

Factotum BOT

Andrea Rustichelli

www.andrea.rustichelli.name

andrea@andrea.rustichelli.name

Ed. 2019.09.02

Cos'è "Factotum BOT"?

Un demo BOT per Telegram
realizzato in Python che implementa
5 funzionalità specifiche.

Perché “Factotum BOT”?

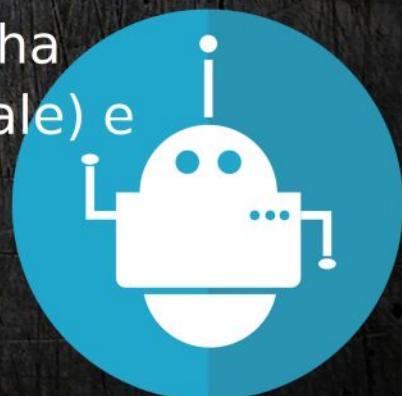
- L'idea di partenza era quella di creare un piccolo sistema di domotica utilizzando un Raspberry controllabile da remoto tramite internet;
- Una possibile soluzione ipotizzata è quella di utilizzare un BOT in Telegram per risolvere tutti i vari problemi che si presentano con una simile impostazione, come ad esempio:
 - Necessità di creare una interfaccia grafica per comunicare con il Raspberry
 - Necessità di raggiungere il Raspberry tramite la rete internet (dynamic DNS)
- Da questo è nata la necessità di sperimentare con i BOT per Telegram in modo da capire quali sono le potenzialità e i limiti di questa architettura
- Capire quale linguaggio di programmazione utilizzare per avere la massima flessibilità e la migliore integrazione con un sistema basato su Raspberry
- Era necessario sviluppare un demo BOT per avere una risposta a tutti gli interrogativi e i dubbi sollevati da un simile progetto...

Un demo **BOT** per Telegram realizzato in Python che implementa 5 funzionalità specifiche.

- Cosa sono i bot -

Il termine BOT è la contrazione, prettamente in ambito informatico, di roBOT, ovvero entità software (o, ancora meglio, software agent) che agiscono come se fossero persone ed eseguono delle attività in modo più o meno automatizzato e con poco intervento umano (se non addirittura senza intervento umano).

I primi bot utilizzavano per l'interazione con l'uomo una semplice logica di programmazione basata sulla presenza, nel testo o nel corso del dialogo, di particolari parole o frasi che poi innescavano la logica applicata nella programmazione. Ovviamente nei nostri giorni questa semplice logica ha lasciato sempre più spazio all'applicazione della IA (intelligenza artificiale) e all'implementazione di tecniche di «machine learning».



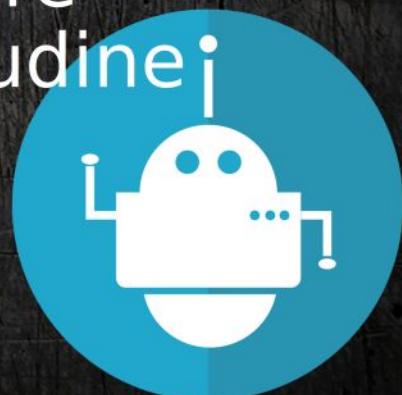
Un demo **BOT** per Telegram realizzato in Python che implementa 5 funzionalità specifiche.

- Tipologia di bot -

Di Bot ne esistono tantissimi tipi, solo per citarne alcuni:

- chatterbot
- web crawler (o web bot, o www bot o internet bot)
- IRC bot (Internet Relay Chat)
- video game bot

Sostanzialmente, visto che con i Bot è possibile fare praticamente tutto, è possibile trovare una moltitudine di definizioni in base alla loro attività/scopo.

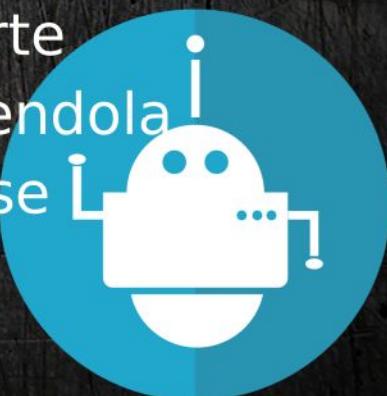


Un demo **BOT** per Telegram realizzato in Python che implementa 5 funzionalità specifiche.

- BOT trivia (1 di 2) -

Uno dei primi interessanti esperimenti con i bot è sicuramente rappresentato da Eliza, un bot creato nel 1966 al MIT dal professor Joseph Weizenbaum con lo scopo di simulare una chiacchierata con una psicoterapeuta di nome Eliza. Di questo bot ne sono state fatte varie versioni anche per ambiente MS DOS nel corso degli anni 90.

Una curiosità che riguarda ELIZA è MELIZA, una sorta di “easter egg” presente in Google Earth che si può attivare posizionandosi su Marte (vicino alla Face of Mars) dove si può trovare una icona che, premendola fa partire una chat con un presunto alieno che implementa le stesse funzioni del bot originale Eliza.



Un demo **BOT** per Telegram realizzato in Python che implementa 5 funzionalità specifiche.

- BOT trivia (2 di 2) -

Nel 2016 la Microsoft aveva rilasciato su Twitter il bot TAY (Thinking About You) che aveva lo scopo di apprendere dagli utenti e interagire in modo autonomo con loro. TAY, dopo solo 16 ore dal lancio, è stato rimosso a causa dei tweet offensivi e dal contenuto polemico pubblicati dal bot.

Infine, merita una menzione d'onore il bot [LSJBOT \(codice sorgente\)](#) creato dal professore svedese Sverker Johansson per creare in modo autonomo delle pagine per la versione Svedese di Wikipedia. Il bot è in grado di generare fino a 10.000 articoli al giorno e questo ha portato l'edizione svedese ad essere la seconda edizione a raggiungere i 2 milioni di articoli.



Un demo BOT per **Telegram** realizzato in Python che implementa 5 funzionalità specifiche.
- Cos' è **Telegram** -

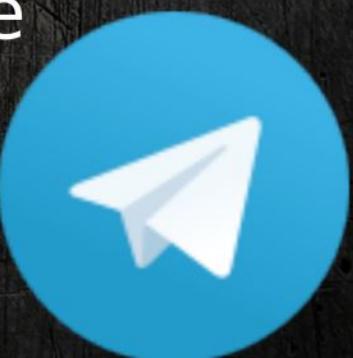
Telegram è un servizio di messaggistica istantanea (come WhatsApp e Hangouts) basato su cloud ed erogato senza fini di lucro dalla società Telegram LLC, una società a responsabilità limitata con sede nel Regno Unito, fondata dall'imprenditore Russo Pavel Durov.

I client ufficiali di Telegram sono distribuiti come software libero per Android, GNU/Linux, iOS, MacOS, Windows e Windows Phone.



Un demo BOT per **Telegram** realizzato in Python che implementa 5 funzionalità specifiche.
- Perché Telegram -

- Gratuito
- Utenti in aumento
- Votato alla sicurezza e alla privacy
- Multiplattaforma (smartphone, tablet, PC e Mac)
- Possibilità di programmare BOT e inserirli nelle chat (anche in chat di gruppo con più persone “reali”)
- Permette di avere già a disposizione il «front end» e concentrarsi unicamente sul «back end»



Un demo BOT per Telegram realizzato in **Python** che implementa 5 funzionalità specifiche.

- Cos' è **Python** -

Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti sviluppato da Guido van Rossum agli inizi degli anni 90.



Un demo BOT per Telegram realizzato in **Python** che implementa 5 funzionalità specifiche.

- Perché Python -

Perché Python è, al momento, uno dei linguaggi di programmazione più interessanti e sul quale la comunità mondiale sta guardando con crescente interesse per lo sviluppo di nuove applicazioni.

Python è il 3° linguaggio di programmazione più utilizzato al mondo e il suo utilizzo è cresciuto rapidamente nel corso degli ultimi anni.

Il crescente interesse per le applicazioni di intelligenza artificiale sta spingendo l'utilizzo di questo linguaggio di programmazione.



Un demo BOT per Telegram realizzato in **Python** che implementa 5 funzionalità specifiche.

- TELEPOT -

TELEPOT è una libreria Python gratuita e semplice e completa per creare BOT per Telegram.

TELEPOT utilizza una classe BOT che non è altro che un oggetto (c.d. “wrapper”) costruito intorno alle BOT API di Telegram.

La libreria al momento non è più in fase di sviluppo pertanto, per progetti futuri, è opportuno valutare la possibilità di utilizzare una delle altre API disponibili per programmare BOT in Telegram utilizzando il Python.

Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

Il BOT:

- 1] se riceve un messaggio di testo, risponde in modo appropriato;
- 2] se riceve una posizione GPS, fornisce l'elenco di alcuni punti geografici di riferimento;
- 3] se l'utente lancia un apposito comando, può visualizzare una «tastiera virtuale» per accedere velocemente a funzioni o a documenti;
- 4] se riceve una foto, prova a cercare se è presente un QR code, a leggerlo e a fornire una risposta in base al contenuto del QR code;
- 5] se riceve un messaggio audio, prova ad analizzarlo (c.d. “speech recognition”) e risponde in modo appropriato.



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- **Funzionalità 1 - Risposta ad un messaggio di testo (1 di 2) -**

La funzione `formula_risposta([...])` prende in input una stringa (chiamata “domanda”) e la divide nelle singole parole che la compongono.

Successivamente le parole della “domanda” vengono confrontate con le parole presenti nelle 3 liste e, se riconosciute, vengono aggiunte in una nuova lista che contiene le parole chiave individuate.

In questo modo, se la nuova lista è vuota, viene visualizzato un messaggio che riporta che non è stato possibile individuare nessuna parola chiave.

Se, al contrario, vengono riconosciute le parole chiave, il BOT invia un messaggio e, optionalmente, può effettuare altre operazioni decise dal programmatore.



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 1 - Risposta ad un messaggio di testo (2 di 2) -

```
HOT_WORD1=["pippo", "Pippo", "PIPPO"]
HOT_WORD2=["pluto", "Pluto", "PLUTO"]
HOT_WORD3=["paperino", "Paperino", "PAPERINO"]

for i in parole:
    if i in HOT_WORD1:
        parole_chiave.append(i)
        bot.sendMessage(chat_id, "Trovata la HOT_WORD1")
    if i in HOT_WORD2:
        parole_chiave.append(i)
        bot.sendMessage(chat_id, "Trovata la HOT_WORD2")
        #bot.sendDocument(chat_id, open('file_name.pdf', 'rb'))
    if i in HOT_WORD3:
        parole_chiave.append(i)
        bot.sendMessage(chat_id, "Trovata la HOT_WORD3")
        #bot.sendMessage(chat_id, "https://www.google.it")
```



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 2 - Posizione geografica (1 di 2) -

Se viene passata la posizione geografica all'interno della chat, Factotum utilizza la funzione `trova_posizione(...)`, si collega al sito <https://www.geoplugin.com/> e utilizza il webservices “Nearby Places” per ottenere i luoghi vicini alla posizione geografica indicata. La posizione geografica deve essere indicata direttamente nell’URL della richiesta al webservice.

Il webservice restituisce un oggetto JSON con alcune informazioni sui luoghi vicini (come il nome del posto, la distanza e la direzione) che vengono visualizzate dal BOT.

L’aspetto più difficile per la gestione dell’output è stata la conversione dell’oggetto JSON in formato “UTF-8” per permettere la corretta gestione dei dati contenuti nella risposta JSON.

L’URL per invocare il webservice è strutturato nel seguente modo:

```
http://www.geoplugin.net/extras/nearby.gp?  
lat=LATITUDINE&lon=LONGITUDINE&format=json
```



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 2 - Posizione geografica (2 di 2) -

```
gps_url="http://www.geoplugin.net/extras/nearby.gp?lat=" + str(gps_lat) +
"&lon=" + str(gps_lon) + "&format=json"

with urlopen(gps_url) as response:
    html_response = response.read()

encoding = response.headers.get_content_charset('utf-8')

decoded_html = html_response.decode(encoding)

data_json=json.loads(decoded_html)
for posti in data_json:
    print(posti["geoplugin_place"],
posti["geoplugin_distanceKilometers"], posti["geoplugin_directionHeading"])

return
```



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- **Funzionalità 3 - Tastiera virtuale o Inline Keyboard (1 di 2) -**

È stato predisposto un apposito comando (funzionalità disponibile nel menù di configurazione dei BOT di Telegram) che, in caso di attivazione visualizza la tastiera virtuale (Inline Keyboard nel gergo di Telegram) tramite una apposita funzione chiamata `attiva_maggiordomo([...])`

In questo modo compaiono 4 tasti (divisi su due righe) che riportano come etichetta il testo presente nell'attributo “text” e che, una volta premuti, restituiscono il messaggio presente nell'attributo “callback_data”.

Proprio tramite questo attributo il programma recepisce la scelta fatta dall'utente e può attivare il codice di programma corrispondente.



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- **Funzionalità 3 - Tastiera virtuale o Inline Keyboard (2 di 2) -**

```
def attiva_maggiordomo(chat_id):  
  
    print("Maggiordomo attivato")  
  
    keyboard = InlineKeyboardMarkup(inline_keyboard=[  
  
        [InlineKeyboardButton(text='Testo key1', callback_data='key1'),  
         InlineKeyboardButton(text='Testo key2', callback_data='key2')],  
  
        [InlineKeyboardButton(text='Testo key3', callback_data='key3'),  
         InlineKeyboardButton(text='Testo key4', callback_data='key4')]  
    ])  
  
    bot.sendMessage(chat_id,"Attivazione del Maggiordomo in  
corso...",reply_markup=keyboard)
```



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 4 - Lettura del QR Code (1 di 2) -

È possibile inviare una immagine a Factotum utilizzando la fotocamera dello smartphone oppure allegando una immagine presente nella galleria. Quando Factotum si accorge che il tipo di input è una immagine, attiva la funzione `leggi_QR_code([...])` che salva in un file temporaneo (e quindi si evita di salvare fisicamente il file sul disco) e poi prova a decodificare l'eventuale QR Code contenuto nell'immagine. Se il BOT non trova nessun codice QR visualizza un apposito messaggio.

Il codice per questa funzionalità è stato tratto dal sito www.allafinedelpalo.it e il collegamento al relativo articolo può essere trovato a questo link.
<http://www.allafinedelpalo.it/implementiamo-un-bot-telegram-in-python-per-leggere-e-creare-qrcode/>



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 4 - Lettura del QR Code (2 di 2) -

```
from pyzbar.pyzbar import decode

raw_img = BytesIO()
bot.download_file(msg['photo'][-1]['file_id'], raw_img)
img = Image.open(raw_img)

qrcodes = decode(img)

if len(qrcodes) > 0:
    for code in qrcodes:
        print("msg: " + str(code.data))
else:
    print("Non ho trovato QRCode nella foto...")
```

Ovviamente è possibile aggiungere un semplice “if” per eseguire del codice o fornire risposte diverse in base al contenuto del QR code riconosciuto.



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- Funzionalità 5 - Speech Recognition di un messaggio vocale (1 di 3) -

- Per la funzionalità di “speech recognition” ho deciso di utilizzare l’omonima libreria e usare Google API per effettuare il riconoscimento del messaggio vocale.
- Quando viene inviato un messaggio audio, viene eseguita la funzione `analizza_audio(...)` che si occupa di salvare il messaggio audio, convertirlo in WAV, effettuare lo “speech recognition” e, se è stato riconosciuto del testo valido, invocare nuovamente la funzione `formula_risposta(...)`
- La prima difficoltà è che il motore per il riconoscimento del testo accetta in input solamente alcuni formati audio come il WAV e il FLAC, mentre i messaggi vocali di Telegram sono in formato OGG.
- Si è quindi reso necessario salvare in locale il file OGG contenente il messaggio vocale e convertire tale file in formato WAV utilizzando la libreria PYDUB scaricata direttamente da GitHub.
- Un volta convertito il file, è stato sufficiente passare il file WAV al modulo dello speech recognizer e indicare di “tradurre” il testo utilizzando la lingua italiana (IT_it) perché di default Google prova a ottenere del testo in lingua inglese.



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

- **Funzionalità 5 - Speech Recognition di un messaggio vocale (2 di 3) -**

Per installare la libreria PYDUB ho scaricato di GitHub direttamente gli ultimi sorgenti e li ho installati sul pc con i seguenti comandi (utilizzando pip3 avevo dei conflitti):

```
git clone https://github.com/jiaaro/pydub.git  
cd pydub  
sudo python3 setup.py install
```

Inoltre, per essere sicuro di non avere problemi nella gestione dei file audio, ho installato la libreria ffmpeg:

```
sudo apt-get install ffmpeg libavcodec-extra
```



Un demo BOT per Telegram realizzato in Python che implementa **5 funzionalità specifiche**.

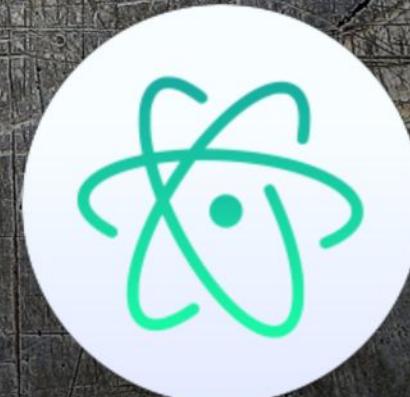
- Funzionalità 5 - Speech Recognition di un messaggio vocale (3 di 3) -

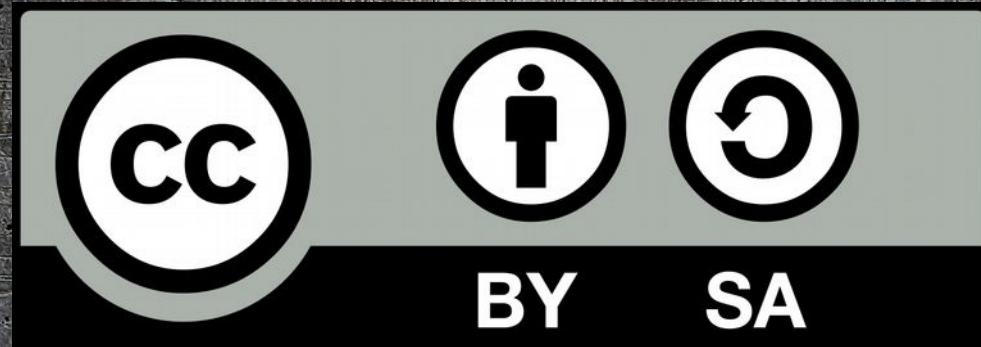
```
bot.download_file(ogg_file_id, "audio_da_telegram.ogg")
file_ogg = AudioSegment.from_ogg("audio_da_telegram.ogg")
file_handle = file_ogg.export("output.wav", format="wav")
r = sr.Recognizer()
with sr.WavFile("output.wav") as source:
    audio = r.record(source)
try:
    domanda=r.recognize_google(audio,language="it_IT")
    print("Trascrizione del file audio: " + domanda)
    formula_risposta(bot,chat_id,msg,domanda)
except LookupError:
    print("Impossibile riconoscere del testo nel file audio analizzato")
```



Ambiente di sviluppo

- SO : Linux Mint 19.1 Tara
- Linguaggio : Python 3.6.7
- IDE : ATOM
- Librerie Python :
 - Telepot
 - Json
 - Pyzbar
 - PIL
 - Pydub
 - speech_recognition
 - io
 - urllib





- Quest'opera è distribuita con Licenza
- [Creative Commons] [Attribuzione]
- [Condividi allo stesso modo] [4.0] [Internazionale]

<http://creativecommons.org/licenses/by-sa/4.0/>