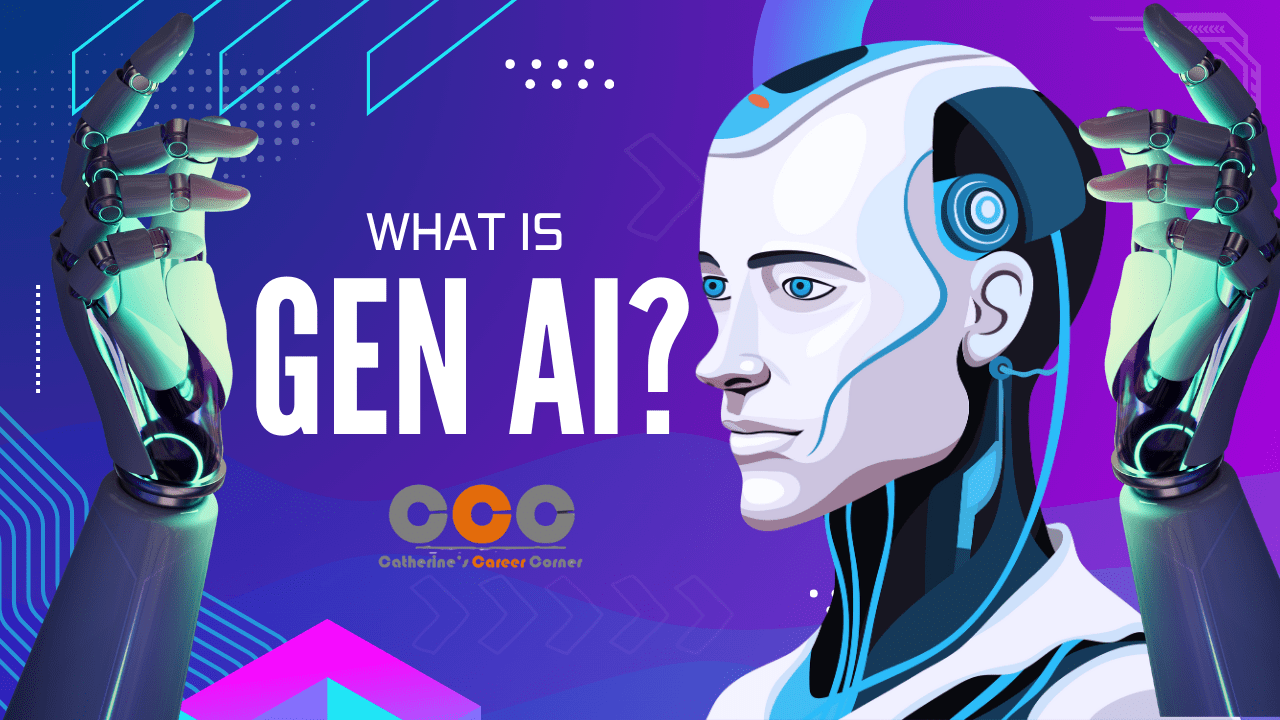
**GENERATIVE AI**

Generative AI is a type of artificial intelligence designed to create new content such as text, images, music or even code by learning patterns from existing data. These models generate original outputs that are often indistinguishable from human-created content. These models use techniques like deep learning and neural networks to generate output.

Unlike discriminative AI which focuses on classifying data into categories like spam vs. not spam, generative AI creates new data such as text, images, audio or video that resembles real-world examples.



How generative AI works

For the most part, generative AI operates in three phases:

* **Training**, to create a foundation model that can serve as the basis of multiple gen AI applications.
* **Tuning**, to tailor the foundation model to a specific gen AI application.
* **Generation**, **evaluation and retuning***,*to assess the gen AI application's output and continually improve its quality and accuracy.

**Training**

Generative AI begins with a foundation model, a deep learning model that serves as the basis for multiple different types of generative AI applications. The most common foundation models today are [large language models (LLMs)](https://www.ibm.com/think/topics/large-language-models), created for text generation applications, but there are also foundation models for image generation, video generation, and sound and music generation as well as multimodal foundation models that can support several kinds content generation.

To create a foundation model, practitioners train a deep learning algorithm on huge volumes of raw, unstructured, unlabeled data e.g., terabytes of data culled from the internet or some other huge data source. During training, the algorithm performs and evaluates millions of ‘fill in the blank’ exercises, trying to predict the next element in a sequence e.g., the next word in a sentence, the next element in an image, the next command in a line of code and continually adjusting itself to minimize the difference between its predictions and the actual data (or ‘correct’ result).

The result of this training is a [neural network](https://www.ibm.com/think/topics/neural-networks) of *parameters,*encoded representations of the entities, patterns and relationships in the data, that can generate content autonomously in response to inputs, or prompts.

This training process is compute-intensive, time-consuming and expensive: it requires thousands of clustered graphics processing units (GPUs) and weeks of processing, all of which costs millions of dollars. Open-source foundation model projects, such as Meta's Llama-2, enable gen AI developers to avoid this step and its costs.

**Tuning**

Metaphorically speaking, a foundation model is a generalist: It knows a lot about a lot of types of content, but often can’t generate specific types of output with desired accuracy or fidelity. For that, the model must be tuned to a specific content generation task. This can be done in a variety of ways.

Fine tuning

[Fine tuning](https://www.ibm.com/think/topics/fine-tuning) involves feeding the model labeled data specific to the content generation application questions or prompts the application is likely to receive, and corresponding correct answers in the desired format. For example, if a development team is trying to create a customer service chatbot, it would create hundreds or thousands of documents containing labeled customers service questions and correct answers, and then feed those documents to the model.

Fine-tuning is labor-intensive. Developers often outsource the task to companies with large data-labeling workforces.

Reinforcement learning with human feedback (RLHF)

In [RLHF](https://www.ibm.com/think/topics/rlhf), human users respond to generated content with evaluations the model can use to update the model for greater accuracy or relevance. Often, RLHF involves people ‘scoring’ different outputs in response to the same prompt. But it can be as simple as having people type or talk back to a chatbot or virtual assistant, correcting its output.

**Generation, evaluation, more tuning**

Developers and users continually assess the outputs of their generative AI apps, and further tune the model even as often as once a week for greater accuracy or relevance. (In contrast, the foundation model itself is updated much less frequently, perhaps every year or 18 months.)

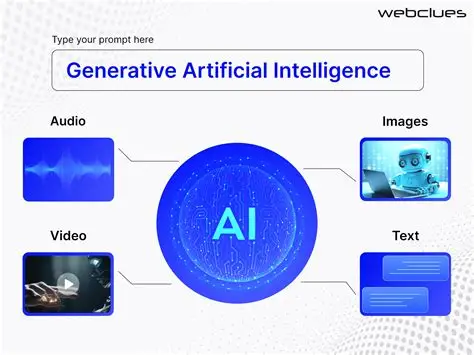
Another option for improving a gen AI app's performance is *retrieval augmented generation*(RAG). RAG is a framework for extending the foundation model to use relevant sources outside of the training data, to supplement and refine the parameters or representations in the original model.RAG can ensure that a generative AI app always has access to the most current information. As a bonus, the additional sources accessed via RAG are transparent to users in a way that the knowledge in the original foundation model is not.

A diagram of a brain

AI-generated content may be incorrect.

Some key technologies behind it include:  
• **Large Language Models (LLMs):** For generating text and code (e.g., GPT models).  
• **GANs (Generative Adversarial Networks):** For realistic image and video generation.  
• **Diffusion Models:** For high-quality image synthesis (used by DALL·E and Stable Diffusion).

**Types of Generative AI**

  
  
**1. Text Generation**  
  
Generative AI can produce human-like written content.  
• **Examples:** Writing essays, emails, poetry, articles, or chat replies.  
• **How it works:** Large Language Models (like GPT) predict the next word in a sequence to build sentences that make sense.  
• **Tools:** ChatGPT, Google Gemini, Claude, Jasper AI.  
  
*Use cases:*  
• Automated content creation  
• Summarization and translation  
• Chatbots and virtual assistants  
  
  
**2.** I**mage Generation**  
  
AI can generate images from text prompts or even enhance existing images.  
• **How it works:** Models learn from millions of images and their descriptions to understand objects, styles, and textures.  
• **Tools:** DALL·E, Midjourney, Stable Diffusion, Adobe Firefly.  
  
*Use cases:*  
• Graphic design and marketing  
• Fashion and product visualization  
• Entertainment and gaming (character or scene design)  
  
  
**3. Audio Generation**  
  
AI models can generate music, voices, or sound effects that sound natural.  
• **How it works:** Models learn sound patterns like pitch, rhythm, and tone from large audio datasets.  
• **Tools:** Suno, ElevenLabs, OpenAI Jukebox.  
  
*Use cases:*  
• Music and podcast creation  
• Voiceovers for videos or virtual assistants  
• Personalized soundscapes and gaming audio  
  
  
**4. Code Generation**  
  
Generative AI can write or complete programming code.  
• **How it works:** Models are trained on vast code repositories to learn syntax, logic, and structure of various programming languages.  
• **Tools:** GitHub Copilot, ChatGPT, Amazon CodeWhisperer.  
  
*Use cases:*  
• Assisting developers in writing or debugging code  
• Automating repetitive coding tasks  
• Learning and documentation support

A diagram of a human head

AI-generated content may be incorrect.  
  
 **Real-World Impact**  
  
Generative AI is transforming industries such as:  
• **Education** – Personalized learning content  
• **Healthcare** – Medical image generation and report drafting  
• **Marketing** – Ad copy, visuals, and campaign ideas  
• **Software development** – Code automation and optimization  
  
  
**Challenges and Ethical Concerns**  
  
While powerful, Generative AI also comes with challenges:  
• **Bias and misinformation** (models reflect data they were trained on)  
• **Deepfakes and misuse** (fake images/videos)  
• **Copyright issues** (using or recreating existing content)  
• **Data privacy** concerns  
  
Responsible usage and regulation are key to ensuring it benefits society positively.