دستیار هوش مصنوعی در پایتون

AI Personal Assistant in python _ Bimex

Producer : Fatemeh Rahmani
professor : Mr.allah Koozegar
Project of Basics of Computer and Programming
University of Guilan

1399-00

به نام یکتای بی همتا

فهرست مطالب Table of Contents

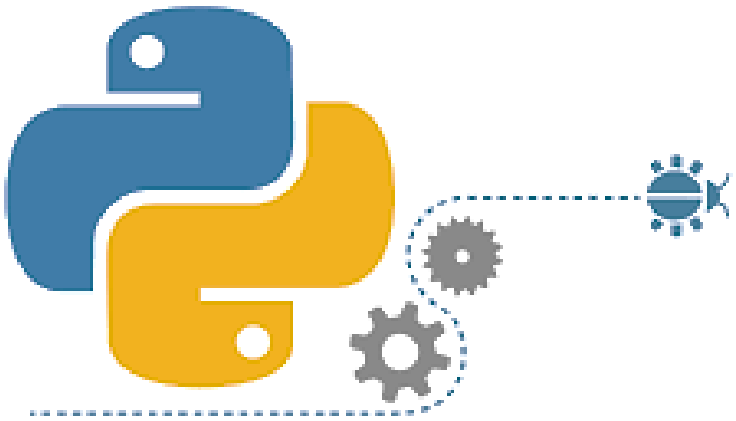
چکیده

دراین نوشتار قصد دارم که به بررسی دستیار مجازی هوش مصنوعی (AI Personal Assistant) در زبان پایتون که به عنوان پروژه پایانی برای این ترم در نظر گرفته شده بپردازم.

در واقع تکنولوژی دستیار شخصی با قابلیت پاسخ به درخواست ها، یک تکنولوژی کاراست که در طی چند سال اخیر در جهان بسیار گسترده شده است و اکنون در سال ۲۰۲۰، بسیاری از نرم افزار های پیام رسان از ربات ها و دستیاران شخصی پشتیبانی می کنند .مثلا ربات های تلگرامی وجود دارند که بر اساس متنی که شما تایپ و ارسال می کنید می توانند پاسخ منحصر به فردی برای شما ارسال کنند.

به عنوان کسی که تازه پا به عرصه ی برنامه نویسی گذاشته میتونه یک پروژه بسیار جذاب والبته کارآمد باشه کهدر این پروژهمن اززبان پایتون (Python) واز IDE پایچارم (PyCharm) که به منظور تسهیل در ویرایش کد های پایتون توسعه داده شده استفاده کردم که البته هرIDE دیگری هم میتونه باشه وزیادی فرقی هم نمیکنه ...

هدف از نوشتن این پروژه تسهیل در انچام کار های مختلف ودرنتیجه افزایش سرعت در پیش بردن انها هست.



مقدمه

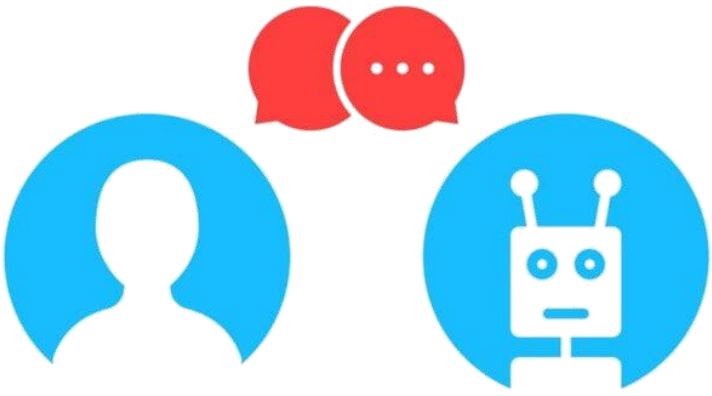
دستیار مجازی هوش مصنوعی (AI Personal Assistant) ، که به آن دستیار هوشمند یا دستیار دیجیتالی هم گفته می‌شود مجموعه ای از اطلاعات و خدماتی به هم پیوسته هست که در آن به صورت آنلاین برنامه قابلیت فهم گفتار طبیعی را دارد و می‌تواند دستورهای صوتی را دریافت نماید وپس از بررسی درصورت صحیح بودن آن به شما پاسخی مناسب بدهدکه این وظایف، سابقا توسط یک دستیار شخصی و یا منشی انجام می‌گرفتند.

امروزه معروف‌ترین دستیارهای هوشمند مجازی شامل الکسای آمازون، سیری اپل، دستیار گوگل و کورتانا مایکروسافت می‌باشند والبته دستیار های دیگری هم وجود دارند ولی این چند نمونه معروف ترینشان هستند.

عبارت دستیار هوشمند بر روی مفهوم دستیار مجازی یا دستیار مجازی شخصی متمرکز است. البته برخی اوقات عبارت دستیار مجازی برای توصیف منشی‌ها و یا دستیارهای شخصی دورکاری نیز گفته می‌شود که یک عامل انسانی است که برای رئیس به صورت دورکار، کارهای برنامه ریزی و ... را انجام می‌دهد.

یک نوع دیگری از دستیارهای مجازی نیز وجود داره که به نوعی در تضاد با این مفهوم هست. این نوع دستیارهای مجازی مشاوران هوشمندی هستند که با کمک هوش مصنوعی به مصرف کنندگان مشاوره می‌دهند. این مشاوران هوشمند حول یک موضوع بخصوص و با محوریت آن به مصرف کنندگان مشاوره می‌دهند .مثل چت بات‌های هوشمندی که روی وبسایت‌های فروشگاهی از آن‌ها استفاده میشه. به طور کلی این نوع دستیارها موضوع محور هستند و دستیارهای مجازی همچون Bimex وظیفه محور.

دستیارهای مجازی معمولا نرم افزارهای مبتنی بر فضای ابری هستند که معمولا نیاز به اتصال اینرتنت دارند. برای مثال هر سه دستیار هوشمند معروف سیری آمازون، کورتانا مایکروسافت و دستیار گوگل که بر روی دستگاه‌های مختلفی در دسترس هستند، برای استفاده از هر کدام از این دستیارهای مجازی برای کار کردن به اتصال اینترنت نیاز دارند که این پروژه رو هم شامل میشه.



در سال‌های اخیر دستگاه‌هایی تولید شده‌اند که به طور اختصاصی به این نرم افزارهای دستیار مجازی اختصاص داده شده‌اند. دستگاه‌های معروف این چنینی که در حال حاضر در بازار موجود هستند و به شرکت‌های بزرگی مانند آمازون، گوگل و مایکروسافت تعلق دارند. برای استفاده و فعال کردن هر کدام از این دستیارها کاربران می‌توانند اسم آن‌ها را صدا کنند و در ادامه کاری را که قرار است دستیار هوشمند برای آن‌ها انجام دهد را به صورت یک دستیار صوتی اعلام می‌کنند.

فناوری‌هایی که به این دستیارهای هوشمند قدرت می‌دهند بر پایه حجم وسیعی از داده هستند. که این داده‌ها به کمک پلتفرم‌های هوش مصنوعی این دستیارها می‌آید. در این دستیارهای هوشمند از تکنولوژهای مبتنی بر هوش مصنوعی دیگری همانند یادگیری ماشین، پردازش گفتار طبیعی و شناسایی گفتار برخوردار هستند. زمانی که کاربر با این دستیارها به تعامل می‌پردازد، هوش مصنوعی از الگوریتم‌هایی پیچیده استفاده می‌کند تا از داده‌ها یاد بگیرد و بتواند در آینده در تشخیص نیاز کاربر بهتر عمل کند.

برخی از افرادی که از دستیارهای مجازی استفاده می‌کنند نسبت به برخی مسائل امنیتی و حریم خصوصی ابراز نگرانی کرده‌اند. برای مثال دستگاه‌هایی همانند الکسای آمازون و گوگل هوم چون به مقدار زیادی از داده‌های شخصی برای پاسخگویی و همواره برای اینکه هنگام نیاز کاربر بتوانند کار کنند در حال شنیدن هستند. فارغ از این‌ها، دستیارهای هوشمند این داده‌ها را حفظ و نگهداری می‌کنند تا از آن‌ها برای آموزش استفاده کنند و هزاران مثال دیگه که با ی سرچ کوچیک به خوبی میتوانید درباره ی آنها اطلاعات کسب کنید.

دستیارهای هوشمند به سرعت در حال پیشرفت هستند تا کارآیی و ارزش بیشتری را برای کاربران به ارمغان بیاورند. با پیشرفت فناوری‌هایی مانند شناسایی گفتار و فهم گفتار طبیعی و همچنین شناخت صدا و... این دستیارها نیز می‌توانند کارهای مهم‌تر و بیشتری را برای کاربرانشان انجام دهند. دستیار هوشمند آینده از فناوری‌های پیشرفته محاسباتی برخوردار هستند که به این ترتیب می‌توانند درخواست‌ها چند مرحله‌ای و کارهای پیچیده‌تری را برای کاربرش انجام دهند.

البته این پروژه فقط یک پروژه دانشگاهی است واستفاده تجاری از آن مد نظر نمی باشد وحتی در فضای وب نیز مورداستفاده قرار نمی گیرد.

در ادامه به بررسی وتحلیل پروژه دستیار مجازی هوش مصنوعی (AI Personal Assistant) می پردازیم....

آشنایی باPython

امروزه تعداد زبان های برنامه نویسی بسیار زیاد است و هر کدام کاربردهای مختلفی دارند. هر کدام از این زبان‌ها مزایا و معایب خودشان را دارند. یکی از زبان های برنامه نویسی مطرح بین برنامه نویسان پایتون است که روز به روز به میزان محبوبیت آن اضافه می شود. از این زبان برنامه نویسی برای انجام کارهایی زیادی از جمله برنامه نویسی هوش مصنوعی، توسعه وب، ساخت اپلیکیشن های موبایل و دسکتاپ استفاده می شود.

پایتون یک زبان برنامه نویسی سطح بالا تفسیر شده برای برنامه نویسی عمومی است. این زبان دارای یک فلسفه طراحی است که بر خواندن کد، به خصوص با استفاده از فضای خالی مهم استوار است. Python دارای یک سیستم نوع پویا و مدیریت حافظه خودکار است و پارادایم های چندگانه برنامه نویسی را پشتیبانی می کند و مفسر پایتون برای بسیاری از سیستم عامل ها در دسترس است.

پایتون یک زبان اسکریپتی است که کدهای آن در پلتفرم های لینوکس، ویندوز، مکینتاش، سیستم عامل‌های موبایل و حتی پلی‌استیشن قابل اجراست و به‌دلیل قابلیت‌های فراوانی که دارد، به یکی از زبان‌های مورد علاقه‌ی برنامه‌نویسان وب تبدیل شده و شرکت‌های بزرگی مثل گوگل، یاهو، اینستاگرام، ناسا، یوتیوب و... در سطح بالایی در حال استفاده از آن هستند.

مزایای پایتون

- حضور ماژول های شخص ثالث

  Python (PyPI)شامل چندین ماژول شخص ثالث است که باعث می شود Python بتواند با بسیاری از زبان ها و سیستم عامل های دیگر ارتباط برقرار کند.

- کتابخانه های پشتیبانی گسترده
- پایتون کتابخانه استاندارد بزرگی را ارائه می دهد که شامل موضوعات مختلف مانند پروتکل اینترنت ، عملیات رشته ، ابزارها و سرویس های وب و رابط های سیستم عامل است. بسیاری از کارهای برنامه نویسی پر کاربرد قبلاً در کتابخانه استاندارد نگاشته شده اند که باعث می شود طول کد به طور قابل توجهی کاهش داده شود.

- منبع باز

زبان پایتون تحت مجوز OSI تأیید شده است که استفاده و توزیع آن را آزاد می کند ، از جمله برای اهداف تجاری. علاوه بر این ، توسعه آن توسط جامعه ای انجام می شود که از طریق میزبانی کنفرانس ها، برای کد آن همکاری می کنند و ماژول های بی شماری را برای توسعه آن فراهم می کنند.

- یادگیری سریع و آسان:

، پایگاه گسترده کاربران و توسعه دهندگان فعال باعث شده است تا یک بانک منابع اینترنتی غنی برای ترغیب توسعه و ادامه پذیرش زبان ایجاد شود.

- ساختار داده های کاربر پسند: :

پایتون دارای ساختار داخلی داده ها و فرهنگ نامه ها است که می تواند برای ساخت سریع داده های زمان اجرا سریع استفاده شود.

- بهره وری و سرعت

پایتون دارای طراحی شی گرا تمیز است ، قابلیت های کنترل پیشرفته یک فرایند را فراهم می کند ، و توانایی های ادغام و پردازش متن دارد ، که همه اینها به افزایش سرعت و بهره وری آن کمک می کند. پایتون گزینه ای مناسب برای ساخت برنامه های پیچیده دارای چند پروتکل تحت شبکه محسوب می شود.

معایب پایتون

- سرعت:

پایتون کندتر از C یا C ++ است. پایتون یک زبان سطح بالا است ، برخلاف C یا C ++ به سخت افزار نزدیک نیست.

- توسعه موبایل:

پایتون یک زبان خیلی خوب برای توسعه موبایل نیست. این یک زبان ضعیف برای محاسبات موبایل است. به همین دلیل است که برنامه های اندکی در تلفن های همراه مانند Carbonnelle در آن ساخته شده اند.

- مصرف حافظه:

پایتون برای کارهای فشرده حافظه گزینه مناسبی نیست. به دلیل انعطاف پذیری انواع داده ها ، مصرف میزان حافظه پایتون نیز زیاد است.

- دسترسی به پایگاه داده:

پایتون با دسترسی به بانک اطلاعات محدودیت هایی دارد. در مقایسه با فن آوری های رایج مانند JDBC و ODBC ، لایه دسترسی به پایگاه داده Python کمی توسعه نیافته و بدوی است.

- خطاهای زمان اجرا:

برنامه نویسان پایتون در زمینه طراحی زبان چندین موضوع را ذکر کردند. از آنجا که این زبان به صورت پویا تایپ می شود ، به آزمایش بیشتری نیاز دارد و دارای خطاهایی است که فقط در زمان اجرا نشان می دهد.

Library ها و کاربردشان

در نوشتن این برنلامه از Library های متعددی و البته با کارایی های متفاوت استفاده شده که در اینجا به بررسی آنها می پردازیم:

Library	کاربرد ماژول ها
speech_recognition	بازشناسی گفتار نیازمند ورودی صوتی است که از طریق PyAudio به راحتی امکان پذیر هست و speech_recognition بازیابی این ورودی را بسیار آسان میکند. ازطرف دیگه، به جای اینکه نیاز به ساخت اسکریپتی برای دسترسی به میکروفون و پردازش فایل صوتی از پایه باشه speech_recognition این کارها را تنها در چند دقیقه برای ما انجام می‌دهد.
pyaudio	بازشناسی گفتار نیازمند ورودی صوتی است درنتیجه وجود ماژول PyAudio برای ثبت ورودی میکروفون لازم هست تا با ارجاع دادن آن فایل های صوتی به speech_recognition برنامه ما تا حدودی از نظر زیر ساختی آماده بشه.
pyttsx3	از ماژول pyttsx3 برای تبدیل متن به صدا استفاده شده تا بتواند به آسانی با کاربر ارتباط برقرارکند.
datetime	ماژول datetime ماژولی است، که اجازه استفاده، ذخیره، و دستکاری ساعت و تاریخ را می‌دهد که در اینجا به منظور تشخیص بازه هایی از روز استفاده

شده.	
با استفاده از ماژول webbrowser بازدید از یک وب سایت یا یک صفحه‌وب برای کاربر امکان پذیر می شود و با نوشتن این برنامه و اجرا شدن آن مرورگر سیستم کاربر به صورت خودکار باز میشود و به آدرسی که از قبل مشخص کردیم هدایت می شود.	webbrowser
ماژول subprocess دارای قابلیت‌های بیشتری نسبت به os برای اجرای دستورات و نمایش خروجی آن‌ها است درنتیجه از آن استفاده شده و دارای توابع بسیاری است که دراین پروژه به منظور باز کردنIDE (visual studio) ، خاموش کردن سیستم (log off) استفاده شده.	subprocess
ماژول playsound که دارای توابع بسیاری هست به منظور پخش موزیک در برنامه بکار رفته و جایگزین خوبی برای pygame میتونه باشه.	playsound
ماژول wolframalpha یک موتور محاسباتی دانش است والبته قدرتمند که این موتور جستجو محاسباتی دارای یک پایگاه دانش فوق العاده هست و از جمله کارهایی که انجام میده: مقایسه اطلاعات تغذیه حل مسائل پیچیده ریاضی مکان شما روز شمار	wolframalpha

تولید پسورد با امنیت بالا نقشه هواشناسی محاسبهBMI و...	
ماژول Requests جزو محبوب ترین کتابخانه های پایتون هست. کاربرد اصلی این کتابخانه این است که این امکان را می دهد تا از طریق HTTP/ درخواست مورد نظر خود را بفرستید و جواب آن را دریافت کنید.	**requests**
ماژول time توابعی مختلفی برای دریافت زمان فعلی سیستم، محاسبه زمان‌ها و ایجاد وقفه در اختیار ما قرار می‌دهد که در اینجا به منظور گرفتن داده های آب و هوا از سایت openwetherبه کار گرفته شده .	**time**
از ماژول Source برای نصب دیگر ماژول ها استفاده شده . و از pipwin به منظور رفع eror نصب pyaudio استفاده شده.	**Source,pipwin**

قابلیت های این پروژه

این پروژه دستیار مجازی هوش مصنوعی که Bimex نام دارد قادر است:

۱. سلام و احوال پرسی کند
۲. خداحافظی کند و برنامه را ببند و ازکاربر درمورد کارایی اش سوال بپرسد که آیا راضی بوده یانه ؟!
۳. درمورد کارهایی که قادر به انجام آن است صحبت کند
۴. خودش را معرفی کند
۵. سازنده ی خودرا به خوبی معرفی کند
۶. درصورتی که کاربربخواهد هوش برنامه را محک بزند و از اون بپرسد که" کیست؟ " قادر به پاسخ گویی است
۷. اسم خودش را می داند و درصورتی که با نام دیگری او را صدا کنند جوابی در خور میدهد
۸. زمان دقیق را در آن واحد بگوید
۹. در ویکی پدیا جست و جو کند
۱۰. یوتیوب را برای کاربر باز کند
۱۱. گوگل کروم را برای کاربر باز کند
۱۲. سایت آمازون را برای کاربر باز کند
۱۳. جیمیل را برای کاربر باز کند
۱۴. اخباری که به تازگی درکشور های جهان اتفاق افتاده رابرای کاربر به نمایش بگذارد
۱۵. در سیستم برای پیدا کردن هر فایلی سرچ کند و آن را برای کاربر نمایش دهد
۱۶. متن مورد نظر کاربر را تایپ کند و آن رادر فایل text برایش ذخیره کند
۱۷. در صورتی که کاربر بخواهد آن رابرایش نمایش دهد وهمچنین برایش بخواند

۱۸. IDEمورد نظر کاربر را باز کند

۱۹. پروژه ای به منظور نوشتن کد در IDEمورد نظر کاربر ایجاد کند

۲۰. به سوالات محاسباتی ، جغرافیایی و... کاربر پاسخ دهد وبرای درک بهتر نتیجه را برایش تایپ کند

۲۱. آب و هوای هرشهری که کاربر بخواهد را به صورت آنلاین و در لحظه برایش باز گو کند

۲۲. مکانی که کابر درخواست کرده رادر GPSپیدا کرده وروی نقشه برایش نمایش دهد

۲۳. موسیقی پخش کند

۲۴. سیستم رابرای کاربر خاموش کند

۲۵. دستوراتی که از جانب کاربر برای برنامه مبهم است را در فایل txt ذخیره کند

شرح چگونگی عملکرد دستیار مجازی هوش مصنوعی

این برنامه با نام **Bimex** با استفاده از ماژول های متفاوت و با مرتبط کردن آن ها به وسیله توابع وساختار های مختلف کار میکند.

که در ابتدا پس از **run**کردن نوشته ی **Loading your computer assistant – Bimex** نمایش داده می شود که نشان دهنده ی **run** شدن صحیح برنامه هست واز طریق تکه کد

```
print('Loading your computer assistant - Bimex')
```

انجام گرفته.

درقسمت بعد به منظور تبدیل متن به صدا و تسهیل برقراری ارتباط میان برنامه و کاربر طبق توضیحات داده شده از ماژول **pyttsx3** استفاده شده وبا کمک توابع مختلف موجود دراین ماژول دستوراتی که در ادامه کد به برنامه داده شده در جواب کاربر اجرا می شوند و از طریق کد

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', 'voices[0].id')
engine.setProperty("rate", 130)
```

انجام میگیرد که دو خط اول به منظور گرفتن متن وتبدیلش به صوت و خط سوم به منظور تعیین صدای برنامه که به ازای **'voices[0].id'** ، صدای مرد را دریافت می کنیم و به ازای **'voices[1].id'** ، صدای زن را.

در خط بعد به منظور تعیین سرعت گفتار **(rate)** از این تکیه کد استفاده شده که هر چه عدد استفاده شده در قسمت دوم بیشتر باشد برنامه با سرعت بیشتر و هرچه این عدد کمترباشد با سرعت پایین تر کلمات را بیان میکند.

درقسمت بعد با ساخت تابع **speak**و با پاس دادن **text** به آن مقدمه بیان دستورات نوشته شده رابه صورت **voices** برای برنامه فراهم میکنیم و از طریق کد

```
def speak(text):
    engine.say(text)
    engine.runAndWait()
```

انجام می شود وبا کمک کد **engine.say(*text*)** به برنامه این امکان را میدهیم که متنی را که نوشته ایم را بگوید و همچنین با استفاده از **runAndWait()** این امکان را به برنامه می دهیم که پس از عمل کردن هر دستور چند ثانیه ای مکث کند و بعد به کارش ادامه بدهد و سایر دستورات را اجرا کند.

سپس با ساخت تابع **beginning** و ایجاد دستوراتی در آن و اسفاده از شروط مختلف از برنامه میخواهیم درابتدای شروع به کاربر پیامی مبنی بر **صبح بخیر good morning** ، **ظهربخیر good afternoon** ، **عصربخیر good evening** را نمایش بدهد و طبق کد

```
def beginning():
    hour = datetime.datetime.now().hour
    if hour>=0 and hour<12:
        speak("Hello,good morning")
        print("Hello,Good Morning")
    elif hour>=12 and hour<18:
        speak("Hello,good afternoon")
        print("Hello,good afternoon")
    else:
        speak("Hello,good evening")
        print("Hello,good evening")
```

پیام های ذکر شده را نمایش می دهد که برای گرفتن داده های زمانی از ماژول **datetime** استفاده کردیم.

وسپس کدی به منظور دریافت **voice**و شناسایی میکروفون پیش فرض سیستم را می نویسیم و آن را درتابع **takecommand**قرار می دهیم وطبق کد

```
def takeCommand():
    r = sr.Recognizer()
    mic = sr.Microphone()
    with mic as source:
        print("Listening...")
        audio = r.listen(source)
```

```
        try:
            statement = r.recognize_google(audio, language='en-in')
            print(f"user said:{statement}\n")

        except Exception as e:
            speak("Please repeat again...")
            return "None,"
        return statement
```

فایل دریافت می شود البته به کمک ماژول **pyaudio** ... و با استفاده از اتصال به اینترنت به پردازش صوت دریافتی می پردازد ودرصورتی که کاربر دستوری به برنامه ندهد ویا به دلیل نویز زیاد قادر به تشخیصش نباشد برنامه از اون می خواهد که دوباره فرمان خود رابیان کند **(Please repeat again..)** و ازانجایی که برایش ناشناخته بوده پیغام **None** بر روی فایلی که به منظور دستورات ناشناخته **(unknown commands.txt)** ساخته شده ظاهر میشود والبته ذخیره هم می شود. و همچنین بعد از این که کاربر فرمان را به برنامه داد ، برنامه فرمان کاربر را پردازش می کند و سپس آن را برروی صفحه تایپ می کند و برای دادن فرمان جدید باید منتظر عبارت **Listening...** باشیم تا برای کاربر نمایش داده شود و سپس او قادر است فرمان جدیدی به برنامه بدهد.

و سپس کد زیر اجرا میشود وتابع **beginning** را فرا می خواند

```
speak("im Bimex your computer assistant")
beginning()
```

و طبق این کد برنامه ابتدا عبارت **im Bimex your computer assistant** را بیان می کند و خودش را معرفی می کند و سپس تابع **beginning** که از قبل توضیحاتش را گفته بودم را فرا می خوانیم تا به برای ما پیغامی متناسب با آن زمان را نمایش دهد.

تا اینجای برنامه تمام مراحل زیر ساختی و پایه انجام شده و فقط مانده دستورات را به برنامه بدهیم تا کارهایی که از آن میخواهیم را درآینده نه چندان دور برایمان انجام دهد ...

وطبق کد زیر

```
if __name__=='__main__':

    while True:
        speak("Tell me how can I help you ?")
        statement = takeCommand().lower()
        if statement==0:
            continue
```

تمام دستورات رادرون یک حلقه Wileقرار میدهیم و همچنین از برنامه میخواهیم در صورتی که فرمان داده شده تهی بود برنامه به کار خود ادامه بدهد و متوقف نشود وپس از انجام هر دستور از کاربر سوال شود که چه کمکی میخواهد؟ (Tell me how can I help you ?)

و البته به منظور کاهش باگ در برنامه از تکه کد **statement = takeCommand().lower()** استفاده شده تا تمامی فرمان هایی که توسط کاربر به برنامه داده می شود با حروف کوچک نوشته شود تا دچار خطا درکارایی برنامه نشود .

وحالا نوبت نوشتن کد هایی به منظور انجام یک دستوراز طرف کاربر رسیده که در ابتدا با این کد ها کار خودمون رو شروع میکنیم..

```
if 'hello' in statement:
    speak('hi Dear user')
    print('hi Dear user')

if 'bye' in statement or 'goodbye' in statement:
    speak('Are you satisfied with my performance?')
    answer = takeCommand()
    if 'yes' in answer:
        speak('thank you, I hope I was able to help you.')

    elif 'no' in answer:
        speak("I'm so sorry, I'm trying to get better in the future.")

    speak('your personal assistant Bimex is shutting down,Good bye')
```

```
    print('your personal assistant Bimex is shutting down,Good bye')
    break

elif 'how are you' in statement:
    speak('fine,I hope I can help you as well as any other advanced robot')
    speak('and how are you, Sir')
    print('fine,I hope I can help you as well as any other advanced robot')
    request = takeCommand()
    if 'fine' in request or 'good' in request:
        speak("It's good to know that your fine")
```

و چون دستورات مختلفی قرار است داده شود به برنامه از if ,elif ,else استفاده میکنیم...

و همانطور که می بینید برنامه قادرهست سلام و احوال پرسی کند ، خداحافظی کند و برنامه را ببند و ازکاربر درمورد کارایی اش سوال بپرسد که آیا راضی بوده یانه ؟! که با کمک حلقه های تو در تو این امکان به برنامه داده شده .

وطبق کد زیر درصورتی که از برنامه درمورد ویژگی هایش و کارهایی که قادر به انجام آنها هست سوال شود می تواند به خوبی برای کاربر توضیح دهد

```
elif 'what can you do' in statement:
    speak('I am Bimex your persoanl assistant.I am programmed to minor tasks
like opening wikipedia,youtube'
          'google chrome,gmail,amazon,your IDE and predict time,predict
weather in different cities,play music,'
          'write the text you want and show it ,search ,talk about siri ,Show
location ,world News,some information'
          'about me and my creator, you can ask me computational or
geographical questions too! and ect')
```

که به راحتی و با کمک تابع **speak** که از قبل چنین دستوراتی را بهش داده بودیم قادر به بیان این متن و معرفی خود است .

می تواند خودش را معرفی کند

```
elif 'what is your name' in statement:
    speak("im Bimex your computer assistant ")
    print("im Bimex your computer assistant ")
```

همچنین می تواند سازنده ی خودرا معرفی کند طبق کدزیر

```
elif 'who made you' in statement or 'who created you' in statement:
    speak("I was built by miss Rahmani  and she is a computer engineering
student")
    print("I was built by miss Rahmani and she is a computer engineering
student")
```

درصورتی که کاربربخواهد هوش برنامه را محک بزند و از اون بپرسد که" کیست؟ "او قادر به پاسخ گویی است طبق کد زیر

```
elif 'who am I' in statement:
    speak("If you talk then definately your human")
    print("If you talk then definately your human")
```

نام خودرا می داند

```
elif 'siri' in statement:
    speak("who is siri? i am the best")
    print("who is siri? i am the best")
```

در نتیجه با هر نام دیگری که اورا صدا کنید میبینید که واکنش نشان می دهد.

می تواند زمان دقیق را در آن واحد بگوید طبق کد زیر

```
elif 'time' in statement:
    strTime=datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"the time is {strTime}")
```

که از ماژول **datetime** برای این منظور استفاده شده و برای بیان ساعت ،دقیقه وثانیه به ترتیب پس از یکدیگر از تکه کد **strftime("%H:%M:%S")** استفاده شده و پس ازآن می تواند ساعت را بیان کند .

وطبق کد های زیر درباره سوالی که برای کاربر به وجود آمده می تواند در ویکی پدیا جست و جو کند ،یوتیوب ،گوگل کروم ،سایت آمازون و البته جیمیل را برای او نیز باز کند

```
elif 'open wikipedia' in statement or 'wikipedia' in statement:
    webbrowser.open_new_tab("https://www.wikipedia.org/")
    speak("wikipedia is open now")
    time.sleep(5)

elif 'open youtube' in statement or 'youtube' in statement:
    webbrowser.open_new_tab("https://www.youtube.com")
    speak("youtube is open now")
    time.sleep(5)

elif 'open google' in statement or 'google' in statement:
    webbrowser.open_new_tab("https://www.google.com")
    speak("Google chrome is open now")
    time.sleep(5)

elif 'open amazon' in statement :
    speak("Amazon is open now")
    webbrowser.open("https://www.amazon.com")

elif 'open gmail' in statement or 'gmail' in statement:
    webbrowser.open_new_tab("gmail.com")
    speak("Gmail is open now")
    time.sleep(5)
```

و پس از اجرای فرمان آن رابازگو کند .. که برای باز کردن این سایت ها از ماژول **webbrowser** استفاده شده و برای باز شدن یک تب جدید از **open_new_tab** استفاده کردم.

و همچنین پس ازاجرای هردستور به مدت تعیین شده در **time.sleep(5)** برنامه متوقف شود تا سایت بالا بیاید وبعد ازاین مدت دوباره به کار خودادامه دهد و از کاربر میخواهد که فرمان جدیدی به او بدهد.

برنامه همچنین می تواند اخباری که به تازگی درکشور های جهان اتفاق افتاده رابرای کاربر به نمایش بگذارد طبق کد زیر

```
elif 'news' in statement:
    news = webbrowser.open_new_tab("https://news.google.com/topstories?hl=en-US&gl=US&ceid=US:en")
    speak('Here are some new news about the countries of the world')
    time.sleep(5)
```

که برای باز کردن این سایت ها از ماژول **webbrowser** استفاده شده و برای باز شدن یک تب جدید از **open_new_tab** استفاده کردم.

وبرای پیدا کرن فایل مورد نظر کاربر درسیستم اش به کمک او بیاید طبق کد زیر

```
elif 'search' in statement:
    statement = statement.replace("search", "")
    webbrowser.open_new_tab(statement)
    speak("tell me what do you want to find?")
    time.sleep(5)
```

و مانند دیگر دستورات بالاعمل می کند.

طبق کد زیر کاربر می تواند از برنامه بخواهد متنی را برایش تایپ کند و آن رادر فایل text برایش ذخیره کند و در صورتی که کاربر بخواهد آن رابرایش نمایش دهد وهمچنین برایش بخواند.

```
elif 'take note' in statement or 'take' in statement:
    speak("ok what should I write?")
    f = open('my_file.txt', 'a+')
    f.write(takeCommand())
    f.close()

elif 'show note' in statement or 'show' in statement:
    speak("Showing note")
    old_file = open(r'my_file.txt', 'r')
    speak(old_file.read())
```

و در صورتی که فایلی وجود نداشت می تواند فایل جدیدی ایجاد کند.

که درابتدا طبق کد **f = open('my_file.txt', 'a+')** یک فایل با نام **my_file.txt** ایجاد می شود و اگر هم ایجاد شده باشد از قبل متن را با کمک دستور **a+** به ادامه آن اضافه می کند و در صورتی که **a+** را بزاریم فایل پس از اجرا شدن دوباره ، پاک نمی شود و پس از تایپ متن توسط کد **f.write(takeCommand())** به آن دستوری را میدهیم که این فایل را ببندد طبق کد **f.close()** . خواندن متنی را هم که تایپ شده دقیقا به همین صورت هست فقط به جای **a+** از مود **r** استفاده میکنیم.

برای کاربر می تواند **IDE** مور نظرش را که دراینجا **Visual Studio** در نظر گرفته شده را طبق کدزیر باز کند و برایش پروژه ای به منظور نوشتن کد ایجاد کند

```
elif 'Visual Studio' in statement or 'Visual' in statement or 'Visual Stadio'
in statement:
    command = r"G:\vsc\Microsoft VS Code\Code.exe"
    subprocess.Popen(command)
    speak("opening visual studio")
    time.sleep(5)
```

که باز شدن برنامه توسط کد subprocess.Popen(command) انجام می شود که در متغیر command آدرس مکان این برنامه را قرار میدهیم و پس از اجرای برنامه باتوجه به time.sleep(5) باتوجه به این زمان صبر می کند تا برنامه باز شود سپس به کار خود ادامه می دهد .

درمورد سوالات محاسباتی و... درسایت قدرتمد wolframalpha برایش جست و جو کند و نتیجه رابرایش طبق کدزیر بازگو کند و همچنین برای درک بهتر برای کاربر تایپ می کند

```
elif 'ask' in statement:
    speak('I can answer to computational and geographical questions and what
question do you want to ask now')
    question=takeCommand()
    app_id="R2K75H-7ELALHR35X"
    client = wolframalpha.Client('R2K75H-7ELALHR35X')
    res = client. statement (question)
    answer = next(res.results).text
    speak(answer)
    print(answer)
```

و طبق کد question=takeCommand() سوالی که برای کاربر به وجو د آمده است را میگیرد و سپس سایت را باتوجه به آیدی ای که از قبل دریافت شده است باز می کند و پس از دریافت نتایج آن را برای کاربر بازگو می کند .

با استفاده از داده های سایت openweather داده های آب و هوایی هرشهری که کاربر بخواهد را به صورت آنلاین و در لحظه برایش باز گو کند

```
elif "weather" in statement:
    api_key = "8ef61edcf1c576d65d836254e11ea420"
    base_url = "https://api.openweathermap.org/data/2.5/weather?"
    speak("what is the city name")
    city_name = takeCommand()
    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
```

```
    x = response.json()
    if x["cod"] != "404":
        y = x["main"]
        current_temperature = y["temp"]
        current_humidiy = y["humidity"]
        z = x["weather"]
        weather_description = z[0]["description"]
        speak(" Temperature in kelvin unit is " +
              str(current_temperature) +
              "\n humidity in percentage is " +
              str(current_humidiy) +
              "\n description  " +
              str(weather_description))
        print(" Temperature in kelvin unit = " +
              str(current_temperature) +
              "\n humidity (in percentage) = " +
              str(current_humidiy) +
              "\n description = " +
              str(weather_description))

    else:
        speak("City Not Found")
```

وطبق کد ، مسیرهایی به منظور گرفتن دما به کلوین ، اب وهوا وهمچنین رطوبت محیط از سایت به کمک IPE ای که از پیش با ثبت نام درسایت گرفته شده ، می دهیم و ازبرنامه میخواهیم که پس از بازگو کردن این اطلاعات آنها را تایپ کند برای کاربر تادقیق تر متوجه آن شود و تمام آِن داده ها با توجه به دستوراتی که البته باتوجه به مسیر آن ها در سایت وجود داشت دریافت می شوند و داده های دیگری را هم میشد به برنامه اضافه کرد ولی همین چند مورد برای دریافت یک اطلاعات کلی و البته جامع کافی هست برای کاربر.

و درصورتی که شهری که کاربر می خواهد پیدا نشد یک پیام مبنی بر "شهر پیدانشد " برایش نمایش داده شود طبق(City Not Found)

مکانی که کابر درخواست کرده رادر GPSپیدا کرده وروی نقشه طبق کد زیر برایش نمایش بدهد

```
elif 'location' in statement:
    speak("Google map is open now")
    webbrowser.open_new_tab("https://www.google.com/maps/@29.6390276,
    52.5241672,18z?hl=fa")
    time.sleep(6)
```

وسایت مورد نظر برای نمایش مکان دقیق توسط کد **webbrowser.open_new_tab** باز می شود و پس از مدت تعیین شده **time.sleep(6)** ادامه دستورات را از کاربر می گیرد .

برای کاربر می تواند طبق کد زیر موسیقی پخش کند

```
elif 'play music' in statement or 'music' in statement or 'play' in
statement:
    speak("music is play now")
    playsound('Conor Maynard- Someone You Loved.mp3')
    speak('Okay, here is your music! Enjoy!')
    time.sleep(6)
```

که در اینجا از ماژول **playsound** به این منظور استفاده شده که البته چون فایل صوتی را درخود فایل پروژه قرار دادم دیگه نیازی به آدرس دهی دقیق نیست و همان نام فایل کافی هست ولی اگر فایل صوتی در فایل پروژه وجود نداشت باید آدرس دقیق مکان آن فایل را قرار میدادیم به جای اسم آن و پس از مدت تعیین شده **time.sleep(6)** ادامه دستورات را از کاربر می گیرد .

می تواند سیستم رابرای کاربر طبق کد زیر خاموش کند

```
elif 'log off' in statement or 'sign out' in statement:
    speak("Ok , your pc will log off in 10 sec make sure you exit from all
applications")
    subprocess.call(["shutdown", "/l"])
```

که دراینجا از ماژول **subprocess** به این منظور استفاده شده.

والبته۱۰ ثانیه صبر می کند تاکاربر برنامه های باز روی دسکتاپ را ببندد و سپس خاموش می شود.

و اگر دستوراتی که کاربر به برنامه می دهد چیزی به غیر از دستوراتی باشد که از قبل برای برنامه تعریف شده است برنامه عذر خواهی کرده وآن دستور را در فایل unknown commands.txt طبق کدزیر ذخیره می کند

```
else:
    fi = open('unknown_commands.txt', 'a+')
    fi.write(takeCommand())
    speak("dear user i'm so sorry ")
    speak("but your command was unknown fore me.")
    fi.close()
```

و البته ابتدا فایل ما توسط fi = open('unknown_commands.txt', 'a+') ابتدا باز می شود واز مود a+ به منظور کارایی بهتر استفاده شده و پس از ذخیره دستورات و معذرت خواهی از کاربر فایل توسط fi.close() بسته می شود .

نکات مهم برای بکارگیری این برنامه

نکاتی که برای استفاده ازاین برنامه لازم هست که رعایت بشه این هست که اول از اتصال اینترنت به سیستم خودتون مطمعن بشید و در مرحله بعد ماژول هایی که به طور پیشفرض بر روی سیستم نصب نیستند رو ازطریق ترمینال و با pip install نصب کنید که این ماژول ها عبارتند از :

- speech_recognition
- pyttsx3
- pyaudio
- source
- webbrowser
- subprocess
- playsound
- wolframalpha
- requests

ماژول pyaudio رو اگر بخواید با استفاده از pip install در ترمینال نصب کنیدحتما به به eror برخورد می کنید درنتیجه ازراه زیر می تونیداین ماژول رو نصب کنید

ابتدا

pip install pipwin

و پس از نصب pipwin با دستور زیر pyaudio را نصب کنید

pipwin install pyaudio

و بعد از انجام این مراحل میکروفون سیستم خودتون رو روشن کنید و به راحتی از Bimex بخوایید که کارهای مختلف رو براتون انجام بده ☺☺

کد نهایی برنامه دستیار مجازی هوش مصنوعی Bimex

```
import speech_recognition as sr
import pyttsx3
import datetime
import webbrowser
import subprocess
from playsound import playsound
import wolframalpha
import requests
import time


print('Loading your computer assistant - Bimex')

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', 'voices[0].id')
engine.setProperty("rate", 130)


def speak(text):
    engine.say(text)
    engine.runAndWait()


def beginning():
    hour = datetime.datetime.now().hour
    if hour >= 0 and hour < 12:
        speak("Hello,good morning")
        print("Hello,Good Morning")
    elif hour >= 12 and hour < 18:
        speak("Hello,good afternoon")
        print("Hello,good afternoon")
    else:
```

```
        speak("Hello,good evening")
        print("Hello,good evening")


def takeCommand():
    r = sr.Recognizer()
    mic = sr.Microphone()
    with mic as source:
        print("Listening...")
        audio = r.listen(source)

        try:
            statement = r.recognize_google(audio, language='en-us')
            print(f"user said:{statement}\n")

        except Exception as e:
            speak("Please repeat again...")
            return "None  ,"
        return statement


speak("im Bimex your computer assistant")
beginning()

if __name__ == '__main__':

    while True:
        speak("Tell me how can I help you ?")
        statement = takeCommand().lower()
        if statement == 0:
            continue

        if 'hello' in statement:
            speak('hi Dear user')
            print('hi Dear user')

        if 'bye' in statement or 'goodbye' in statement:
            speak('Are you satisfied with my performance?')
```

```
            answer = takeCommand()
            if 'yes' in answer:
                speak('thank you, I hope I was able to help you.')

            elif 'no' in answer:
                speak("I'm so sorry, I'm trying to get better in the
future.")

            speak('your personal assistant Bimex is shutting down,Good bye')
            print('your personal assistant Bimex is shutting down,Good bye')
            break

        elif 'how are you' in statement:
            speak('fine,I hope I can help you as well as any other advanced
robot')
            speak('and how are you, Sir')
            print('fine,I hope I can help you as well as any other advanced
robot')
            request = takeCommand()
            if 'fine' in request or 'good' in request:
                speak("It's good to know that your fine")

        elif 'what can you do' in statement:
            speak('I am Aria your persoanl assistant.I am programmed to minor
tasks like opening wikipedia,'
                  'youtube,google chrome,gmail,amazon,your IDE and predict
time,predict weather in different cities'
                  'play music,write the text you want and show it
,search,Show location ,world News'
                  'some information about me and you can ask me computational
or geographical questions too! and ect')

        elif 'what is your name' in statement:
            speak("im Bimex your computer assistant ")
            print("im Bimex your computer assistant ")

        elif 'who made you' in statement or 'who created you' in statement:
            speak("I was built by miss Rahmani")
```

```
            print("I was built by miss Rahmani")

        elif 'who am I' in statement:
            speak("If you talk then definately your human")
            print("If you talk then definately your human")

        elif 'siri' in statement:
            speak("who is siri? i am the best")
            print("who is siri? i am the best")

        elif 'time' in statement:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"the time is {strTime}")

        elif 'open wikipedia' in statement or 'wikipedia' in statement:
            webbrowser.open_new_tab("https://www.wikipedia.org/")
            speak("wikipedia is open now")
            time.sleep(5)

        elif 'open youtube' in statement or 'youtube' in statement:
            webbrowser.open_new_tab("https://www.youtube.com")
            speak("youtube is open now")
            time.sleep(5)

        elif 'open google' in statement or 'google' in statement:
            webbrowser.open_new_tab("https://www.google.com")
            speak("Google chrome is open now")
            time.sleep(5)

        elif 'open amazon' in statement:
            speak("Amazon is open now")
            webbrowser.open_new_tab("https://www.amazon.com")
            time.sleep(5)

        elif 'open gmail' in statement or 'gmail' in statement:
            webbrowser.open_new_tab("https://www.gmail.com")
            speak("Gmail is open now")
            time.sleep(5)
```

```
        elif 'news' in statement:
            news =
webbrowser.open_new_tab("https://news.google.com/topstories?hl=en-
US&gl=US&ceid=US:en")
            speak('Here are some new news about the countries of the world')
            time.sleep(5)

        elif 'search' in statement:
            statement = statement.replace("search", "")
            webbrowser.open_new_tab(statement)
            speak(" tell me what do you want to find?")
            time.sleep(5)

        elif 'take note' in statement or 'take' in statement:
            speak("ok what should I write?")
            f = open('my_file.txt', 'a+')
            f.write(takeCommand())
            f.close()

        elif 'show note' in statement or 'show' in statement:
            speak("Showing note")
            old_file = open(r'my_file.txt', 'r')
            speak(old_file.read())

        elif 'Visual Studio' in statement or 'V S code' in statement or
'usual' in statement:
            command = r"G:\vsc\Microsoft VS Code\Code.exe"
            subprocess.Popen(command)
            speak("visual studio is open now")
            time.sleep(5)

        elif 'ask' in statement:
            speak('I can answer to computational and geographical questions
and what question do you want to ask now')
            question = takeCommand()
            app_id = "R2K75H-7ELALHR35X"
            client = wolframalpha.Client('R2K75H-7ELALHR35X')
```

```
            res = client. statement (question)
            answer = next(res.results).text
            speak(answer)
            print(answer)

        elif "weather" in statement:
            api_key = "8ef61edcf1c576d65d836254e11ea420"
            base_url = "https://api.openweathermap.org/data/2.5/weather?"
            speak("what is the city name")
            city_name = takeCommand()
            complete_url = base_url + "appid=" + api_key + "&q=" + city_name
            response = requests.get(complete_url)
            x = response.json()
            if x["cod"] != "404":
                y = x["main"]
                current_temperature = y["temp"]
                current_humidiy = y["humidity"]
                z = x["weather"]
                weather_description = z[0]["description"]
                speak(" Temperature in kelvin unit is " +
                      str(current_temperature) +
                      "\n humidity in percentage is " +
                      str(current_humidiy) +
                      "\n description  " +
                      str(weather_description))
                print(" Temperature in kelvin unit = " +
                      str(current_temperature) +
                      "\n humidity (in percentage) = " +
                      str(current_humidiy) +
                      "\n description = " +
                      str(weather_description))

            else:
                speak("City Not Found")

        elif 'location' in statement:
            speak("Google map is open now")
```

```
webbrowser.open_new_tab("https://www.google.com/maps/@29.6390276,52.5241672,1
8z?hl=fa")
            time.sleep(6)

        elif 'play music' in statement or 'music' in statement or 'play' in
statement:
            speak("music is play now")
            playsound('Conor Maynard- Someone You Loved.mp3')
            speak('Okay, here is your music! Enjoy!')
            time.sleep(6)

        elif 'log off' in statement or 'sign out' in statement:
            speak("Ok , your pc will log off in 10 sec make sure you exit
from all applications")
            subprocess.call(["shutdown", "/l"])

        else:
            fi = open('unknown_commands.txt', 'a+')
            fi.write(takeCommand())
            speak("dear user i'm so sorry ")
            speak("but your command was unknown fore me.")
            fi.close()

time.sleep(3)
```