Artificial Intelligence

Definition:

In simple terms, we can define AI as a machine that can simulate human thought process and can take actions based on those thoughts and even draw conclusions. It should also be able to correct itself, if it makes a mistake. This also means that AI based computer would be able to make a decision in a given situation like human beings and in some cases even better

According to Niti Aayog:

AI refers to the ability of machines to perform cognitive tasks like thinking, perceiving, learning, problem solving and decision making. Initially conceived as a technology that could mimic human intelligence, AI has evolved in ways that far exceed its original conception. With incredible advances made in data collection, processing and computation power, intelligent systems can now be deployed to take over a variety of tasks, enable connectivity and enhance productivity.

Some common examples of AI applications that you might encounter in your surroundings:

1. **Virtual Assistants:** Virtual assistants like Amazon's Alexa, Apple's Siri, Google Assistant, and Microsoft's Cortana utilize AI to understand and respond to voice commands, perform tasks, answer questions, and provide personalized recommendations.
2. **Recommendation Systems:** Platforms such as Netflix, Spotify, and Amazon use AI algorithms to analyze user behavior and preferences, recommending content, products, or services tailored to individual tastes.
3. **Social Media:** AI algorithms are employed by social media platforms like Facebook, Twitter, and Instagram for content moderation, personalized content curation, targeted advertising, and sentiment analysis.
4. **Navigation Apps:** Navigation apps like Google Maps and Waze leverage AI to analyze real-time traffic data, provide optimal routes, estimate arrival times, and offer alternative routes to avoid congestion.
5. **Online Shopping:** E-commerce platforms such as Amazon and Alibaba utilize AI for product recommendations, dynamic pricing, fraud detection, inventory management, and customer service chatbots.
6. **Healthcare:** AI technologies are increasingly used in healthcare for medical imaging analysis, disease diagnosis, drug discovery, personalized treatment recommendations, and remote patient monitoring.
7. **Finance:** In the financial sector, AI is employed for fraud detection, algorithmic trading, credit scoring, risk assessment, customer service chatbots, and personalized financial advice.
8. **Smart Home Devices:** AI powers smart home devices such as thermostats, security cameras, doorbell cameras, and lighting systems, enabling automation, energy efficiency, and remote control via voice commands or mobile apps.
9. **Autonomous Vehicles:** AI is at the core of self-driving cars and autonomous vehicles, enabling them to perceive their surroundings, navigate safely, and make real-time driving decisions based on sensor data and machine learning algorithms.
10. **Language Translation:** AI-driven language translation services like Google Translate and Microsoft Translator use neural machine translation techniques to translate text and speech between multiple languages accurately.

Advantages of AI:

1. **Automation:** AI enables the automation of repetitive and mundane tasks, freeing up human resources to focus on more creative and strategic activities. This leads to increased productivity and efficiency in both individual tasks and entire workflows.
2. **Accuracy and Precision:** AI systems can process vast amounts of data with high accuracy and precision, minimizing errors and improving the quality of decision-making. This is particularly beneficial in tasks such as data analysis, diagnostics, and quality control.
3. **24/7 Availability:** AI-powered systems can operate continuously without the need for breaks, resulting in 24/7 availability of services such as customer support, virtual assistants, and online transactions, enhancing convenience for users across different time zones.
4. **Personalization:** AI algorithms can analyze user data and behavior to provide personalized recommendations, content, and services tailored to individual preferences. This enhances user experience and satisfaction by delivering relevant and timely information.
5. **Predictive Analytics:** AI enables organizations to leverage predictive analytics to forecast trends, anticipate customer needs, and identify potential risks or opportunities. This proactive approach facilitates better decision-making and strategic planning.
6. **Efficient Resource Utilization:** AI-driven optimization algorithms can optimize resource allocation, scheduling, and logistics, leading to cost savings, reduced waste, and improved resource utilization in various sectors such as transportation, manufacturing, and energy management.
7. **Enhanced Healthcare:** AI technologies, including machine learning and computer vision, are transforming healthcare by assisting in medical imaging analysis, disease diagnosis, drug discovery, personalized treatment planning, and remote patient monitoring, leading to improved patient outcomes and reduced healthcare costs.
8. **Improved Safety:** AI-powered systems enhance safety in various domains, including autonomous vehicles, manufacturing, and cybersecurity, by detecting anomalies, predicting potential hazards, and implementing preventive measures to mitigate risks.
9. **Innovative Applications:** AI fosters innovation by enabling the development of new products, services, and business models that were previously unfeasible. This includes applications such as natural language processing, robotics, virtual reality, and augmented reality, opening up new possibilities for creativity and exploration.
10. **Global Competitiveness:** Organizations that harness the power of AI gain a competitive edge in the global marketplace by leveraging advanced technologies to drive innovation, improve efficiency, and deliver superior products and services.

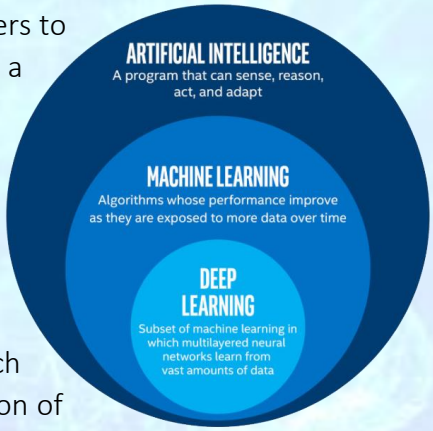
Disadvantages of AI:

1. **Job Displacement:** AI-driven automation has the potential to replace human workers in certain tasks and industries, leading to job displacement and unemployment. This particularly affects jobs that involve repetitive or routine tasks, such as manufacturing, customer service, and administrative roles.
2. **Bias and Discrimination:** AI algorithms can inadvertently perpetuate biases present in training data, leading to discriminatory outcomes in decision-making processes. This can exacerbate societal inequalities related to race, gender, ethnicity, and other factors if not carefully monitored and mitigated.

3. **Privacy Concerns:** AI technologies often rely on vast amounts of personal data to function effectively, raising concerns about privacy and data security. Unauthorized access, data breaches, and misuse of personal information pose significant risks to individuals' privacy rights.
4. **Lack of Transparency:** AI models, particularly complex deep learning algorithms, can be opaque and difficult to interpret, leading to a lack of transparency in decision-making processes. This lack of transparency raises concerns about accountability, fairness, and trust in AI systems.
5. **Overreliance on Technology:** Overreliance on AI systems without human oversight or intervention can lead to overconfidence and complacency, particularly in safety-critical domains such as autonomous vehicles and healthcare. Human supervision and intervention are essential to ensure the safety and reliability of AI-driven systems.
6. **Ethical Dilemmas:** AI raises complex ethical dilemmas, such as the use of autonomous weapons, surveillance technologies, and predictive policing algorithms. These raise concerns about accountability, human rights violations, and the erosion of civil liberties.
7. **Social Isolation:** The proliferation of AI-driven technologies, such as virtual assistants and social robots, may contribute to social isolation and disconnection by reducing face-to-face interactions and interpersonal relationships.

AI, ML & DL

AI-Artificial Intelligence (AI) Refers to any technique that enables computers to mimic human intelligence. It gives the ability to machines to recognize a human's face; to move and manipulate objects; to understand the voice commands by humans, and also do other tasks. The AI-enabled machines think algorithmically and execute what they have been asked for intelligently.



ML-Machine Learning (ML) It is a subset of Artificial Intelligence which enables machines to improve at tasks with experience (data). The intention of Machine Learning is to enable machines to learn by themselves using the provided data and make accurate Predictions/ Decisions.

DL-Deep Learning (DL) It enables software to train itself to perform tasks with vast amounts of data. In Deep Learning, the machine is trained with huge amounts of data which helps it in training itself around the data. Such machines are intelligent enough to develop algorithms for themselves. Deep Learning is the most advanced form of Artificial Intelligence out of these three. Then comes Machine Learning which is intermediately intelligent and Artificial Intelligence covers all the concepts and algorithms which, in some way or the other mimic human intelligence.

Types of AI:

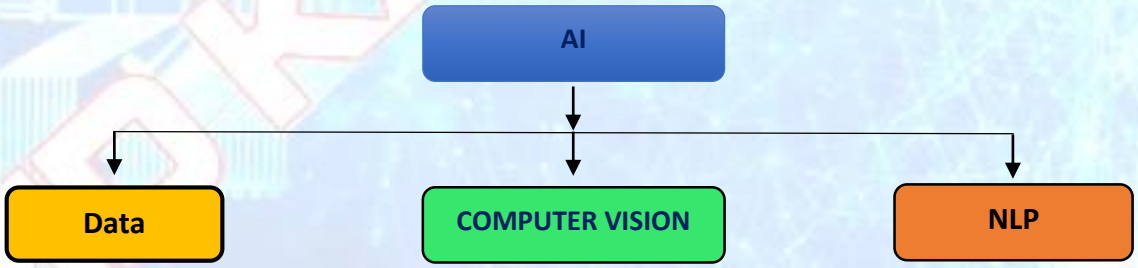
Narrow AI - Narrow AI, also known as Weak AI, refers to AI systems that are designed and trained for specific tasks or domains. These systems excel at performing a single task or a narrow range of tasks but lack the general intelligence and adaptability of humans. Examples of narrow AI include virtual assistants like Siri and Alexa, recommendation systems, image recognition systems, and language translation tools.

General AI- General AI, also known as Strong AI or Artificial General Intelligence (AGI), refers to AI systems that possess human-like intelligence and are capable of understanding, learning, and reasoning across a wide range of tasks and domains. AGI systems would be able to perform any intellectual task that a human can do, and potentially surpass human intelligence. However, true AGI remains a theoretical concept and has not yet been achieved.

Super AI -"Super AI" typically refers to an advanced form of artificial intelligence that surpasses human-level intelligence and capabilities in various domains. This concept often appears in science fiction, where super AI may possess abilities such as self-awareness, creativity, emotional intelligence, and problem-solving skills far beyond what humans can achieve.

In reality, while significant advancements have been made in AI technology, we haven't reached the level of "super AI" yet. Current AI systems excel at specific tasks but lack the general intelligence and adaptability of human beings. Researchers are continuously working on developing more advanced AI systems, but achieving true super AI remains a distant goal with profound ethical and societal implications.

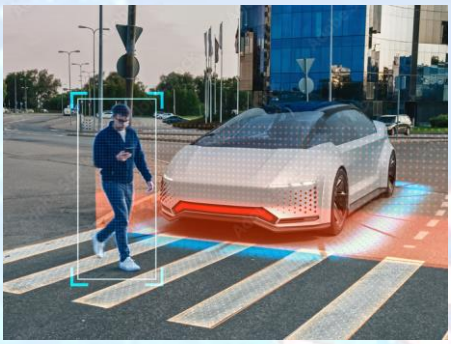
AI Domains



Data Science

Data sciences is a domain of AI related to data systems and processes, in which the system collects numerous data, maintains data sets and derives meaning/sense out of them. Data science plays a crucial role in AI by providing the foundation for creating models and algorithms. It involves collecting, cleaning, analyzing, and interpreting large datasets to extract insights and make informed decisions.

Computer Vision

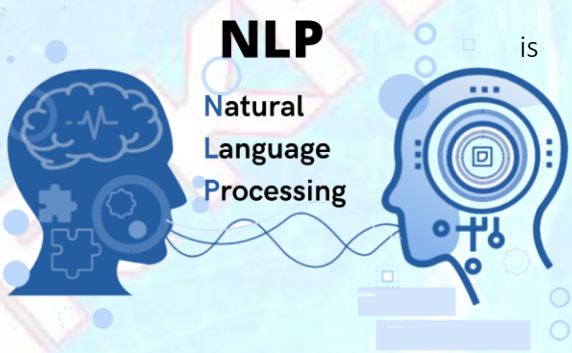


Computer vision is a field of artificial intelligence that focuses on enabling machines to interpret and make decisions based on visual data, similar to how humans use their eyes and brains to understand their surroundings. It involves the development of algorithms and systems that can analyze images and videos to extract meaningful information.

Examples of Computer Vision

Self-Driving cars/ Automatic Cars
Face Lock in Smartphones

Natural Language Processing



Natural Language Processing, abbreviated as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using natural language. Natural language refers to language that is spoken and written by people, and natural language processing (NLP) attempts to extract information from the spoken and written word using algorithms.

Examples of Natural Language Processing

Email filters-Email filters are one of the most basic and initial applications of NLP online. It started out with spam filters, uncovering certain words or phrases that signal a spam message.

Smart assistants-Smart assistants like Apple's Siri and Amazon's Alexa recognize patterns in speech, then infer meaning and provide a useful response.

AI ethics

AI ethics refers to the moral principles, guidelines, and standards governing the development, deployment, and use of artificial intelligence technologies. As AI becomes increasingly integrated into various aspects of society, ethical considerations become paramount to ensure that AI systems are developed and deployed in a responsible, fair, and beneficial manner. Some key areas of AI ethics include:

1. **Fairness and Bias:** Ensuring that AI systems are fair and unbiased is crucial to prevent discrimination and ensure equal treatment across different demographic groups. This involves identifying and mitigating biases in training data, algorithms, and decision-making processes to promote fairness and equity.
2. **Transparency and Explainability:** AI systems should be transparent and explainable, enabling users to understand how decisions are made and the reasoning behind them. Providing explanations for AI decisions fosters trust, accountability, and oversight, particularly in high-stakes applications such as healthcare, criminal justice, and finance.

3. **Privacy and Data Protection:** Protecting individuals' privacy and data rights is essential when developing and deploying AI systems that collect, process, and analyze personal data. Adhering to privacy regulations, implementing data anonymization techniques, and obtaining informed consent from users are critical to safeguarding privacy and preventing misuse of personal information.
4. **Accountability and Responsibility:** Establishing clear lines of accountability and responsibility is necessary to address the potential ethical implications and consequences of AI systems. Developers, organizations, and policymakers should take responsibility for the ethical design, development, and deployment of AI technologies, including addressing unintended consequences and harmful outcomes.
5. **Safety and Security:** Ensuring the safety and security of AI systems is paramount to prevent accidents, errors, or malicious exploitation. Implementing robust testing, validation, and cybersecurity measures can mitigate risks associated with AI failures, vulnerabilities, or adversarial attacks.
6. **Human-Centered Design:** Prioritizing human well-being, autonomy, and dignity in the design and deployment of AI systems is essential to enhance societal benefit and minimize harm. Human-centered design approaches involve involving diverse stakeholders, considering ethical implications, and prioritizing the needs and values of end-users.
7. **Social Impact and Equity:** Assessing and mitigating the social impact of AI technologies on individuals, communities, and society at large is crucial to promote inclusive and equitable outcomes. Addressing issues such as job displacement, economic inequality, and access to AI technologies can help mitigate negative consequences and promote social good.
8. **Global Collaboration and Governance:** Promoting international collaboration and cooperation is essential to address the global nature of AI ethics challenges and ensure alignment with shared ethical principles and values. Establishing robust governance mechanisms, standards, and regulations can help guide the responsible development and use of AI technologies on a global scale.

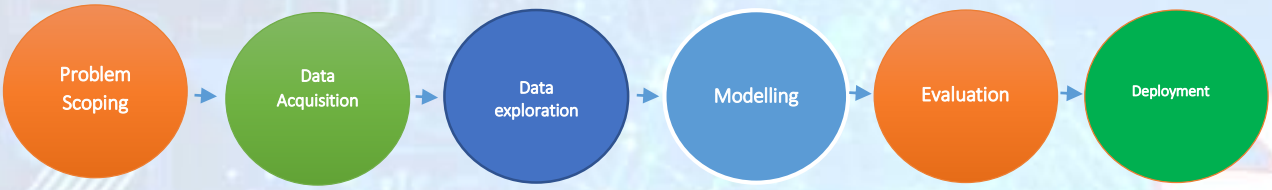
AI Bias

Another aspect to AI Ethics is bias. Everyone has a bias of their own no matter how much one tries to be unbiased, we in some way or the other have our own biases even towards smaller things. Biases are not negative all the time. Sometimes, it is required to have a bias to control a situation and keep things working. When we talk about a machine, we know that it is artificial and cannot think on its own. It can have intelligence, but we cannot e xpect a machine to have any biases of its own. Any bias can transfer from the developer to the machine while the algorithm is being developed.

AI Access

Since Artificial Intelligence is still a budding technology, not everyone has the opportunity to access it. The people who can afford AI enabled devices make the most of it while others who cannot are left behind. Because of this, a gap has emerged between these two classes of people and it gets widened with the rapid advancement of technology.

AI Project Cycle



Problem Scoping

It is a fact that we are surrounded by problems. They could be small or big, sometimes ignored or sometimes even critical. Many times, we become so used to a problem that it becomes a part of our life. Identifying such a problem and having a vision to solve it, is what Problem Scoping is about.

Sustainable Development Goals

The Sustainable Development Goals (SDGs) are a collection of 17 global goals set by the United Nations General Assembly in 2015 as part of the 2030 Agenda for Sustainable Development. They build upon the success of the Millennium Development Goals (MDGs) and aim to address a range of interconnected global challenges, including poverty, inequality, environmental degradation, and climate change. The SDGs are designed to be integrated and indivisible, recognizing that progress in one area often depends on progress in others.

Artificial intelligence (AI) can play a significant role in advancing the Sustainable Development Goals (SDGs) by providing innovative solutions to complex global challenges. Here's how AI can contribute to each of the SDGs:

1. No Poverty: AI-powered tools can help governments and organizations better target poverty reduction efforts by analyzing socioeconomic data, identifying vulnerable populations, and optimizing resource allocation.
2. Zero Hunger: AI-driven precision agriculture techniques can improve crop yield prediction, optimize resource usage, and detect pests and diseases early, helping to ensure food security and reduce food waste.
3. Good Health and Well-being: AI applications in healthcare, such as medical imaging analysis, predictive analytics for disease diagnosis and outbreak detection, and personalized treatment recommendations, can improve healthcare delivery and outcomes.
4. Quality Education: AI-based educational technologies, such as personalized learning platforms, intelligent tutoring systems, and language translation tools, can enhance access to quality education and support lifelong learning for all.
5. Gender Equality: AI can help identify and address gender biases in various domains, promote gender-inclusive policies and practices, and support women's empowerment through access to information and opportunities.
6. Clean Water and Sanitation: AI can assist in monitoring water quality, optimizing water distribution systems, and predicting water-related disasters, contributing to sustainable water management and sanitation infrastructure.
7. Affordable and Clean Energy: AI can optimize energy distribution, improve energy efficiency, facilitate renewable energy integration into the grid, and support the development of smart grids and energy storage systems.
8. Decent Work and Economic Growth: AI-driven automation and digitalization can create new job opportunities, enhance productivity, and foster inclusive economic growth, provided that measures are taken to mitigate potential job displacement and inequalities.
9. Industry, Innovation, and Infrastructure: AI technologies can accelerate innovation, optimize supply chains, improve infrastructure planning and maintenance, and promote sustainable industrialization and infrastructure development.
10. Reduced Inequality: AI can help identify and address disparities in access to resources and opportunities by analyzing data, informing policy decisions, and designing targeted interventions to reduce inequality within and among countries.
11. Sustainable Cities and Communities: AI-driven urban planning, transportation optimization, waste management, and public service delivery can contribute to building sustainable, resilient, and inclusive cities and communities.
12. Responsible Consumption and Production: AI can support sustainable consumption and production patterns by optimizing resource use, reducing waste, and promoting circular economy practices across industries.
13. Climate Action: AI can enhance climate modeling, support renewable energy integration, optimize resource management, and facilitate climate adaptation and mitigation efforts to address the impacts of climate change.
14. Life Below Water: AI-powered monitoring and surveillance systems can help protect marine ecosystems, combat illegal fishing, and promote sustainable fisheries management to conserve life below water.
15. Life on Land: AI can aid in monitoring and managing terrestrial ecosystems, combating deforestation and desertification, protecting biodiversity, and promoting sustainable land use practices.

16. **Peace, Justice, and Strong Institutions**: AI can support conflict prevention, peacebuilding, and the rule of law by analyzing social media data for early warning signs of conflict, facilitating access to justice, and enhancing transparency and accountability in governance.
17. **Partnerships for the Goals**: AI can facilitate collaboration and knowledge sharing among stakeholders, support data-driven decision-making, and strengthen partnerships for achieving the SDGs.

4Ws Problem Canvas

Who?

The "Who" block helps in analysing the people getting affected directly or indirectly due to it. Under this, we find out who the 'Stakeholders' to this problem are and what we know about them. Stakeholders are the people who face this problem and would be benefitted with the solution.

What?

Under the "What" block, you need to look into what you have on hand. At this stage, you need to determine the nature of the problem. What is the problem and how do you know that it is a problem? Under this block, you also gather evidence to prove that the problem you have selected actually exists. Newspaper articles, Media, announcements, etc are some examples.

Where?

Now that you know who is associated with the problem and what the problem actually is; you need to focus on the context/situation/location of the problem. This block will help you look into the situation in which the problem arises, the context of it, and the locations where it is prominent.

Why?

You have finally listed down all the major elements that affect the problem directly. Now it is convenient to understand who the people that would be benefitted by the solution are; what is to be solved; and where will the solution be deployed. These three canvases now become the base of why you want to solve this problem. Thus, in the "Why" canvas, think about the benefits which the stakeholders would get from the solution and how it will benefit them as well as the society.

Data Acquisition

Data acquisition is the process of collecting and gathering data from various sources. As the term clearly mentions, this stage is about acquiring data for the project. Let us first understand what is Data. Data can be a piece of information or facts and statistics collected together for reference or analysis. Whenever we want an AI project to be able to predict an output, we need to train it first using data.

Types of Data:

Training Data- Training data refers to the subset of data used to train machine learning models. It consists of input features and their corresponding target labels (for supervised learning) or just input features

(for unsupervised learning). Training data serves as the foundation for the model to learn patterns, relationships, and correlations between the input features and the target labels.

Testing Data- Testing data, also known as test data or validation data, is a separate subset of data used to evaluate the performance and generalization ability of machine learning models after they have been trained on the training data. Testing data helps assess how well a trained model can make predictions on new, unseen data that it hasn't encountered during training.

There can be various ways in which you can collect data. Some of them are :

1. Surveys and Questionnaires: Surveys and questionnaires are used to gather data from a large number of respondents about their opinions, behaviors, preferences, and other relevant information.
2. Interviews: Interviews involve direct communication between an interviewer and interviewee(s) to collect detailed information on a particular topic. Interviews can be structured, semi-structured, or unstructured depending on the level of formality and flexibility needed.
3. Observational Studies: Observational studies involve systematically observing and recording behaviors, events, or phenomena in natural settings without intervention. This method is often used in fields like anthropology, sociology, and psychology.
4. Experiments: Experiments involve manipulating variables under controlled conditions to observe the effects on other variables. This method allows researchers to establish cause-and-effect relationships.
5. Data Mining: Data mining involves extracting patterns and insights from large datasets using computational techniques such as machine learning, statistical analysis, and pattern recognition.
6. Web Scraping: Web scraping involves automatically extracting data from websites. This method is often used to collect data from online sources such as social media, e-commerce websites, and news websites.
7. Sensor Data Collection: Sensors can be used to collect data from the physical environment, such as temperature, humidity, pressure, and movement. This method is commonly used in fields like environmental monitoring, healthcare, and manufacturing.
8. Focus Groups: Focus groups involve small groups of participants discussing a topic guided by a moderator. This method is useful for collecting qualitative data and exploring attitudes and perceptions.

System Map

A system map shows the components and boundaries of a system and the components of the environment at a specific point in time. With the help of System Maps, one can easily define a relationship amongst different elements which come under a system.

Data exploration

Data exploration is the process of examining, cleaning, and visualizing data to discover patterns, trends, and relationships. It's a crucial step in the data analysis workflow that helps analysts and data scientists gain insights into the underlying structure of the data before applying more advanced analytics techniques.

Data Visualization

Data visualization is a powerful way to interpret and communicate data insights through graphical representation. It helps to make complex data more understandable and accessible. Here are some common types of data visualizations and their typical uses:

1. **Bar Charts**: Useful for comparing quantities across different categories. They can be vertical or horizontal and are great for showing relative sizes and trends over time.
2. **Line Charts**: Ideal for displaying data trends over time. They are often used to track changes in data points and to identify patterns or trends.
3. **Pie Charts**: Show the proportional contributions of different categories to a whole. They are best used when you want to illustrate how parts make up a whole.
4. **Histograms**: Display the distribution of numerical data by showing the frequency of data within certain ranges or bins. They are useful for understanding the distribution and spread of your data.
5. **Scatter Plots**: Reveal relationships or correlations between two numerical variables. They are useful for identifying patterns, trends, and outliers.
6. **Heat Maps**: Use color to represent data values in a matrix format. They are effective for showing the intensity of data across two dimensions.
7. **Box Plots**: Summarize data distribution through quartiles and highlight outliers. They provide a visual representation of the data's spread and central tendency.
8. **Bubble Charts**: An extension of scatter plots where data points are represented by bubbles of varying sizes. They can show relationships between three variables.
9. **Treemaps**: Display hierarchical data using nested rectangles. They are effective for visualizing proportions within a hierarchy and making comparisons between different categories.
10. **Gantt Charts**: Used for project management to show the timeline of tasks or activities. They help visualize project schedules and track progress.
11. **Geographical Maps**: Useful for visualizing data that has a spatial component. They can show regional differences and patterns on a map.

Modelling

In the context of AI, modeling refers to the process of creating mathematical representations of real-world phenomena or patterns in data using machine learning or deep learning techniques.

Artificial intelligence models apply different algorithms to relevant data inputs to achieve the tasks, or output, they've been programmed for.

Generally, AI models can be classified as follows :

Rule Based Approach

Refers to the AI modelling where the rules are defined by the developer. The machine follows the rules or instructions mentioned by the developer and performs its task accordingly.

Learning Based Approach

Refers to the AI modelling where the machine learns by itself. Under the Learning Based approach, the AI model gets trained on the data fed to it and then is able to design a model which is adaptive to the change in data. That is, if the model is trained with X type of data and the machine designs the algorithm around it, the model would modify itself according to the changes which occur in the data so that all the exceptions are handled in this case.

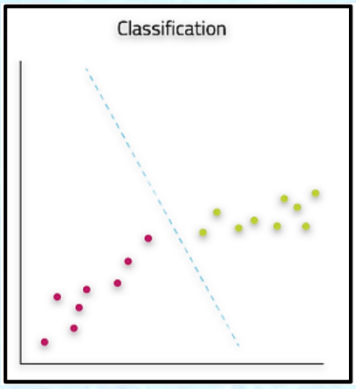
Supervised Learning

In a supervised learning model, the dataset which is fed to the machine is labelled. In other words, we can say that the dataset is known to the person who is training the machine only then he/she is able to label the data. A label is some information which can be used as a tag for data. For example, students get grades according to the marks they secure in examinations. These grades are labels which categorise the students according to their marks.

There are two types of Supervised Learning models :

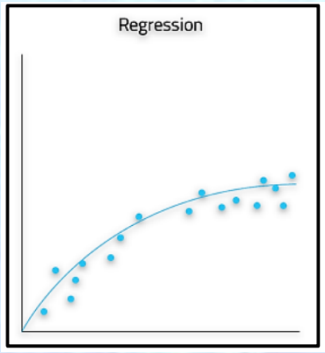
Classification:

Classification is a fundamental task in machine learning and artificial intelligence where the goal is to categorize input data into one of several predefined classes or categories. It's widely used across various domains, including image recognition, natural language processing, medical diagnosis, and spam filtering, among others.



Regression

Regression is another fundamental task in machine learning and statistical modeling, distinct from classification. While classification aims to predict the class labels of input data, regression focuses on predicting continuous numerical values. It's commonly used for tasks such as predicting house prices, stock prices, temperature, sales forecasts, and more.

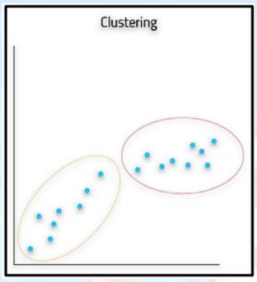


Unsupervised Learning

An unsupervised learning model works on unlabelled dataset. This means that the data which is fed to the machine is random and there is a possibility that the person who is training the model does not have any information regarding it. The unsupervised learning models are used to identify relationships, patterns and trends out of the data which is fed into it. It helps the user in understanding what the data is about and what are the major features identified by the machine in it.

Unsupervised learning models can be further divided into two categories :

Clustering



Refers to the unsupervised learning algorithm which can cluster the unknown data according to the patterns or trends identified out of it. The patterns observed might be the ones which are known to the developer or might even come up with some unique patterns out of it. it

Dimensionality Reduction

We humans are able to visualise upto 3-Dimensions only but according to a lot of theories and algorithms, there are various entities which exist beyond 3-Dimensions. For example, in Natural language Processing, the words are considered to be N-Dimensional entities. Which means that we cannot visualise them as they exist beyond our visualisation ability. Hence, to make sense out of it, we need to reduce their dimensions. Here, dimensionality reduction algorithm is used.

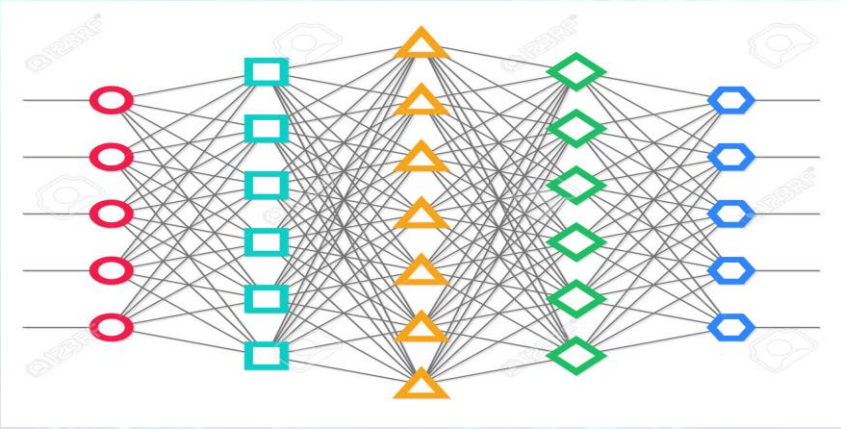
Evaluation

Evaluation is a crucial step in the development of AI models, whether they are used for classification, regression, clustering, or any other task. It involves assessing the performance of the model to understand how well it generalizes to new, unseen data and whether it meets the project's success criteria.

Deployment

Deployment is a critical stage in the AI project cycle where the trained and validated model is put into a production environment where it can make real-time predictions or decisions based on new data.

Neural Networks



Neural networks, often referred to as artificial neural networks (ANNs), are a class of machine learning algorithms inspired by the structure and function of the human brain. They consist of interconnected nodes, called neurons or units, organized into layers. Each neuron receives input signals, processes them using an activation function, and produces an output signal. Neural networks are capable of learning complex patterns in data through a process called training, where they adjust their parameters based on examples provided in a dataset. ***The key advantage of neural networks are that they are able to extract data features automatically without needing the input of the programmer.***

Here are some key components and concepts related to neural networks:

1. Neurons (Nodes): Neurons are the basic computational units in a neural network. They receive inputs from other neurons, perform a computation, and produce an output. Neurons are organized into layers, including input, hidden, and output layers.

2. Weights and Biases: Each connection between neurons in adjacent layers is associated with a weight, which determines the strength of the connection. Additionally, each neuron typically has an associated bias term, which allows the network to learn more complex functions.

3. Activation Function: The activation function of a neuron determines its output based on the weighted sum of its inputs. Common activation functions include the sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax functions.

4. Layers: A neural network is organized into layers, with each layer containing a group of neurons. The input layer receives the initial input data, while the output layer produces the final output of the network. Hidden layers, located between the input and output layers, perform intermediate computations.

5. Feedforward and Backpropagation: In the feedforward phase, input data is passed through the network, and predictions are generated. During backpropagation, the error between the predicted output and the true output is calculated and used to update the network's weights and biases, typically using optimization techniques such as gradient descent.

6. Training Data: Neural networks are trained using labeled datasets, where each example is associated with a known input and output. The network learns to approximate the mapping from inputs to outputs by adjusting its parameters to minimize the prediction error on the training data.

7. Loss Function: The loss function measures the difference between the predicted output of the neural network and the true output for a given example. Common loss functions include mean squared error (MSE), categorical cross-entropy, and binary cross-entropy.

Some of the features of a Neural Network are listed below:

- The Artificial Neural Network systems are modelled on the human brain and nervous system.
- They are able to automatically extract features without feeding the input by programmer.
- Every node of layer in a Neural Network is compulsorily a machine learning algorithm.
- It is very useful to implement when solving problems for very huge datasets.

Python

- Introduction
- Working with python
- Python Fundamentals
- Data Types
- Control Statements
- String
- List
- Tuple
- Dictionary

Introduction

- Python is a High level Programming language, developed by Guido Van Rossum in 1991.
- It is free to use.
- It is an Interpreted language.
- It is portable language which means it can run on any platform.
- Syntax are less in python compared to c, c++.
- It has wide range of built in functions, modules & libraries.

Working with python

+ Installation of python is available on www.python.org, which carry the python installation package.

Working Modes in Python Programming

Python works in two modes:

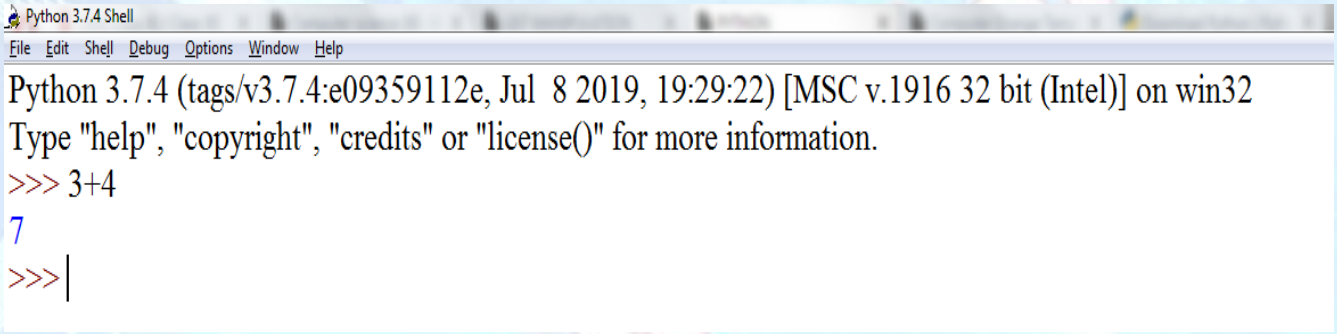
(i) Interactive Mode

(ii) Script Mode

Interactive Mode: User gives one command at a time and python executes the command and produces the output.

To work with Interactive mode, click on **Start→All programs→python3.7.x→IDLE**

A screen will appear shown as below:

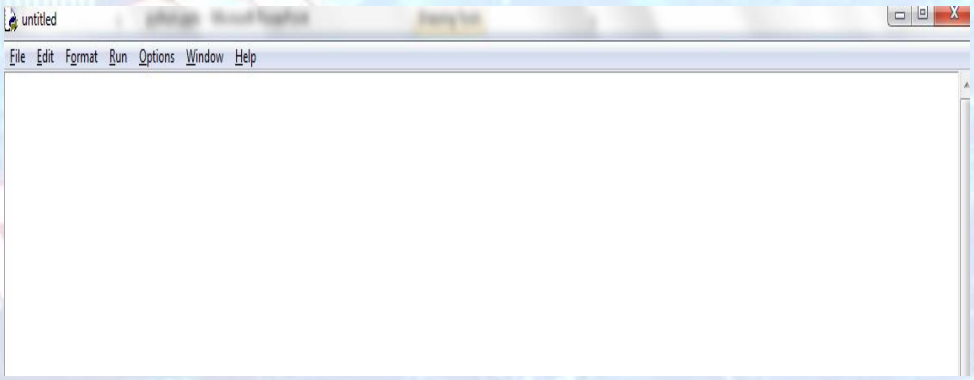


Interactive mode : In Interactive mode, user gives commands to python command prompt(>>>>) which indicates that python is waiting for user response.

Script Mode: In this mode, multiple statements are written and saved as a file in .py extension and execute it.

- To work with script mode, open IDLE. In IDLE click **File→New File**

A screen will appear shown as below:



- To save the file in script mode, click **File→Save** & save the file.
- To execute the file click **Run→Run Module** and get the output if there is no error.

Python fundamentals

- The fundamental and smallest unit of a program is token.
- There are five categories in token:

  Keywords

  Identifiers

  Literals

  Operators

  Punctuators

Keywords-: Keywords are the reserved words and have special meaning for python interpreter. Every keyword is assigned specific work and it can be used only for that purpose. Keywords can not be any identifiers.

Some of the keywords in python is-:

and	del	from	not
while	as	elif	global
or	with	assert	else
if	pass	yield	break
except	import	print	class
exec	in	raise	continue
finally	is	return	def
for	lambda	try	

Identifiers-: These are the names given to different parts of program like variables, objects, classes, functions etc.

Identifier forming rules of Python are :

- Is an arbitrarily long sequence of letters and digits
- The first character must be letter or underscore
- Upper and lower case are different
- The digits 0-9 are allowed except for first character
- It must not be a keyword
- No special characters are allowed other than underscore is allowed.
- Space not allowed
  - Valid identifiers: myfile, _file, no2ab
  - Invalid identifiers: 2no, na me,cs-12
  -

Literals/Values:

Literals are data items that have a fixed value. Python supports several kinds of literals:

- String Literal
- Numeric Literals
- Boolean Literals
- Special Literals – *None*
- Literal Collections

String Literals

- It is a collection of character(s) enclosed in a double or single quotes
- Examples of String literals
  - "Python"
  - '123456'
  - 'Hello Everyone'
  - Size of the string is determined by len() function.

Example: name='rajesh'

len(name)=6

Numeric Literals

- The numeric literals in Python can belong to any of the following numerical types:
  1) **Integer Literals:** it contain at least one digit and must not contain decimal point. It may contain (+) or (-) sign.
  2) **Floating point Literals:** also known as real literals. Real literals are numbers having fractional parts. It is represented in two forms Fractional Form or Exponent Form
- Numeric values with commas are not considered for int or float value, rather Python treats them as **tuple**.
- User can check the type of literal using type() function.

```
>>> a=10
>>> b=10.5
>>> print(a)
10
>>> print(b)
10.5
>>> c='Railway'
>>> print(c)
Railway
>>> print(type(a))
<class 'int'>
>>> print(type(b))
<class 'float'>
>>> print(type(c))
<class 'str'>
>>> d=1,4,8
>>> print(type(d))
<class 'tuple'>
```

Boolean Literals

- A Boolean literals in Python is used to represent one of the two Boolean values i.e. True or False
- These are the only two values supported for Boolean Literals

Special Literals *None*

Python has one special literal, which is None. It indicate absence of value. In other languages it is knows as NULL. It is also used to indicate the end of lists in Python.

Operators

Operators are symbol that perform specific operation when applied on variables i.e. operator operates on operand.

Some operator requires two operand and some requires only one operand to operate.

Example: a + b is an expression, a & b are operands and + is an operator.

Types of Operators

Unary operators: are those operators that require one operand to operate upon.

Binary Operators: These are those operators that require two operand to operate upon. Following are some Binary operators

Arithmetic Operators

Operator	Action
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
**	Exponent
//	Floor division

Conditional Operators / **Relation Operators**

- **It compares the values**

Operator	Meaning	Expression	Result
>	Greater Than	20 > 10	True
		15 > 25	False
<	Less Than	20 < 45	True
		20 < 10	False
==	Equal To	5 == 5	True
		5 == 6	False
!=	Not Equal to	67 != 45	True
		35 != 35	False
>=	Greater than or Equal to	45 >= 45	True
		23 >= 34	False
<=	Less than or equal to	13 <= 24	True
		13 <= 12	False

Logical Operators

Operator	Description	Example
and	Retrun true if both conditions are true	(4<5 and 5!=7), return true because both conditions are true
or	Retrun true if at least one condition is true	(5==9 or 'ram'>'rajesh'), return true because one condition is true.
not	reverse the condition	not(5==5), return false because it reverse true.

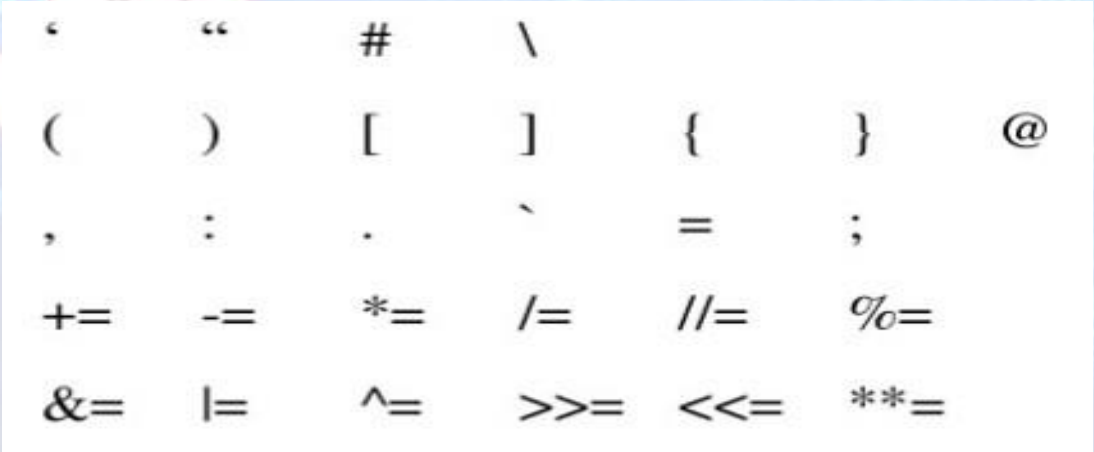
Assignment Operators

Operator	Description	Example	Example
=	Assigns values from right side operands to left side operand	a=b	a=5
+=	Add 2 numbers and assigns the result to left operand.	a+=b	a+=6 a=a+6
/=	Divides 2 numbers and assigns the result to left operand.	a/=b	a=a/b
*=	Multiply 2 numbers and assigns the result to left operand.	a*=b	a=a*b
-=	Subtracts 2 numbers and assigns the result to left operand.	a-=b	a=a-b
%=	modulus 2 numbers and assigns the result to left operand.	a%=b	a=a%b
//=	Perform floor division on 2 numbers and assigns the result to left operand.	a//=b b//=a	a=a//b b=b//a

Punctuators / Delimiters

- Used to implement the grammatical and structure of a Syntax.

Following are the python punctuators.



Input & Output in python:

input() function in python is used to allow user to give input values via keyboard in form of string.

Syntax is-: variable = input(<message to display>)

- In python, we use int() and float() functions to convert the string values in integer and float respectively.
- Example:

a=input("enter name")

b=int(input("enter your age"))

c=float(input("enter your percentage"))

so a will take string values, b will take integer values & c will take float values from user.

- Note: float values are fractional values like 7.5, 89.5, while string values are set of characters like 'ram', 'rakesh'. String values should always enclosed in ' ' or " ", otherwise it will give an error.

Conversion from one type to another

Python allows converting value of one data type to another data type.

If it is done by the programmer then it will be known as type conversion or type casting

If it is done by compiler automatically then it will be known as implicit type conversion.

Explicit type conversion To perform explicit type conversion, Python provide functions like int(), float(), str(), bool().

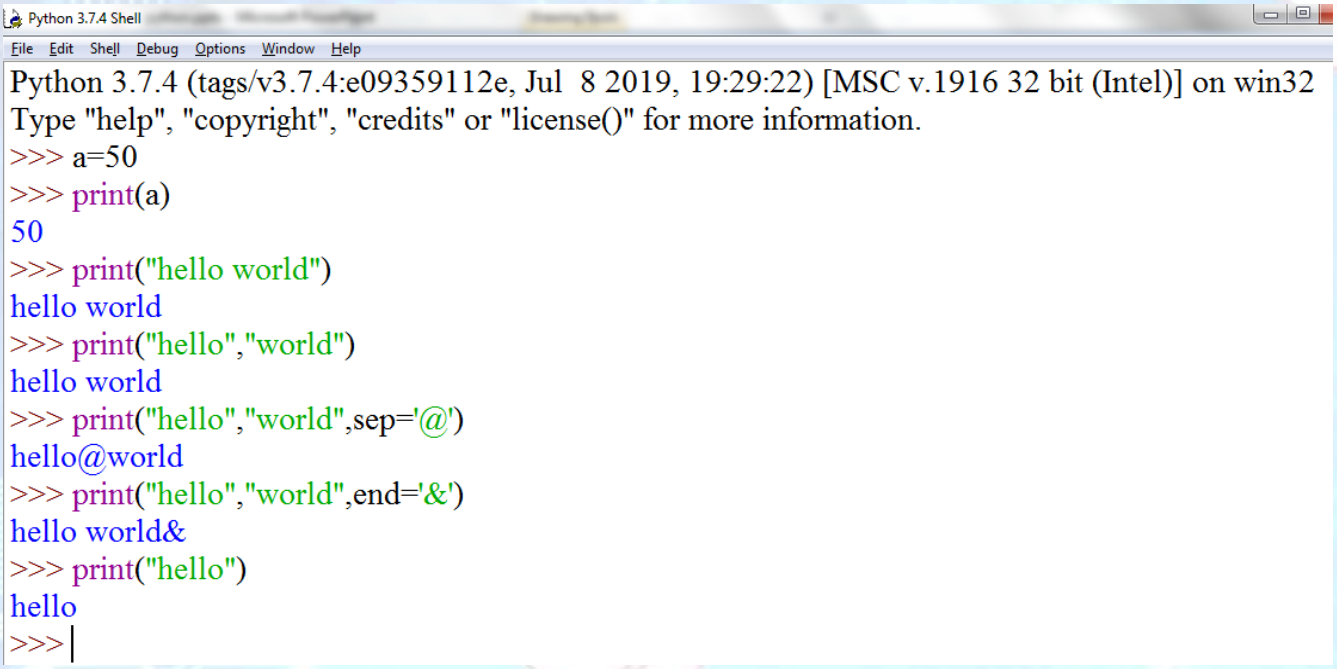
```
>>> a=input("enter price")
enter price50
>>> print(type(a))
<class 'str'>
>>> b=int(input("enter price"))
enter price50
>>> print(type(b))
<class 'int'>
>>>
```

```
>>> a=25.10
>>> b=int(a)
>>> print(b)
25
>>> c=float(b)
>>> print(c)
25.0
>>> d=str(b)
>>> print(d)
25
>>> print(type(d))
<class 'str'>
>>> print(type(b))
<class 'int'>
>>>
```

print() function in python is used to give output to user.

Syntax of print function: print(expression/value, sep, end)

- If sep(seperator) is not defined in print then by default it will take whitespace.
- If end keyword is not defined then it will take new line character by default.

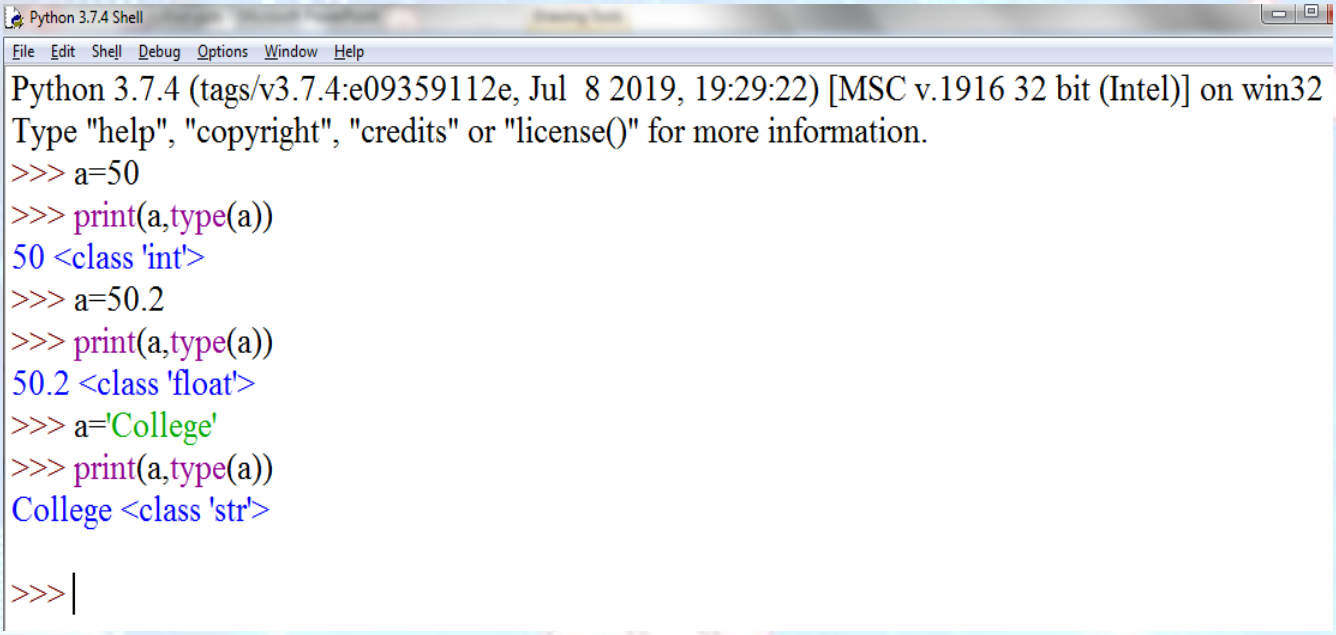


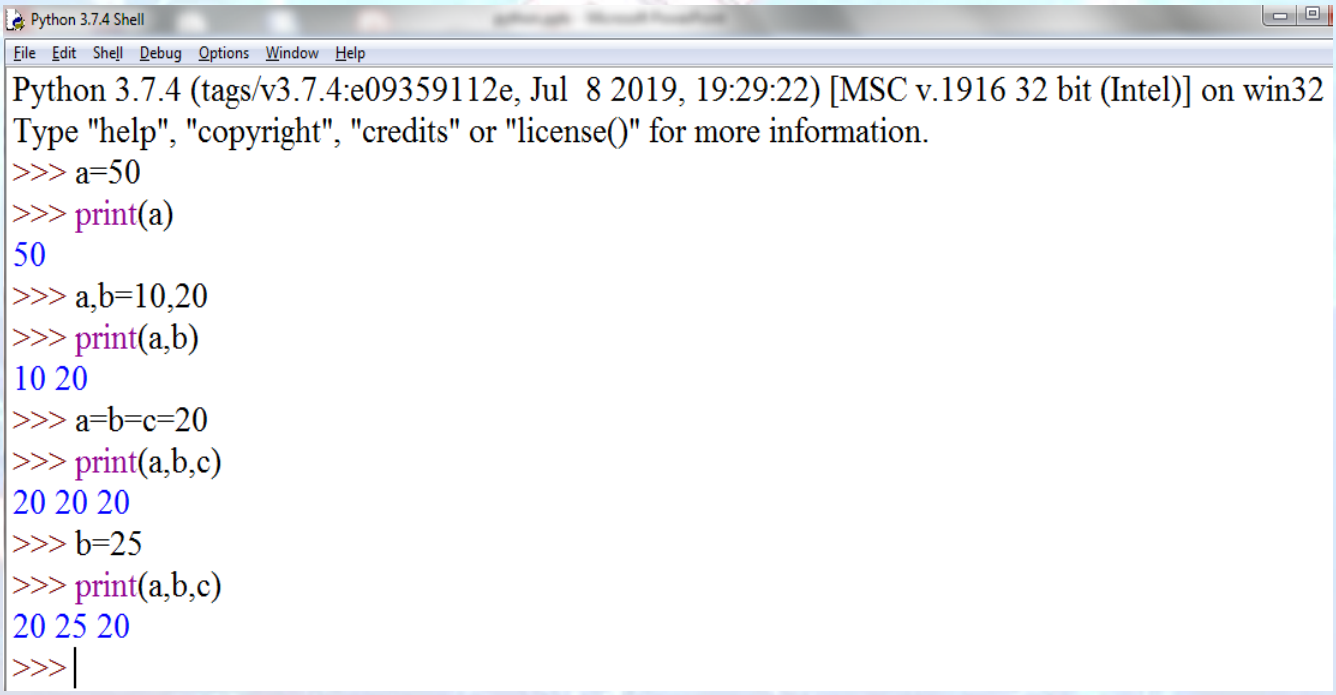
Open a new script file and type the following code & Execute the code:

```
num1=int(input("Enter Number 1 "))
num2=int(input("Enter Number 2 "))
 a = num1 + num2
 b=  num1 -  num2
 c=   num1 *  num2
 d=   num1/num2
print("Add =",a)
print("Subtraction =",b)
print("Multiplication =",c)
print("Division =",d)
```

Variable: variable is a location or container that holds some value. Python is dynamic typed programming language which means user don't have to declare the type of variable. When a value is assigned to a variable, the type is automatically assigned and data type of variable can be changed.

Lets see through example in which a is variable and value of a and type is changing:





In above example, we can see that we can assign multiple values to multiple variable, same value to multiple variables, so multiple assignment is possible in python.

DATA TYPES

Data type in Python specifies the type of data we are going to store in any variable, the amount of memory it will take and type of operation we can perform on a variable. Data can be of many types e.g. character, integer, real, string etc.

Python supports following data types:

- Numbers (int, float, complex)
- String
- List
- Tuple
- Dictionary

NUMBERS

Number data types are used to store numeric values. Numbers in Python can be of following types:

(i) Integers

a) Integers(signed)

b) Booleans

(ii) Floating point numbers

(iii) Complex Numbers

Integers

Integers allows to store whole numbers only and there is no fraction parts.

Integers can be positive and negative e.g. 100, 250, -12, +50

There are two integers in Python:

1) **Integers(signed) :** it is normal integer representation of whole numbers. Integers in python can be on any length, it is only limited by memory available. In Python 3.x int data type can be used to store big or small integer value whether it is +ve or –ve.

2) **Booleans:** it allows to store only two values **True** and **False**. The internal value of boolean value True and False is **1 and 0** resp. We can get boolean value from 0 and 1 using **bool()** function.

```
>>> int(True)
1
>>> bool(0)
False
>>>
```

FLOATING POINT NUMBERS

It allows to store numbers with decimal points. For e.g. 2.14. The decimal point indicate that it is not an integer but a float value. 100 is an integer but 100.5 is a float value.

COMPLEX NUMBERS

Python represent complex numbers in the form A+Bj. To represent imaginary numbers, Python uses j or J in place of i. Both real and imaginary parts are of type **float**

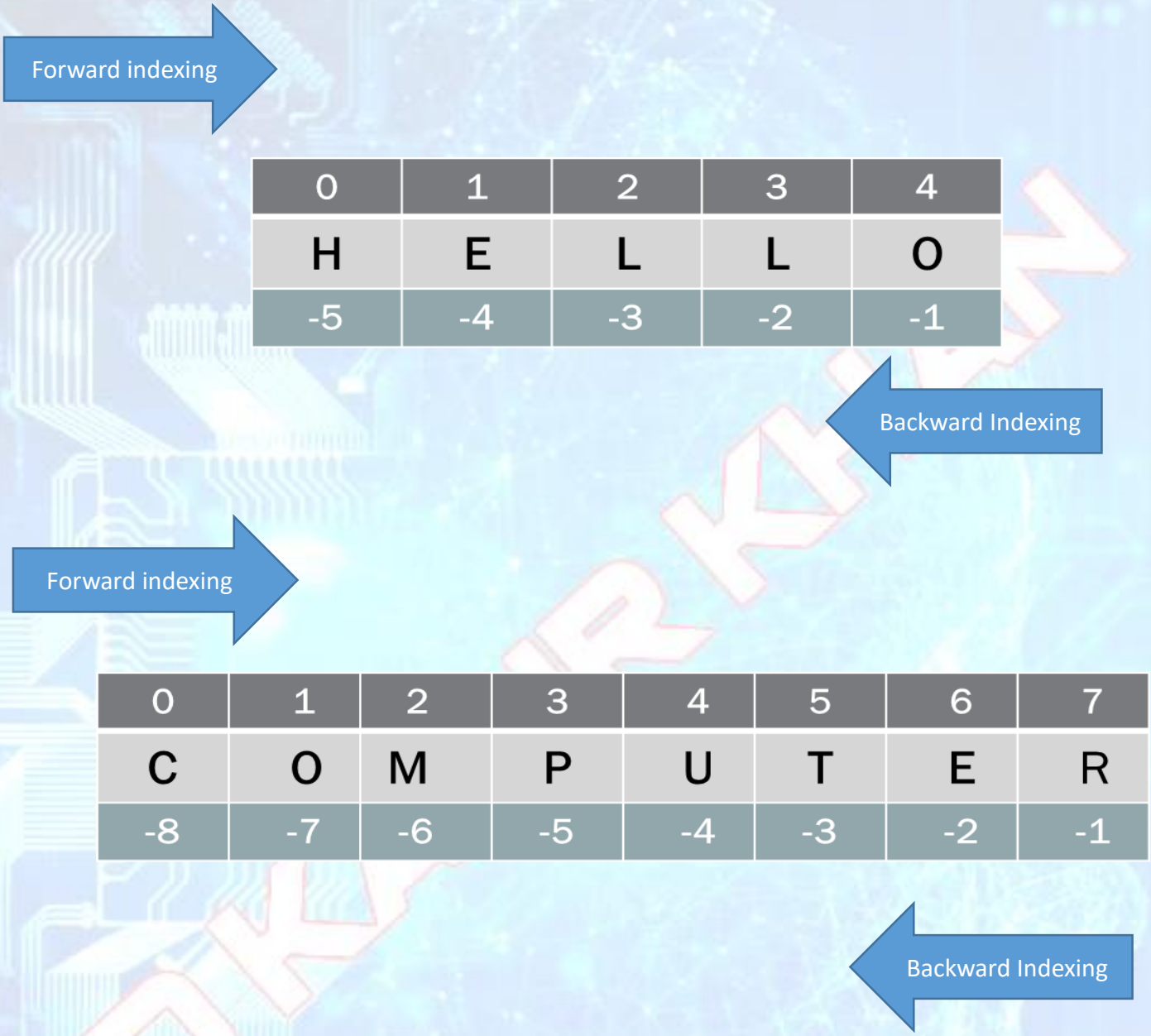
Python allows to retrieve real and imaginary part of complex number using attributes: **real** and **imag**. If the complex number is a then we can write **a.real** or **a.imag**

```
>>> a=61.57
>>> print(a)
61.57
>>> print(int(a))
61
>>> a=6+5j
>>> print(a)
(6+5j)

>>> print(a.real)
6.0
>>> print(a.imag)
5.0
>>>
```

STRING

- String is a collection of any valid characters in a quotation marks (single quotation or double quotation)
- Each character of String in Python is a Unicode character.
- Strings are used to store information like name, address, descriptions. For example: “hello”, “welcome“, “computer science”.
- In Python, string is a sequence of characters and each character can be individually access using **index**.
- From beginning, the first character in String is at index 0 and last will be at **len-1**.
- From backward direction, last character will be at index **-1** and first character will be at **–len.**



To access individual character of String, we can use the syntax:

StringName[index position]

```
>>> name="Rajesh"
>>> print(name)
Rajesh
>>> print(name[1])
a
>>> print(name[-2])
s
>>> print(name[5])
h
>>>
```

User cannot change the individual letters of string by assignment because string in python is immutable and Python will raise an error "**object does not support Item assignment**"

>>>name="Hello"

>>>name[1]='t' **# error**

However user can assign string to another string. Like

>>>name="Computer Science"

>>>name="Information technology"

>>>> print(name)

Information technology

LIST & TUPLES

- List and Tuple are compound data types i.e. they allows to store multiple values under one name of different data types.
- The main difference between List and Tuple is **List can be changed/modified i.e. mutable type** whereas **Tuple cannot be changes or modified i.e. immutable type**.

List: A list in python represents a list of comma-separated values of any data type between **square brackets**.

Example: [1,2,4,8,9], ['hello', 'everyone'], [1,'h','y',4.5]

```
>>> list1=[1,2,2,4]
>>> print(list1)
[1, 2, 2, 4]
>>> list1=['e','hello']
>>> print(list1)
['e', 'hello']
>>> list1=[1,2.5,'h','everyone']
>>> print(list1)
[1, 2.5, 'h', 'everyone']
>>>
```

- The values stored in List are internally indexed numbering from 0 onwards. i.e. first element will be at position 0 and second will be at 1 and so on.
- User can check the number of items in list using len() function

```
>>> list1=[1,4.2,'shyam','t']
>>> print(list1)
[1, 4.2, 'shyam', 't']
>>> print(list1[1])
4.2
>>> print(list1[2][1])
h
>>> print(len(list1))
4
>>>
```

Tuple: A tuple in python represents a list of comma-separated values of any data type between **parenthesis**. It is immutable data type i..e it can not be modified.

Like List, Tuple values are also internally numbered from 0 and so on for forward indexing and -1, -2 from backward indexing.

Example:

>>> favorites=("Blue","Cricket","Gajar Ka Halwa")

>>> print(favorites)

("Blue","Cricket","Gajar Ka Halwa")

>>>print(favorites[1])

Cricket

>>>print(favorites[-2])

Cricket

>>>favorites[0]='Football' #Error, tuple does not support assignment i.e. immutable

```
>>> grocery=['Rajma', 'Biscuit', 'Oil', 'Rice']
>>> print(grocery)
['Rajma', 'Biscuit', 'Oil', 'Rice']
>>> grocery[2]='Ghee'
>>> print(grocery)
['Rajma', 'Biscuit', 'Ghee', 'Rice']
>>>
```

DICTIONARY

- Dictionary is another feature of Python.
- It is an unordered set of comma separated **key:value** pairs. Dictionary Items are defined in Curly Brackets { }.
- Keys defined in Dictionary cannot be same i.e. no two keys can be same.
- Keys should be of immutable data types i.e. string, number or tuple.
- Dictionary is mutable. i.e. We can modify dictionary elements.
- In dictionary, values can be accessed using keys. The syntax is:

dictionary-name[key]

- User can modify dictionary elements in following syntax:

dictionary-name[key]=new value

```
>>> dict1={1:'Aakash',2:'Rajesh'}
>>> dict2={(1,2):5,'Oil':500}
>>> print(dict1)
{1: 'Aakash', 2: 'Rajesh'}
>>> print(dict2)
{(1, 2): 5, 'Oil': 500}
>>> print(dict1[2])
Rajesh
>>> print(dict2[(1,2)])
5
>>> dict2[1,2]=10
>>> print(dict2)
{(1, 2): 10, 'Oil': 500}
>>> dict3={[1,2]:'Rakesh',5:10}
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    dict3={[1,2]:'Rakesh',5:10}
TypeError: unhashable type: 'list'
```

Conditional Statements

While coding in Python, a lot of times we need to take decisions. For example, if a person needs to create a calculator with the help of a Python code, he/she needs to take in 2 numbers from the user and then ask the user about which function he/she wishes to operate. Now, according to the user's choice, the selection of function would change. In this case, we need the machine to understand what should happen when. This is where conditional statements help. Conditional statements help the machine in taking a decision according to the condition which gets fulfilled. There exist different types of conditional statements in Python.

There are basically three types of selection statement:

1. if statement
2. if….else statement
3. Nested if…else statement (if…..elif statement)

if statement: It is the basic form of selection statement. It checks the condition & if the condition is true, then statements under if will get executed & if the condition is false, then will not be executed.

```
Syntax:   if condition:
              statement1
              statement2
          statement3
```

if….else statement: It checks the condition & if condition is true then some statements will be executed and if condition is false then some other statements will be executed.

```
Syntax:   if condition:
              statement1(s)
          else:
              statement2(s)
```

program of senior citizen is example of if…else statement.

Some program exercise:

1. Write a program in python to decide whether a given number is even or odd.
2. Write a program in python to decide whether a number is divisible by 5 or not.

if….elif statement: it is used when multiple chain of conditions have to be checked. Each elif condition must be followed by condition and then followed by colon. After elif condition user can give else condition and else condition will get executed if all above condition evaluates to false.

```
Syntax:   if condition:
              statement1(s)
          elif condition:
              statement2(s)
          elif condition:
              statement3(s)
          else:
              statement4(s)
```

Lets take an example of finding a day based on number, for example if we will input the number as 1, it should give Monday, 2 for Tuesday and so on.

```
File  Edit  Format  Run  Options  Window  Help
a=int(input("enter number"))
if a==1:
    print("It is monday")
elif a==2:
    print("It is tuesday")
elif a==3:
    print("It is wednesday")
elif a==4:
    print("It is thursday")
elif a==5:
    print("It is friday")
elif a==6:
    print("It is saturday")
elif a==7:
    print("It is sunday")
else:
    print("Invalid choice")
```

```
Type "help", "copyright", "credits" o
>>>
==========
enter number4
It is thursday
>>>
==========
enter number2
It is tuesday
>>>
==========
enter number7
It is sunday
>>>
==========
enter number15
Invalid choice
>>>
```

Loop Statement

These statements are used to execute the statement(s) in repeated manner until the condition is true.

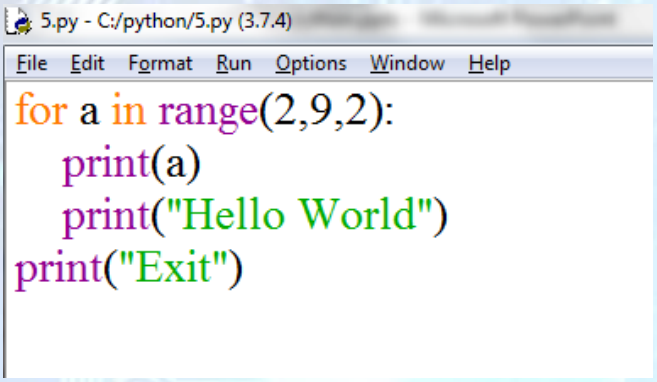
There are two types of loop statements:

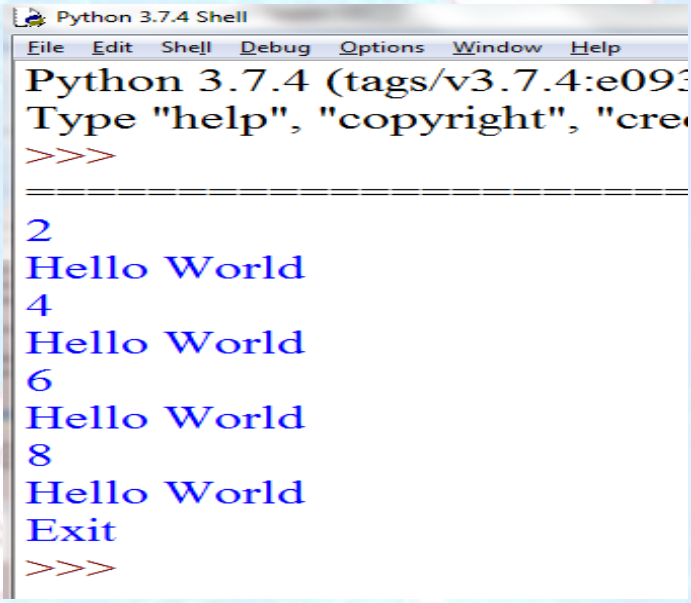
1. for loop
2. While loop
3. Do while loop

for loop-: It is used to iterate over a sequence like string, list. User can execute a set of statement, once for each item in a sequence.

Syntax:- for value in sequence: statement

- Lets take an example





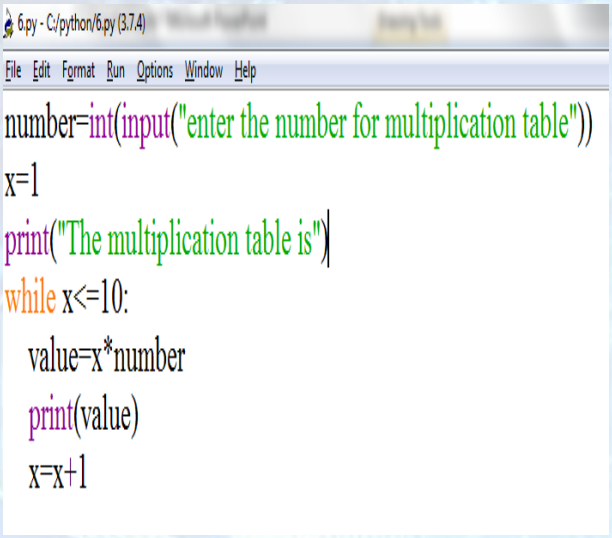
In above example, range(2,9,2) will generate a list of values [2,4,6,8] and each time **variable a** corresponds to single item one by one, so value of a will be 2,4,6,8 and for each time "Hello World" will be executed and after fourth time control will come out from the loop and print "Exit". print("Exit") is outside of loop so control will printed only once when it will come out from loop.

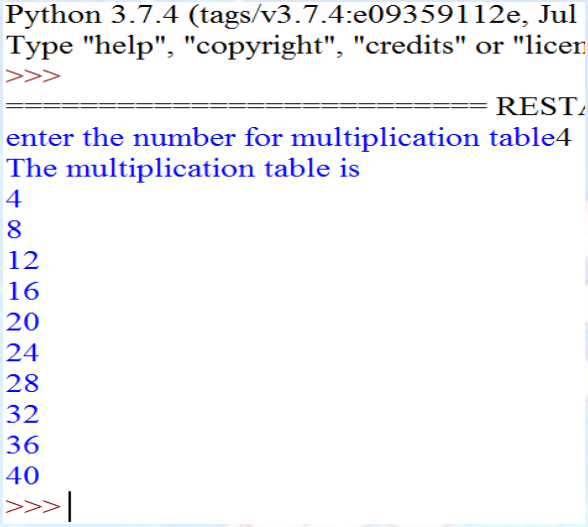
while loop-: It is used to execute a block of statement as long as a given condition is true & when the condition become false, the control will come out of the loop. The condition is checked every time at the beginning of the loop.

Syntax: while condition:

statement(s)

Lets take an example of printing multiplication table of a number which is given by user.





In above example, we have input a number from user using input() function & then initialize x to 1. In while loop we put the condition that loop body will execute again and again until value of x becomes more than 10 & we have multiplied x to number, but the main point is after printing the value we have to increase value of x by 1, so when it will go for printing the value the value of x is now 2 and value of 8 will get printed and so on.