مقدمه ای برنامه نویسی پایتون

۱-۱- علوم تجربی و علوم اطلاعات و آنالیز

علوم تجربی و علوم اطلاعات و داده دو حوزه مختلف در علوم هستند با تمرکز و کاربردهای متفاوت. در اینجا تفاوت‌های اصلی بین آنها را بررسی می‌کنیم:

- علوم تجربی (علوم طبیعی)

علوم تجربی به تحقیقات و مطالعه‌هایی از طبیعت و جهان طبیعی می‌پردازند. این حوزه شامل رشته‌هایی مانند فیزیک، شیمی، زیست‌شناسی، علوم زمین، و علوم اجتماعی به شیوه‌ای تجربی است که با استفاده از مشاهدات، آزمایش‌ها و تجزیه و تحلیل‌های دقیق، قوانین و نظریه‌های جدید را برای توضیح پدیده‌های طبیعی ارائه می‌دهند. هدف اصلی علوم تجربی درک و تبیین قوانین طبیعی و توسعه نظریه‌هایی است که به طور دقیق رفتار طبیعت را توصیف کنند.

- علوم اطلاعات و داده

علوم اطلاعات و داده مدیریت، تحلیل و بهره‌برداری از داده‌ها برای انتقال اطلاعات و دانش می‌پردازند. این حوزه شامل رشته‌هایی مانند علوم کامپیوتر، مهندسی نرم‌افزار، علوم داده و هوش مصنوعی است که با استفاده از روش‌های مختلف مانند مدل‌سازی، الگوریتم‌های ماشینی، و پردازش داده‌های بزرگ(Big Data) ، اطلاعات مفید و الهام‌بخش را از داده‌ها استخراج می‌کنند. هدف علوم اطلاعات و داده ارتقاء فرآیندها و تصمیم‌گیری‌ها از طریق استفاده از داده‌ها و اطلاعات به صورت هوشمند و تحلیلی است.

بنابراین، در حالی که علوم تجربی به بررسی و توضیح پدیده‌های طبیعت می‌پردازند، علوم اطلاعات و داده بر روی تحلیل و آنالیز داده‌ها با هدف استخراج اطلاعات مفید تمرکز دارند. به عبارت دیگر وظیفه افرادی که در دنیای علوم تجربی فعالیت می کنند جمع آوری داده ها می باشد ولی افرادی که در نیای علوم اطلاعات فعالیت دارند با انالیز داده های ثبت شده به دنبال اطلاعات و دانش جدید هستند.

۱-۲- تفاوت دنیای برنامه نویسی و بیولوژی

تفاوت‌های بین دنیای برنامه‌نویسی و بیولوژی بسیار جالب و متنوع است. در ادامه، تفاوت‌های اصلی این دو حوزه را تشریح می‌دهم:

- **از دیدگاه طبیعت علمی و موضوع**

برنامه‌نویسی: دنیای برنامه‌نویسی معمولاً در زمینه علوم کامپیوتر و مهندسی نرم‌افزار قرار دارد. این حوزه بر روی توسعه نرم‌افزارها، طراحی الگوریتم‌ها، برنامه‌نویسی وب، و برنامه‌نویسی سیستم تمرکز دارد و آنها الگوریتم‌ها و کدهای را برای حل مسائل و ایجاد برنامه‌های کاربردی مختلف ایجاد می‌کنند.

بیولوژی: از سوی دیگر، بیولوژی به مطالعه ساختار، عملکرد و تکامل سلول‌ها، افراد، اجتماعات و سیستم‌های زنده می‌پردازد. این شاخه علمی شامل زیرمجموعه‌های مختلفی مانند بیوشیمی، ژنتیک، علوم زیستی مولکولی، اکولوژی و زیست‌شناسی تکاملی است.

- **ازدیدگاه روش‌های کار**

برنامه‌نویسی: در برنامه‌نویسی، توسعه‌دهندگان از زبان‌های برنامه‌نویسی مختلف مانندC ، Java، Pythonو... استفاده می‌کنند. با استفاده از زبان‌های برنامه‌نویسی مختلف، الگوریتم‌ها و ساختارهای داده، و پلتفرم‌های نرم‌افزاری، نرم‌افزارها و سیستم‌های کاربردی را ایجاد می‌کنند.

بیولوژی: در زمینه بیولوژی، محققان از روش‌های متنوعی مانند آزمایش‌های آزمایشگاهی، تحلیل‌های زیستی ، و بررسی‌های میدانی برای مطالعه سیستم‌های زیستی استفاده می‌کنند. به عبارت اکثرا در این حوزه فعالیت های آزمایشگاهی از اولویت بالایی برخوردار است.

- **کاربردها و تأثیرات**

برنامه‌نویسی: در دنیای برنامه‌نویسی، تولید نرم‌افزارهای کاربردی، سیستم‌های عامل، برنامه‌های وب، بازی‌های رایانه‌ای، و نرم‌افزارهای هوش مصنوعی از جمله کاربردهای متداول هستند.

بیولوژی: بیولوژی کاربردهای متنوعی از جمله درمان بیماری‌ها، بهبود محیط زیست، تولید محصولات کشاورزی و دامپروری، تولید دارو، تکنولوژی ژنتیک، و مطالعه تکامل دارد.

در نهایت، برنامه‌نویسی و بیولوژی دو حوزه جذاب و مهم علمی هستند که هر کدام به نحوی برای پیشرفت انسانی و بهبود جامعه اساسی هستند، اما هر یک دارای رویکردها، روش‌ها، و کاربردهای خاص خود هستند. یک بیولوژیست از تجهیزات با قیمت بالا و هزینه بالا استفاده می کند و اگر جایی بخواهید این موارد رو یاد بگیرید خیلی به شما اجازه کار به تجهیزات نمی دهند و بخواهید خود شما کار کنید با توجه به هزینه بسیار بالا جای برای آزمون خطا ندارید ولی در برنامه نویسی تقریبا هزینه وجود ندارد و می تواند با آزمون خطا های مختلف به جواب مورد نظر دست یافت. همچنین تمرکز اصلی بیولوژیست ها تولید داده های مختلف است ولی برنامه نویس تمرکز اصلی بر روی آنالیز داده های موجود است. علاوه بر این موارد، برنامه نویسی یک فرآیند پیوسته است و اگر مبحثی را یاد نگیرید در بقیه موارد هم به مشکل خواهید خورد ولی در زیست شناسی شاید فردی یک تکنیک آزمایشگاهی را به خوبی بلد باشد ولی تکنیک دیگری را بلد نیست و مشکلی برایش وجود ندارد. به عبارت دیگر برنامه نویسی با توجه به پیوسته بودن و مهارتی بودن نیازمند تمرین بسیار زیاد است.

هر دو حوزه از نظر تکنولوژی و علمی اهمیت بسیاری دارند و هر کدام دارای نقش مهمی در توسعه جامعه و پیشرفت انسانی هستند. در حال حاضر، تلاش برای ادغام این دو حوزه نیز در حال افزایش است، به عنوان مثال در زمینه بیوانفورماتیک که اجتماع دو حوزه بیولوژی و علوم کامپیوتر است.

۱-۳- چرا یک بیولوزیست نیاز به یادگیری برنامه نویسی دارد؟

برای پاسخ به این سوال لازم است ابتدا به هرم دانش بشری (DIKW) آشنا شوید. این هرم به مدل‌هایی اشاره دارد که برای ارتباط ساختاری بین داده، اطلاعات، دانش (علم) و حکمت (خرد یا فرزانگیبا بینش) به کار می‌روند. معمولاً هر لایه با تعریف الگویی بر روی لایه پایین‌تر تعریف می‌گردد.

هرمDIKW ، یک مدل مفهومی است که برای توصیف مراحل تبدیل داده‌ها به دانش و حکمت در یک سازمان یا محیط دیگر استفاده می‌شود. این مدل به طور سلسله مراتبی نشان می‌دهد که چگونه داده‌ها ابتدا تبدیل به اطلاعات، سپس به دانش و در نهایت به بینش می‌شوند. هرم دانش یک مدل ساده شده و قابل‌فهم برای نشان‌دادن رابطه بین داده (Data)، اطلاعات (information)، دانش (knowledge) و خرد (wisdom) است. DIKW در واقع سرواژه و مخفف این چهار کلمه است.

هرم دانش یک مدل است. مدل‌ها در واقع یک تمثال یا نمایش ساده از واقعیت هستند. یک مدل به ما کمک می‌کنند تا متوجه بشویم چیزها چطور کار می‌کنند و اجزاء آن‌ها چه ارتباط منطقی با هم دارند.در اینجا هرم دانش به ما کمک می‌کند تا فرایند تبدیل داده به دانش و خرد را بهتر درک کنیم. هرم دانش یک مدل سلسه‌مراتبی (hierarchy) است؛ یعنی هرکدام از بخش‌های آن نسبت به بقیه در جایی بالاتر یا پایین‌تر قرار می‌گیرند و ارتباط آنها با هم بر همین اساس تعریف می‌شود .هرم دانش به چهار بخش یا مرحله تقسیم می‌شود :داده، اطلاعات، دانش، خرد یا حکمت. کار در مرحله اول با داده خام شروع می‌شود و مرحله‌های بعد با انجام کاری روی داده آن را به اطلاعات، دانش، و خرد تبدیل می‌کنیم. هرکدام از این مراحل پیش نیازی برای رسیدن به مرحله بعد هستند.

داده: اگر به تعریف رسمی مراجعه کنیم داده‌ها شامل حقایق(facts) ، آمار، کمیت یا کیفیت چیزی هستند؛ اما به‌طورکلی داده می‌تواند شامل هر چیزی باشد (از تراکنش‌های بانکی گفته تا دمای هوا). داده اولین مرحله، نقطه آغاز و پایه هرم دانش است. در واقع بدون داده هیچ کاری انجام نمی‌شود از طرفی داده خام به‌خودی‌خود هیچ معنا و مفهومی را منتقل نمی‌کند. برای درک داده باید آن را در زمینه (context) خاص خودش بررسی کرد. به این عدد دقت کنید ۱۴۰۱۱۲۲۰ این عدد مثالی از یک داده خام است. به‌خودی‌خود مفهومی را منتقل نمی‌کند؛ اما اگر آن را در زمینه مناسب مثلاً تقویم سالیانه! بگنجانیم تاریخ ۲۰/۱۲/۱۴۰۱

اطلاعات: در پله دوم هرم به اطلاعات می‌رسیم. قبلاً گفتیم که برای درک داده باید آن را در زمینه (context) خودش بررسی کرد. این زمینه می‌تواند پرسیدن سؤال‌هایی مثل چه چیزی؟ کجا؟ و کی اتفاق افتاده باشد. این سؤال‌ها به ما کمک می‌کند داده‌های بی‌معنی را به اطلاعات بامعنی تبدیل کنیم. اطلاعات حاصل پردازش داده خام است. داده نامنظم و مملو از خطا است؛ اما اطلاعات دسته‌بندی شده، صحیح، قابل‌فهم، و دارای چهارچوب و زمان‌بندی است. بر عکس داده خام، اطلاعات معنا و مفهومی را به مخاطب منتقل می‌کند.

دانش: دانش سومین مرحله در هرم DIWK است. دانش را می‌توان جمع اطلاعات بعلاوه تجربه و تخصص دانست. در واقع دانش درک رابطه بین اطلاعات مختلف است. دانش به سؤالاتی که درباره چگونگی یک موضوع است پاسخ می‌دهد. سؤالاتی مثل: اطلاعات مختلف چگونه به هم ارتباط دارند و این ارتباط چطور ارزش بیشتری را ایجاد می‌کند؟ چگونه از اطلاعات برای رسیدن به اهداف خود استفاده کنیم؟ اگر به اطلاعات صرفاً به‌عنوان توصیفی از حقایق نگاه نکنیم و یاد بگیریم چطور برای رسیدن به اهدافمان از آنها استفاده کنیم اطلاعات را به دانش تبدیل کرده‌ایم.

حکمت یا خرد: خرد آخرین مرحله و در رأس هرم دانش است. خرد؛ یعنی دانش بعلاوه عمل، استفاده از دانش برای تصمیم‌گیری در مورد آینده ما را وارد قلمرو خرد می‌کند. خرد به ما می‌گوید بهتر است چه کاری را انجام بدهیم یا انجام ندهیم! در واقع خرد دانش کاربردی و عملی است.

داده و اطلاعات درباره گذشته هستند؛ اما دانش و خرد با زمان حال و آینده گره‌خورده‌اند .داده درباره مشاهده پدیده ها و اتفاقات است؛ اطلاعات آنها را توصیف می‌کند، دانش درک روابط بین اتفاقات را میسر می‌کند و خرد به ما می‌گوید چه کاری باید انجام دهیم.

برای اینکه بهتر مفهوم تبدیل داده به خرد را متوجه شویم از یک مثال با داده‌ای ساده استفاده می‌کنیم.

داده: به جدول داده‌های زیر نگاه کنید. این جدول دو سطر و سه ستون دارد. اما معنی و مفهوم خاصی را نمی‌رساند.

قرمز	20	50
آبی	30	50

تبدیل داده به اطلاعات: بیایید داده جدول بالا را در کانتکس خودش قرار بدهیم. این جدول داده‌های مربوط به دو ماشین را نشان می‌دهد. ستون اول از چپ مسافت طی شده بر حسب کیلومتر، ستون دوم زمانی است که برای طی مسافت صرف شده و ستون سوم رنگ آن خودرو را نشان می‌دهد. با دانستن این داده‌های جدید جدول زیر را می‌توانیم ایجاد کنیم.

قرمز	20min	50km
آبی	30min	50km

استفاده از دانش: فرض کنید اطلاعات دیگری هم به دست آورده‌ایم، داده‌های بالا برای مسیری است که ما می‌شناسیم (مثلاً تهران – کرج) با استفاده از تجربه خودمان می‌توانیم بفهمیم که راننده خودرو قرمز با سرعتی بیش از سرعت مجاز در این مسیر رانندگی کرده است (۱۱۰ کیلومتر برای بیشینه سرعت مجاز) درحالی‌که راننده آبی از سرعت متوسط ۱۰۰ کیلومتر بر ساعت تجاوز نکرده است.

خرد یا حکمت: خرد؛ یعنی دانش بعلاوه عمل، شما اگر کارگذار یک شرکت بیمه بودید باتوجه‌به اطلاعات بالا چه می‌کردید؟ قطعاً در محاسبه بیمه سالیانه برای راننده خودرو قرمز مبلغ بالاتری را در نظر می‌گرفتید؛ چون احتمال تصادف راننده‌ای که بالاتر از سرعت مجاز می‌راند بیشتر است.

مثالی از هرم DIKW در رابطه با بیولوژی ممکن است به صورت زیر باشد:

داده(Data): سری‌های DNA متشکل از نوکلئوتیدها (آدنین، تیمین، گوانین، سیتوزین).

اطلاعات(Information) : توالی خاصی از نوکلئوتیدها که به عنوان یک ژن مشخص شده و اطلاعات درباره ساختار و وظیفه آن در سلول را فراهم می‌کند.

دانش(Knowledge) : فهم اینکه ژن خاصی چگونه ترجمه می‌شود و به چه نحوی بر روی فعالیت‌های سلولی تأثیر می‌گذارد، به عنوان مثال، ژنومیک و پروتئومیک.

بینش(Wisdom) : توانایی انتخاب و استفاده از این دانش برای توسعه درمان‌های جدید، دستیابی به پیشرفت‌های درمانی و شناخت بهتر از بیماری‌ها و راه‌های پیشگیری از آنها، به طوری که به بهبود سلامت و کیفیت زندگی انسان‌ها کمک می‌کند.

در این مثال، داده‌ها (سری‌های DNA) به اطلاعات (توالی خاصی از نوکلئوتیدها) تبدیل می‌شوند. سپس، با دانش (فهم اینکه ژن خاصی چگونه ترجمه می‌شود و به چه نحوی بر روی فعالیت‌های سلولی تأثیر می‌گذارد)، می‌توانیم به بینش (استفاده از این دانش برای توسعه درمان‌های جدید و بهبود سلامت انسان‌ها) برسیم.

به طور خلاصه، داده‌ها از خودی خود معنای خاصی ندارند، اما با تحلیل و تفسیر آن‌ها، می‌توان اطلاعاتی را از آن‌ها استخراج کرد. برای این کار، ما به ابزارهایی مانند آمار توصیفی و تحلیلی نیاز داریم. بیولوژیست‌ها مهارت‌های خوبی در تولید داده‌ها و تبدیل آن‌ها به اطلاعات دارند. این اطلاعات می‌توانند در سطح بالاتری به دانش منجر شوند که به ما امکان تصمیم‌گیری و اجرای اقدامات می‌دهد. در نهایت، سطح حکمت یا Wisdom، ما را به انجام اقدامات و تصمیم‌گیری‌های مفید و معنادار می‌رساند. اما مرز بین این سطوح در بسیاری از موارد مبهم است.آیا می توانیم از این داده ها یک سطح بالاتری از اطلاعات را استخراج کنیم. معمولا بیولوژیست ها در این دو کار مهارت های خوبی دارند ۱ –تولید داده ها (مانند کاشت سلولو نمونه برداری از بافت ها و ... (که در ژن ایران همش را می توانید دقیق یاد بگیرید)۲- تبدیل داده ها به اطلاعات در سطحی که نیاز به آمار توصیفی و تحلیلی پایه. در سطح بالاتر از این با کنار هم قرار دادن این اطلاعات می توان دانشی بدست آورد به عنوان مثال بدست آورد که بعضی از افراد کلسترول بالایی دارند و این منجر به چاقی یا فشار خون بالا می شد. یعنی یک زیر گروه براساس این پارامتر ها می توانیم استخراج کنیم.

بعد از آشنایی با این هرم می توان چنین بیان کرد وضعیت فعلی این هرم از حالت نرمال خارج شده است و دلیل این وضعیت افزایش تعداد داده های موجود نسبت به سایر سطوح این هرم است. این وضعیت ناشی از توسعه تکنولوژی و ابزار های جمع آوری داده است. در مقابل سایر سطوج نسبت به سطح داده رشد نکرده است. به همین دلیل نیاز به ابزار ها و افرادی متخصص هستیم که بتوان با استفاده از این ابزار ها و تکنیک ها بتوانند داده ها را به اطلاعات، دانش و حکمت تبدیل کنند.
تحلیل و تفسیر داده‌ها از جمله فرآیندهای کلیدی است که در فرآیند تبدیل داده‌ها به دانش نقش اساسی دارد. در این فرآیند، داده‌های اولیه که معمولاً به صورت بی‌ساختار و بدون معنی هستند، تحت تحلیل قرار می‌گیرند و به اطلاعات تبدیل می‌شوند. این اطلاعات با ترتیب دادن، خلاصه‌سازی، دسته‌بندی و استخراج الگوها، به یک ساختار منظم و معنی‌دار تبدیل می‌شوند که به انسان‌ها قابل فهم باشد. با انجام تحلیل و تفسیر داده‌ها، دانش از داده‌ها به دست می‌آید. این دانش نتیجه فهم بهتر از الگوها، روابط و تغییرات در داده‌ها است که از طریق تحلیل آن‌ها به دست می‌آید. با داشتن دانش، افراد می‌توانند ارتباطات پنهان و الگوهای زیربنایی در داده‌ها را شناسایی کرده و تصمیم‌گیری‌های بهتری انجام دهند. بنابراین، تحلیل و تفسیر داده‌ها نه تنها به فرآیند تبدیل داده‌های بی‌ساختار به اطلاعات ساختارمند کمک می‌کند، بلکه اطلاعات به دانش تبدیل می‌شود که به افراد امکان می‌دهد روابط پنهان و الگوهای موجود در داده‌ها را به دقت شناسایی کنند و استفاده‌های مفیدی از آن‌ها را درک کنند. . هرم دانش در عین سادگی و سرراست بودن یک اصل مهم را نشان می‌دهد راه رسیدن به خرد از طریق داده انجام می‌شود. به طور خلاصه از دلایل اینکه بیولوژیست‌ها نیاز به یادگیری محاسبات عددی، برنامه‌های محاسباتی و برنامه‌نویسی دارند می توان به موارد زیر اشاره کرد:
تحلیل داده‌ها: بیولوژیست‌ها برای تحلیل داده‌های بزرگ و پیچیده که در آزمایشگاه‌ها و مطالعات میدانی بدست می‌آیند، نیاز به استفاده از روش‌های محاسباتی دارند. این شامل تجزیه و تحلیل داده‌های ژنتیکی، ساختارهای پروتئینی، اطلاعات اکولوژیک و...
مدل‌سازی ریاضی: در بسیاری از زمینه‌های بیولوژی، ایجاد مدل‌های ریاضی برای توصیف و پیش‌بینی فرایندها و الگوهای زیستی بسیار مهم است. این امر نیازمند دانش در زمینه محاسبات عددی و برنامه‌نویسی است تا بتوانند مدل‌های موثری ایجاد کنند.
پردازش داده‌های بزرگ: با پیشرفت تکنولوژی، حجم داده‌های زیست‌شناسی به صورت چشمگیری افزایش یافته است. برای مدیریت، تحلیل و استفاده از این داده‌ها، بیولوژیست‌ها نیازمند آشنایی با ابزارها و روش‌های محاسباتی هستند.

توسعه نرم‌افزارهای کاربردی : برای توسعه نرم‌افزارهای کاربردی برای بیولوژی، از جمله نرم‌افزارهای مدل‌سازی، آنالیز داده‌ها، و شبیه‌سازی‌های زیستی، نیاز به مهارت در برنامه‌نویسی و محاسبات عددی دارند.
پیشرفت در تحقیقات: استفاده از روش‌های محاسباتی و برنامه‌نویسی در تحقیقات بیولوژی، می‌تواند به بیولوژیست‌ها کمک کند تا به نتایج بهتری برسند و مسائل پیچیده‌تر را حل کنند.

به طور کلی، دانش در زمینه محاسبات عددی و برنامه‌نویسی برای بیولوژیست‌ها ابزارهای قدرتمندی را فراهم می‌کند که به آنها کمک می‌کند تا در تحقیقات خود موثرتر عمل کنند و به دست‌آوردهای بهتری دست یابند.

با توجه به موارد گفته شده لزوم یاد گیری ابزار ها و تکنیک ها محاسباتی به خصوص برنامه نویسی آشکار شد. البته باید به این نکته تووجه کنید که در انتخاب ابزار های مختلف حتما باید مسیر و هدف مشخص شود و بر اساس این مسیر و هدف ابزار و تکنیک را انتخاب کنید. همچنین با توجه به رشد علوم و بین رشته ای شدن توصیه می شود که نرم افزار های اکسل، SPSS، prism و زبان های برنامه نویسی پایتون و R را به عنوان اولویت برای یاد گیری قرار دهید.

در این جزوه هدف یادگیری زبان برنامه نویسی پایتون است که در ادامه به مفاهیم و بخش های مختلف آن آشاره می کنیم.

۱-۴- زبان برنامه نویسی پایتون

زبان‌های برنامه‌نویسی، مجموعه‌ای از قواعد و دستورالعمل‌ها هستند که توسط برنامه‌نویسان برای نوشتن برنامه‌های کامپیوتری استفاده می‌شوند. این زبان‌ها به ماشین‌ها و کامپیوترها دستورات را می‌دهند که چگونه عمل کنند و وظایف مختلف را اجرا کنند. زبان‌های برنامه‌نویسی متنوعی وجود دارند که هر کدام ویژگی‌ها، قواعد و استفاده‌های خاص خود را دارند. به عنوان مثال:

Python
Java
C++
JavaScript
C#
Swift
Ruby
PHP
Go
Rust

هر کدام از این زبان‌ها ویژگی‌ها و کاربردهای مختلفی دارند و بسته به نیازها و شرایط مورد استفاده قرار می گیرند. همانطور که از عنوان این بخش مشخص است، این عبارت از سه کلمه زبان، برنامه نویسی و پایتون تشکیل شده است که در ادامه هر از این اجزا را بررسی می کنیم.

۱-۴-۱- زبان های کامپیوتری

زبان‌ها، به عنوان ابزاری برای ارتباط، انتقال اطلاعات، و انتقال دستورات بین افراد مورد استفاده قرار می‌گیرند. هر زبان از یک سری حروف الفبا و گرامر و دستورالعمل هایی تشکیل شده است که با یاد گیری این دستورالعمل ها می توان به آن زبان تسلط پیدا کرد. ولی استفاده از زبان و برقراری ارتباط به افراد نیاز مند تمرین و کسب مهارت می باشد و فقط با مطالعه و گذراندن کلاس نمی توان با افراد مختلف ارتباط برقرار کرد. پس به طور خلاصه می توان چنین بیان کرد که یاد گیری زبان یک مهارت می باشد.

در زمینه ی زبان های کامپیوتر نیز، زبان ها این امکان می‌دهند با کامپیوترها ارتباط برقرار کنند و دستورات لازم برای انجام کارهای مختلف را به آنها بدهند. با توجه به عملکرد و نحوه کار کامپیوتر ها برعکس انسان ها، هیچ درکی از مفهوم ندارد و برای برقراری ارتباط و انجام کارهای مختلف توسط کامپیوتر، باید دستورات را به صورت دقیق و گام به گام به کامپیوتر ها بدهید. زبان های کامپیوتر را معمولاً به سه دسته‌ی مختلف بر اساس سطح انتزاع و فاصله از زبان ماشینی دسته‌بندی می‌کنند:

- **زبان‌های سطح پایین(Low-Level Languages)**

این زبان‌ها نزدیکترین به زبان ماشینی هستند. این دسته زبان‌ها برای برنامه‌نویسی سیستم‌های سخت‌افزاری مانند درایورها و سیستم‌عامل‌ها استفاده می‌شود این زبان ها برای کامپیوتر قابل درک تر هستند ولی برای انسان قابلیت درک سخت‌تری داند. همچنین نیاز به مدیریت حافظه دارند. زبان های سطح پائین سرعت اجرای بسیار بالا است.. مثال‌ها: زبان ماشین، زبان اسمبلی

- **زبان‌های سطح متوسط(Mid-Level Languages)**

این زبان‌ها در میانه بین زبان‌های سطح پایین و زبان‌های سطح بالا قرار دارند. مانند زبان‌های سطح بالا قابلیت تجرید یا انتزاع دارند. برای کامپیوتر قابل درک نیستند و باید به زبان ماشین تبدیل شوند ولی برای انسان قابل درک هستند و شبیه زبان انگلیسی است. مانند زبان‌های سطح پایین می‌توانند حافظه را مدیریت کنند و در صورت لزوم نیز مانند زبان‌های سطح بالا می‌توان به مدیریت حافظه کاری نداشت. نسبت به زبان‌های سطح بالا سرعت بالاتری دارند اما نسبت به زبان‌های سطح پایین سرعت پایین‌تری دارند. زبان های C و ++C را می توان جر این دسته از زبان ها در نظر گرفت.

- **زبان‌های سطح بالا(High-Level Languages)**

این زبان‌ها از سطح بالاتری از انتزاع برخوردار هستند و به برنامه‌نویس امکان می‌دهند که با تمرکز بر روی الگوریتم‌ها و منطق برنامه‌نویسی، بدون نگرانی از جزئیات سطح پایین، کد بنویسد. زبان‌های سطح بالا به زبان انگلیسی بسیار نزدیک هستند. بنابراین این زبان‌ها برای انسان قابل درک هستند اما برای کامپیوتر قابل درک نیستند. زبان‌های سطح بالا باید توسط یک نرم‌افزار میانی یا کامپایلر به زبان سطح پایین یعنی «زبان ماشین» تبدیل شوند تا برای کامپیوتر نیز قابل درک باشند. سرعت این زبان‌ها نسبت به زبان اسمبلی و زبان ماشین پایین‌تر هستند و به دلیل ترجمه به زبان سطح پایین، باعث می‌شود سرعتشان پایین‌تر باشد. ایده‌ی زبان‌های سطح پایین به این دلیل بود که انسان‌ها به راحتی بتوانند برنامه نویسی کنند و کار دشواری مانند زبان اسمبلی و زبان ماشین نداشته باشند. زبان های برنامه نویسی پایتون، R و ... جز این دسته از زبان ها کامپیوتر هستند. زبان پایتون هم به این دلیل که جز این دسته می باشدف جز ساده ترین زبان ها در یاد گیری و استفاده می باشد و یکی از دلایل محبوبت زبان برنامه نویسی پایتون می باشد.

همانطور که بیان شد یاد گیری زبان یک مهارت می باشد. به عنوان مثال شما با یاد گیری اجزای یک ماشین آیا می توانید بگویید که رانندگی بلد هستید؟ خیر. به این علت است که رانندگی یک مهارت است و تنها راه یادگیری تمرین بسیار می باشد. زبان های کامپیوتر به خصوص برنامه نویسی نیز چنین است، با یاد گیری دستورالعمل های یک زبان شما فقط می توانید به کد نویسی تسلط پیدا کنید و برای یادگیری برنامه نویسی باید تمرین بسیار کنید.

۱-۴-۲- برنامه نویسی

برنامه‌نویسی به فعالیت طراحی و توسعه نرم‌افزارها با استفاده از زبان‌های برنامه‌نویسی اشاره دارد. این فعالیت شامل تدوین الگوریتم‌ها، نوشتن کدها، تست و اجرای برنامه‌ها و در نهایت پشتیبانی از آنها می‌شود. برنامه‌نویسان در فرآیند برنامه‌نویسی از تکنیک‌ها و اصول مختلفی استفاده می‌کنند تا نرم‌افزارهایی را توسعه دهند که مشکلات موجود را حل کرده و نیازهای کاربران را برطرف کنند. از طرفی، کد نویسی یکی از فعالیت‌های اصلی در فرآیند برنامه‌نویسی است. در واقع، کد نویسی به فرآیند ترجمه الگوریتم‌ها و ایده‌ها به کدهای ماشینی قابل اجرا اشاره دارد. بنابراین، کد نویسی فقط یکی از مراحل فرآیند برنامه‌نویسی است و شامل نوشتن دستورات و ساختارهای مورد نیاز برای تعامل با کامپیوتر است. به طور خلاصه، برنامه‌نویسی فرآیند کلی طراحی و توسعه نرم‌افزار است که

شامل مراحلی از جمله تدوین الگوریتم‌ها، طراحی ساختار برنامه، کد نویسی، تست و اجرا و پشتیبانی می‌شود، در حالی که کد نویسی فقط یکی از مراحل این فرآیند است که برای ترجمه الگوریتم‌ها و ایده‌های برنامه‌نویس به زبان ماشینی استفاده می‌شود.

- **برنامه نویسی**
  - درک صورت مسئله
  - حل مسئله
  - الگوریتم

به شکل کلی، برنامه‌نویسی یک فرآیند است که شامل چند مرحله مهم است که در زیر توضیح داده شده‌اند:

درک صورت مسئله: در این مرحله، برنامه‌نویس باید مسئله‌ی مورد نظر را به دقت درک کند. این شامل فهم دقیق اهداف و نیازهای برنامه، ورودی‌ها و خروجی‌های مورد انتظار، محیط کاربری و دیگر موارد مرتبط است. برای این منظور، برنامه‌نویس می‌تواند با مشتری یا کارفرما مصاحبه کند، مستندات مربوط به پروژه را مطالعه کند و یا تحقیقاتی در این زمینه انجام دهد.

حل مسئله: پس از درک دقیق مسئله، برنامه‌نویس باید راه‌حلی مناسب برای آن مسئله پیدا کند. این شامل تحلیل و ارزیابی روش‌های مختلف حل مسئله، انتخاب روش مناسب، و طراحی راه‌حلی که نیازها و معیارهای مورد نظر را برآورده کند، می‌شود.

طراحی الگوریتم: الگوریتم مرحله‌ی مهمی در فرآیند برنامه‌نویسی است. الگوریتم به معنای یک مجموعه از گام‌ها و دستورات است که به ترتیب مشخص مسئله را حل می‌کنند. برنامه‌نویس باید الگوریتمی را طراحی کند که مسئله را به شکلی کامل و صحیح حل کند. الگوریتم ممکن است به صورت نوشتاری (مانند روشن‌نویسی) یا به صورت نمودارهای فلوچارت یا نمودارهای دیگر توصیف شود.

در کل، این مراحل متمرکز بر فهم مسئله، تعیین راه‌حل مناسب و تبدیل آن به یک الگوریتم منطقی و قابل اجرا هستند که به برنامه‌نویس کمک می‌کند تا نرم‌افزار مورد نظر را به صورت کارآمد و صحیح ایجاد کند.

- **کد نویسی >>>** ترجمه الگوریتم به زبان کامپیوتر

کد نویسی به فعالیت ترجمه‌ی الگوریتم‌ها و ایده‌های برنامه‌نویس به زبان برنامه‌نویسی مورد استفاده توسط کامپیوترها اشاره دارد. به طور دقیق‌تر، کد نویسی فرآیند ترجمه‌ی دستورات و ساختارهای مرتبط با حل مسئله به زبان ماشینی است که توسط کامپیوترها قابل اجراست. برنامه‌نویسان به وسیله کد نویسی، الگوریتم‌ها و روش‌های حل مسئله را به دستورات و دستورات کامپیوتری تبدیل می‌کنند تا برنامه‌ی خود را ایجاد کنند. این دستورات شامل دستورهای کنترلی مانند شرط‌ها و حلقه‌ها، دستورات عملیاتی مانند اعمال ریاضی و منطقی، و ساختارهای داده‌ای مختلف مانند آرایه‌ها، لیست‌ها و سایر عناصر است. در واقع، کد نویسی به عمل ترجمه‌ی زبان انسانی برنامه‌نویس به زبانی است که کامپیوترها قادرند تا اجرای آن را انجام دهند. این فرآیند شامل نوشتن دستورالعمل‌ها، تعیین ساختار و ترتیب اجرای برنامه، و رفع اشکالات و اشتباهات ممکن در کد می‌شود.

۱-۴-۳- پایتون و اهمیت آن

پایتون (Python) یک زبان برنامه‌نویسی متن‌باز و عمومی است که توسط گوئیدو وان روسوم (Guido van Rossum) در اواخر دهه ۱۹۸۰ ایجاد شده است. این زبان از ویژگی‌های ساختاری و عملی برای نوشتن کدهای مختلف بهره می‌برد و مورد استفاده گسترده‌ای در زمینه‌های مختلف از جمله توسعه وب، علوم داده، هوش مصنوعی، اپلیکیشن‌های دسکتاپ و موارد دیگر دارد. پایتون

به عنوان یکی از زبان‌های برنامه‌نویسی پرکاربرد و محبوب، اهمیت و دلایل محبوبیت زیادی دارد. در زیر به برخی از این دلایل اشاره شده است:

سادگی و خوانایی: پایتون با داشتن سینتکسی ساده و خوانا، برنامه‌نویسان را قادر می‌سازد به سرعت کدهای خود را بنویسند و از آنها استفاده کنند. این ویژگی باعث می‌شود پایتون به عنوان یک زبان برنامه‌نویسی آموزشی نیز بسیار مورد استفاده قرار گیرد.

پویایی و انعطاف‌پذیری: پایتون یک زبان برنامه‌نویسی پویا و انعطاف‌پذیر است که به برنامه‌نویسان امکان می‌دهد به سرعت و با قابلیت تغییر، کدهای خود را توسعه دهند.

پایداری و پشتیبانی: پایتون یک زبان پایدار است که توسط یک جامعه بزرگی از برنامه‌نویسان حمایت می‌شود. این باعث می‌شود که پایتون دائماً به‌روز و پشتیبانی شود و از ویژگی‌های جدید و بهبودهای مختلفی برخوردار باشد.

مجموعه‌کتابخانه‌های بزرگ: پایتون دارای مجموعه‌کتابخانه‌های بسیار بزرگ و گسترده‌ای است که برای حل مسائل مختلف در زمینه‌های مختلف مورد استفاده قرار می‌گیرد. این کتابخانه‌ها به برنامه‌نویسان امکان می‌دهند که بدون نوشتن کد از ابتدا، از ابزارها و تکنولوژی‌های موجود استفاده کنند و زمان توسعه را به شکل قابل توجهی کاهش دهند.

کاربردهای گسترده: پایتون به عنوان یک زبان برنامه‌نویسی چندمنظوره شناخته می‌شود و در زمینه‌های مختلفی از جمله توسعه وب، علوم داده، هوش مصنوعی، توسعه اپلیکیشن‌های دسکتاپ و موبایل، توسعه بازی‌ها و موارد دیگر مورد استفاده قرار می‌گیرد.

زبان‌های برنامه‌نویسی در حوزه بیولوژی و علوم زیستی از اهمیت بسیار زیادی برخوردارند و برای انجام مجموعه‌ای از فعالیت‌ها و تحقیقات مورد استفاده قرار می‌گیرند. برخی از کاربردهای زبان‌های برنامه‌نویسی در بیولوژی عبارتند از:

مدل‌سازی و شبیه‌سازی: زبان‌های برنامه‌نویسی مورد استفاده قرار می‌گیرند برای ایجاد مدل‌های ریاضی و شبیه‌سازی‌های کامپیوتری از فرآیندهای زیستی، اعم از تعاملات مولکولی، پروتئین‌ها، سلول‌ها، و حتی ایجاد مدل‌های اکولوژیک.

تحلیل داده‌های بیولوژیکی: زبان‌های برنامه‌نویسی مورد استفاده قرار می‌گیرند برای پردازش و تحلیل داده‌های بیولوژیکی مانند داده‌های ژنومیک، پروتئومیک، متابولومیک و داده‌های انجمنی.

بیوانفورماتیک: زبان‌های برنامه‌نویسی برای توسعه ابزارها و نرم‌افزارهای بیوانفورماتیکی مورد استفاده قرار می‌گیرند که به محققان در تحلیل داده‌های بیولوژیکی و درک بهتر فرآیندهای زیستی کمک می‌کنند.

مدیریت داده‌های زیستی: زبان‌های برنامه‌نویسی برای ایجاد سیستم‌های مدیریت داده‌های بیولوژیکی مورد استفاده قرار می‌گیرند که امکان مدیریت و ذخیره سازی انواع مختلف داده‌های زیستی را فراهم می‌کنند.

توسعه نرم‌افزارهای آزمایشگاهی: زبان‌های برنامه‌نویسی برای توسعه نرم‌افزارها و ابزارهای مورد استفاده در آزمایشگاه‌های بیولوژیکی و تحقیقاتی مورد استفاده قرار می‌گیرند که به محققان کمک می‌کنند فرآیندهای آزمایشی را مدیریت و اطلاعات بیشتری از آزمایشات به دست بیاورند.

تحلیل توالی DNAو RNA: برنامه نویسی در تحلیل توالی DNAو RNAبسیار مفید است. برای مثال برای تحلیل توالی ژنوم، تعیین عملکرد ژن ها و شناسایی ویژگی های زیستی، از زبان های برنامه نویسی مانند پایتون و Rاستفاده می شود.

طراحی و توسعه وب سایت های زیستی: برای طراحی و توسعه وب سایت های زیستی مانند پایگاه داده های زیستی، سیستم های اطلاعاتی زیستی و غیره از زبان های برنامه نویسی مانند پایتون، PHPو JavaScriptاستفاده می شود.

پردازش تصویر زیستی: برای پردازش تصاویر زیستی که از میکروسکوپ ها و دستگاه های تصویربرداری بدست می آیند، از زبان های برنامه نویسی مانند پایتون استفاده می شود. این زبان ها به بیولوژیست‌ها امکان می دهند تا تصاویر را پردازش کرده و ویژگی های مختلف آنها را استخراج کنند. به عنوان مثال، برای شناسایی سلول‌های سرطانی در تصاویر میکروسکوپی، از الگوریتم‌های پردازش تصویر برای تشخیص و محاسبه ویژگی‌های سلول‌ها استفاده می شود.

به طور کلی، برنامه نویسی در زمینه بیولوژی به بیولوژیست‌ها امکان می دهد تا داده ها را به صورت دقیق تر و سریع تر تحلیل کنند و مدل های ریاضی برای توصیف فرآیندهای زیستی ایجاد کنند. همچنین، برنامه نویسی به آنها امکان می دهد تا ابزارهای تحقیقاتی خود را بر اساس نیازهای خود طراحی کنند و در نهایت، به دستیابی به نتایج بهتر و سریع تر در تحقیقات خود کمک می کند.

۱-۵- محیط های یکپارچه توسعه نرم افزار (IDE)

محیط‌های یکپارچه توسعه نرم‌افزار یا همان IDE (Integrated Development Environments) ، نرم‌افزارهایی هستند که برای توسعه، تست، اجرا و ادیت کدهای برنامه‌نویسی مورد استفاده قرار می‌گیرند. این محیط‌ها عموماً شامل مجموعه‌ای از ابزارها و ویژگی‌هایی هستند که کار برنامه‌نویس را در فرآیند توسعه نرم‌افزار بهبود می‌بخشند. برخی از ویژگی‌ها و ابزارهایی که در یک IDE معمولاً موجود هستند عبارتند از:

ویرایشگر کد: ابزاری که به برنامه‌نویس امکان می‌دهد کدهای برنامه‌نویسی خود را ویرایش و تغییر دهد، و شامل ویژگی‌هایی مانند تشدید رنگ، تمیزکاری خودکار و تکمیل خودکار کد می‌باشد.

پیش‌نمایش کد: امکان مشاهده پیش‌نمایش کد و خطاها قبل از اجرای برنامه.

اجرا و دیباگ کد: امکان اجرای کدها به صورت مستقیم در IDE و انجام عملیات دیباگ (رفع اشکال) بر روی آنها.

ایجاد و مدیریت سند: ابزارهایی برای ایجاد و مدیریت انواع سندها و فایل‌های مورد نیاز پروژه.

پایتون وابسته به یک IDE نیست و در هر محیطی که بتوان کد نویسی کرد می توان برنامه های پایتون را اجرا کرد. پس باید به این نکته توجه شود که شما در این دوره برنامه نویسی پایتون یاد میگیرید و تفاوتی برای شما محیط برنامه نویسی نخواد داشت. به عبارت دیگر تمام کد هایی که در یک محیط می تویسید در محیط های دیگر برنامه نویسی اجزا می شود. انواع محیط‌های برنامه نویسی پایتون وجود دارد که در اینجا فقط چند مورد از مهمترین ها را مورد بررسی قرار می دهیم.

- **محیط رسمی برنامه نویسی پایتون:**

محیط رسمی برنامه‌نویسی پایتون که توسط تیم توسعه‌دهندگان پایتون ارائه شده است، اسم آن IDLE است IDLE .مخفف "Integrated Development and Learning Environment"می‌باشد و به‌عنوان محیط توسعه رسمی پایتون شناخته می‌شود. IDLEیک محیط گرافیکی است که به برنامه‌نویسان پایتون امکان می‌دهد کدهای خود را ویرایش، اجرا و تست کنند. این محیط شامل ویژگی‌هایی مانند ویرایشگر کد، پیش‌نمایش کد، پشتیبانی از انواع مختلف زبان‌های برنامه‌نویسی، دیباگ کد و ... می‌باشد. IDLE به‌صورت پیش‌فرض با نصب پایتون همراه می‌شود و بنابراین هیچ نیازی به نصب جداگانه نیست. این ابزار مناسب برای کاربران مبتدی و کاربرانی است که به دنبال یک محیط ساده و راحت برای توسعه و آزمون کدهای پایتون هستند. برای دانلود این محیط می توانید به وب سایت https://www.python.org/ مراجعه کنید

- **توزیع آناکوندا:**

آناکوندا یک توزیع محیط برنامه‌نویسی برای زبان‌های برنامه‌نویسی متن باز است که برای کار با داده‌های بزرگ و علوم داده بسیار مناسب است. این توزیع شامل مجموعه‌ای از ابزارها، کتابخانه‌ها و بسته‌های نرم‌افزاری مرتبط با علوم داده، محاسبات علمی و توسعه نرم‌افزار است . برخی از ویژگی‌های آناکوندا عبارتند از:

مدیریت بسته‌ها: آناکوندا امکان مدیریت و نصب بسته‌ها و کتابخانه‌های مختلف پایتون را فراهم می‌کند. این به شما امکان می‌دهد که به راحتی بسته‌های مورد نیاز خود را نصب و به‌روزرسانی کنید.

محیط توسعه یکپارچه: آناکوندا شامل محیط توسعه یکپارچه‌ای است که به برنامه‌نویسان امکان می‌دهد کدهای پایتون خود را ویرایش، اجرا و تست کنند.

کتابخانه‌های علم داده: آناکوندا شامل بسیاری از کتابخانه‌ها و ابزارهای محبوبی است که برای کار با داده‌های بزرگ و تحلیل داده‌ها استفاده می‌شوند، از جملهNumPy ، Pandas، Matplotlibو.scikit-learn

آناکوندا به‌طور گسترده توسط دانشمندان داده، محققان و برنامه‌نویسان استفاده می‌شود و به عنوان یکی از توزیع‌های محبوب برای کار با داده‌های بزرگ و توسعه نرم‌افزارهای علمی و تحقیقاتی مورد توجه قرار دارد.

برای نصب محیط آناکوندا به وب سایت https://www.anaconda.com/ مراجعه کنید. با نصب این توزیع محیط های برنامه نویسی Jupyter Notebook و Spyder نصب می شود.

- **Jupyter Notebook**

Jupyter Notebookیک برنامه متن‌باز است که به برنامه‌نویسان امکان می‌دهد کدهای خود را در قالب یک محیط تعاملی و توضیحی نوشته و اجرا کنند. این برنامه امکان ایجاد و اشتراک‌گذاری دفترچه‌های Jupyter (یا همان نوت‌بوک‌ها) را فراهم می‌کند که حاوی کدهای برنامه‌نویسی، متن توضیحی، فرمول‌های ریاضی، تصاویر تعاملی و نمودارها می‌باشند. برخلاف محیط‌های برنامه‌نویسی سنتی که کد را به صورت خط به خط اجرا می‌کنند، Jupyter Notebookبه برنامه‌نویسان امکان می‌دهد که کدهای خود را به صورت بلوک‌های اجرایی (یا همان سلول‌ها) بنویسند که می‌توانند به صورت تکراری اجرا شوند و نتایج آن‌ها در همان دفترچه نوشته و نمایش داده شود. بعضی از ویژگی‌های مهم Jupyter Notebook عبارتند از:

تعاملی بودن: برنامه‌نویسان می‌توانند کد خود را به صورت تعاملی اجرا و نتایج آن‌ها را مشاهده کنند، این امر امکان آزمون و تجزیه و تحلیل کدها را بسیار آسان می‌کند.

پشتیبانی از زبان‌های مختلف: پشتیبانی از بیشتر زبان‌های برنامه‌نویسی محبوب را فراهم می‌کند، اما بیشتر از همه برای زبان‌های محاسبات علمی مانند پایتون، Rو Julia مورد استفاده قرار می‌گیرد.

قابلیت اشتراک‌گذاری: کاربران می‌توانند دفترچه‌های Jupyter خود را در اینترنت به اشتراک بگذارند، این امکان به برنامه‌نویسان اجازه می‌دهد که کدها، توضیحات و نتایج خود را به دیگران نمایش دهند و با همکاران و همگروهی‌هاشان همکاری کنند.

پشتیبانی از تولید خروجی چند رسانه: امکان ایجاد خروجی‌های چند رسانه‌ای مانندHTML ، PDF، اسلایدرهای تعاملی، ویدئو و ... را داراست که برای ارائه نتایج و گزارش‌های تحقیقاتی بسیار مفید است.

در کل، Jupyter Notebook یک ابزار قدرتمند و گسترده است که برای توسعه نرم افزار و الگوریتم ها به خصوص در حوزه انالیز داده استفاده می شود.

- **Spyder**

Spyderیک محیط توسعه یکپارچه (IDE) برای زبان برنامه‌نویسی پایتون است که به‌طور خاص برای کارهای علمی و محاسباتی طراحی شده است. این IDE شامل ابزارها و ویژگی‌هایی است که به برنامه‌نویسان علم داده و پژوهشگران امکان می‌دهد تا به راحتی کدهای پایتونی خود را توسعه، تست، و بهینه‌سازی کنند. برخی از ویژگی‌های مهم Spyder عبارتند از:

ویرایشگر کد پیشرفته: دارای یک ویرایشگر کد پیشرفته است که امکاناتی مانند تشدید رنگ، تکمیل خودکار کد، نمایش نوار ترسیم و ... را فراهم می‌کند.

محیط توسعه تعاملی: به برنامه‌نویسان امکان می‌دهد کدهای خود را به صورت تعاملی اجرا کرده و نتایج را به صورت فوری مشاهده کنند.

پشتیبانی از کتابخانه‌های علم داده: این IDE شامل بسیاری از کتابخانه‌های محبوب مورد استفاده برای علم داده مانندNumPy ، Pandas، Matplotlibو ... می‌باشد.

پیکربندی سفارشی: به برنامه‌نویسان امکان می‌دهد تا محیط خود را با توجه به نیازهای خود پیکربندی و تنظیم کنند.

پشتیبانی از تجزیه و تحلیل داده‌های بزرگ: ابزارها و ویژگی‌های مختلفی دارد که برای تجزیه و تحلیل داده‌های بزرگ و پردازش علمی مورد استفاده قرار می‌گیرد.

به طور کلی، Spyder یک IDE کارآمد و قدرتمند برای برنامه‌نویسان پایتون در حوزه علم داده و محاسبات علمی است که امکانات گسترده‌ای را برای توسعه کدهای پایتونی فراهم می‌کند.

- **Google Colab**

یک سرویس رایگان ارائه شده توسط گوگل است که به برنامه‌نویسان امکان اجرای کدهای پایتونی را در محیطی ابری و با استفاده از منابع محاسباتی گوگل فراهم می‌کند. این سرویس بر مبنای محیط Jupyter Notebook است و برای تحلیل داده، آموزش مدل‌های یادگیری ماشین، انجام پروژه‌های علم داده و ... استفاده می‌شود. برخی از ویژگی‌های Google Colab عبارتند از:

رایگان بودن و منابع محاسباتی قوی: به‌صورت رایگان ارائه می‌شود و از منابع محاسباتی گوگل از جمله پردازنده و حافظه استفاده می‌کند که برای اجرای کدها و آنالیز داده‌ها بسیار قوی است.

تعاملی بودن و قابلیت اشتراک‌گذاری: مشابهJupyter Notebook ، Google Colabبه برنامه‌نویسان امکان می‌دهد کدهای خود را به صورت تعاملی اجرا کرده و نتایج را به‌صورت فوری مشاهده کنند. همچنین این ابزار به شما امکان می‌دهد دفترچه‌های خود را با دیگران به‌اشتراک بگذارید.

پشتیبانی از کتابخانه‌های علم داده: شامل بسیاری از کتابخانه‌های محبوب برای علم داده مانندNumPy ، Pandas، Matplotlibو ... می‌باشد.

انجام تجزیه و تحلیل داده‌های بزرگ: با استفاده ازGoogle Colab ، می‌توانید داده‌های بزرگ را آنالیز و پردازش کنید و از منابع محاسباتی قوی گوگل بهره‌برداری کنید.

پشتیبانی از GPU و TPU : امکان استفاده از منابع پردازشی مانند GPU و TPU را برای تسریع در عملیات‌های محاسباتی فراهم می‌کند که برای آموزش مدل‌های یادگیری ماشین بسیار مفید است.

با توجه به ویژگی‌های بالا، Google Colab یک ابزار قدرتمند و رایگان برای تحلیل داده، آموزش مدل‌های یادگیری ماشین، و انجام پروژه‌های علم داده می‌باشد که توسط برنامه‌نویسان و پژوهشگران با توجه به امکانات و منابع قدرتمند آن مورد استفاده قرار می‌گیرد. برای دسترسی به این محیط از وب سایت https://colab.research.google.com/ استفاده کنید.

GENRAN