

প্রোগ্রামিং: পরিচিতি

প্রোগ্রামিং হলো মেশিন বা কম্পিউটারকে নির্দিষ্ট কাজের জন্য নির্দেশনা দেওয়ার প্রক্রিয়া। এই নির্দেশনাগুলো একটি প্রোগ্রামের আকারে তৈরি হয়, যা কম্পিউটার বুঝতে পারে এবং সেই অনুযায়ী কার্যকর করতে পারে। সহজ ভাষায় বলতে গেলে, প্রোগ্রামিং হলো কম্পিউটারের জন্য এমন একটি প্রক্রিয়া, যার মাধ্যমে একটি নির্দিষ্ট সমস্যার সমাধান বা কাজ সম্পন্ন করার জন্য নির্দেশনা প্রদান করা হয়।

প্রোগ্রামিং ভাষা

প্রোগ্রামিং লেখার জন্য বিভিন্ন ভাষা রয়েছে, যেমন:

C
C++
Python
Java

উদাহরণ

যদি আমরা একটি প্রোগ্রাম তৈরি করি যা কম্পিউটারকে নির্দিষ্ট সংখ্যাগুলো গুনতে বলে, তাহলে সেই প্রোগ্রামে কম্পিউটারকে কীভাবে এবং কতবার গুনতে হবে সে সম্পর্কিত নির্দেশনা প্রদান করা হয়।

পিক্সেল কী?

পিক্সেল হলো একটি ছোট ডট বা বিন্দু, যা কম্পিউটার বা ডিজিটাল ডিসপ্লেতে পরিমাপের একটি মৌলিক একক হিসেবে ব্যবহৃত হয়। প্রতিটি পিক্সেল একটি নির্দিষ্ট রঙ ধারণ করে এবং এই ছোট ডটগুলো একসঙ্গে মিলিত হয়ে একটি পূর্ণাঙ্গ চিত্র তৈরি করে।

পিক্সেলের বৈশিষ্ট্য

প্রতিটি পিক্সেল একক রঙ ধারণ করে।

পিক্সেলের সংখ্যা এবং ঘনত্ব কোনো চিত্রের রেজোলিউশন নির্ধারণ করে।

উচ্চ রেজোলিউশনের ডিসপ্লেতে পিক্সেলের ঘনত্ব বেশি থাকে, ফলে ছবির মান উন্নত এবং পরিষ্কার হয়। উদাহরণস্বরূপ, একটি 4K রেজোলিউশন ডিসপ্লেতে পিক্সেলের সংখ্যা বেশি হওয়ায় ছবির মানও অনেক উন্নত হয়।

মেশিন কোড কী?

মেশিন কোড হলো কম্পিউটারের নিজস্ব ভাষা, যা ০ এবং ১ (বাইনারি সংখ্যা) দিয়ে তৈরি। এটি সরাসরি কম্পিউটারের প্রসেসর বা CPU দ্বারা বোঝা এবং কার্যকর করা যায়।

মেশিন কোডের বৈশিষ্ট্য

- বাইনারি ভাষা: এটি শুধুমাত্র ০ এবং ১ নিয়ে গঠিত।
- দ্রুত কার্যক্ষমতা: কম্পিউটারের হার্ডওয়্যার সরাসরি এই কোড বুঝতে পারে এবং কার্যকর করতে পারে।
- লিখতে কঠিন: মেশিন কোড বোঝা বা লিখতে অত্যন্ত জটিল এবং সময়সাপেক্ষ।

উচ্চ স্তরের ভাষার ব্যবহার

মেশিন কোডের এই জটিলতার কারণে প্রোগ্রামাররা সাধারণত উচ্চ স্তরের প্রোগ্রামিং ভাষা যেমন C, Python, বা Java ব্যবহার করেন। এই ভাষাগুলো সহজবোধ্য এবং এগুলো পরে কম্পাইলারের মাধ্যমে মেশিন কোডে রূপান্তরিত হয়।

অনুবাদক প্রোগ্রাম: পরিচিতি

অনুবাদক প্রোগ্রাম হলো এমন একটি সফটওয়্যার, যা প্রোগ্রামারদের লেখা কোডকে কম্পিউটার বা মেশিনের জন্য বোধগম্য ভাষায় অনুবাদ করে। যেহেতু কম্পিউটার কেবলমাত্র মেশিন ল্যাঙ্গুয়েজ (০ এবং ১) বুঝতে পারে, তাই প্রোগ্রামারদের লেখা উচ্চ স্তরের ভাষাগুলো (যেমন Python, C, Java) সরাসরি কম্পিউটার বোঝে না। অনুবাদক প্রোগ্রাম সেই ভাষাগুলোকে মেশিন কোডে রূপান্তরিত করে, যাতে কম্পিউটার তা বুঝে কার্যকর করতে পারে।

অনুবাদক প্রোগ্রামের ধরন

অনুবাদক প্রোগ্রাম প্রধানত তিনটি শ্রেণিতে বিভক্ত:

১. কম্পাইলার

কাজের ধরন: পুরো প্রোগ্রামকে একবারে মেশিন কোডে অনুবাদ করে এবং একটি আলাদা এক্সিকিউটেবল ফাইল তৈরি করে।
উদাহরণ: C, C++, এবং Java প্রোগ্রামিং ভাষায় ব্যবহৃত হয়।
সুবিধা: দ্রুত কার্যক্ষম। প্রোগ্রাম একবার কম্পাইল হয়ে গেলে কম্পাইলার আর প্রয়োজন হয় না।
অসুবিধা: পুরো প্রোগ্রাম লেখার পর ত্রুটি সনাক্ত হয়, যা সংশোধন করতে সময় বেশি লাগে।

২. ইন্টারপ্রেটার

কাজের ধরন: প্রোগ্রামের কোডকে এক লাইন বা একটি নির্দেশনায় অনুবাদ করে তাৎক্ষণিকভাবে কার্যকর করে।
উদাহরণ: Python, JavaScript, এবং Ruby প্রোগ্রামিং ভাষায় ব্যবহৃত হয়।
সুবিধা: ত্রুটি সনাক্ত করা দ্রুত এবং ডিবাগিং সহজ।
অসুবিধা: প্রতিটি লাইনের কোড বারবার অনুবাদ হওয়ায় প্রোগ্রামের কার্যক্ষমতা তুলনামূলক ধীর।

৩. অ্যাসেম্বলার

কাজের ধরন: অ্যাসেম্বলি ভাষায় লেখা কোডকে মেশিন কোডে রূপান্তর করে।
বৈশিষ্ট্য: অ্যাসেম্বলি ভাষা হলো মেশিন কোডের কাছাকাছি নিম্ন স্তরের ভাষা।
উদাহরণ: মাইক্রোকন্ট্রোলার বা ইলেকট্রনিক ডিভাইসের ক্ষেত্রে ব্যবহৃত হয়।

অনুবাদক প্রোগ্রামের ভূমিকা

কোড অনুবাদ: প্রোগ্রামারের কোডকে মেশিনের বোধগম্য ভাষায় রূপান্তর করে।
ত্রুটি সনাক্তকরণ: কোডে কোনো ভুল থাকলে তা নির্দেশ করে।
কোড অপটিমাইজেশন: কার্যক্ষমতা বাড়ানোর জন্য কোড অপটিমাইজ করে।
অনুবাদক প্রোগ্রাম ছাড়া উচ্চ স্তরের প্রোগ্রামিং ভাষায় লেখা কোড কম্পিউটারে কার্যকর করা সম্ভব নয়।

Python প্রোগ্রামিং শুরু করার ধাপ

Python হলো ইন্টারপ্রেটার-ভিত্তিক একটি প্রোগ্রামিং ভাষা, যা সরাসরি কোড লাইনে লাইনে কার্যকর করে। Python প্রোগ্রামিং শুরু করতে হলে প্রথমে Python Interpreter ইনস্টল করতে হবে।

Python Interpreter ইনস্টল করার ধাপ

১. Python ওয়েবসাইটে যান
Python ডাউনলোড করার জন্য www.python.org ওয়েবসাইটে যান।
২. ডাউনলোড এবং ইনস্টল করুন
আপনার অপারেটিং সিস্টেম অনুযায়ী Python ডাউনলোড করুন। ইনস্টল করার সময় “Add Python to PATH” চেকবক্সটি অবশ্যই সিলেক্ট করুন, যাতে এটি স্বয়ংক্রিয়ভাবে PATH-এ যুক্ত হয়।
৩. ইন্সটলেশন চেক করুন
ইনস্টল হয়ে গেলে Command Prompt বা Terminal খুলে নিচের কমান্ডগুলো ব্যবহার করে নিশ্চিত করুন:

```
python --version
```

অথবা, শুধু python লিখে এন্টার চাপুন। যদি পাইথনের ভার্সন প্রদর্শিত হয়, তবে ইন্সটলেশন সফল হয়েছে। Python ইন্সটলেশন সফল হলে আপনি পাইথনের কোড লিখে তা সরাসরি চালাতে পারবেন।

সংক্ষেপে পার্থক্য: কম্পাইলার বনাম ইন্টারপ্রেটার

| বৈশিষ্ট্য | কম্পাইলার | ইন্টারপ্রেটার |
|------------------|----------------------------------|---------------------------------|
| কাজের পদ্ধতি | পুরো প্রোগ্রাম একবারে অনুবাদ করে | লাইনে লাইনে অনুবাদ করে |
| ত্রুটি সনাক্তকরণ | একবারে ত্রুটি সনাক্ত হয় | তাৎক্ষণিকভাবে ত্রুটি সনাক্ত হয় |
| গতিসম্পন্নতা | দ্রুত কার্যক্ষম | তুলনামূলক ধীর |
| উদাহরণ | C, C++ | Python, JavaScript |

এই বৈশিষ্ট্যগুলোর ভিত্তিতে প্রোগ্রামিং ভাষার ধরণ অনুযায়ী অনুবাদক প্রোগ্রামের ব্যবহার নির্ধারণ করা হয়।

8. Python Shell চালু করুন:

আপনি এখন পাইথনের IDLE অথবা যে কোনো কোড এডিটর (যেমন Thonny, Visual Studio Code, PyCharm) ব্যবহার করে Python কোড লিখে এবং চালাতে পারবেন। Python Interpreter ইনস্টল করলে আপনি সহজেই Python প্রোগ্রামিং শুরু করতে পারবেন।

ধন্যবাদ আপনার আগ্রহের জন্য! এখন লিনাক্স এবং ম্যাক অপারেটিং সিস্টেমে Python Interpreter ইনস্টল করার জন্য অতিরিক্ত কিছু নির্দেশনা দেওয়া হলো:

লিনাক্সে Python ইন্টারপ্রেটার ইনস্টল করা:

লিনাক্সের বেশিরভাগ ডিস্ট্রিবিউশনে (যেমন Ubuntu, Fedora) Python পূর্বেই ইনস্টল থাকে। তবে, যদি এটি ইনস্টল না থাকে বা আপনি আপডেটেড ভার্সন চান, তাহলে নিচের পদক্ষেপগুলো অনুসরণ করতে পারেন:

1. ইনস্টলেশন চেক করা

টার্মিনালে নিচের কমান্ডটি লিখুন:

```
python3 --version
```

যদি Python ভার্সন প্রদর্শিত হয়, তাহলে Python ইতোমধ্যে ইনস্টল করা রয়েছে। অন্যথায়, ইনস্টল করার জন্য নিচের পদক্ষেপ অনুসরণ করুন।

2. Python ইনস্টল করা (Debian/Ubuntu এর জন্য):

```
sudo apt update
```

```
sudo apt install python3
```

3. Python ইনস্টল করা (Fedora এর জন্য):

```
sudo dnf install python3
```

4. প্যাকেজ ম্যানেজার pip ইনস্টল করা:

Python প্যাকেজ ম্যানেজার pip ইনস্টল করতে:

```
sudo apt install python3-pip
```

এখন Python এবং pip ইনস্টল হয়ে গেলে, আপনি Python প্রোগ্রামিং শুরু করতে পারবেন।

ম্যাকে Python ইন্টারপ্রেটার ইনস্টল করা:

ম্যাক অপারেটিং সিস্টেমে Python সাধারণত ডিফল্টভাবে ইনস্টল থাকে, তবে আপনি আপডেটেড সংস্করণও ইনস্টল করতে পারেন।

1. ইনস্টলেশন চেক করা

টার্মিনালে লিখুন:

```
python3 --version
```

যদি Python এর সংস্করণ দেখা যায়, তাহলে এটি ইনস্টল রয়েছে। না থাকলে, নিচের পদক্ষেপ অনুসরণ করুন।

2. Homebrew ব্যবহার করে Python ইনস্টল করা:

Homebrew একটি জনপ্রিয় প্যাকেজ ম্যানেজার, যার মাধ্যমে Python ইনস্টল করা খুবই সহজ।

প্রথমে Homebrew ইনস্টল করুন (যদি ইতিমধ্যে না থাকে):

```
/bin/bash -c "$(curl -fsSL
```

```
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

এখন Python ইনস্টল করুন:

```
brew install python
```

3. pip ইনস্টলেশন চেক করা:

Python এর সাথে pip সাধারণত স্বয়ংক্রিয়ভাবে আসে। টার্মিনালে লিখুন:

```
pip3 --version
```

এখন আপনার Python এবং pip প্রস্তুত, এবং আপনি যেকোনো টার্মিনালে Python কোড লিখে চালাতে পারবেন।

IDE (Integrated Development Environment) ইনস্টলেশন:

IDE একটি সফটওয়্যার অ্যাপ্লিকেশন যা প্রোগ্রামিং ও সফটওয়্যার ডেভেলপমেন্টের জন্য বিভিন্ন টুল একত্রিত করে। এটি প্রোগ্রামারদের কোড লেখা, সম্পাদনা, কম্পাইল এবং ডিবাগ করতে সহায়তা করে। IDE সাধারণত কোড লেখার জন্য এডিটর, কোডের ত্রুটি চিহ্নিত করার জন্য ডিবাগার এবং কোড কম্পাইল বা রান করার জন্য ইন্টারপ্রেটার নিয়ে তৈরি হয়।

IDE এর প্রধান বৈশিষ্ট্য:

1. কোড এডিটর: কোড লেখার জন্য একটি ব্যবহারযোগ্য ইন্টারফেস, যা সিনটাক্স হাইলাইটিং এবং অটো-কমপ্লিট সাপোর্ট করে।
2. ডিবাগিং টুলস: প্রোগ্রামে ত্রুটি চিহ্নিত করতে সহায়ক।
3. কম্পাইলার বা ইন্টারপ্রেটার: কোডকে মেশিন ভাষায় অনুবাদ করে প্রোগ্রামটি কার্যকরী করে।
4. ভার্সন কন্ট্রোল ইন্টিগ্রেশন: গিট বা অন্যান্য ভার্সন কন্ট্রোল সিস্টেমের সাথে ইন্টিগ্রেশন সাপোর্ট।
5. গ্রাফিক্যাল ইউজার ইন্টারফেস (GUI): ব্যবহারকারীদের জন্য সহজ এবং কার্যকরী ইউজার ইন্টারফেস।

জনপ্রিয় IDE গুলি:

1. Visual Studio Code (VS Code): শক্তিশালী এবং ওপেন সোর্স কোড এডিটর, যা Python সহ বিভিন্ন ভাষার জন্য এক্সটেনশন সাপোর্ট করে।
2. PyCharm: Python প্রোগ্রামিং ভাষার জন্য পূর্ণাঙ্গ IDE।
3. Eclipse: Java, C++, এবং অন্যান্য ভাষার জন্য ব্যবহৃত একটি ওপেন সোর্স IDE।
4. IntelliJ IDEA: Java এবং অন্যান্য ভাষার জন্য শক্তিশালী IDE।
5. Xcode: macOS এবং iOS অ্যাপ্লিকেশন ডেভেলপমেন্টের জন্য ব্যবহৃত IDE।

IDE এর তুলনা:

| IDE | বিশেষত্ব | ব্যবহারকারী |
|----------|---|----------------------|
| Thonny | সহজ, শিক্ষানবিশদের জন্য উপযুক্ত | নতুন প্রোগ্রামার |
| PyCharm | শক্তিশালী, পেশাদার Python IDE | পেশাদার ডেভেলপার |
| VSCodium | ওপেন সোর্স, একাধিক ভাষার জন্য এক্সটেনশন সাপোর্ট | সফটওয়্যার ডেভেলপার |
| ZED | হালকা, ক্লাউড সাপোর্ট, লাইভ কোডিং সেশন | ক্লাউড-বেজড ডেভেলপার |

IDE এর সুবিধা:

1. কোড লেখার গতি বৃদ্ধি: অটো-কমপ্লিট, সিনটাক্স হাইলাইটিং এবং অন্যান্য বৈশিষ্ট্য দ্বারা কোড দ্রুত এবং সঠিকভাবে লেখা যায়।
2. ত্রুটি শনাক্তকরণ: ডিবাগার এবং সেমান্টিক চেকিংয়ের মাধ্যমে কোডের ত্রুটি দ্রুত শনাক্ত করা যায়।
3. ডেভেলপমেন্ট পরিবেশ একত্রিত করা: কম্পাইলিং, ডিবাগিং এবং টেস্টিং এর মতো টুল একসাথে থাকা কারণে ডেভেলপমেন্ট প্রক্রিয়া সহজ এবং দ্রুত হয়।

IDE ইনস্টল করার জন্য জনপ্রিয় বিকল্পসমূহ:

Thonny: সহজ এবং শিক্ষানবিশদের জন্য উপযুক্ত।
PyCharm: শক্তিশালী এবং পেশাদার Python IDE।
VSCodium: ওপেন সোর্স, দ্রুত এবং একাধিক ভাষার জন্য এক্সটেনশন সাপোর্ট।
ZED: হালকা, ক্লাউড সাপোর্ট সহ লাইভ কোডিং সেশন প্রদান করে।

এই IDE গুলির মাধ্যমে আপনি Python প্রোগ্রামিংকে আরও উন্নত এবং কার্যকরীভাবে শিখতে পারবেন।

আপনি যদি Thonny, PyCharm, VSCodium, বা ZED IDE ইনস্টল করতে চান, তাহলে নিচে প্রতিটি IDE-এর ইনস্টলেশন পদ্ধতি বিস্তারিতভাবে ব্যাখ্যা করা হয়েছে। আপনি যে কোনটি পছন্দ করে ইনস্টল করতে পারেন।

1. Thonny IDE ইনস্টল করা

লিনাক্স/উবুন্টু:

1. টার্মিনাল খুলুন।
2. নিচের কমান্ডটি রান করুন:

```
sudo apt update  
sudo apt install thonny
```

ম্যাক:

1. Homebrew ব্যবহার করে ইনস্টল করতে:

```
brew install --cask thonny
```

Windows:

1. Thonny এর Windows সংস্করণ ডাউনলোড করতে [এখানে ক্লিক করুন](#) এবং ইনস্টল করুন।

2. PyCharm IDE ইনস্টল করা

লিনাক্স/উবুন্টু:

- টার্মিনাল খুলুন।
- নিচের কমান্ডটি রান করুন:

```
sudo snap install pycharm-community --classic
```

ম্যাক:

- Homebrew ব্যবহার করে ইনস্টল করতে:

```
brew install --cask pycharm-ce
```

Windows:

- PyCharm এর Community Edition বা Professional Edition ডাউনলোড করতে [এখানে ক্লিক করুন](#) এবং ইনস্টল করুন।

3. VSCodium ইনস্টল করা

লিনাক্স/উবুন্টু:

- টার্মিনাল খুলুন।
- নিচের কমান্ডটি রান করুন:

```
sudo apt update  
sudo apt install codium
```

ম্যাক:

- Homebrew ব্যবহার করে ইনস্টল করতে:

```
brew install --cask vscodium
```

Windows:

- VSCodium এর Windows সংস্করণ ডাউনলোড করতে [এখানে ক্লিক করুন](#) এবং ইনস্টল করুন।

4. ZED IDE ইনস্টল করা

লিনাক্স/উবুন্টু, ম্যাক, এবং Windows:

- ZED IDE ইনস্টল করতে অফিসিয়াল সাইট থেকে ডাউনলোড করুন:
[ZED Official Website](#)
- ডাউনলোড করার পর ইনস্টলেশন নির্দেশনা অনুসরণ করুন।

ইনস্টলেশন পরবর্তী পদক্ষেপ:

- ইনস্টলেশন শেষ হলে, IDE ওপেন করুন।
- আপনার পছন্দের প্রোগ্রামিং ভাষা (যেমন Python) নির্বাচন করুন।
- একটি নতুন প্রোজেক্ট শুরু করুন এবং কোড লিখতে শুরু করুন।
- প্রয়োজনে প্লাগইন বা এক্সটেনশন ইনস্টল করুন।

কোড উদাহরণ: Allahu Akbar শব্দটি output দেখানোর জন্য কোড লিখুন

```
print("Allahu Akbar")
```

ব্যাখ্যা:

print() ফাংশনটি ব্যবহার করা হয় কোনো আউটপুট প্রদর্শনের জন্য। এখানে "Allahu Akbar" একটি স্ট্রিং হিসেবে আউটপুট করা হচ্ছে। স্ট্রিং প্রকাশের জন্য সাধারণত ডাবল কোট (" ") বা সিঙ্গেল কোট (' ') ব্যবহার করা হয়।

বাগ (Bug):

যে কোনো ভুল বা ত্রুটি যেটি কোডের কাজকে ব্যাহত করে, তাকে বাগ বলা হয়।

ডিবাগ (Debug):

বাগ ঠিক করার প্রক্রিয়া। কোডে ভুল থাকলে, তা চিহ্নিত করা এবং সংশোধন করার কাজকে ডিবাগিং বলা হয়।

পাইথন ফাইল টাইপ (File Type):

পাইথন ফাইলের এক্সটেনশন হল .py। এর মাধ্যমে পাইথন কোড সংরক্ষণ করা হয়।

কমেন্ট লাইনের ভূমিকা এবং তার ব্যবহার

কমেন্ট লাইন হল কোডের সেই অংশ যা কম্পিউটার বা প্রোগ্রামিং ভাষার ইন্টারপ্রিটার/কম্পাইলার দ্বারা উপেক্ষা করা হয়, এবং এটি শুধুমাত্র ডেভেলপার বা প্রোগ্রামারদের জন্য ব্যবহার করা হয়। কমেন্ট লাইন মূলত কোডের ব্যাখ্যা, নির্দেশিকা, অথবা পরিকল্পনা দিতে ব্যবহৃত হয়।

কেন কমেন্ট লাইন ব্যবহার করা হয়?

- কোড বুঝতে সহজ করা
 - জটিল কোডের অংশ ব্যাখ্যা করার জন্য কমেন্ট ব্যবহার করা হয়, যাতে কোডের কাজ এবং উদ্দেশ্য সহজেই বোঝা যায়।
- সহযোগিতা সহজ করা
 - যখন একাধিক ডেভেলপার একটি প্রোজেক্টে কাজ করেন, তখন কমেন্ট লাইন অন্যদের জন্য কোড বুঝতে সহজ করে তোলে। এটি দলের সদস্যদের মধ্যে যোগাযোগকে উন্নত করে।
- ডিবাগিং সহায়ক
 - কোডের নির্দিষ্ট অংশ সাময়িকভাবে অকার্যকর বা মন্তব্যের মধ্যে রাখা যায়, যা ডিবাগিং প্রক্রিয়া সহজ করে।
- পরিকল্পনা
 - প্রোজেক্টের জন্য ভবিষ্যতে কী কাজ করা হবে, তার পরিকল্পনা এবং নোট রাখতে কমেন্ট ব্যবহার করা হয়।

কমেন্ট লাইনের ধরন

১. Single-line Comment (এক লাইনের কমেন্ট)

- এক লাইনের মধ্যে লেখা হয় এবং কোডের চলমান অংশের পর স্পষ্ট ব্যাখ্যা দেওয়ার জন্য ব্যবহৃত হয়।
- Python-এ # দিয়ে শুরু হয়।

উদাহরণ:

```
# এটি একটি Single-line Comment
print("Hello, World!") # এটি Hello, World! প্রিন্ট করবে
```

২. Multi-line Comment (একাধিক লাইনের কমেন্ট)

- একাধিক লাইনে লেখা যায় এবং বৃহত্তর ব্যাখ্যা দেওয়ার জন্য ব্যবহার করা হয়।
- Python-এ ''' ''' বা """ """ এর মাধ্যমে এই কমেন্ট তৈরি করা হয়।

উদাহরণ:

```
'''
এটি একটি Multi-line Comment।
এখানে অনেক কিছু লেখা যেতে পারে।
যেমন, কোডের একটি দীর্ঘ ব্যাখ্যা।
'''

print("Hello, World!")
```

বিভিন্ন ভাষায় কমেন্ট করার নিয়ম

- Python:
 - Single-line: #
 - Multi-line: ''' ''' বা """ """
- C / C++ / Java:
 - Single-line: //
 - Multi-line: /* */
- HTML:
 - <!-- -->
- JavaScript:
 - Single-line: //
 - Multi-line: /* */

উপসংহার

কমেন্ট লাইন কোডের উন্নত ব্যাখ্যা এবং কার্যকরী সহায়তার জন্য অত্যন্ত গুরুত্বপূর্ণ। এটি কেবল কোডের ফাংশনালিটির ব্যাখ্যা করার জন্য নয়, বরং কোডের উন্নতির জন্য পরিকল্পনা এবং সহযোগিতার ক্ষেত্রেও সহায়ক।

ভেরিয়েবল:

ভেরিয়েবল হচ্ছে একটি নির্দিষ্ট নাম যা ডেটা সংরক্ষণের জন্য ব্যবহৃত হয়।

ভেরিয়েবল নামের নিয়ম:

1. ভেরিয়েবল নামের মধ্যে A-Z, a-z, 0-9, এবং underscore (_) ব্যবহার করা যাবে।

উদাহরণ: RollNo, Roll_No

2. ভেরিয়েবল নামের মধ্যে স্পেস থাকতে পারবে না। উদাহরণ: Roll No ভুল, Roll_No সঠিক।
3. ভেরিয়েবল নামের শুরুতে সংখ্যা রাখা যাবে না। উদাহরণ: 10Roll ভুল, Roll10 সঠিক।
4. একই নামের একাধিক ভেরিয়েবল ব্যবহার করা যাবে না।
5. পাইথনের কিওয়ার্ড (যেমন int, if, else) কে ভেরিয়েবল নাম হিসেবে ব্যবহার করা যাবে না।

ডেটা টাইপ:

ডেটা টাইপ নির্দেশ করে ডেটার ধরণ। পাইথনে বিভিন্ন ধরনের ডেটা টাইপ রয়েছে।

বেসিক ডেটা টাইপ:

1. int: পূর্ণসংখ্যা। উদাহরণ: 5, -3
2. float: ভগ্নাংশ সংখ্যা। উদাহরণ: 3.14, -0.5
3. str: স্ট্রিং বা অক্ষরের সমষ্টি। উদাহরণ: "Hello", "Python"
4. bool: বুলিয়ান মান (সত্য বা মিথ্যা)। উদাহরণ: True, False

অপারেটর (Arithmetic):

অ্যারিথমেটিক অপারেটরগুলি গণনা বা অঙ্কের কাজের জন্য ব্যবহৃত হয়।

1. যোগ (+): দুটি সংখ্যার যোগফল। উদাহরণ: $5 + 3 = 8$
2. বিয়োগ (-): একটি সংখ্যা থেকে অন্যটি বিয়োগ করা। উদাহরণ: $5 - 3 = 2$
3. গুণ (*): দুটি সংখ্যার গুণফল। উদাহরণ: $5 * 3 = 15$
4. ভাগ (/): একটি সংখ্যাকে অন্যটির দ্বারা ভাগ করা। উদাহরণ: $6 / 3 = 2.0$
5. ভাগশেষ (%): ভাগশেষ বের করা। উদাহরণ: $5 \% 3 = 2$

ইনপুট: input()

input() ফাংশন ব্যবহার করে ইউজার থেকে স্ট্রিং হিসেবে ডেটা ইনপুট নেওয়া যায়।

Syntax:

```
variable = input() # যা string হিসেবে data input করাব
```

উদাহরণ:

```
Roll = input() # ইউজার থেকে একটি স্ট্রিং ইনপুট নিন
```

ডেটা টাইপ রূপান্তর:

পাইথনে ডেটার ধরণ (data type) রূপান্তর করতে type casting ব্যবহার করা হয়।

স্ট্রিং থেকে ইনপুট রূপান্তর:

int() - পূর্ণসংখ্যা (integer) তৈরি করতে।

float() - দশমিক সংখ্যার জন্য।

str() - স্ট্রিং (string) রূপান্তর করতে।

bool() - বুলিয়ান মান (True/False) রূপান্তর করতে।

আউটপুট: print()

print() ফাংশন দিয়ে কনসোলে আউটপুট প্রদর্শন করা হয়।

Syntax:

```
print(variable) # যেকোনো ভেরিয়েবলের মান আউটপুট করতে
```

প্রোগ্রাম উদাহরণ:

1. দুটি পূর্ণসংখ্যা ইনপুট নিয়ে যোগফল প্রদর্শন:

```
a = int(input()) # প্রথম পূর্ণসংখ্যা ইনপুট নিন  
b = int(input()) # দ্বিতীয় পূর্ণসংখ্যা ইনপুট নিন  
s = a + b # যোগফল হিসাব করুন  
print(s) # যোগফল প্রদর্শন করুন
```

2. দুটি পূর্ণসংখ্যা ইনপুট নিয়ে বিয়োগফল প্রদর্শন:

```
a = int(input()) # প্রথম পূর্ণসংখ্যা ইনপুট নিন  
b = int(input()) # দ্বিতীয় পূর্ণসংখ্যা ইনপুট নিন  
d = a - b # বিয়োগফল হিসাব করুন  
print(d) # বিয়োগফল প্রদর্শন করুন
```

3. দুটি পূর্ণসংখ্যা ইনপুট নিয়ে গুণফল প্রদর্শন:

```
a = int(input()) # প্রথম পূর্ণসংখ্যা ইনপুট নিন  
b = int(input()) # দ্বিতীয় পূর্ণসংখ্যা ইনপুট নিন  
m = a * b # গুণফল হিসাব করুন  
print(m) # গুণফল প্রদর্শন করুন
```

4. দুটি পূর্ণসংখ্যা ইনপুট নিয়ে ভাগফল প্রদর্শন:

```
a = int(input()) # প্রথম পূর্ণসংখ্যা ইনপুট নিন  
b = int(input()) # দ্বিতীয় পূর্ণসংখ্যা ইনপুট নিন  
d = a / b # ভাগফল হিসাব করুন  
print(d) # ভাগফল প্রদর্শন করুন
```


5. দুটি পূর্ণসংখ্যা ইনপুট নিয়ে ভাগশেষ প্রদর্শন:

```
a = int(input()) # প্রথম পূর্ণসংখ্যা ইনপুট নিন
b = int(input()) # দ্বিতীয় পূর্ণসংখ্যা ইনপুট নিন
r = a % b # ভাগশেষ হিসাব করুন
print(r) # ভাগশেষ প্রদর্শন করুন
```

প্রোগ্রামের অংশ:

1. Input: ব্যবহারকারীর কাছ থেকে ডেটা নেওয়া
2. Process: ইনপুট ডেটা প্রক্রিয়া করা
3. Output: ফলাফল প্রদর্শন করা

সূত্র সংশ্লিষ্ট প্রোগ্রাম

১. দুটি সংখ্যার যোগফল প্রদর্শন:

```
a = int(input()) # প্রথম সংখ্যা ইনপুট
b = int(input()) # দ্বিতীয় সংখ্যা ইনপুট
s = a + b # যোগফল হিসাব
print(s) # যোগফল আউটপুট
```

উদাহরণ প্রোগ্রাম:

২. দুটি সংখ্যার গড় নির্ণয়ের প্রোগ্রাম:

```
a = float(input()) # প্রথম সংখ্যা ইনপুট (ভগ্নাংশ)
b = float(input()) # দ্বিতীয় সংখ্যা ইনপুট (ভগ্নাংশ)
Avg = (a + b) / 2 # গড় হিসাব
print(Avg) # গড় প্রদর্শন
```

৩. আয়তাক্ষত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম:

```
l = float(input("Enter Length: ")) # দৈর্ঘ্য ইনপুট
w = float(input("Enter Width: ")) # প্রস্থ ইনপুট
A = l * w # ক্ষেত্রফল হিসাব
print("Area of Rectangle is ", A) # আয়তাক্ষত্রের ক্ষেত্রফল আউটপুট
```

৪. ত্রিভুজের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম:

```
b = float(input("Enter Base: ")) # ভিত্তি ইনপুট
h = float(input("Enter Height: ")) # উচ্চতা ইনপুট
A = (b * h) / 2 # ক্ষেত্রফল হিসাব
print("Area of Triangle is ", A) # ত্রিভুজের ক্ষেত্রফল আউটপুট
```

৫. সেলসিয়াস থেকে ফারেনহাইট নির্ণয়ের প্রোগ্রাম:

```
C = float(input("Enter The Temp Of Degree Celsius: ")) # সেলসিয়াস ইনপুট
F = 9 * C / 5 + 32 # ফারেনহাইটে রূপান্তর
print("The Temp in Degree Fahrenheit is: ", F) # ফারেনহাইট আউটপুট
```

৬. ফারেনহাইট থেকে সেলসিয়াস নির্ণয়ের প্রোগ্রাম:

```
F = float(input("Enter The Temp Of Degree Fahrenheit: ")) # ফারেনহাইট ইনপুট
C = (5 / 9) * (F - 32) # সেলসিয়াসে রূপান্তর
print("The Temp in Degree Celsius is: ", C) # সেলসিয়াস আউটপুট
```

৭. নিষ্ক্ষেপিত বস্তুর বিচরণকাল নির্ণয়ের প্রোগ্রাম:

```
u = float(input("Enter Initial Velocity: ")) # প্রাথমিক গতি ইনপুট
T = 2 * u / 9.8 # বিচরণকাল হিসাব
print("Your answer is: ", T) # বিচরণকাল আউটপুট
```

৮. মোলসংখ্যা নির্ণয়ের প্রোগ্রাম:

```
w = float(input("Enter The Value Of W: ")) # ভরের ইনপুট
m = float(input("Enter The Value Of M: ")) # মোট ভরের ইনপুট
n = w / m # মোলসংখ্যা হিসাব
print("Your answer is: ", n) # মোলসংখ্যা আউটপুট
```

৯. বর্গমূল নির্ণয়ের প্রোগ্রাম:

```
n = int(input("Enter a Number: ")) # একটি সংখ্যা ইনপুট
print(n**0.5) # বর্গমূল প্রদর্শন
```

১০. একটি সংখ্যা জোড় নাকি বিজোড় নির্ণয়ের প্রোগ্রাম:

```
n = int(input("Enter a number: ")) # সংখ্যাটি ইনপুট করুন
if (n % 2 == 0): # যদি সংখ্যাটি ২ দিয়ে বিভাজ্য হয়
    print(n, "is an even number") # জোড় সংখ্যা
else:
    print(n, "is an odd number") # বিজোড় সংখ্যা
```

১১. লজিক্যাল অপারেটর (Logical Operators):

এবং: and
অথবা: or
না: not

১২. শর্তসাপেক্ষ অপারেটর (Conditional Operators):

< : ছোট
<= : ছোট বা সমান
> : বড়
>= : বড় বা সমান
== : সমান
!= : সমান নয়

8. elif এর Syntax:

```
if (condition):  
    # যদি শর্ত সত্য হয়, এই অংশটি কার্যকর হবে  
    statement_of_true  
elif (condition):  
    # যদি প্রথম শর্ত মিথ্যা হয় এবং দ্বিতীয় শর্ত সত্য হয়  
    statement_of_true  
else:  
    # যদি উপরের কোন শর্তই সত্য না হয়  
    statement_of_false
```

৫. একটি ছাত্রের মার্ক ইনপুট নিয়ে তার গ্রেড নির্ণয়ের প্রোগ্রাম:

```
mark = float(input("Enter Mark: ")) # মার্ক ইনপুট করুন  
  
if (mark >= 80 and mark <= 100):  
    print("A+")  
elif (mark >= 70 and mark < 80):  
    print("A")  
elif (mark >= 60 and mark < 70):  
    print("A-")  
elif (mark >= 50 and mark < 60):  
    print("B")  
elif (mark >= 40 and mark < 50):  
    print("C")  
elif (mark >= 33 and mark < 40):  
    print("D")  
elif (mark >= 0 and mark < 33):  
    print("F")  
else:  
    print("Syntax Error")
```

৬. ক্যালকুলেটর প্রোগ্রাম:

```
n1 = float(input("Enter 1st Number: ")) # প্রথম সংখ্যা ইনপুট  
n2 = float(input("Enter 2nd Number: ")) # দ্বিতীয় সংখ্যা ইনপুট  
op = input("Enter The Arithmetic Operator {+, -, *, /, %, **}: ") # অপারেটর ইনপুট  
  
result = None # ফলাফল শুরুতে কিছুই নেই  
if (op == "+"):  
    result = n1 + n2 # যোগফল  
elif (op == "-"):  
    result = n1 - n2 # বিয়োগফল  
elif (op == "*"):  
    result = n1 * n2 # গুণফল  
elif (op == "/"):  
    result = n1 / n2 # ভাগফল  
elif (op == "%"):  
    result = n1 % n2 # ভাগশেষ  
else:  
    print("Operation is incorrect") # ভুল অপারেটর  
print(n1, op, n2, "=", result) # আউটপুট প্রদর্শন
```

১. while লুপের মৌলিক গঠন:

লুপের মূল অংশ:

1. Initialization (শুরু)
2. Condition (শর্ত)
3. Body (অংশ)
4. Update (পরিবর্তন)

while লুপের Syntax:

```
# Initialization
while condition:
    # Body
    # Update
```

২. "Allahu Akbar" তিনবার আউটপুট করার প্রোগ্রাম:

```
i = 1 # শুরু
while i <= 3: # শর্ত
    print(i, "Allahu Akbar") # আউটপুট
    i = i + 1 # পরিবর্তন
```

৩. ১ থেকে n পর্যন্ত সংখ্যা আউটপুট করার প্রোগ্রাম:

```
n = int(input("Enter last number: ")) # ইনপুট
i = 1 # শুরু
while i <= n: # শর্ত
    print(i) # আউটপুট
    i += 1 # পরিবর্তন
```

৪. ৩, ৬, ৯,...n পর্যন্ত সংখ্যা আউটপুট করার প্রোগ্রাম:

```
n = 10 # সর্বশেষ সংখ্যা
i = 1 # শুরু
while i <= n: # শর্ত
    print(i) # আউটপুট
    i += 3 # পরিবর্তন
```

৫. ২, ৪, ৬, ৮, ১০... পর্যন্ত সংখ্যা আউটপুট করার প্রোগ্রাম:

```
n = 10 # সর্বশেষ সংখ্যা
i = 2 # শুরু
while i <= n: # শর্ত
    print(i) # আউটপুট
    i += 2 # পরিবর্তন
```

৬. ২, ৪, ৮,... ২০ পর্যন্ত সংখ্যা আউটপুট করার প্রোগ্রাম:

```
n = 20 # সর্বশেষ সংখ্যা
i = 2 # শুরু
while i <= n: # শর্ত
    print(i) # আউটপুট
    i *= 2 # পরিবর্তন
```

৭. প্রথম সংখ্যা থেকে শেষ সংখ্যা পর্যন্ত নির্দিষ্ট ধাপে সংখ্যা আউটপুট করার প্রোগ্রাম (এবং বৃদ্ধি):

```
m = int(input("Enter 1st Number: ")) # প্রথম সংখ্যা ইনপুট
n = int(input("Enter last Number: ")) # শেষ সংখ্যা ইনপুট
d = int(input("Enter difference: ")) # পার্থক্য ইনপুট

while m <= n: # শর্ত
    print(m) # আউটপুট
    m += d # পরিবর্তন
```

৮. শেষ সংখ্যা থেকে প্রথম সংখ্যা পর্যন্ত নির্দিষ্ট ধাপে সংখ্যা আউটপুট করার প্রোগ্রাম (এবং হ্রাস):

```
m = int(input("Enter 1st Number: ")) # প্রথম সংখ্যা ইনপুট
n = int(input("Enter last Number: ")) # শেষ সংখ্যা ইনপুট
d = int(input("Enter difference: ")) # পার্থক্য ইনপুট

while m >= n: # শর্ত
    print(m) # আউটপুট
    m -= d # পরিবর্তন
```

৯. ১ থেকে ১০০ পর্যন্ত সংখ্যা যোগফল বের করার প্রোগ্রাম:

```
s = 0 # যোগফল শুরু
i = 1 # শুরু
while i <= 100: # শর্ত
    s = s + i # যোগফল
    i += 1 # পরিবর্তন
print(s) # আউটপুট
```

১০. ১ থেকে n পর্যন্ত সংখ্যার বর্গফল যোগফল বের করার প্রোগ্রাম:

```
n = int(input("Enter last Number: ")) # শেষ সংখ্যা ইনপুট
s = 0 # যোগফল শুরু
i = 1 # শুরু
while i <= n: # শর্ত
    s += i * i # বর্গফল যোগ
    i += 1 # পরিবর্তন
print(s) # আউটপুট
```