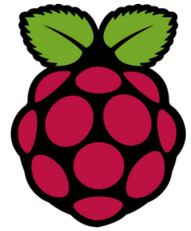
Linux Day 2023

# LAB
# RASPBERRY PI 

- ChatBot basato su Python e OpenAI API
- Assistente vocale (plus)

LINUX DAY

bitO1

# Step 1: Prepariamo la micro SD

- Scarica PI Imager dal sito ufficiale (www.raspberrypi.com)
- Installalo sul tuo computer
- Inserisci la micro SD che desideri utilizzare nel PC
- Avvia Raspberry PI Imager
- Sotto la voce “Sistema operativo”: scegli uno tra i SO proposti (es. Raspberry PI OS 32)
- Seleziona alla voce “Scheda SD” il dispositivo che intendi formattare
- Premi il pulsante “Scrivi”


Raspberry PI Imager - Software ufficiale

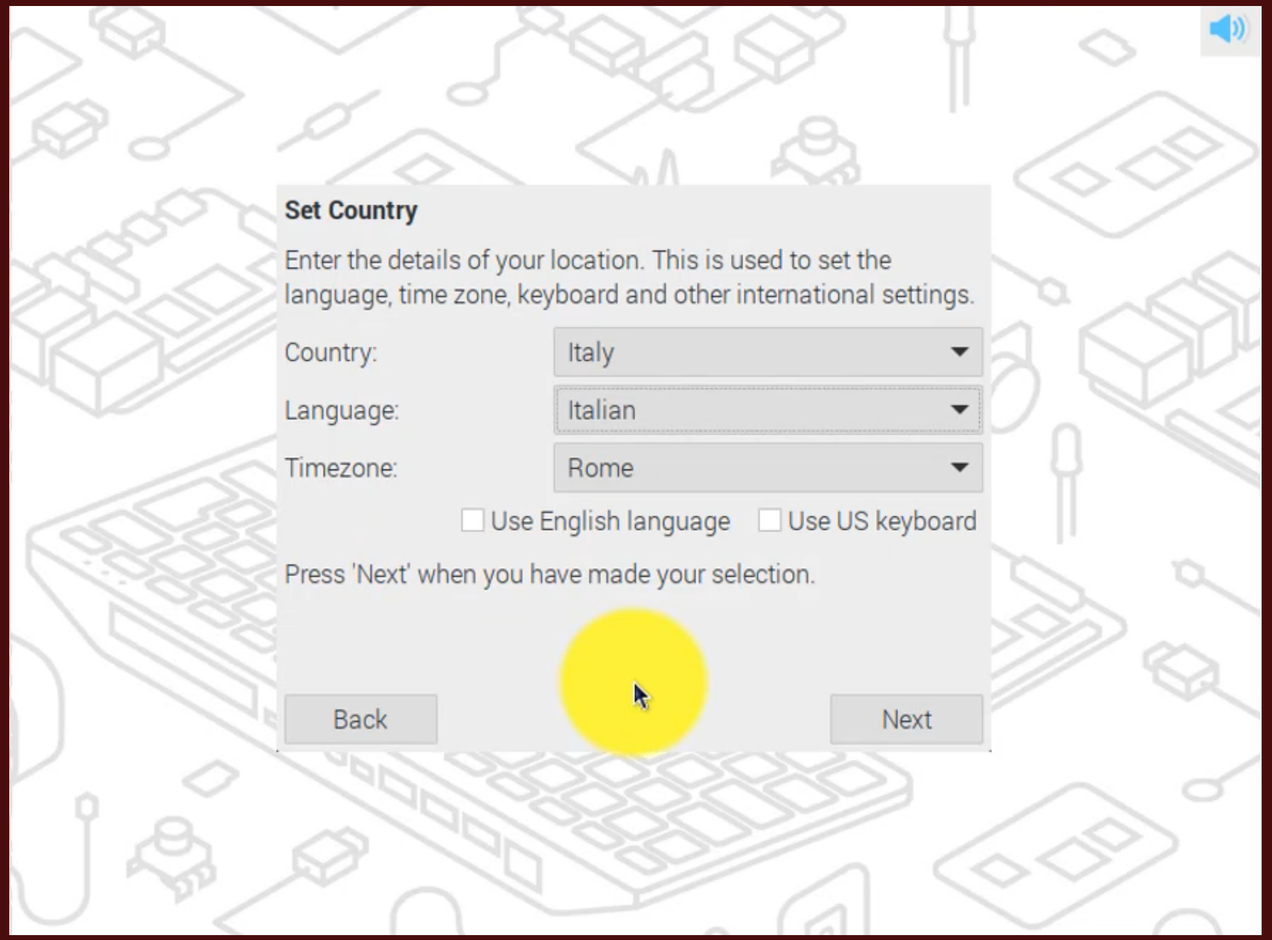
bit01

# Step 2: Colleghiamoci a Raspberry PI

- Inserisci la micro SD appena creata;
- Collega il Raspberry PI all'alimentazione con l'apposito cavo usb;
- Collega un monitor con cavo HDMI;
- Inserisci i restanti componenti (mouse, tastiera ed ethernet se non si dispone di un wifi)

bit01

# Step 3: Avviamo l'installazione del sistema




Al primo avvio dovremo installare il sistema operativo. Ci verrà richiesto di scegliere la lingua, creare un utente, settare una password, ecc.

bit01

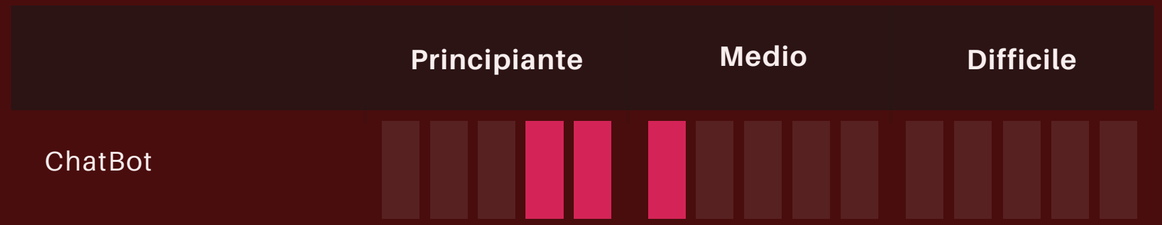
# Raspberry PI

ChatBOT sfruttando uno script Python e le API di OpenAI

bit01

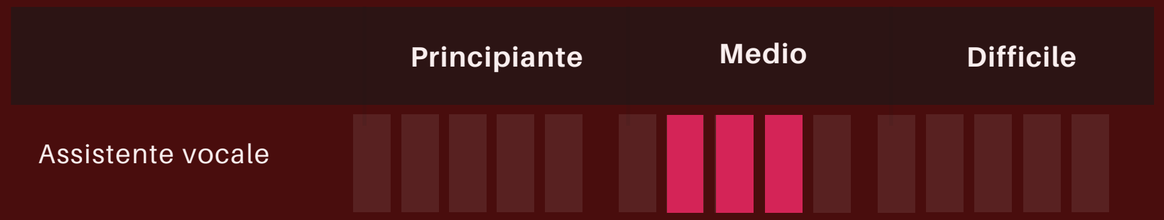
# Cosa ci occorre?

## Progetto principale: ChatBot




- OpenAI API;
- Script Python;
- Raspberry PI 3b+;
- Connessione internet (wifi o ethernet).

## Abilitare funzioni vocali:




- OpenAI API;
- Script Python;
- Raspberry PI 3b+;
- WebCam USB con microfono integrato;
- Connessione internet (wifi o ethernet).

bit01

# Application Programming Interface

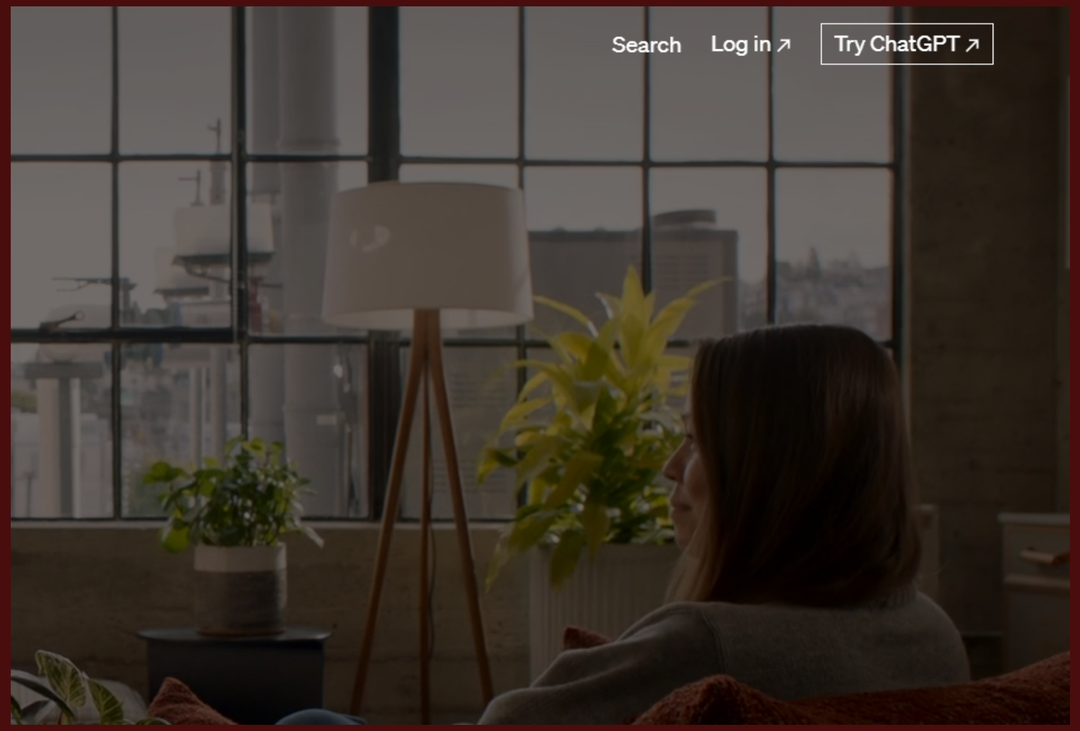
Le API rappresentano una scorciatoia per consentire a delle applicazioni di interagire con altre applicazioni.

Registrandoci ad **OpenAI.com per la prima volta** abbiamo la possibilità di usufruire, per un periodo di tempo limitato, di **5$** di benvenuto; da sfruttare come Developers. Useremo questo credito per interfacciarci con i prodotti **OpenAI** tramite le API.

Dopo aver generato la **API Key** da utilizzare nei nostri progetti (la KEY va mantenuta segreta), nella sezione USAGE di OpenAI, sulla tua sinistra, potrai monitorare il credito utilizzato; questo servirà ad evitare di sforare il budget incorrendo in costi aggiuntivi).

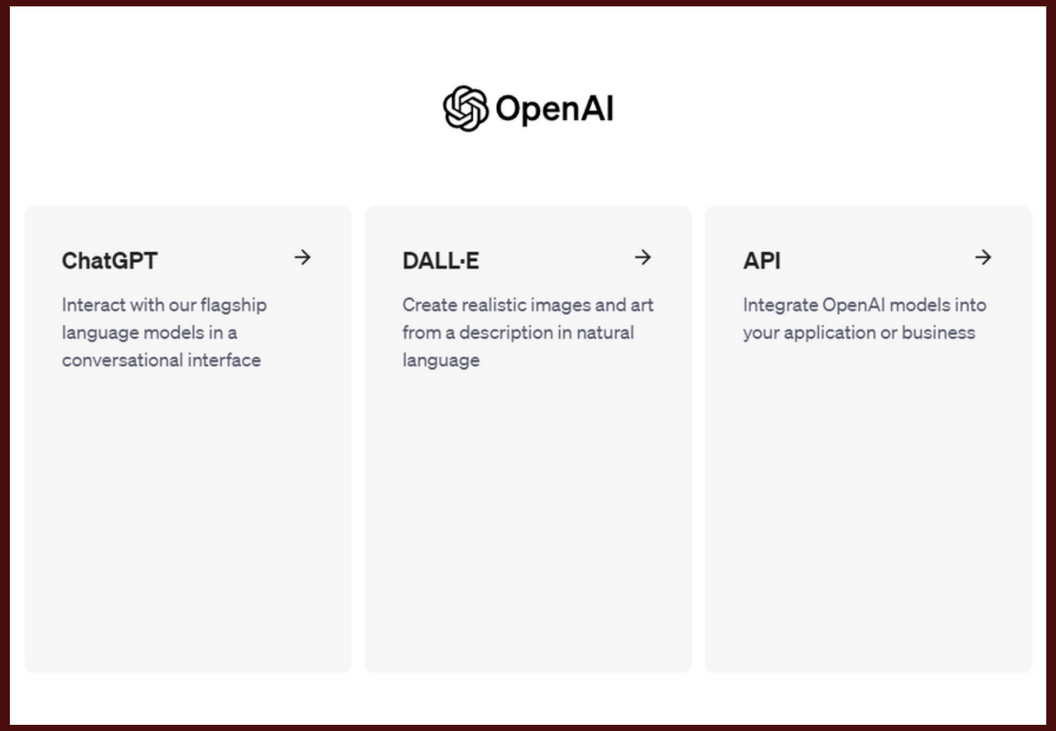
bit01

# Step 4: Ottenere la API Key




1.

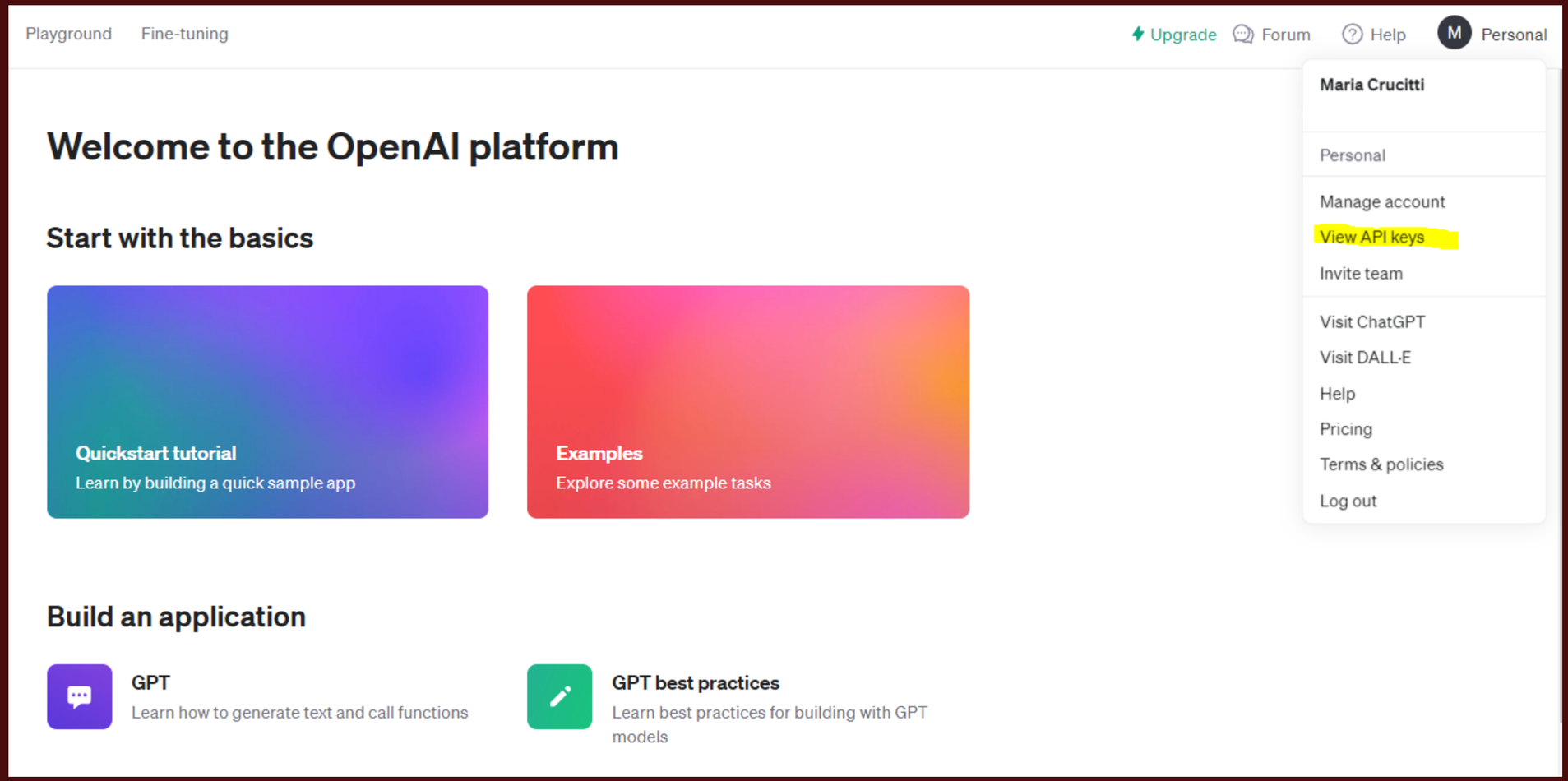
Quando ci registriamo ad OpenAI.com per la prima volta, effettuando il Log-in...




2.

Ci troveremo davanti a questa schermata. Clicchiamo sull'ultima voce: API.

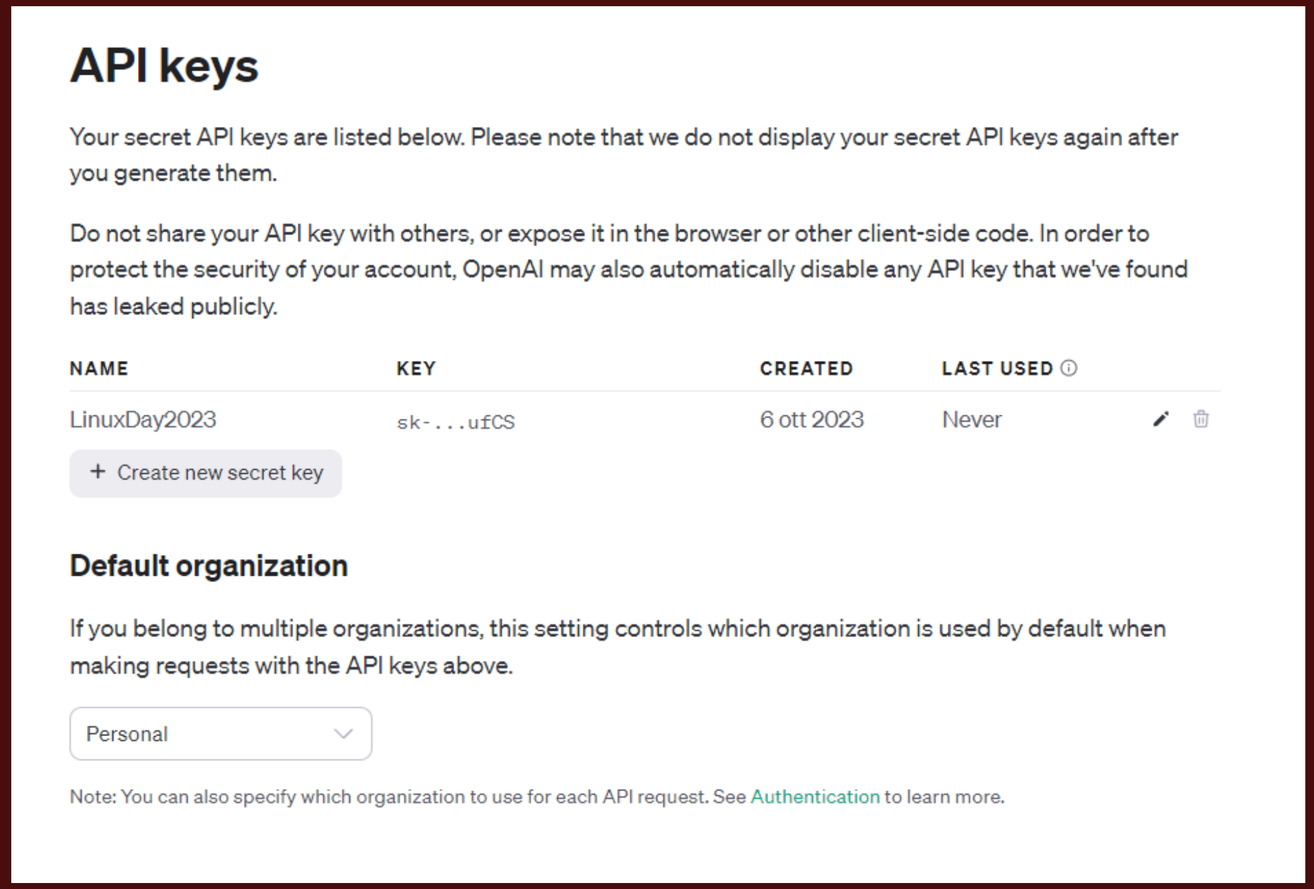
bit01




3.

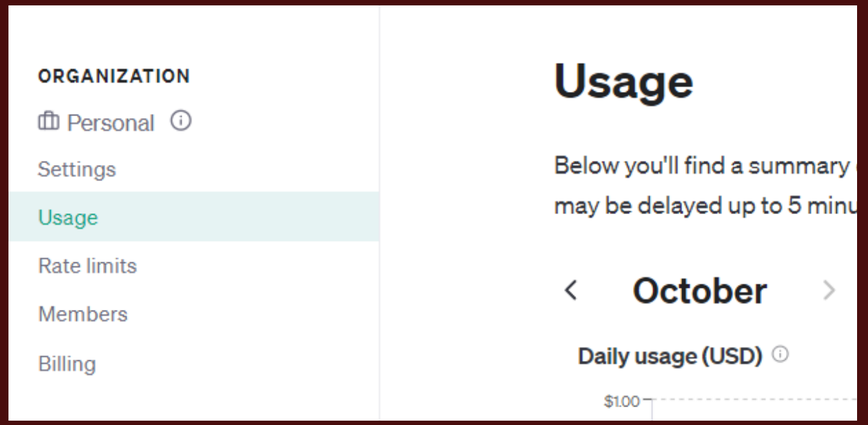
in alto a destra scegliamo: Personal > View API Keys >

bit01




NOTA

Tieni sempre d'occhio la sezione USAGE, con il credito rimanente




4.

Andiamo ora a Creare una API Key cliccando il pulsante: + Create new secret key. Non resta che copiare e salvare la key appena generata all'interno del nostro computer.

bit01

# Raspberry PI

Come creare e lanciare lo script...

bit01

# Step 5: Aggiorna il sistema ed installa openAI...

Apriamo il terminale e lanciamo i seguenti comandi, uno per volta.

```
sudo apt update
sudo apt upgrade
pip3 install openai
```

Raspberry supporta python nativamente, ma assicurati di controllare che sul tuo Raspberry sia presente l'ultima versione - python3; altrimenti procedi con "sudo apt-get install python3".

Ricordati infine di cambiare l'audio settandolo in HDMI. In alto a destra, dall'apposita icona del SO, tramite interfaccia grafica sarà possibile settarlo.

bit01

# Step 5: lo script in python...

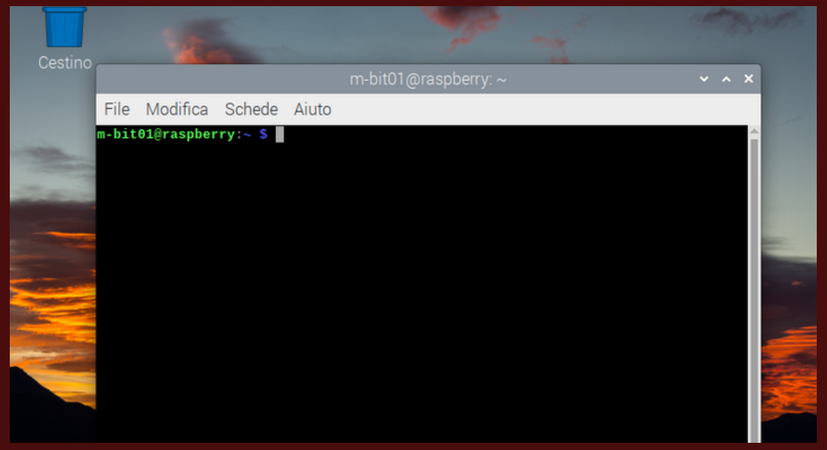
Apriamo un editor di testo (es. nano) ed inseriamo il seguente codice da salvare con estensione .py

```
import openai
def chatBot(text):
    openai.api_key = "Inserisci qui la tua API KEY"
    response = openai.Completion.create(
        engine = "text-davinci-003",
        prompt = text,
        temperature = 0.3,
        max_tokens = 200,
    )
    return print(response.choices[0].text)
def main():
    while True:
        print('Chiedimi quello che vuoi:\n')
        domanda = input()
        chatBot(domanda)
        print('\n')
main()
```

Una volta salvato, lancialo scrivendo:

```
python3 nomeScript.py
```

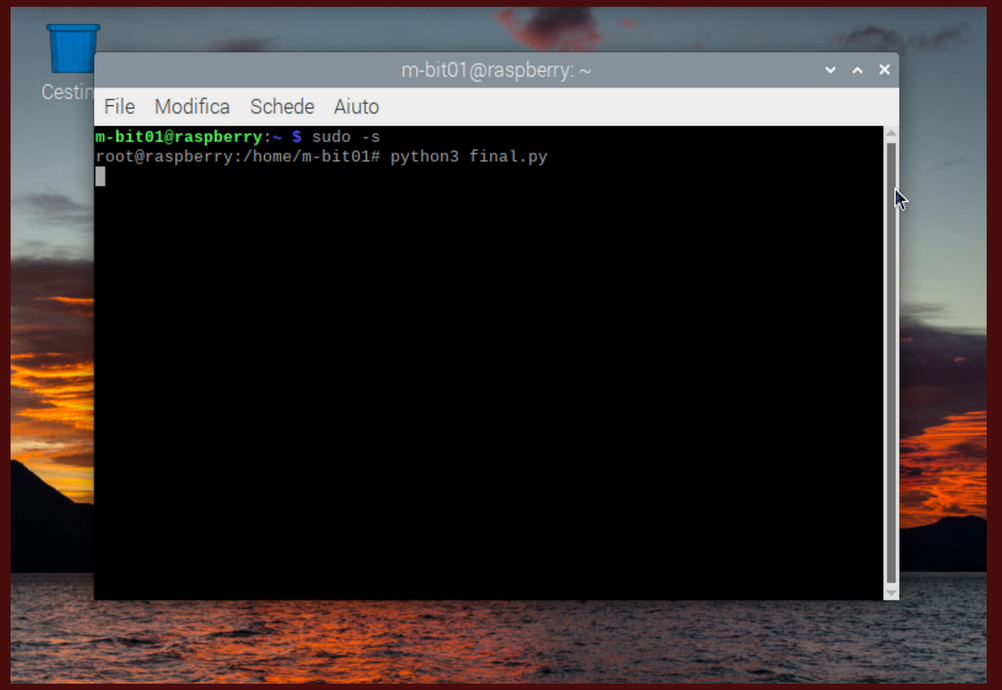
Testiamo il codice chiedendo qualcosa:




bit01

# Assistente Vocale

Cambiando script, e con qualche passaggio in più, possiamo abilitare i comandi vocali (bonus project):




bit01

# Raspberry PI

Assistente vocale...

bit01

# Script per l’assistente vocale

Per il test ho usato una **Smart-TV**, ricordiamoci l’audio del raspberry va settato in HDMI.

```
pip install openai
pip install pyaudio
pip install SpeechRecognition
pip install gtts
sudo apt-get install mpg321
```

Nb. all’occorrenza potrebbero essere richieste durante l’esecuzione altre librerie da installare sempre con: apt-get. Ad esempio, mpg321

bit01

# Script

```
import openai
import speech_recognition as sr
from gtts import gTTS
import os


# Configurazione per OpenAI
openai.api_key = "API_KEY"
model_engine = "text-davinci-003"
temperature = 0.5

# Configurazione per sintesi vocale
language = "it"

# Configurazione per riconoscimento vocale
r = sr.Recognizer()
r.energy_threshold = 4000
r.pause_threshold = 0.6

# Variabili per mantenere il contesto della
conversazione
conversation_history = []
max_history_length = 5
```

```
# Funzione per sintetizzare la risposta vocale
def speak(text):
    tts = gTTS(text=text, lang=language)
    tts.save("response.mp3")
    os.system("mpg321 response.mp3")

# Funzione per riconoscere il comando vocale
def listen():
    with sr.Microphone() as source:
        audio = r.listen(source)
    try:
        command = r.recognize_google(audio,
language="it-IT")
        print("Hai detto: " + command)
        return command
    except:
        return ""
```

bit01

# Script

```
# Funzione per ottenere una risposta da OpenAI
def get_openai_response(prompt, temperature):
    response = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=1024,
        n=1,
        stop=None,
        temperature=temperature,
    )
    message = response.choices[0].text.strip()
    return message

# Loop principale per la conversazione

while True:
    # Richiedi un comando vocale all'utente
    prompt = "\nUtente: "
    user_input = listen()

    # Se il comando vocale è stato riconosciuto, aggiungilo alla
conversazione e chiedi a OpenAI di generare una risposta
    if user_input:
        conversation_history.append(user_input)
        conversation_history = conversation_history[-
max_history_length:]
        context = "\n".join(conversation_history[-
max_history_length:])
        prompt += context + user_input + "\nAI: "
        response = get_openai_response(prompt, temperature)

        # Sintetizza la risposta vocale e riprodurla
        print("AI: " + response)
        speak(response)

    # Se non viene riconosciuto alcun comando vocale, continua ad
ascoltare
    else:
        print("Scusa, non ho capito. Puoi ripetere?")
```

bit01

# In sintesi

Per il funzionamento di un assistente vocale è richiesta l'elaborazione del suono.
Grazie al **riconoscimento vocale** (Speech Recognition o **SST** che sta per "speech to text") saremo in grado di generare un prompt scritto dalla nostra voce, questo sarà dato in input all'AI (come accadeva nel caso del chatBOT).
Una volta elaborate le nostre richieste, per avere un audio in risposta, dovremo avvalerci di un processo che si chiama **sintesi vocale** (la TTS - text to speech); il suo obiettivo è di ricostruire una voce il più possibile simile a quella umana.

Le librerie utilizzate sono:
openai - interfaccia python per utilizzare openAI ed i suoi servizi;
pyaudio - tra i suoi utilizzi c'è quello di registrare audio da un microfono e riprodurre audio;
speechRecognition - serve al riconoscimento vocale;
gtts - google text to speech, servirà a generare un audio;
mpg321 - lo script "assistenteVocale" genera un file mp3 locale, questa libreria è utile per riprodurre gli mp3

NB. alcune librerie viste funzionano sia in modalità online che offline. Altre richiedono la connessione.

bit01

# Riferimenti

HTML.it
https://bit.ly/3FkoLRv

GITHUB
https://github.com/ThomasVuNguyen/chatGPT-Voice-Assistant

DOMSORIA.COM
https://bit.ly/3FlmLbK

bit01

# GRAZIE PER L'ATTENZIONE


bit01

LAB - MARIA CRUCITTI