

Chapter 1: Programming and Introduction to Python

1. Setting the Stage

Activity 1: Match the Concepts

Match each programming-related word in **Column A** with the correct definition in **Column B**.



Column A: Terms	Column B: Definitions
1. Algorithm	(A) Named storage locations that hold data.
2. Code	(B) A set of written commands that a computer understands.
3. Syntax	(C) Code structures that repeat a set of instructions.
4. Debugging	(D) A program that translates code into machine-readable instructions.
5. Python	(E) The rules for writing code correctly in a programming language.
6. Compiler	(F) A high-level programming language known for simplicity.
7. Variables	(G) A step-by-step set of instructions to solve a problem.
8. Loops	(H) Finding and fixing errors in a program.

Activity 2: Think & Share

If humans use different languages, what do computers use? How do they 'understand' instructions?

Activity 3: Fact or Myth?

Decide if each statement is **true** or **false** and explain why.

"Python is one of the easiest programming languages to learn." T ☐ F ☐

"You must be a math genius to code." T ☐ F ☐

"Programming languages are like human languages - they evolve over time."

T ☐ F ☐

"Python was named after a snake." T ☐ F ☐

"Once you learn one programming language, you can't switch to another."

T ☐ F ☐

2. Take a Listen



Activity 4: Complete the Passage

Fill in the blanks using the following words to complete the sentences correctly:

*Kickstarter - Dodgeball - Passions - Hour of Code -
Interactable item Allah Ben - Pick up - Virtual Reality - Fundamental*

1. VR stands for _____, which is a computer-generated three-dimensional environment.
2. The students wanted to create a _____ game in virtual reality.
3. The main problem they faced at the beginning was that they could not _____ the ball.
4. To fund their project, they launched a _____ campaign and raised almost \$5,000.
5. The students struggled with coding because their experience was limited to a school program called _____.

6. One major mistake Ben made was renaming a code from "interactable item" to _____, which caused 200 error messages.
7. Eventually, they learned that breaking their goals into _____ interactions helped them progress.
8. The students believe that coding should be taught in three parts: learning on their own, receiving occasional guidance, and connecting to their _____.

Activity 5: Choose the Best Answer

Choose the correct answer from the options.

1. What did the students initially assume about making a VR game?
 - a) It would be very expensive
 - b) It would be easy and fun
 - c) It would take years to complete
 - d) It would not require much coding
2. What software did they need to purchase to develop their game?
 - a) Unreal Engine
 - b) Python
 - c) They didn't specify
 - d) JavaScript
3. What did the students do when they encountered their first major coding problem?
 - a) Gave up and started a new project
 - b) Hired a professional coder
 - c) Searched the internet for a solution
 - d) Asked their teacher for help
4. How did Ben accidentally break the code?
 - a) He deleted important files
 - b) He renamed a critical function incorrectly
 - c) He copied and pasted the wrong code
 - d) He installed the wrong software
5. What was the main issue with the "Hour of Code" program in schools?
 - a) It lasted only one hour per year
 - b) It was too difficult for students

- c) It was only available to certain students
 - d) It required expensive software
6. How did they fix their coding mistake?
- a) They rewrote the entire code from scratch
 - b) They hired a professional
 - c) They figured it out by breaking down the problem
 - d) They gave up on that part of the project
7. What lesson did they learn from their experience?
- a) Coding is impossible without formal education
 - b) Mistakes help you learn and improve
 - c) Coding should always be left to professionals
 - d) Learning coding in school is enough
8. What is one of their recommendations for teaching coding?
- a) Children should rely on teachers for all help
 - b) Coding should be learned only in college
 - c) Children should learn by doing and experimenting
 - d) Coding should not be taught in schools

Activity 6: Fill in the Missing Words

Watch the video once more and complete the sentences with relevant words or phrases

1. The students initially believed that making a VR game would be _____ and _____.
2. Their first major coding challenge was that they couldn't _____ the ball in the game.
3. They lacked experience because their school's coding program, called _____, only lasted for one hour per year.
4. To fund their project, they used _____ and managed to raise almost \$ _____.
5. When Ben renamed a key function incorrectly, they received over _____ error messages.
6. They discovered that breaking down their coding goals into _____ interactions helped them progress.
7. One of their key takeaways was that making mistakes in coding helped them _____.

- 8.They believe that coding should be taught by letting children learn _____,
with occasional guidance from adults.

3. Read & Reflect

History of Python

Python is a widely used general-purpose, high-level programming language. It was initially designed by **Guido van Rossum** in **1991** and developed by Python Software Foundation. It was mainly developed to emphasize code readability, and its syntax allows programmers to express concepts in fewer lines of code.



Who Invented Python?

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application-based work in December of 1989 at Centrum Wiskunde & Informatica (CWI) which is situated in the Netherlands. It was started as a hobby project because he was looking for an interesting project to keep him occupied during Christmas.

The programming language in which [Python](#) is said to have succeeded is ABC Programming Language, which had interfacing with the [Amoeba Operating](#)



[System](#) and had the feature of exception handling. He had already helped create ABC earlier in his career and had seen some issues with ABC but liked most of the features. After that what he did was very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and created a good scripting language that had removed all the flaws.

Why Python called Python?

The inspiration for the name came from the BBC's TV Show - '**Monty Python's Flying Circus**', as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the "Benevolent dictator for life" (BDFL) until he stepped down from the position as

the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.

Evolution of Python

When it was released in 1991, it used a lot fewer codes to express the concepts, when we compare it with [Java](#), [C++](#) & [C](#). Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. It had more than enough capability to provide classes with inheritance, several core data types of exception handling, and functions. For various purposes such as developing, scripting, generation, and , this language is utilized. Due to its elegance and simplicity, top technology organizations like Dropbox, Google, Quora, Mozilla, Hewlett-Packard, Qualcomm, IBM, and Cisco have implemented Python. Python has come a long way to become the most popular coding language in the world. Python has just turned 30 and just recently at pycon22(python conference) a new feature was released by Anaconda foundation. It's known as [pyscript](#). With this, now Python can be written and run in the browser like [javascript](#) which was previously not possible, but it still has that unknown charm & X factor.

Python has been an inspiration for many other coding languages such as [Ruby](#), Cobra, Boo, CoffeeScript ECMAScript, Groovy, Swift, Go, OCaml, [Julia](#), etc.

Python history shows how it has grown to become a key player in the programming world due to its ease of use and strong community support. Python is not only easy to learn but also supported by a community that keeps making it better. Learning Python can really boost your career, thanks to its popularity and wide application across different fields.

What are the uses of Python?

One more question arises that, [Is Python Programming or Scripting?](#) According to Linux Journal readers, Python is a programming as well as a scripting language. GeeksforGeeks. (2025, January). History of Python. GeeksforGeeks. Retrieved February 10, 2025, from <https://www.geeksforgeeks.org/history-of-python/>

Activity 7: The History of Python

Complete the following passage using the correct words from the word bank. **Each blank corresponds to a single word or phrase.** Ensure your choices fit grammatically and contextually.

• *Word Bank: Amoeba, Benevolent Dictator for Life, ABC, simplicity, 2018, exception, scripting, Guido van Rossum, browser, Python Software Foundation, high-level, Swift, versatile, Christmas, evolving, Ruby, design, Google, fewer, Monty Python's Flying Circus, Mozilla, British, PyScript, snake*

The History of Python

Python is a (1)_____ programming language known for its (2)_____ and readability. It was initially designed by (3)_____ in 1991 and later developed under the (4)_____ Foundation. The idea for Python originated in the (5)_____, when its creator sought an engaging project to work on during (6)_____.

Python is considered the successor to the (7)_____ Programming Language, which interfaced with the (8)_____ Operating System and featured (9)_____ handling. However, its creator found some limitations in ABC and improved upon them by (10)_____ its syntax and fixing its flaws.

Interestingly, the language's name was not inspired by the (11)_____, but rather by a (12)_____ TV show called (13)_____, which its creator enjoyed. He held the title of (14)_____ (BDFL) until his resignation in (15)_____.

Python gained popularity due to its (16)_____ philosophy, emphasizing readability and developer productivity. Unlike Java or C++, Python required (17)_____ lines of code to express the same concepts. This efficiency made it widely adopted by top companies such as (18)_____ and (19)_____.

Over the years, Python has influenced various modern languages, including (20)_____ and (21)_____. With the introduction of (22)_____ by the Anaconda Foundation at PyCon22, Python can now be executed in the (23)_____ like JavaScript. This has further cemented its role as a (24)_____ language in programming and scripting.

Activity 8: Fact Check

Determine whether the statements below are True or False. If False, correct them.

1. Python is a low-level programming language.
2. Python's syntax allows programmers to write more lines of code compared to other languages.
3. Facebook does not use Python.
4. Python is used for machine learning and AI applications.
5. Google uses Python only for front-end development.

Activity 9: Python Applications

Match the Python application with its correct use case.

Python Application	Use Case
AI and Machine Learning	A. Data analysis and backend services
Google	B. Running the world's largest deployment of Django
Instagram	C. One of the main server-side languages used by a tech giant
Spotify	D. Developing intelligent algorithms using stable and flexible programming

Activity 10: Justify Your Answer

Instructions: Answer the following questions in 2-3 sentences.

1. Why is Python a popular choice for AI and machine learning?
2. What advantages does Python provide to companies like Spotify and Instagram?
3. How does Python's readability impact productivity in software development?

Activity 11: Tech Terminology

Fill in the blanks with the correct term from the list below:

TensorFlow, CPU, Service end, GPU, Django, Spotify, TPU, Scikit-learn, Frontend development.

1. The _____ (Central Processing Unit) is the brain of a computer, responsible for executing instructions.
2. A _____ (Graphics Processing Unit) is optimized for parallel processing and is commonly used in gaming and AI applications.
3. A _____ (Tensor Processing Unit) is a hardware accelerator developed by Google specifically for machine learning tasks.
4. _____ is an open-source machine learning framework developed by Google, often used in AI and deep learning applications.

5. _____ is a popular Python library for machine learning that provides simple and efficient tools for data mining and analysis.
6. _____ is a well-known music streaming service that uses machine learning for personalized recommendations.
7. _____ is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
8. _____ refers to the part of web development that deals with the user interface and experience, usually involving HTML, CSS, and JavaScript.
9. The term _____ refers to the backend part of a system where core processing and logic take place.

Activity 12: Mastering Tech Terms

Instructions:

Select the correct answer for each question.

1. What does **CPU** stand for?
 - a) Central Programming Unit
 - b) Core Processing Utility
 - c) Central Processing Unit
 - d) Computer Processing Unit
2. Which hardware component is primarily used for parallel processing in gaming and AI applications?
 - a) CPU
 - b) GPU
 - c) TPU
 - d) RAM
3. What is **TPU** mainly designed for?
 - a) Image rendering
 - b) Machine learning acceleration
 - c) Web development
 - d) Storage management
4. Which of the following is an open-source machine learning framework developed by Google?
 - a) Scikit-learn
 - b) TensorFlow
 - c) Spotify
 - d) Django

5. _____ is a popular Python library used for machine learning and data analysis.
 - a) TensorFlow
 - b) Spotify
 - c) Scikit-learn
 - d) Django
6. Which of the following is a music streaming service that uses AI for personalized recommendations?
 - a) TensorFlow
 - b) Django
 - c) Spotify
 - d) Scikit-learn
7. What is **Django** primarily used for?
 - a) Machine learning
 - b) Web development
 - c) Video processing
 - d) Hardware acceleration
8. Which field focuses on designing and building the user interface of websites and applications?
 - a) Backend development
 - b) Frontend development
 - c) Database management
 - d) Cybersecurity
9. The **service end** of a system typically refers to:
 - a) The part that interacts directly with users
 - b) The backend where logic and data processing occur
 - c) The hardware of a computer
 - d) A cloud storage solution
10. Which of the following best describes **GPU**?
 - a) A hardware accelerator for deep learning
 - b) A processing unit optimized for graphics tasks
 - c) A cloud-based AI service
 - d) A framework for frontend development

4. Grammar in Action

Activity 13: Spot the Difference

Read the two sentences in each pair. Identify which one sounds more formal and why.

1. (a) The company decided to expand its services.
(b) The company made a decision to expand its services.
2. (a) The government analyzed the situation carefully.
(b) The government conducted an analysis of the situation carefully.



Nominalization

Rules in Use

- ◆ Definition: Turning verbs/adjectives into nouns.
- ◆ Examples: decide → decision, analyze → analysis.
- ◆ Key Structures: Verbs → Nouns, Adjectives → Nouns.

Usage & Application

- ✓ Academic Writing: More formal, structured writing.
- ✓ Professional Communication: Clarity in reports.
- ✓ Technical Contexts: Precision in legal & scientific texts.

Common Pitfalls

- ⚠ Overuse: Can make writing too complex.
- ⚠ Loss of Clarity: Avoid unnecessary nominalization.
- ⚠ Grammar Hacks: Use strategically for conciseness.

Activity 14: Nominalizing Python's Core Ideas

Transform the verbs in the following sentences into nouns, creating new sentences with nominalizations.

1. Guido van Rossum *created* Python in the late 1980s.
2. Many programmers *use* Python for web development because of its frameworks like Django and Flask.
3. Python *emphasizes* code readability by using significant indentation.
4. Google *adopted* Python early on, which contributed to its popularity.
5. Python *handles* memory management automatically with garbage collection.
6. Python is often *used* for data science because it has libraries such as NumPy and Pandas.
7. The Python community *develops* new packages and libraries continuously.
8. Python *supports* multiple programming paradigms, including object-oriented, imperative, and functional styles.
9. Python *interprets* code at runtime, making it a dynamic language.
10. Python *simplifies* complex tasks, which makes it popular for scripting and automation.

Activity 15: Pythonic Nouns

For each of the following sentences related to Python programming, choose the best word from the options provided to complete the sentence.

1. Python *supports* multiple programming paradigms.
Python's _____ of multiple programming paradigms makes it a versatile language.
(a) support
(b) supporting
(c) supporter
(d) supports
2. Developers often *refactor* code to improve its readability.
Code _____ is a common practice to enhance readability.
(a) refactor
(b) refactoring
(c) refactored
(d) refactors
3. The interpreter *executes* Python code line by line.
The _____ of Python code by the interpreter happens line by line.

- (a) execute
 - (b) execution
 - (c) executing
 - (d) executes
4. Libraries like NumPy *enable* efficient numerical computation.
The _____ of efficient numerical computation is made possible by libraries like NumPy.
- (a) enable
 - (b) enabling
 - (c) enablement
 - (d) enables
5. Python *encourages* code reuse through its module system.
Python's _____ of code reuse is facilitated by its well-designed module system.
- (a) encourage
 - (b) encouraging
 - (c) encouragement
 - (d) encourages
6. Effective testing *validates* the correctness of software.
The thorough _____ of software correctness is essential for building reliable applications.
- (a) validate
 - (b) validating
 - (c) validation
 - (d) validates
7. Type hints *clarify* the intended data types in Python 3.
The _____ provided by type hints enhances code readability and maintainability in Python 3.
- (a) clarify
 - (b) clarifying
 - (c) clarification
 - (d) clarifies
8. Good API design *abstracts* away implementation details.
Effective API design achieves _____ of underlying implementation complexities, promoting modularity.

- (a) abstract
- (b) abstracting
- (c) abstraction
- (d) abstracts

9. Version control systems *track* changes to the codebase.

The meticulous _____ of codebase modifications is a vital function of version control systems.

- (a) track
- (b) tracking
- (c) tracker
- (d) tracks

10. Secure coding practices *mitigate* potential vulnerabilities.

The consistent _____ of potential vulnerabilities is a key element of secure software development.

- (a) mitigate
- (b) mitigating
- (c) mitigation
- (d) mitigates

5. Speak & Debate

Activity 16: The Future of Programming: Human vs. AI Programmers" Debate

- AI tools like ChatGPT and GitHub Copilot are already writing code.
- Some believe AI will replace human programmers, while others think humans will always be needed.



Will AI replace human programmers in the future?

Do you think AI will completely take over programming? Or do you believe humans will always be necessary? Why?

- Some believe Humans will always be better programmers than AI.
- Others think AI will eventually replace human programmers.

Guiding Questions for Discussion:

For those who believe humans will always be needed:

- Why is human creativity important in programming?
- What are some limitations of AI in writing code?
- Can AI fully understand user needs or ethical issues?

For those who believe AI will take over programming:

- How can AI be faster and more efficient than humans?
- Can AI learn and improve beyond human capabilities?
- What are the financial benefits of using AI over human programmers?

6. Writing with Might



Write a well-structured paragraph answering the question: **Why is programming an essential skill in today's digital world?**

- Begin with a clear **topic sentence** stating your main idea.
- Provide **at least two reasons** why learning to code is important.
- Use **examples** to support your argument.
- Conclude by summarizing your main point.