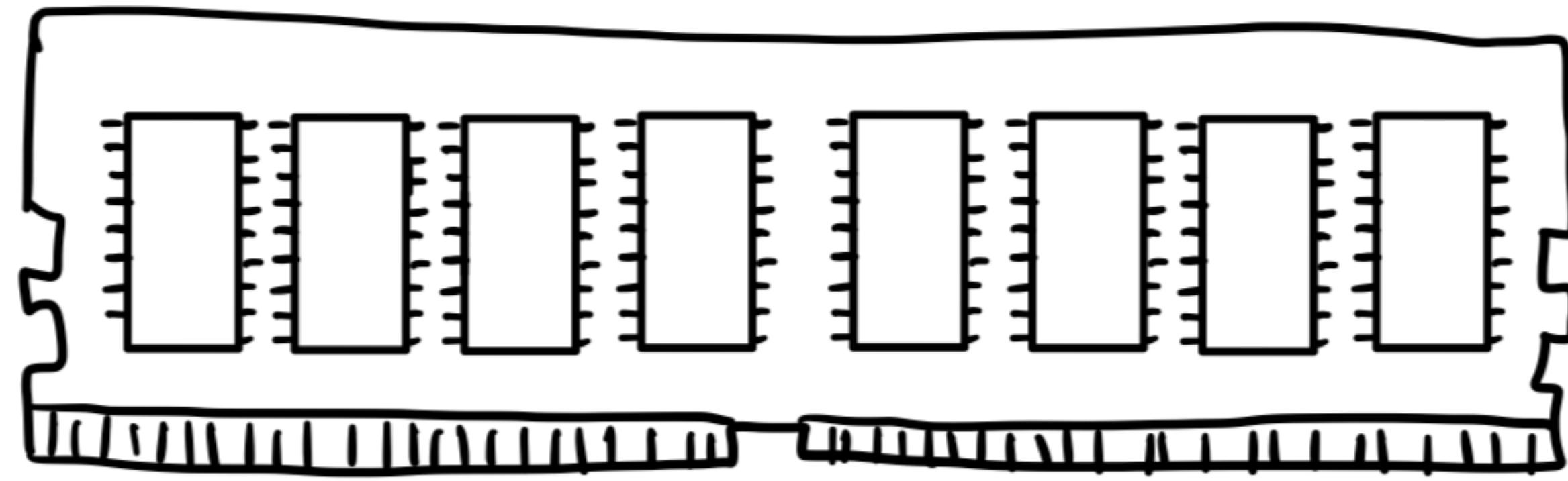
Clicca sul tasto **Start** (l'icona a forma di bandierina collocata nell'angolo inferiore sinistro dello schermo o al centro nel caso di Windows 11) e segui una delle procedure seguenti, variabili in base alla versione di Windows in tuo possesso.

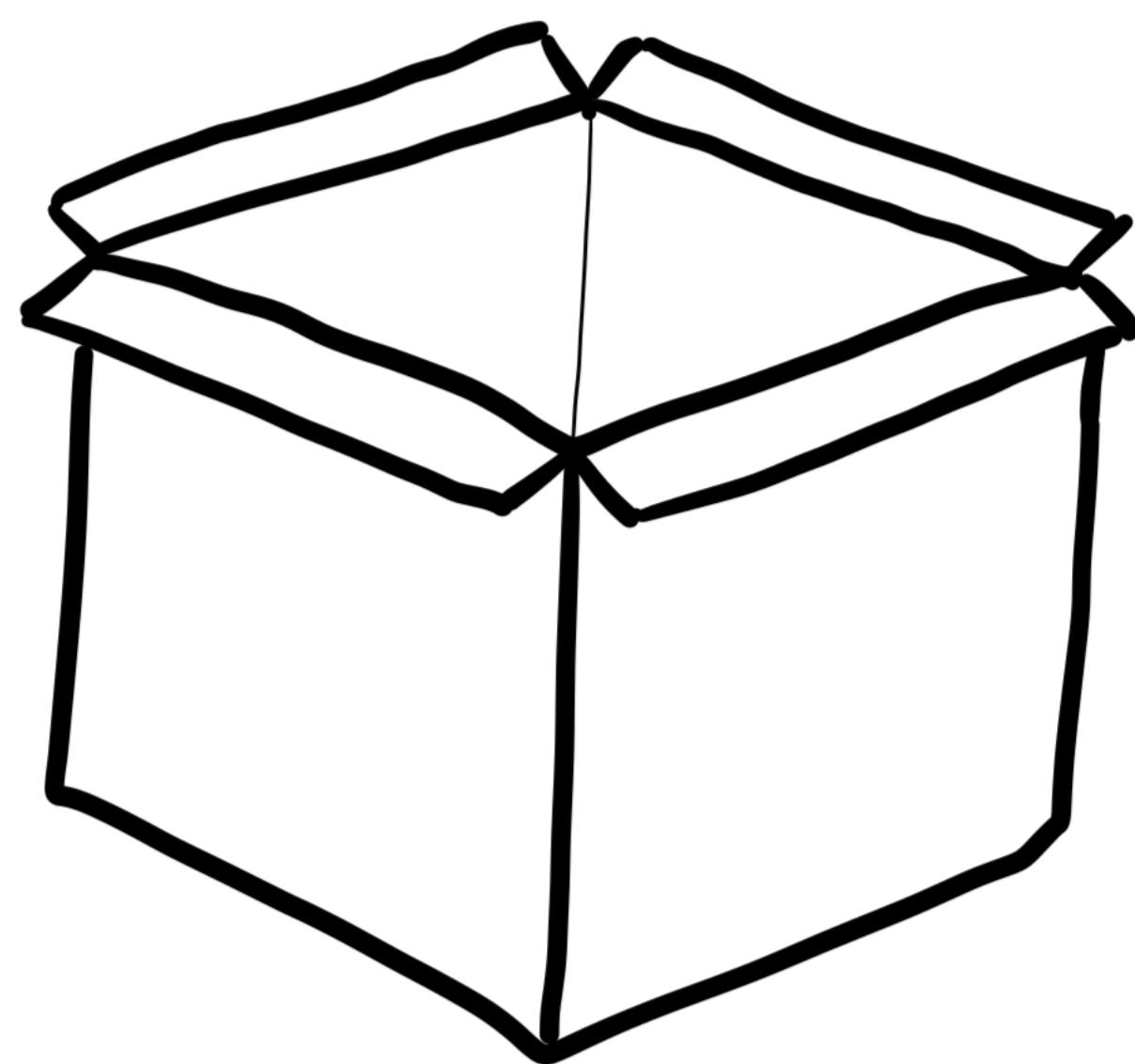
- **Windows 11:** premi sulla **barra di ricerca** in alto, digita “prompt” e seleziona il riquadro **Prompt dei comandi**. In alternativa, puoi seguire il percorso menu **Start > Tutte le app > Strumenti di Windows > Prompt dei comandi**.
- **Windows 10:** scorri in basso la sezione contenente le cartelle dei programmi (cioè la parte sinistra del menu) fino a trovare la cartella **Sistema Windows**, fai clic su quest'ultima e ripeti l'operazione sulla voce **Prompt dei comandi**.
- **Windows 8.1:** una volta entrato nella **Start Screen**, muovi il mouse e clicca sul pulsante (**↓**) che compare in basso a sinistra per espandere la sezione **Tutte le app**; infine, clicca sull'icona **Prompt dei comandi** collocata nel riquadro **Sistema Windows**.
- **Windows 7:** clicca sulla voce **Tutti i programmi**, poi sulla cartella **Accessori** e, per concludere, seleziona la voce **Prompt dei comandi**.

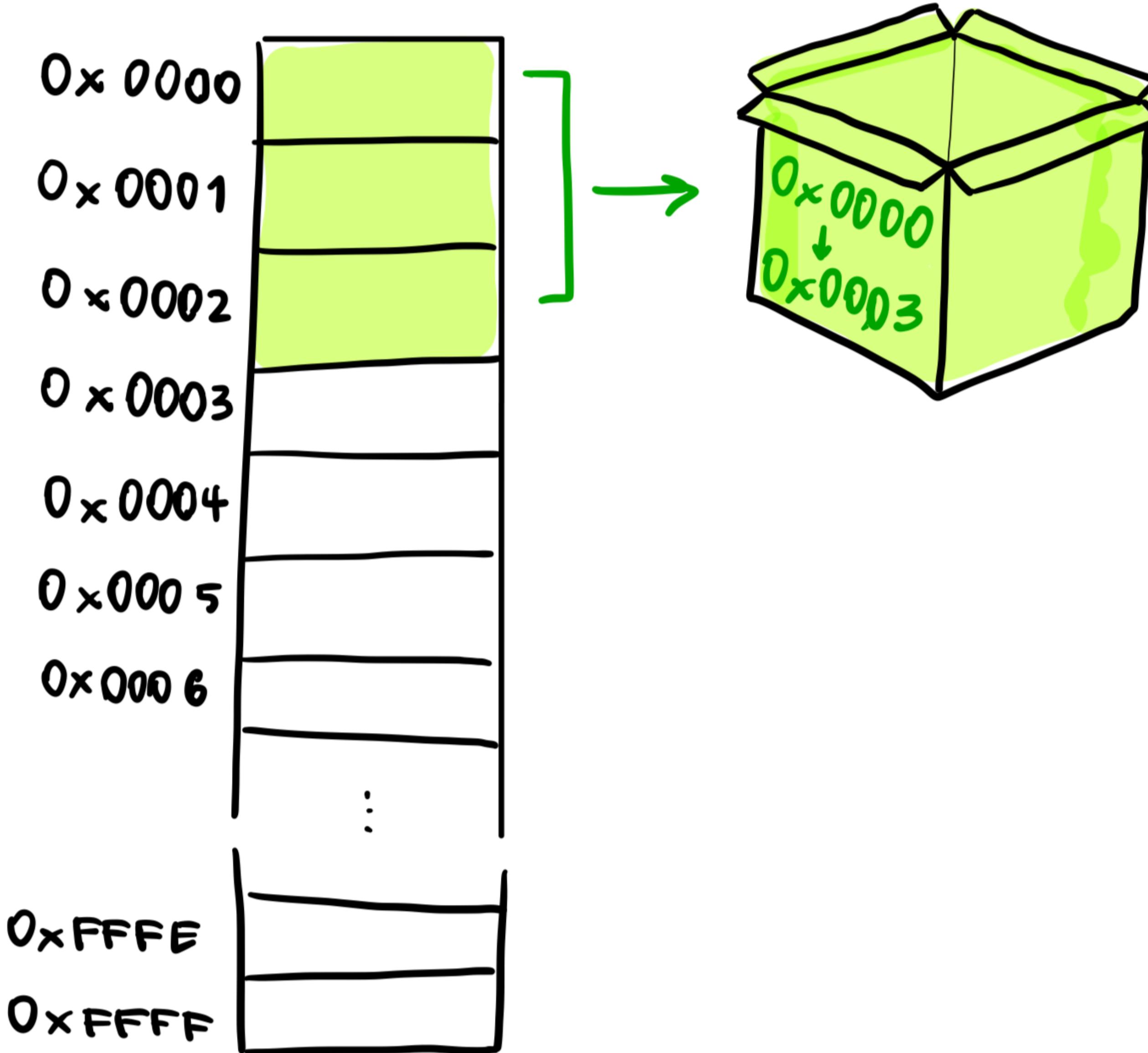
## macOS

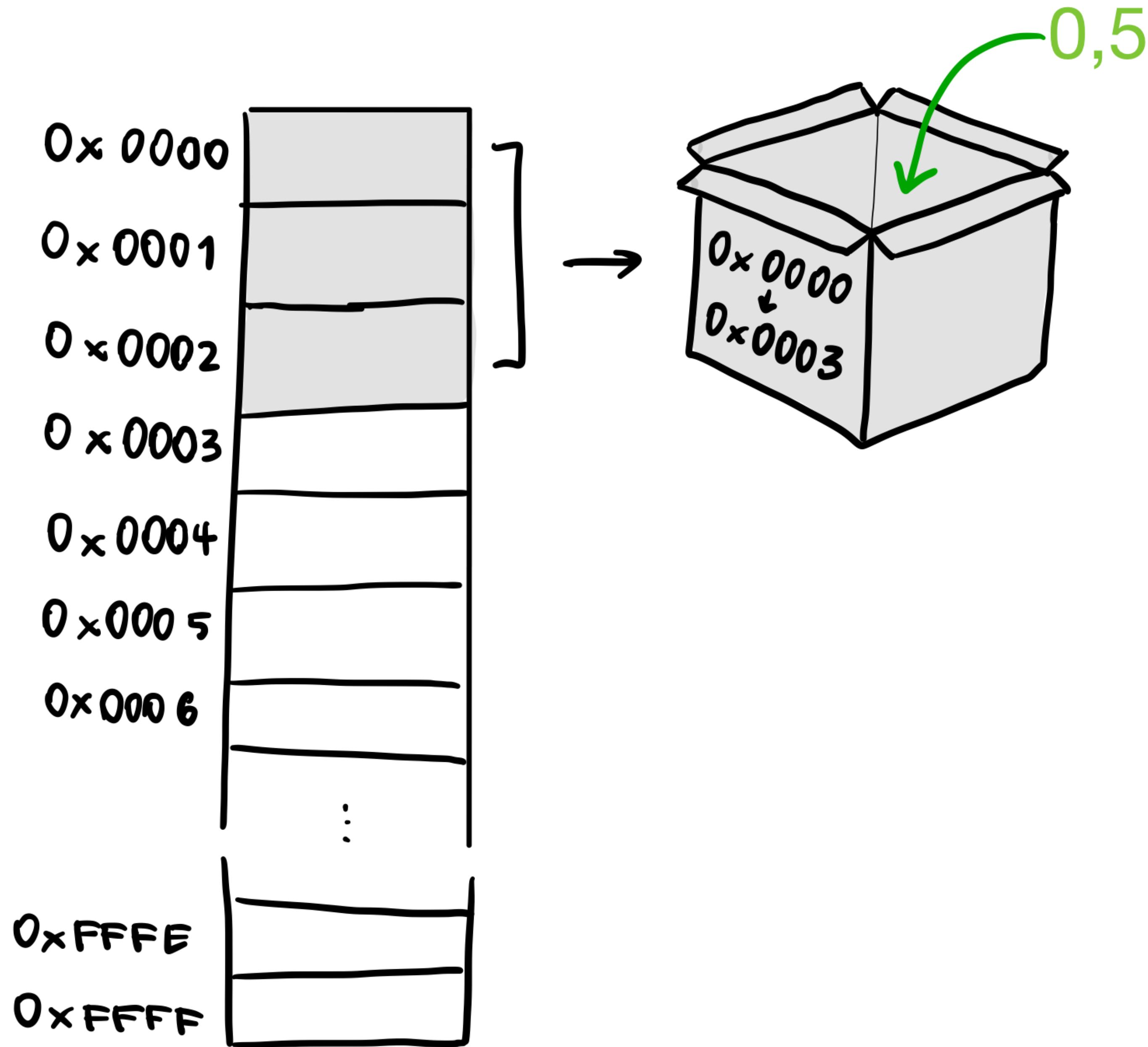
Apri una finestra **Finder**, recati nel menu **Vai > Utility** legato al **Finder** e fare clic sull'icona del **Terminale**.

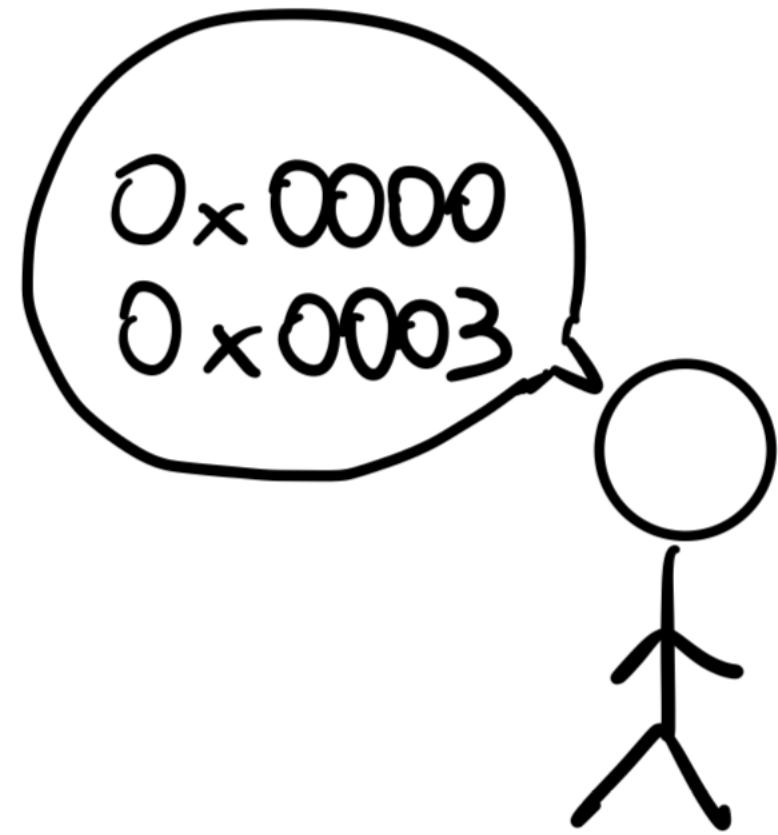
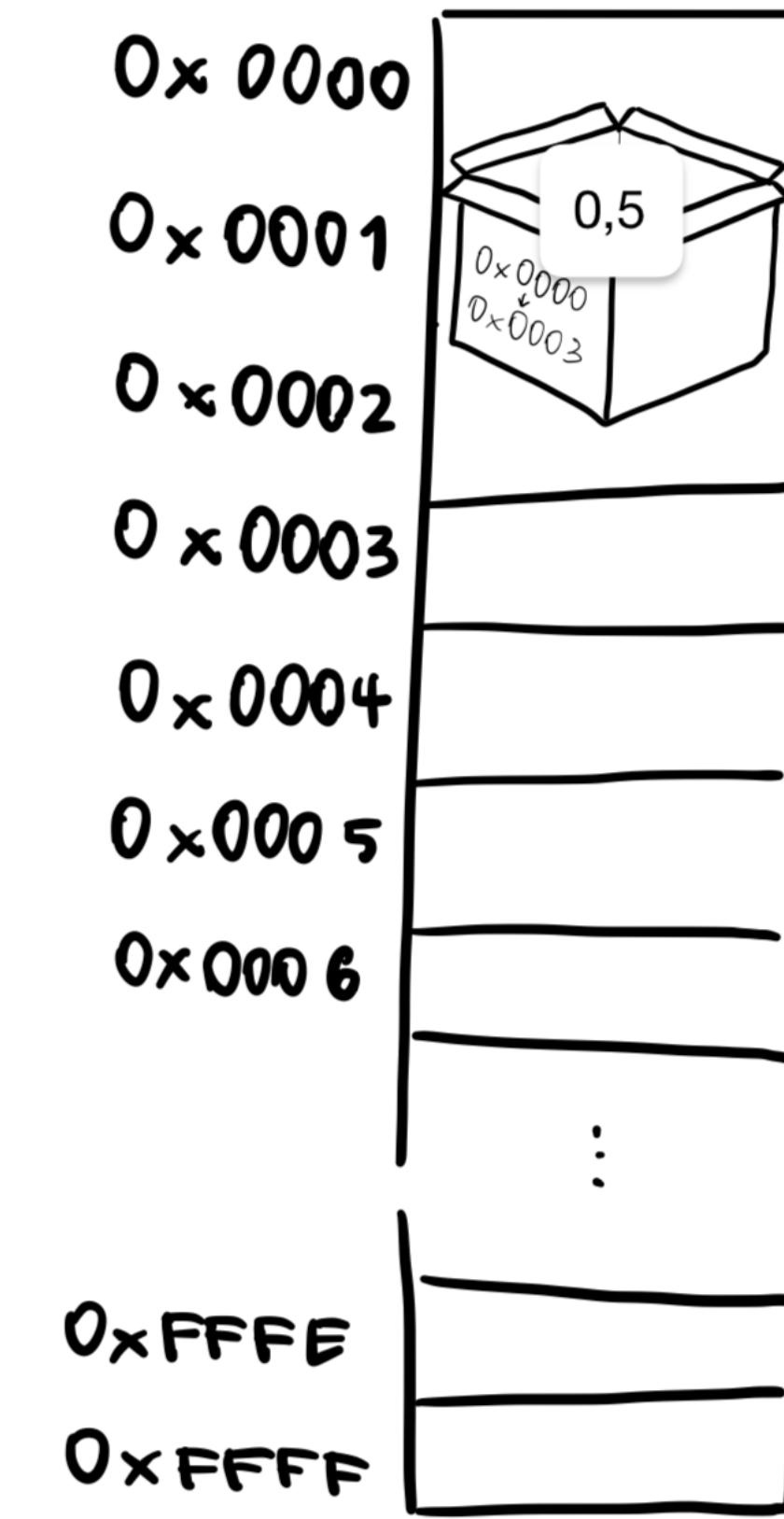
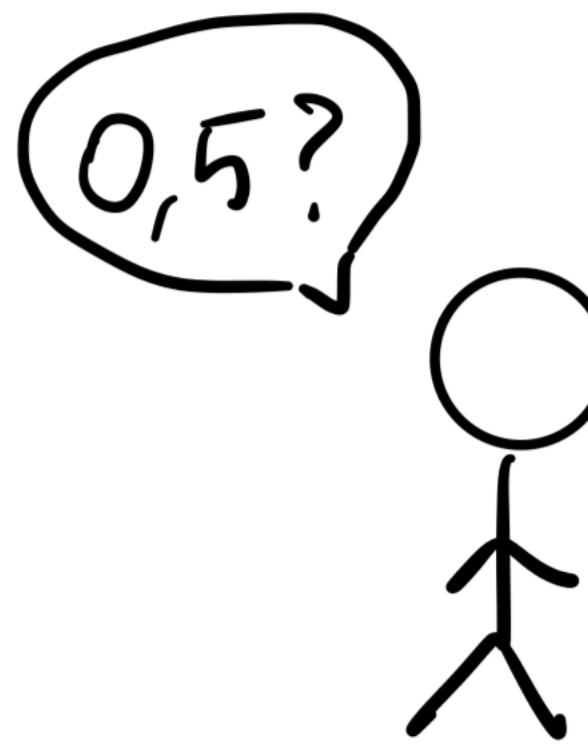
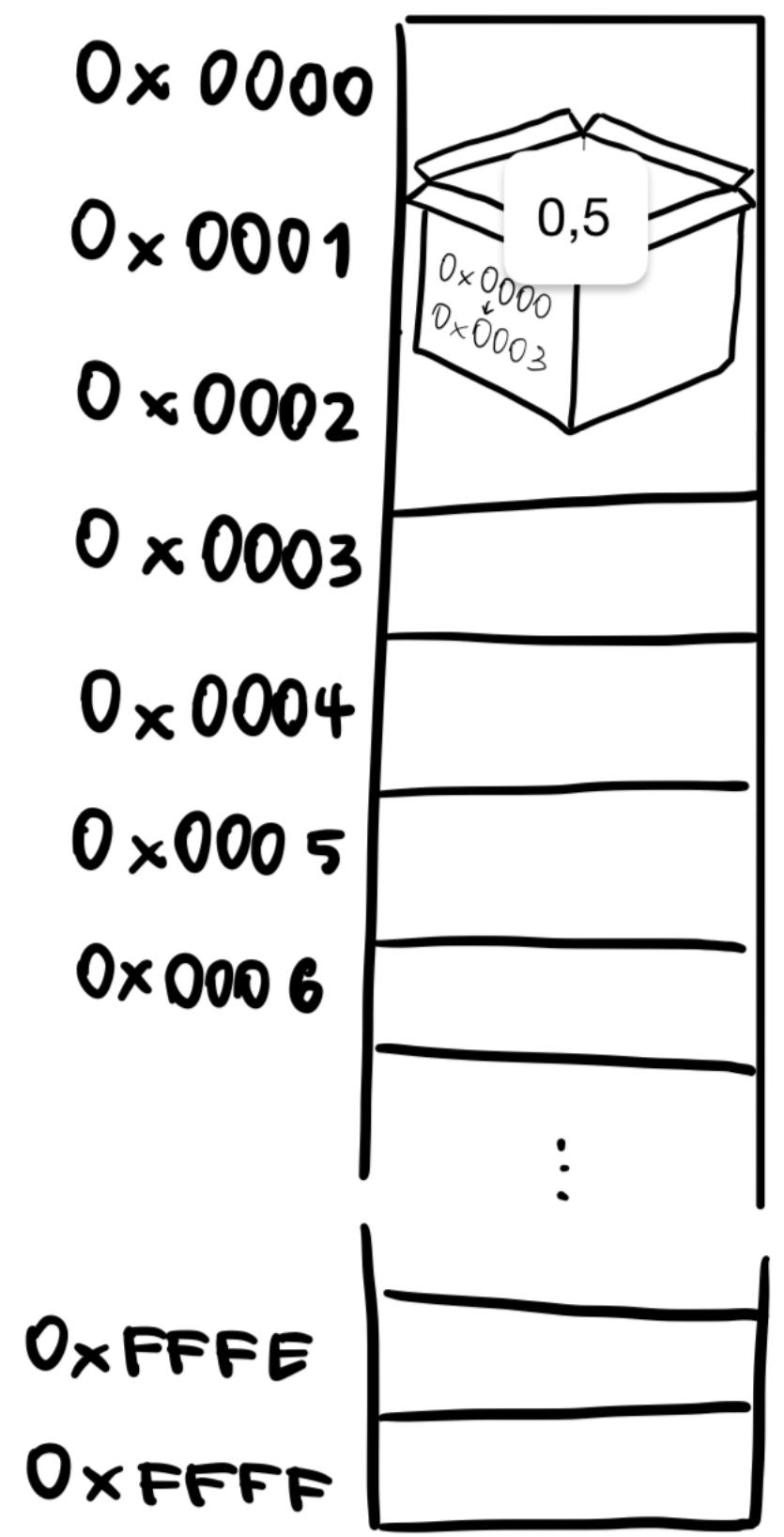
Fonte: [Salvatore Aranzulla](#)

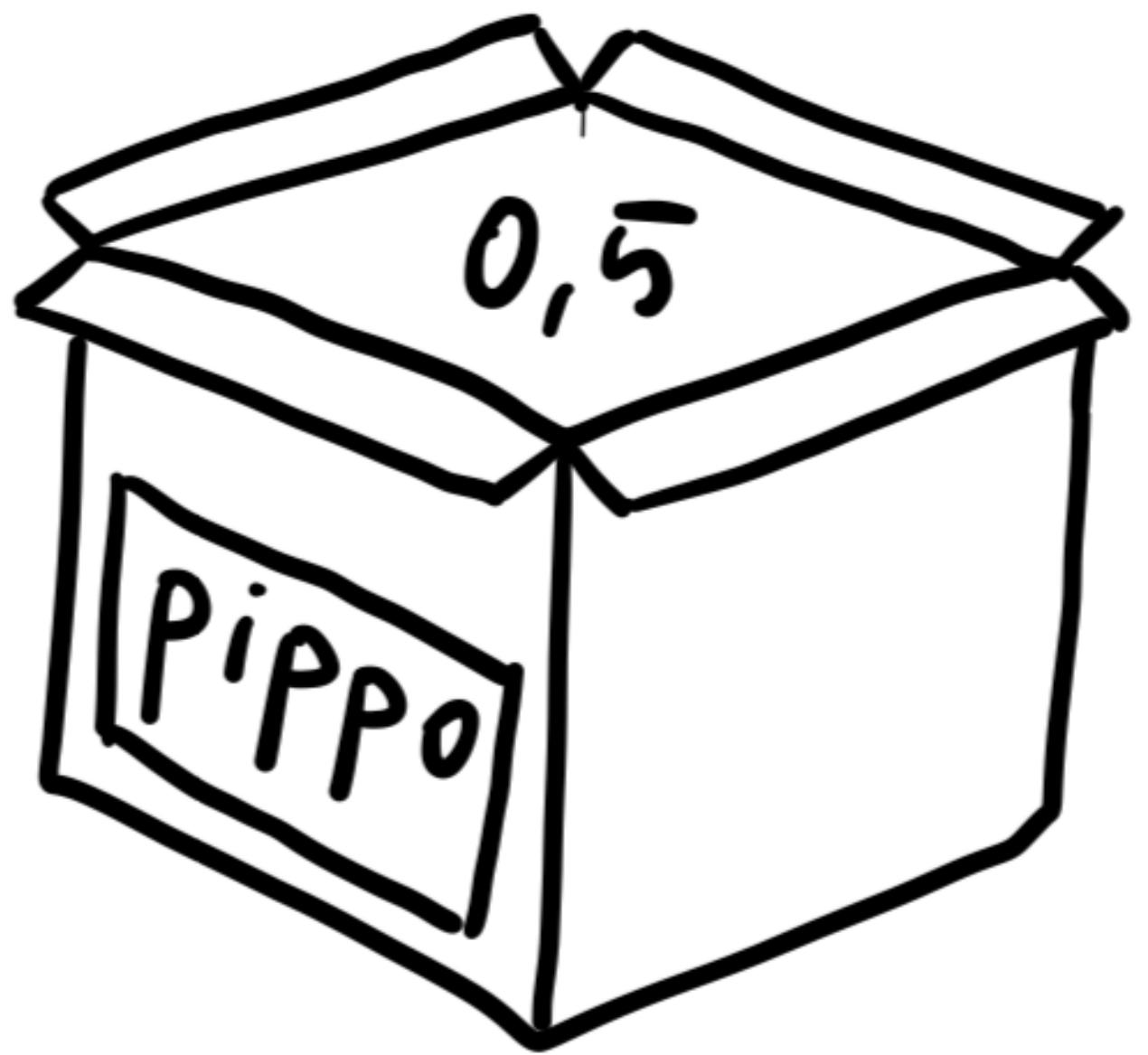












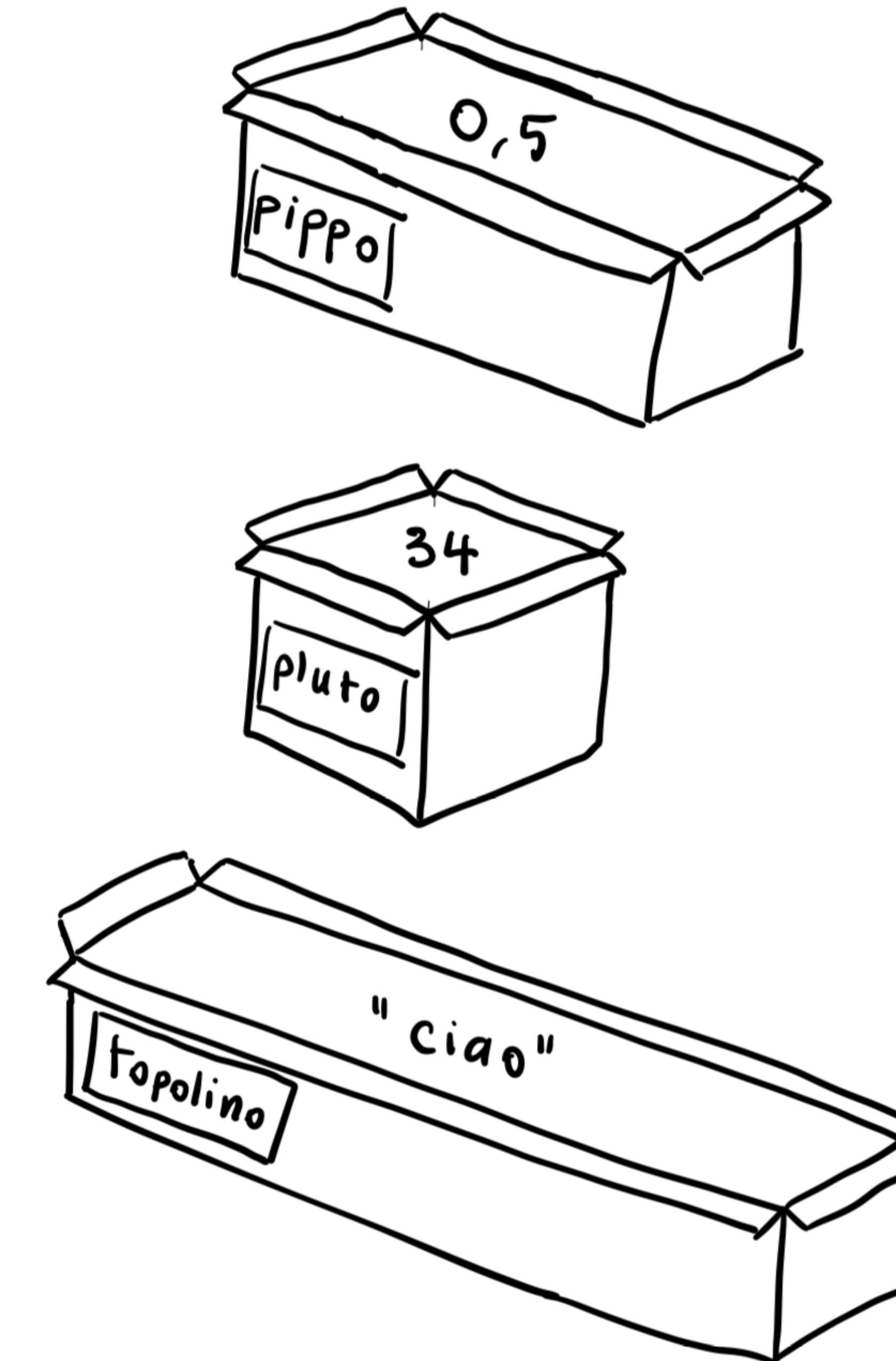
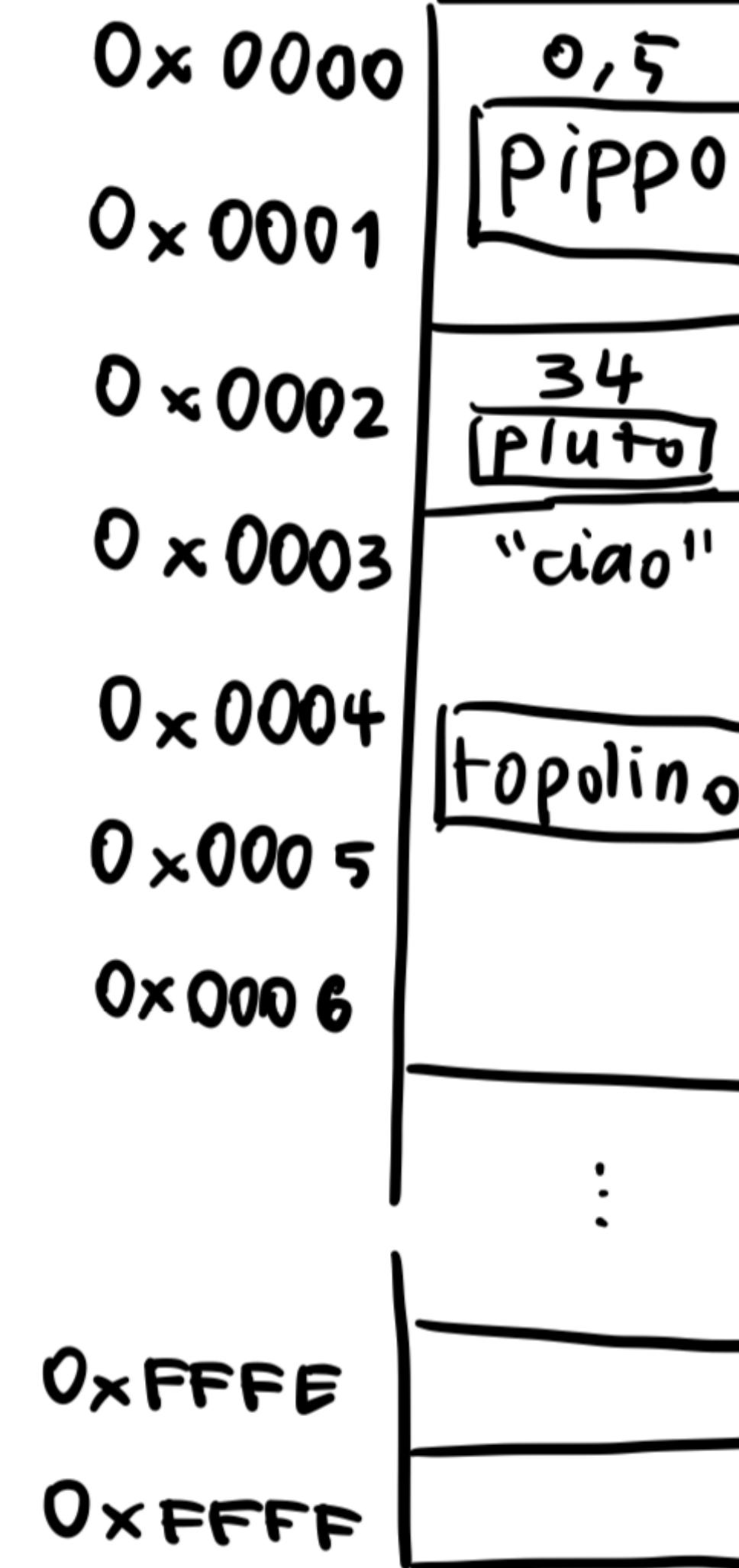
0x 0000



0x 0003



PIIPPO



```
int myNum;  
myNum = 15;
```

C# ▾

↓ TIPO ↓ NOME  
**int** myNum → DICHIAZIONE  
myNum = **15**; → ASSEGNAIMENTO  
    ↑ VALORE

```
myNum = 15
```

```
def saluta():
    print("Ciao")
```

Python ▾

## FIRMA DELLA FUNZIONE

```
def saluta():
    print("Ciao")
```

CORPO DELLA FUNZIONE  
NON RITORNA NESSUN VALORE

```
def saluta_nome(nome):
    nome = " " + nome
    print("Ciao" + nome)
```

FIRMA → NOME FUNZIONE  
def saluta\_nome(nome):

ARGOMENTO → nome

```
    nome = " " + nome
    print("Ciao" + nome)
```

CORPO

NESSUN VALORE RESTITUITO

```
def area_rettangolo(lato1, lato2):
    area = lato1 * lato2
    return area
```

ARGOMENTO1 → lato1  
ARGOMENTO2 → lato2

```
def area_rettangolo(lato1, lato2):
    area = lato1 * lato2
    return area
```

CORPO

VALORE RESTITUITO : LA FUNZIONE TERMINA

DIPENDENZE  
PROGETTO 1

pandas v 0.1.8  
numpy v 1.4.6  
jupiter v 2.0.7

---

DIPENDENZE  
PROGETTO 2

tqdm v 1.0.1  
numpy v 1.5.0

---

DIPENDENZE  
PROGETTO 3

pandas v 2.0.1  
jupiter v 4.6.1

PACCHETTI INSTALLATI  
DOPO IL PROGETTO 1

pandas v 0.1.8  
numpy v 1.1.6  
jupyter v 2.0.7

PACCHETTI INSTALLATI  
DOPO IL PROGETTO 2

pandas v 0.1.8  
numpy v 1.5.0  
jupyter v 2.0.7  
tqdm v 1.0.1

PACCHETTI INSTALLATI  
DOPO IL PROGETTO 3

pandas v 2.0.1  
numpy v 1.5.0  
jupyter v 4.6.1  
tqdm v 1.0.1

### PROGETTO 1

- └ main.py
- └ utils.py
- └ venv
  - └ pandas v 0.1.8
  - └ numpy v 1.4.6
  - └ jupiter v 2.0.7

### PROGETTO 2

- └ main2.py
- └ utils2.py
- └ venv
  - └ tqdm v 1.0.1
  - └ numpy v 1.5.0

### PROGETTO 3

- └ main3.py
- └ utils3.py
- └ venv
  - └ pandas v 2.0.1
  - └ jupiter v 4.6.1