শিখন অভিজ্ঞতা-৪

সমস্যার সমাধান চাই প্রোগ্রামিংয়ের জুড়ি নাই

আমরা সপ্তম শ্রেণিতে একটি বাস্তব সমস্যাকে সমাধানের জন্য সুডো কোড তৈরি করা শিখেছিলাম। সুডো কোড দেখে আমরা ধারণা পাব সমাধানের উপায়টি কেমন হবে। তবে কম্পিউটারকে সেই নির্দেশ বুঝিয়ে দিয়ে কম্পিউটারকে দিয়ে সমাধানটি করতে চাইলে আমাদের প্রয়োজন হবে একটি প্রোগ্রাম ডিজাইন করা। তাই এই শিখন অভিজ্ঞতায় আমরা দেখব যন্ত্রের জন্য নির্দিষ্ট যুক্তি সাজিয়ে কীভাবে একটি প্রোগ্রামিং ভাষা ব্যবহার করে প্রোগ্রাম ডিজাইন করতে পারি। তারপরে একটি সমস্যাকে নির্বাচন করে আমরা সেটি সমাধান করব প্রোগ্রাম ডিজাইন করে। এই প্রোগ্রাম ডিজাইনের সময়ে কিছু ত্রুটি এবং ঝুঁকিও তৈরি হতে পারে। সেগুলোকে কীভাবে সমাধান করা যায় সেটাও খুঁজে বের করার চেষ্টা করব আমরা এই অভিজ্ঞতায়।

সেশন-১ ও ২: যন্ত্রের মধ্যে বিভিন্ন ইনপুটে লজিকের সমন্বয়

সপ্তম শ্রেণিতে আমরা জেনেছিলাম কম্পিউটার বাইনারি সংখ্যা অর্থাৎ ০ আর ১ ছাড়া আর কোনো ডিজিট বা অংঙ্ক বুঝতে পারে না। আচ্ছা আমরা যখন সংখ্যা গণনা করি, প্রাথমিকভাবে মোট কয়টি ডিজিট গুণতে পারি বল তো? আমরা কিন্তু ০ থেকে ৯ পর্যন্ত মোট ১০টি ডিজিট গুণতে পারি। বাকিসব সংখ্যাকে ০ থেকে ৯ এর মাধ্যমেই প্রকাশ করা হয়। এই ১০ টি ডিজিট দিয়ে সব সংখ্যা আমরা কেন প্রকাশ করি? কারণ মানুষ যখন গণনা করা শুরু করেছিল, হাতের ১০ আঙ্গুলের কারণে যেকোনো কিছু ১০ সংখ্যা বিশিষ্ট তথা ডেসিম্যাল (Decimal) পদ্ধতি দিয়ে গণনা শুরু করেছিল।

তার মানে আমাদের দুই হাতে ১০টি আঙুলের পরিবর্তে যদি আরও কম বেশি আঙুল থাকত, তাহলে আমাদের ডিজিট সংখ্যাও কিন্তু পাল্টে যেত। এবারে একটু কম্পিউটারের কথা ভাবা যাক। কম্পিউটার তো ০ আর ১ অর্থাৎ দুই সংখ্যা বিশিষ্ট বাইনারি (Binary) সংখ্যা পদ্ধতি ব্যবহার করে। এর কারণ কি তাহলে?

কারণ কম্পিউটার বা যেকোনো ইলেক্ট্রনিক ডিভাইসে চিন্তা করা হয় ডিভাইসটি আমি চালু করতে পারি অথবা বন্ধ করতে পারি। অর্থাৎ আমার কাছে অপশন দুটি।

- ডিভাইস বন্ধ = ০
- ডিভাইস চালু = ১

এই কারণে ০ আর ১ অর্থাৎ দুই সংখ্যা বিশিষ্ট পদ্ধতি বা বাইনারি পদ্ধতি চালু হয়।

কম্পিউটারে যতরকম তথ্য জমা রাখা হয়, সবকিছু বাইনারি সংখ্যা হিসেবে জমা থাকে। আবার কম্পিউটার যতরকম হিসাবনিকাশ করে অর্থাৎ যোগ, বিয়োগ, গুণ ভাগ ইত্যাদি সবই বাইনারি সংখ্যাতেই করে থাকে।

এখন কম্পিউটার তো যেকোনো সংখ্যাকে ০ আর ১ দিয়ে উপস্থাপন করতে চাইবে। কিন্তু আমরা তো যেকোনো সংখ্যাকে ০ থেকে ৯ দিয়ে প্রকাশ করতে অভ্যস্ত। তাই হঠাৎ করে চোখের সামনে ০১১১০০০১১০ এমন একটা সংখ্যা দেখলে আমরা তো বুঝতে পারি না এর অর্থ কি। তবে খুব সহজেই চাইলে ডেসিম্যাল থেকে বাইনারি এবং বাইনারি থেকে ডেসিম্যাল রূপান্তর করা সম্ভব। আমরা ৮ম শ্রেণির গণিত বইয়ে বাইনারি সংখ্যা ও এর বিভিন্ন গাণিতিক অপারেশন সম্পর্কে আরও বিস্তারিত জানতে পারব।

আমরা এর আগে সপ্তম শ্রেণিতে প্রবাহচিত্র সম্পর্কে জানার সময় ইনপুট ও আউটপুট সম্পর্কে জেনেছিলাম। সপ্তম শ্রেণিতে আমরা রোবট দিয়ে আগুন নেভানোর উদাহরণ দেখেছিলাম। আমাদের রোবট প্রথমে ক্যামেরা দিয়ে আগুন শনাক্ত করেছিল। এটি ছিল ইনপুট। এরপর রোবটটি পানি ঢেলেছিল, এই পানি ঢালা ছিল আউটপুট।

অর্থাৎ একটি ইনপুটের জন্য আমরা একটি আউটপুট পেয়েছিলাম। এটাকে আমরা লিখতে পারি,

আগুন শনাক্ত ⟶ পানি ঢালা

তার মানে, আগুন শনাক্ত হবার উপর পানি ঢালা নির্ভরশীল। যদি আগুন শনাক্ত হয়, তাহলে রোবট পানি ঢালবে। যদি আগুন শনাক্ত না হয়, রোবট কিন্তু পানি ঢালবে না।

আবার একটি নেতিবাচক বা বিপরীত ঘটনাকেও আমরা প্রকাশ করতে পারি লজিকাল নট (Logical NOT) দিয়ে।

¬ চিহ্ন দিয়ে লজিকাল নট বুঝানো হয়।

তাহলে আমরা রোবটটির জন্য এটাও লিখতে পারি,

আগুন শনাক্ত ⟶ ¬পানি ঢালা

অর্থাৎ আগুন শনাক্ত না হলে, রোবট পানি ঢালবে না।

আবার আমরা চাইলে চলকের মাধ্যমেও এটি উপস্থাপন করতে পারি।

আমরা যদি লেখি,

ক = আগুন শনাক্ত

খ = পানি ঢালা

তাহলে লিখতে পারি,

ক ⟶ খ

¬ ক ⟶ ¬ খ

যখন একটি ইনপুট বা আউটপুট ঘটে, সেটিকে আমরা ১ দিয়ে প্রকাশ করতে পারি।

আবার যখন ইনপুট বা আউটপুট না ঘটে, সেটিকে ০ দিয়ে প্রকাশ করা যায়।

এভাবে ইনপুট ও আউটপুটের বিভিন্ন পরিণতিকে ১ ও ০ এর মাধ্যমে একটি টেবিলে প্রকাশ করা যায়। এই টেবিল সবসময় আমাদের সত্য জানিয়ে দেয়, তাই একে ট্রুথ টেবিল (Truth Table) বলা হয়।

একটি ইনপুট গ (একটি সুইচ) এবং আউটপুট ঘ (একটি বাতি) এরজন্য চিন্তা করি –

গ ⟶ ¬ ঘ

¬ গ ⟶ ঘ

অর্থাৎ, ইনপুট এবং আউটপুট বিপরীত আচরণ করছে। সুইচ চালু করলে বাতি নিভে যাবে, সুইচ বন্ধ করলে বাতি জ্বলবে। এমন একটি ঘটনার জন্য ট্রুথ টেবিল নিচে পূরণ কর-

ইনপুট গ	আউটপুট ঘ
১	০
০	১

এটি হলো লজিকাল নটের জন্য সাধারণ একটি ট্রুথ টেবিল।

এবারে আমরা যদি রোবট দিয়ে আগুন শনাক্ত হলে পানি ঢালার উদাহরণের কথা আবার ভাবি।
যদি,
ক = আগুন শনাক্ত
খ = পানি ঢালা
তাহলে,
ক ⟶ খ
¬ ক ⟶ ¬ খ
এক্ষেত্রে ট্রুথ টেবিল কেমন হবে নিচের ছকে পূরণ করে ফেলি

ইনপুট ক	আউটপুট খ

কিন্তু এমন যদি হয় দুটি ইনপুট ক ও খ আছে, যাদের উপর আউটপুট গ নির্ভরশীল ?

ধরি,

ক = ১ম ইনপুট (প্রথম সুইচ)

খ = ২য় ইনপুট (দ্বিতীয় সুইচ)

গ = আউটপুট (একটি বাতি)

যদি আমরা চাই দুটি সুইচ একসাথে জ্বালালে তখনই কেবল বাতি জ্বলবে। অর্থাৎ, দুইটি ইনপুট সচল হলেই কেবল আউটপুট পাব, এমন ক্ষেত্রে ব্যবহার করা হয় লজিকাল এন্ড (Logical AND), যাকে ^ চিহ্ন দিয়ে

প্রকাশ করা হয়। এক্ষেত্রে নিচের মত ঘটনা ঘটবে -

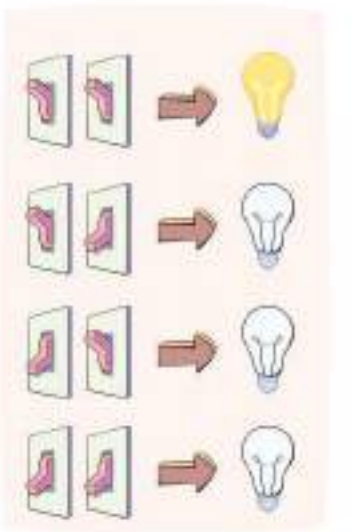
ক ^ খ ⟶ গ

ক ^ ¬ খ ⟶ ¬ গ

¬ ক ^ খ ⟶ ¬ গ

¬ ক ^ ¬ খ ⟶ ¬ গ

লক্ষ্য করেছো, একটি ইনপুট সচল না হলেই আউটপুট আর সচল হতে পারছে না।



একইভাবে এই ঘটনার জন্য ট্রুথ টেবিল নিচে পূরণ করে ফেল–

ইনপুট ক	ইনপুট খ	আউটপুট গ = ক ^ খ

আবার যদি আমরা চাই দুটি সুইচ থাকবে, কিন্তু অন্তত একটি সুইচ সচল হলেই বাতি জ্বলে উঠবে। অর্থাৎ, দুটি ইনপুটের যেকোনো একটি ইনপুট সচল হলেই আউটপুট পাব, এমন ক্ষেত্রে ব্যবহার করা হয় লজিকাল অর (Logical OR), যাকে | চিহ্ন দিয়ে প্রকাশ করা হয়। এক্ষেত্রে নিচের মত ঘটনা ঘটবে –

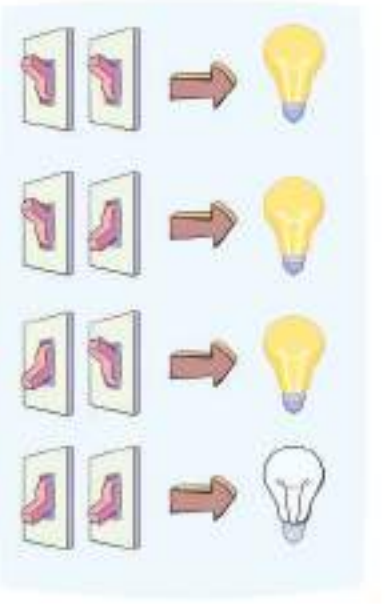
ক | খ ⟶ গ

ক | ¬ খ ⟶ গ

¬ ক | খ ⟶ গ

¬ ক | ¬ খ ⟶ ¬ গ

লক্ষ্য করেছো, শুধুমাত্র দুটি ইনপুটই অচল থাকলে তখন আউটপুট সচল হতে পারছে না। এছাড়া অন্য সবসময়ই আউটপুট সচল হচ্ছে।



একইভাবে এই ঘটনার জন্য ট্রুথ টেবিল

নিচে পূরণ করে ফেল–

ইনপুট ক	ইনপুট খ	আউটপুট গ = ক | খ

আমরা যখন ২টি ইনপুট নিয়ে কাজ করছি, তখন সবসময় মোট ৪টি সম্ভাব্য ঘটনা পাচ্ছি। কিন্তু কেন এমন হচ্ছে? আসলে আমাদের কাছে প্রতিটি ইনপুটের জন্য দুটিই সম্ভাব্য মান আছে। আমরা হয় ইনপুটের মান পাব ০ অথবা ১

কয়েকটি ইনপুটের জন্য মোট সম্ভাব্য ঘটনার সংখ্যা হয় = (মোট ইনপুট)^ মোট সম্ভাব্য মান

তাহলে আমাদের ২টি ইনপুটের জন্য মোট সম্ভাব্য ঘটনা = ২^২ = ৪টি

একইভাবে ইনপুট যদি আমরা ৩টি নিতাম, মোট সম্ভাব্য ঘটনা হতো = ২^৩ = ৮টি

সেশন-৩: প্রোগ্রাম ডিজাইনের সূচনা

কম্পিউটারকে যেকোনো নির্দেশ দিতে গেলে কম্পিউটার বুঝতে পারে এমন ভাষায় নির্দেশ লিখতে হয়। সপ্তম শ্রেণিতে সুডো কোড লেখা শিখেছিলাম আমরা। সুডো কোডের মাধ্যমে আমরা একটি নির্দেশমালা তৈরি করেছিলাম যা অনুসরণ করে একটি রোবট আগুন নিভাতে পারবে। কিন্তু সুডো কোড সরাসরি কম্পিউটার বা রোবটের কাছে দিলে সেটি আমাদের কম্পিউটার কিংবা রোবট বুঝতে পারবে না। আমরা তো আগেই জেনেছি কম্পিউটারসহ যেকোনো ইলেক্ট্রনিক ডিভাইস শুধুমাত্র ০ আর ১ কে বুঝতে পারে। এদিকে শুধু ০ আর ১ দিয়ে নিজেদের নির্দেশগুলো লিখে ফেলাও আমাদের জন্য কঠিন। তাহলে কম্পিউটারের সাথে কীভাবে আমরা যোগাযোগ করব? এমন কিছু ভাষা আছে, যেখানে ওই ভাষার রীতিনীতি অনুসরণ করে নির্দেশ লিখলে কম্পিউটার সেই ভাষাকে সহজেই বাইনারিতে অথবা হেক্সাডেসিম্যালে রূপান্তর করে নিয়ে নির্দেশগুলো বুঝতে পারে। এই ভাষাগুলোকে বলা হয় প্রোগ্রামিং ভাষা। মানুষ নিজেদের মধ্যে কথা বলার জন্য বাংলা,ইংরেজি, ফ্রেঞ্চ, ল্যাটিন, স্প্যানিশ ইত্যাদি কতরকমের ভাষা ব্যবহার করে! ঠিক তেমনই অনেক রকম প্রোগ্রামিং ভাষা আছে। যেমন - সি, সি++, পাইথন, জাভা ইত্যাদি।

আমরা এরকম যেকোনো একটি প্রোগ্রামিং ভাষা শিখলে সেই ভাষার মাধ্যমে কম্পিউটারকে প্রয়োজনমত বিভিন্ন নির্দেশ দিতে পারব। আমাদের কম্পিউটারের একটি সফটওয়্যার অ্যাপ্লিকেশনে প্রথমে আমরা নির্দিষ্ট

কোনো প্রোগ্রামিং ভাষায় নির্দেশগুলো লিখে ফেলব। কম্পিউটারে এমন একটি রূপান্তর ব্যবস্থা থাকে যা সেই প্রোগ্রামিং ভাষার নির্দেশগুলোকে মেশিন কোডে রূপান্তর করে।

মেশিন কোড কি? মূলত ০ আর ১ এর সমন্বয়ে তৈরি বাইনারি কোডকেই মেশিন কোড বলা হয়, যা আমাদের কম্পিউটার বুঝতে পারে। এই রূপান্তরের ফলে আমাদের নির্দেশগুলো কম্পিউটার বুঝতে পারবে এবং সেই নির্দেশ অনুসরণ করে একটি কাজ সম্পন্ন করতে পারবে।

কম্পিউটারে থাকা প্রোগ্রামিং ভাষার রূপান্তর ব্যবস্থা আবার দুই রকমের হতে পারে-

ক) কিছু রূপান্তর ব্যবস্থায় আমরা যতগুলো নির্দেশ দিব, যদি নির্দেশগুলো নির্ভুল হয় তাহলে সবগুলো নির্দেশ একসাথে মেশিন কোডে রূপান্তর হবে। এই রূপান্তর ব্যবস্থাকে বলা হয় কম্পাইল (Compile) করা। আর যে সফটওয়্যার রূপান্তর করল, সেই রুপান্তরকারী হচ্ছে একটি কম্পাইলার (Compiler)। তবে কম্পাইলার যদি পুরো নির্দেশের কোথাও ভুল পায়, তাহলে রূপান্তর করতে পারে না। সবগুলো নির্দেশ নির্ভুল দিলে তখনই রুপান্তরের কাজটি করতে পারে।

খ) কিছু রূপান্তর ব্যবস্থায় আমরা যত নির্দেশই দেই না কেন, সব একসাথে রূপান্তর হবে না। একটি একটি করে নির্দেশ ধারাবাহিকভাবে রূপান্তর হতে থাকবে। এই রূপান্তর ব্যবস্থাকে বলা হয় ইন্টারপ্রেট (Interprete) করা। আর যে রূপান্তর কাজটি করছে তাকে বলা হয় ইন্টারপ্রেটার (Interpreter)। ইন্টারপ্রেটার একটি একটি করে নির্দেশ রূপান্তর করতে থাকবে। কোনো নির্দেশে ভুল পেলে সেই নির্দেশে আসার পর থেমে যাবে।

এবারে নিচের ছকে সঠিক অপশনে টিক চিহ্ন দেই–

প্রোগ্রামিং ভাষার বিবরণ	রূপান্তর ব্যবস্থা	
	কম্পাইলার	ইন্টারপ্রেটার
সি প্রোগ্রামিং ভাষায় আমরা যতগুলো নির্দেশ দিব সব একসাথে রূপান্তরিত হবে।		
পাইথন প্রোগ্রামিং ভাষায় আমরা যতগুলো নির্দেশ দিব একটি একটি করে রূপান্তরিত হবে।		

কিন্তু এই যে অজস্র প্রোগ্রামিং ভাষা আছে, এরমধ্যে কোনটি আমরা শিখব? যেকোনো একটি প্রোগ্রামিং ভাষা প্রথমে শিখলেই হলো। কারণ সব প্রোগ্রামিং ভাষার মূল গঠন একইরকমের, শুধু ভাষাগুলোতে বিভিন্ন নির্দেশ লেখার নিয়ম একটু ভিন্ন থাকে। যেমন সি প্রোগ্রামিং ভাষায় প্রতিটি নির্দেশ (স্টেটমেন্ট) শেষ হবার পর একটি সেমিকোলন চিহ্ন দিতে হয়, কিন্তু পাইথনে এই কাজ করতে হয় না। এরকম কিছু পার্থক্য থাকলেও চিন্তার কিছু নেই। তুমি একটি প্রোগ্রামিং ভাষা শিখে নিলে এরপর অন্য প্রোগ্রামিং ভাষাগুলো শেখা খুব সহজ হয়ে যাবে তোমার জন্য। আমরা এই বইয়ে পাইথন নিয়ে কাজ শুরু করব। তুমি চাইলে পাইথন শেখার পর খুব সহজে অন্য প্রোগ্রামিং ভাষাও শিখে নিতে পারবে।

পাইথনের যাত্রা শুরু:

সহজে শেখার জন্য পাইথন বেশ মজার একটি প্রোগ্রামিং ভাষা।

পাইথন ভাষায় নির্দেশ লেখার জন্য সবার আগে আমাদের কিছু কাজ করতে হবে-

১। সবার আগে আমাদের পাইথন অ্যাপ্লিকেশন ডাউনলোড করে কম্পিউটারে ইন্সটল করতে হবে।

এই লিংকে চলে যাই - https://www.python.org/downloads/

এরপর সেখান থেকে সবচেয়ে সাম্প্রতিক ভার্সন নামিয়ে নেই।

২। অ্যাপ্লিকেশন নামানো হয়ে গেলে এটি ইন্সটল করে ফেলি। ইন্সটলের সময় নিচের ছবির মতো একটি উইন্ডো আমরা দেখতে পাব -

আমরা ইন্সটল উইন্ডোর নিচে থাকা অপশনগুলো ক্লিক করে টিক চিহ্ন দিয়ে দিব। তারপর Install Now অপশনে ক্লিক করব। এসময় ইন্সটল হবার অনুমতি চাইলে সেটাও অনুমতি দিয়ে দিব।

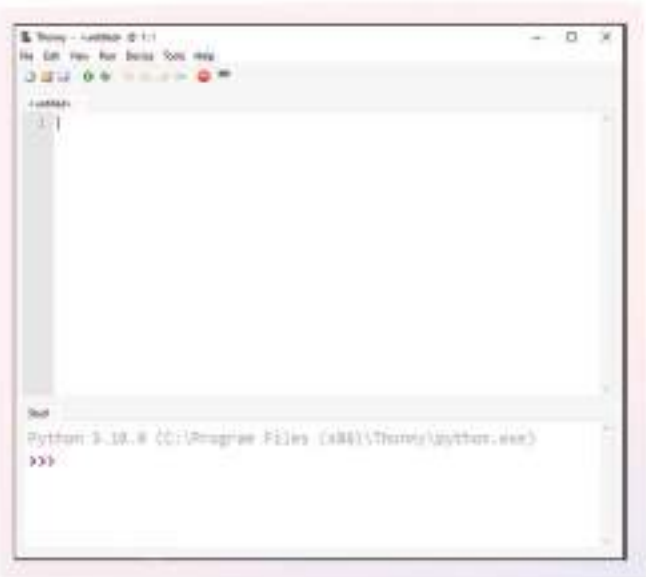
৩। এরপর আমাদের মেসেজ দেখাবে যে আমাদের সেটআপ সফল হয়েছে।

৪। পাইথন আমাদের কম্পিউটারে যুক্ত হলো। কিন্তু আমাদের আরেকটি সফটওয়্যার এপ্লিকেশন লাগবে যেখানে আমরা আমাদের নির্দেশ লিখে কম্পিউটারকে বুঝিয়ে দিব। সেজন্য এই লিংকে যাই - https://thonny.org/

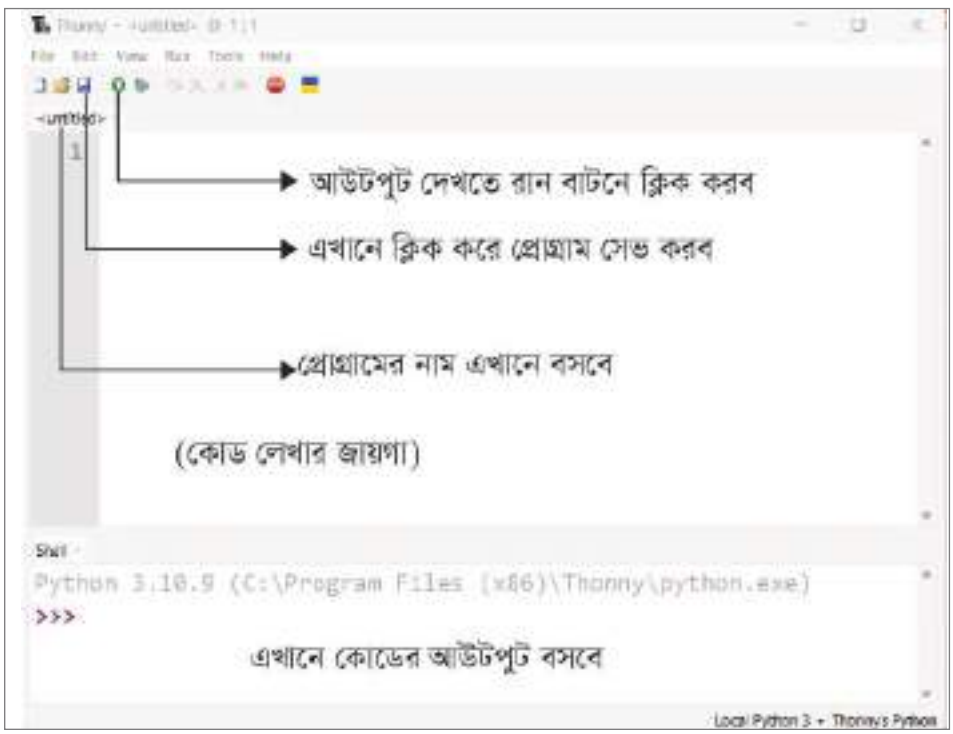
এই লিংক থেকে thonny সফটওয়ারটি নামিয়ে নেই ও ইন্সটল করে ফেলি।

৫। thonny সফটওয়্যার এরপর চালু করি। নিচের মতো উইন্ডো দেখতে পাব–

Let's go! বাটনে ক্লিক করলে নিচের মতো উইন্ডো আসবে–



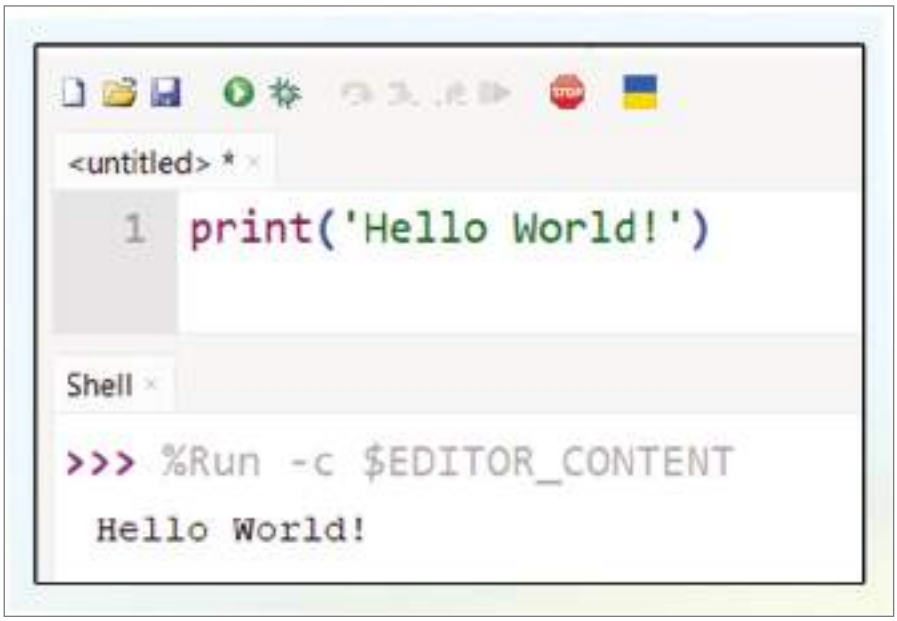
৬। এই উইন্ডোতে থাকা গুরুত্বপূর্ণ কয়েকটি অংশ বুঝে নেই–



৭। এবারে একটা কাজ করি। আমরা একটা প্রোগ্রাম লিখি, যার কাজ হবে আউটপুট হিসেবে Hello World! প্রিন্ট করা। আউটপুট হিসেবে কোনো কিছু প্রিন্ট করতে হলে print () ব্যবহার করতে হয়। আমরা যেই টেক্সট প্রিন্ট করতে চাই, সেটা print () এর ভিতরে Single Quotation (' ') দিয়ে তারমধ্যে লিখব। তাহলে Hello World! প্রিন্ট করতে লেখি-

print('Hello World!')

এরপর রান বাটনে ক্লিক করলে নিচে আউটপুট হিসেবে Hello World! লেখা উঠবে।



৮। এবারে সেভ বাটনে ক্লিক করে প্রোগ্রামের একটি নাম দিয়ে ফাইলটি সেভ করি। তখন আমাদের ফাইলের নামও প্রদর্শন করবে প্রোগ্রামের উপরে।

কোনো টেক্সট প্রদর্শন করা কত সহজ দেখেছ? আমরা যেমন ইংরেজি টেক্সট প্রিন্ট করলাম, একইভাবে বাংলা লিখেও তুমি সেটা প্রিন্ট করতে পারবে।

যেমন, নিচের লাইন লিখে রান করে দেখ তো কি দেখা যায়-

print ('আমি এখন পাইথন শিখছি')

তুমিও কি এমন বিভিন্ন টেক্সট প্রদর্শন করতে পারবে?

নিচের ছকে কোনো টেক্সট প্রদর্শন করতে কি প্রোগ্রাম লিখতে হবে তা পূরণ কর–

যা টেক্সট প্রদর্শন করব	প্রোগ্রাম যা লিখতে হবে
তোমার নিজের নাম ইংরেজিতে ও বাংলায় লিখে প্রিন্ট কর	
I love Bangladesh	
আমি ৮ম শ্রেণিতে পড়ি	
প্রোগ্রামিং শিখতে ভারি মজা	

সেশন-৪ ও ৫: ভ্যারিয়েবল ভারি মজার

আগের সেশনে আমরা দেখলাম যেকোনো টেক্সট কত সহজে প্রিন্ট করা যায়। আমাদের প্রোগ্রামে যেকোনো সময় print () ফাংশন ব্যবহার করে এই কাজটি করা সম্ভব।

আবার কোনো তথ্য যদি প্রোগ্রামের ভিতর সঞ্চয় করতে হয় তাহলে আমরা ব্যবহার করতে পারব একটি ভ্যারিয়েবল (Variable)। ভ্যারিয়েবল হলো একটি বাক্সের মত, যার ভিতরে নির্দিষ্ট একটি তথ্য জমা রাখা যায়। আবার ভ্যারিয়েবল শব্দটির অর্থ পরিবর্তনশীল। মানে আমরা চাইলে প্রোগ্রামে একটি লাইনে ভ্যারিয়েবলের মধ্যে একটি তথ্য জমা রাখার পর অন্য লাইনে সেই তথ্য পরিবর্তন করে ভিন্ন আরেকটি তথ্য জমা রাখতে পারি।

আমাদের সবার তো একটা নির্দিষ্ট নাম আছে তাই না? এই নাম দিয়ে সবাই আমাদের চিনতে পারে। একইভাবে প্রতিটি ভ্যারিয়েবলের একটি নির্দিষ্ট নাম দিতে হয়, যেই নাম দিয়ে পুরো প্রোগ্রামে আমরা ভ্যারিয়েবলটি চিনতে পারব ও ব্যবহার করতে হবে।

যেমন, আমরা যদি চাই number নামে একটি ভ্যারিয়েবল তৈরি করব, যেখানে ভ্যারিয়েবলের মান হিসেবে 7 জমা রাখতে চাই। তাহলে আমরা লিখব –

number = 7

তবে ভ্যারিয়েবলের নাম দেওয়ার সময় আমাদের কিছু দিক খেয়াল রাখতে হবে–

১। প্রথমত ভ্যারিয়েবলের নাম সবসময় একটি শব্দ হবে। অর্থাৎ একাধিক শব্দ দিয়ে আমরা ভ্যারিয়েবলের নাম লিখতে পারব না। তবে চাইলে দুটি শব্দের মধ্যে থাকা স্পেসগুলো বাদ দিয়ে তাদের একটি শব্দ হিসেবে ভ্যারিয়েবলের নাম দেওয়া যাবে। আবার চাইলে দুটি শব্দের মধ্যে থাকা স্পেস বাদ দিয়ে তাদের মধ্যে একটি

আন্ডারস্কোর (_) চিহ্ন দিয়েও ভ্যারিয়েবলটির নামকরণ করা যাবে।

ভ্যারিয়েবলের সঠিক নামকরণ	ভ্যারিয়েবলের ভুল নামকরণ
MyVariable	My Variable
this_variable_is_cool	this variable is cool

২। একটি ভ্যারিয়েবলের নামের প্রথম অক্ষর অবশ্যই a-z অথবা A-Z অথবা আন্ডারস্কোর (_) হতে হবে। প্রথম অক্ষর কোন সংখ্যা (0-9) বা অন্য কোনো প্রতীক চিহ্ন (যেমন * বা - ইত্যাদি) হতে পারবে না। প্রথম অক্ষরের পর বাকি অক্ষরগুলোতে যেকোনো সংখ্যা (0-9) বা a-z বা A-Z বা আন্ডারস্কোর (_) ব্যবহার করা যাবে। তবে ভ্যারিয়েবলের নামে অন্য কোনো প্রতীক চিহ্ন যেমন @,$, %, ^ ইত্যাদি ব্যবহার করা যাবে না।

ভ্যারিয়েবলের সঠিক নামকরণ	ভ্যারিয়েবলের ভুল নামকরণ
z1yan	z!yan
a	8a
_variable	$variable
My_namE	@My_name

৩। পাইথন একটি কেস সেন্সেটিভ (Case Sensitive) প্রোগ্রামিং ভাষা। তাই একই অক্ষর ছোট হাতের ও বড় হাতের হলে পাইথন তাদের দুটি ভিন্ন ভ্যারিয়েবল হিসেবে বিবেচনা করবে।

যেমন, My_variable আর my_variable দুটি ভিন্ন ভ্যারিয়েবল হিসেবে বিবেচিত হবে।

এবারে নিচের ছক থেকে কোনো ভ্যারিয়েবলের নামকরণ সঠিক হয়েছে এবংকোনো ভ্যারিয়েবলের নামকরণ ভুল হয়েছে সেটি শনাক্ত কর–

ভ্যারিয়েবলের নাম	সঠিক/ভুল
Bd_cap1tal	
8class_Section_C	
d1gital_T3chn0logY	
Ch@tta0gram	
tiiigeeeeer	
Robotics learning	

আবার, ভ্যারিয়েবলের মধ্যে তথ্য জমা করার জন্য আমরা = চিহ্ন ব্যবহার করি। একে বলা হয় ভ্যালু অ্যাসাইন করা।

ধরি, আমাদের একটি ভ্যারিয়েবল আছে count

count এর মান যদি 5 রাখতে চাই, তাহলে প্রোগ্রামে আমরা লিখব-

count = 5

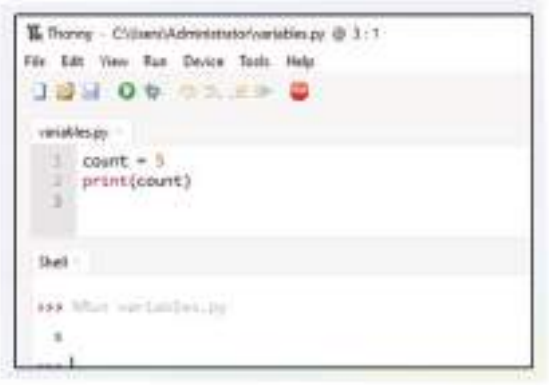
এরপর এই মানটি প্রিন্ট করতে লিখব,

print(count)

তাহলে নিচের প্রোগ্রামটি আমরা যদি লিখে রান করি,

```
count = 5
print(count)
```

আমরা নিচের মতো আউটপুট পাব -



অর্থাৎ, count ভ্যারিয়েবলের যে মান, সেটি প্রিন্ট করা হয়েছে।

আবার, একটি ভ্যারিয়েবলের মান পুরো প্রোগ্রামে একাধিকবার পরিবর্তন করা সম্ভব। ভ্যারিয়েবলের মধ্যে নতুন মান এসাইন করা হলে আগের মান মুছে যায় ও সর্বশেষ এসাইন করা মানটি জমা থাকে।
নিচের প্রোগ্রামটি যদি রান করি, আউটপুট তাহলে কি হবে?

```
value_now = 1
print(value_now)
value_now= 2
print(value_now)
value_now=3
print(value_now)
```

এই প্রোগ্রামের আউটপুট নিচের ঘরে লিখে ফেলি–

লক্ষ করি, উপরের প্রোগ্রামে একই ভ্যারিয়েবল value_now কে আমরা বারবার প্রিন্ট করেছি। কিন্তু একেক সময়ে ভ্যারিয়েবলটির ভেতরে জমা থাকা তথ্য ভিন্ন ছিল, তাই প্রিন্ট করার পর ভিন্ন মান পেয়েছি।

আবার, ভ্যারিয়েবলের মধ্যে সবসময় একইরকম তথ্য বা ডেটা আমরা জমা রাখি না। যেই তথ্য জমা রাখব সেটি কেমন তথ্য, তার উপর ভিত্তি করে কয়েকটি তথ্যের প্রকারভেদ বা ডেটাটাইপ (Data Type) আছে-

ক) int : ভ্যারিয়েবলে আমরা পূর্ণসংখ্যা জমা রাখতে পারি। পূর্ণসংখ্যাকে ইংরেজিতে integer number (ইন্টিজার নাম্বার) বলা হয়। তাই ভ্যারিয়েবলে পূর্ণসংখ্যা জমা রাখলে তার ডেটাটাইপকে বলা হয় int. এখানে int হলো integer এর সংক্ষিপ্ত রূপ। এমন একটি উদাহরণ হলো-

a = 5

খ) float : ভ্যারিয়েবলে আমরা দশমিক যুক্ত সংখ্যা বা ভগ্নাংশ জমা রাখতে পারি। এমন সংখ্যাকে ইংরেজিতে floating number (ফ্লোটিং নাম্বার) বলা হয়। তাই ভ্যারিয়েবলে ভগ্নাংশ বা দশমিক যুক্ত সংখ্যা জমা রাখলে তার ডেটাটাইপকে বলা হয় float. এখানে float হলো floating এর সংক্ষিপ্ত রূপ। এমন একটি উদাহরণ হলো-

a = 5.09

গ) str: ভ্যারিয়েবলে যদি কোনো টেক্সট বা অক্ষর জাতীয় তথ্য জমা রাখতে চাই, তাহলে সেটিকে string (স্ট্রিং) বলা হয়। আর এ ধরনের তথ্য str ডেটাটাইপের অন্তর্ভুক্ত হয়। এখানে str হলো string এর সংক্ষিপ্ত রূপ। আমরা ভ্যারিয়েবলে যেই টেক্সট রাখতে চাই তা (Single Quotation) এরমধ্যে জমা রাখব।

এমন উদাহরণ নিচে দেখি-

a='c'

b='This is a string variable'

ঘ) bool: কোন ভ্যারিয়েবলে যদি সত্য (True) অথবা মিথ্যা (False) কে তথ্য হিসেবে জমা রাখতে চাই, তাহলে সেটি হবে বুলিয়ান (boolean) তথ্য। এধরনের তথ্যকে বলা হয় bool ডেটাটাইপের তথ্য। এখানে bool হলো boolean এর সংক্ষিপ্ত রূপ। bool ডেটাটাইপে দুটি মাত্র তথ্য রাখা যায়- True ও False

এমন উদাহরণ নিচে দেখি-

a= True

আমরা চারটি ডেটাটাইপ জানলাম - int, float, str এবং bool

এগুলো হলো পাইথনে প্রধান চারটি ডেটাটাইপ। এছাড়াও আরও কিছু ডেটাটাইপ আছে, যদি কখনও প্রোগ্রাম লেখার সময় প্রয়োজন হয় আমরা সেগুলো সম্পর্কে তখন জানতে পারব। ভ্যারিয়েবলের মধ্যে আমরা যেই তথ্য জমা রাখি, সেটা জমা হয় কম্পিউটারের মেমোরিতে। তাই যখন আমরা প্রোগ্রামের মধ্যে কোথাও ভ্যারিয়েবল ব্যবহার করব, কম্পিউটার মেমোরিতে জমা থাকা ভ্যারিয়েবলটির মান তখন প্রোগ্রামে ব্যবহার হবে।

মজার জিনিস হলো, কোনো ভ্যারিয়েবলের ডাটটাইপ কোনটি, সেটি সহজেই বের করা যায় type() এর মাধ্যমে। আমরা যদি একটি ভ্যারিয়েবলকে type() ফাংশনের ভিতরে রেখে প্রিন্ট করি, তাহলে ওই ভ্যারিয়েবলের ডেটাটাইপ পেয়ে যাব।

যেমন নিচের প্রোগ্রামটি যদি রান করি,

```
my_variable = 23.07
print(my_variable)
print(type(my_variable))
```

আউটপুট পাব নিচের মতো -

```
23.07
<class 'float'>
```

অর্থাৎ, আমরা বুঝতে পারলাম my_variable নামক ভ্যারিয়েবলের কাছে জমা করা তথ্য 23.07 এবং এটি একটি float ডেটাটাইপের অন্তর্ভুক্ত ভ্যারিয়েবল।

এবারে আমরা বাসায় নিচের ভ্যারিয়েবলগুলোর ডেটাটাইপ বের করার চেষ্টা করি –

প্রোগ্রাম	ডেটাটাইপ
Ab = True	
my_value = 'Variable have some data types'	
f = 23	
status_is = 'False'	
number_now = 12.789	
section = 'b'	

সেশন-৬: ইনপুট নেওয়া শুরু করি

আগের সেশনে আমরা ভ্যারিয়েবলে তথ্য অ্যাসাইন করা শুরু করেছি। আমরা যদি প্রোগ্রামের ব্যবহারকারীর থেকে একটি তথ্য ইনপুট নিতে চাই তাহলে কি করব? কাজটি করা খুবই সহজ input () ব্যবহার করে।

আমরা যদি নিচের মতো লেখি-

my_input= input ()

তাহলে my_input ভ্যারিয়েবলটি আমাদের থেকে একটি ইনপুট গ্রহণ করবে। তবে, input () প্রোগ্রামের ভিতরে ইনপুট হিসেবে কোনো সংখ্যা, অক্ষর ইত্যাদি যেটাই গ্রহণ করুক না কেন, সেটিকে str ডেটাটাইপে গ্রহণ করবে।

চল একটি কাজ করি, একটি ভ্যারিয়েবল ইনপুট নিয়ে সেটি প্রিন্ট করি। এমন একটি প্রোগ্রাম লেখা খুব সহজ। নিচের মতো করে একটি প্রোগ্রাম লিখে রান করি–

```
my_input = input()
print(my_input)
print(type(my_input))
```

এই প্রোগ্রামের আউটপুট তাহলে কি হবে? তুমি যা ইনপুট দিবে সেটিই কিন্তু আউটপুট হিসেবে প্রিন্ট হবে। কিন্তু খেয়াল করে দেখ, তুমি কোনো পূর্ণসংখ্যা বা ভগ্নাংশ ইনপুট দিলেও সেটির ডেটাটাইপ হিসেবে str প্রিন্ট হচ্ছে। তার মানে, তুমি যেই তথ্যই জমা দাও, input() তথ্যটিকে একটি স্ট্রিং হিসাবে গ্রহণ করবে। কিন্তু তুমি যদি চাও ইনপুটটি যেন স্ট্রিং হিসেবে জমা না হয়ে ইন্টিজার বা ফ্লোট ডেটা হিসেবে জমা হয় তাহলে তোমাকে তথ্যটি ওই নির্দিষ্ট ডেটাটাইপে রূপান্তর করে নিতে হবে। একে বলা হয় টাইপ কাস্টিং (Type Casting) করা।

আমাদের চারটি প্রধান ডেটাটাইপের মধ্যে int, float, str ও bool এ রূপান্তর করা সহজ।

ডেটাটাইপ	ওই ডেটাটাইপে রূপান্তর করতে লিখব
int	int()
float	float()
str	str()
bool	bool()

আমরা যখন কোনো তথ্য ইনপুট নিচ্ছি, input() ফাংশনকে সরাসরি int ডেটাটাইপে রূপান্তর করা যাবে নিচের মতো প্রোগ্রাম করে-

```
my_input = int(input())
print(my_input)
print(type(my_input))
```

প্রোগ্রামটি রান করার পর খেয়াল করে দেখো int(input()) লেখার কারণে my_input ভ্যারিয়েবলটি int ডেটা টাইপে রূপান্তরিত হচ্ছে।

তুমি কি একইভাবে my_input ভ্যারিয়েবলকে float ডেটা টাইপে রূপান্তর করার একটি প্রোগ্রাম লিখতে পারবে? নিচের ঘরে এমন একটি প্রোগ্রাম লিখে ফেল–

আবার, আমরা চাইলে একটি তথ্য ইনপুট নেবার সময় নির্দিষ্ট কমান্ড বা নির্দেশনা দিয়ে দিতে পারব। সেজন্য input() এর ভিতরে " দিয়ে সেই কমান্ড লিখে দিতে পারি।

যেমন, আমরা যদি লেখি-

```
my_input = input('Provide a Sentence as an input:')
print(my_input)
print(type(my_input))
```

তাহলে প্রোগ্রামটি রান করার পর প্রথমে আমাদের কাছে একটি ইনপুট কমান্ড প্রদর্শন করবে-
Provide a Sentence as an input:

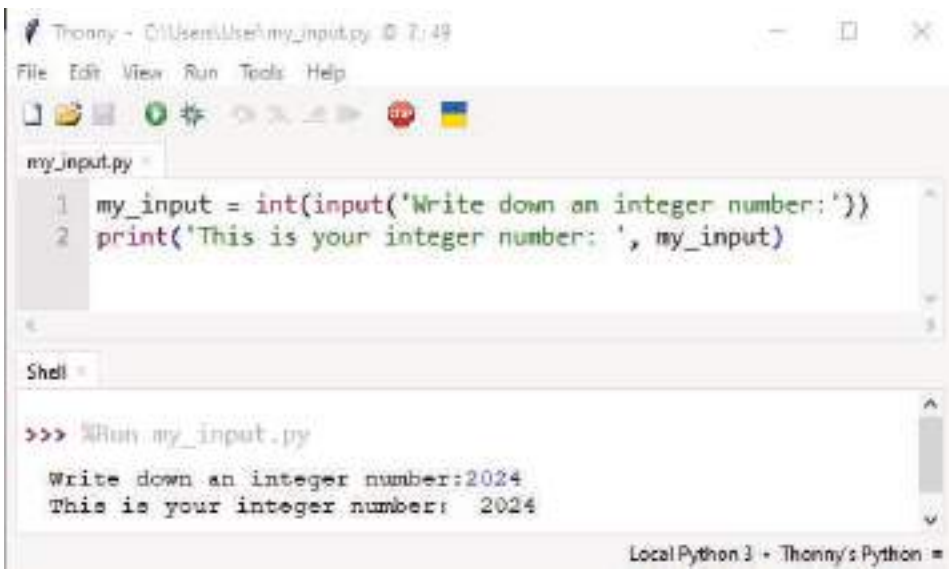
তারপর আমরা আমাদের ইনপুট দিলে সেই ইনপুট প্রিন্ট করবে এবং ডেটা টাইপ হিসেবে str দেখাবে।

আবার আমরা যদি চাই, আমাদের ইনপুট নেওয়া তথ্য প্রিন্ট করার আগে ও পরে আরও শব্দ বা বাক্য প্রিন্ট করতে পারি। এজন্য যেই শব্দ বা বাক্য প্রিন্ট করতে চাই, সেটি print() এর মধ্যে " এর ভিতরে লিখব ও তারপর , (কমা) চিহ্ন দিয়ে আমাদের ভ্যারিয়েবল লিখে দিব।

যেমন, আমরা যদি নিচের প্রোগ্রামটি লেখি,

```
my_input = int(input('Write down an integer number:'))
print('This is your integer number: ', my_input)
```

তাহলে নিচের মতো আউটপুট দেখতে পাব-



এখানে খেয়াল করে দেখো, print() এর মধ্যে যখন আমরা টেক্সট প্রিন্ট করছি, তখন সেটি ' ' বা একক উদ্ধৃতির ভিতরে লিখেছি। আর যখন আমরা একটি ভ্যারিয়েবল প্রিন্ট করেছি, সেটিকে কোন ' ' বা একক উদ্ধৃতির মধ্যে না রেখে সরাসরি ভ্যারিয়েবলটির নাম লিখেছি।

এবারে, নিচের প্রোগ্রামটি লেখার চেষ্টা করি-

সমস্যা - এমন একটি প্রোগ্রাম ডিজাইন কর, যা প্রথমে একটি ভ্যারিয়েবলকে পূর্ণসংখ্যা হিসেবে ইনপুট গ্রহণ করবে। এরপর আরেকটি ভ্যারিয়েবলকে দশমিক যুক্ত সংখ্যা ইনপুট হিসেবে গ্রহণ করবে। এরপর সংখ্যা দুটি প্রিন্ট করবে এবং তাদের ডেটাটাইপ প্রিন্ট করবে।
হিন্ট- ইনপুট দুটিকে অবশ্যই যথাক্রমে int ও float হিসেবে টাইপ কাস্টিং করে নিতে হবে।

সমাধান-

সেশন-৭: দুটি সংখ্যার ইনপুট নিয়ে করি যোগ, বিয়োগ, গুণ, ভাগ এবং মডুলাস

আগের সেশনে আমরা ইনপুট নেওয়া শিখেছি input() ফাংশন ব্যবহার করে। আমরা চাইলে দুটি সংখ্যা ইনপুট নিয়ে তাদের মধ্যে গাণিতিক অপারেশন (Arithmetic Operation) করতে পারি।

এক্ষেত্রে নিচের অপারেটরগুলো ব্যবহার করতে পারি আমরা সহজেই-

অপারেটর	কাজ
+	এটি যোগ (Addition) অপারেটর। এই অপারেটর ব্যবহার করে অপারেটরটির দুপাশে থাকা দুটি ভ্যারিয়েবলের যোগফল বের করা যাবে।
-	এটি বিয়োগ (Subtraction) অপারেটর। এই অপারেটর ব্যবহার করে অপারেটরটির দুপাশে থাকা দুটি ভ্যারিয়েবলের বিয়োগফল বের করা যাবে।
*	এটি গুণ (Multiplication) অপারেটর। এই অপারেটর ব্যবহার করে অপারেটরটির দুপাশে থাকা দুটি ভ্যারিয়েবলের গুণফল বের করা যাবে।
/	এটি ভাগ (Division) অপারেটর। এই অপারেটর ব্যবহার করে অপারেটরটির বামপাশে থাকা ভ্যারিয়েবলকে ডানপাশে থাকা ভ্যারিয়েবল দিয়ে ভাগ করে ভাগফল বের করা যাবে।
%	এটি মডুলো (Modulo) অপারেটর। এই অপারেটর ব্যবহার করে অপারেটরটির বামপাশে থাকা ভ্যারিয়েবলকে ডানপাশে থাকা ভ্যারিয়েবল দিয়ে ভাগ করে ভাগশেষ বের করা যাবে।

আমরা যদি দুটি সংখ্যা ইনপুট নিয়ে তাদের যোগফল প্রিন্ট করতে চাই, আমাদের জন্য সহজ হবে প্রথমে সুডো কোড তৈরি করে , তারপর সুডো কোডের ধাপগুলো অনুসরণ করে একটি প্রোগ্রাম ডিজাইন করি। আমাদের দুটি সংখ্যা ইনপুট নিয়ে যোগফল বের করার সুডো কোড নিচের মতো হবে-

```
ক = প্রথম ইনপুট নেই
খ = দ্বিতীয় ইনপুট নেই
গ = ক+খ
গ সংখ্যাটি প্রিন্ট করি
```

আমরা এর আগে একটি পূর্ণসংখ্যা কীভাবে ইনপুট নিতে হয় সেটি দেখেছি। কোনো ভ্যারিয়েবল প্রিন্ট কীভাবে করতে হয় সেটিও দেখেছি। চল, এখন দেখি কীভাবে আমরা প্রোগ্রাম ডিজাইন করে দুটি পূর্ণসংখ্যা ইনপুট নিয়ে তাদের যোগফল প্রিন্ট করতে পারি–

```
num1 = int(input('Enter the first integer:'))
num2 = int(input('Enter the second integer:'))
result = num1 + num2
print('The sum of', num1, 'and', num2,'is', result)
```

তুমি কি প্রোগ্রামের লাইনগুলো বুঝতে পারছ?

num1 = int('input('Enter the first integer:'))

এই লাইনে আমরা একটি ইনপুট নিয়েছি input() ব্যবহার করে। এসময় ইনপুটের মধ্যে আমরা কমান্ড দিয়ে দিয়েছি যে আমরা একটি ইন্টিজার বা পূর্ণ সংখ্যাকে ইনপুট হিসাবে চাচ্ছি। ইনপুট তো সবকিছু স্ট্রিং হিসাবে ইনপুট নেয়, তাই না? একারণে ইনপুটকে টাইপ কাস্টিং করে পূর্ণসংখ্যায় রূপান্তর করতে int() ব্যবহার করেছি। এরপর এই পূর্ণসংখ্যা ইনপুট পেলে সেটি জমা করেছি num1 ভ্যারিয়েবলের ভিতরে।

একইভাবে পরের লাইনে লিখেছি

num2 = int(input('Enter the second integer:'))

এই লাইনে একইভাবে num2 ভ্যারিয়েবলে দ্বিতীয় পূর্ণসংখ্যাটি ইনপুট নিয়েছি।

এরপর আমাদের সংখ্যা দুটি যোগ করে আরেকটি ভ্যারিয়েবল result এর মধ্যে আমরা জমা রাখব। এজন্য আমরা পরের লাইনে লিখেছি-

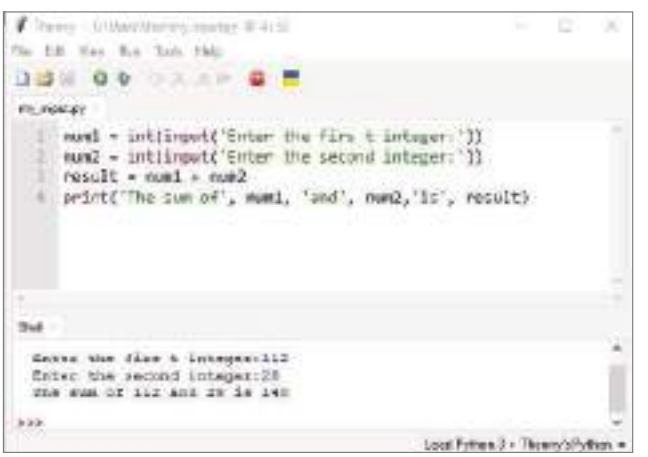
result = num1 + num2

সবশেষে আমরা সংখ্যা দুটির যোগফল প্রিন্ট করেছি এভাবে-

print('The sum of', num1, 'and', num2,'is', result)

কত সহজ ও মজার না যোগ করার এই প্রোগ্রামটি?

উপরের প্রোগ্রামটি রান করলে তুমি দুটি সংখ্যা ইনপুট দিতে পারবে পছন্দমতো এবং তারপর সংখ্যা দুটির যোগফল বের করতে পারবে। নিচের ছবিতে আমরা এমন দুটি সংখ্যা ইনপুট দিয়ে যোগফল বের করার উদাহরণ দেখতে পাচ্ছি-



তুমি কি একইভাবে গুণ করার একটি প্রোগ্রাম ডিজাইন করতে পারবে?

প্রথমে নিচের ছকে দুটি সংখ্যা ইনপুট নিয়ে তাদের গুণফল প্রিন্ট করার একটি সুডো কোড লিখে ফেলো–

এবারে তোমার তৈরি করা সুডো কোড অনুসরণ করে একটি প্রোগ্রাম ডিজাইন কর নিচের ছকে এবং সেটিকে রান করে দেখ।

আমরা এভাবে দুটি সংখ্যা ইনপুট নিয়ে তাদের যোগফল, বিয়োগফল, গুণফল, ভাগফল এবং ভাগশেষ বের করতে পারব খুব সহজেই একটি প্রোগ্রাম ডিজাইন করে।

বাড়ির অনুশীলনী

আমদের আশেপাশে এমন অনেক সমস্যা আছে যা নির্দিষ্ট সংখ্যার উপর গাণিতিক অপারেশন করে সমাধান করা যায়। যেমন আমরা চাইলে সেলসিয়াস থেকে ফারেনহাইটে তাপমাত্রার মান রূপান্তর করতে পারি।আমরা চাইলে পাঁচটি পণ্য কিনে সেগুলোর মোট দাম হিসাব করতে পারি ইত্যাদি।এবারে আমরা এমন তিনটি সমস্যা খুঁজে বের করি, যেখানে আমাদের এরকম কোন গাণিতিক হিসাব নিকাশ করে ফলাফল বের করতে হয়। এমন সমস্যাগুলো নিচের ছকে লিখে ফেলি -

১।

২।

৩।

সেশন-৮ ও ৯: সমস্যা সমাধানের জন্য প্রোগ্রাম ডিজাইন

এবারে আমাদের শিক্ষক আমাদের কয়েকটি দলে ভাগ করে দিবেন। প্রতিটি দলে সর্বোচ্চ ৫-৬ জন শিক্ষার্থীকে নির্ধারণ করবেন শিক্ষক। আমরা প্রত্যেকে যে ৩টি করে সমস্যা বের করেছি, দলের বাকিদের সাথে সমস্যাগুলো নিয়ে আলোচনা করি। এর আগে আমরা দেখেছি কীভাবে একটি প্রোগ্রাম ডিজাইন করা যায় ও বিভিন্ন ইনপুট নিয়ে যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদি প্রয়োগ করে সেখান থেকে আউটপুট বের করা যায়।

তাই এবারে আমরা দলের সবাই মিলে আলোচনা করে একটি সমস্যা নির্ধারণ করি, যেই সমস্যা সমাধানের জন্য আমরা প্রোগ্রাম ডিজাইন করব।

প্রোগ্রাম ডিজাইন করার আগে প্রথমে এই সমস্যা সমাধানের জন্য একটি সুডো কোড তৈরি করি -

আমাদের দলের নির্ধারিত বাস্তব সমস্যা
প্রোগ্রামে যেই ইনপুটগুলো নিতে হবে তার তালিকা -
কী কী গাণিতিক অপারেশন করতে হবে?

আমাদের তৈরি করা সুডো কোড

এই পর্যায়ে সুডো কোড অনুসরণ করে একটি পাইথন প্রোগ্রাম ডিজাইন করি। অবশ্যই আমাদের প্রোগ্রামে input(), print(), গাণিতিক অপারেশন ব্যবহার করব।

আমাদের তৈরি করা পাইথন প্রোগ্রাম নিচের ছকে লিখে ফেলি -

সেশন-১০: বিভিন্ন ইনপুটের ত্রুটির জন্য ঝুঁকি শনাক্ত ও সমাধান করা

আমাদের তৈরি করা প্রোগ্রামে যদি ভুল ইনপুট দেই তাহলে সঠিক আউটপুট বা ফলাফল পাব না। এগুলোই আমাদের প্রোগ্রামের ত্রুটি সংঘটিত হবার ঝুঁকি। প্রোগ্রাম ডিজাইনের সময়ে তাই এটাও খেয়াল রাখা জরুরি কীভাবে এরকম বিভিন্ন ঝুঁকি শনাক্ত করা যায় এবং সেটি সমাধান করা যায়।

তাই বিভিন্ন রকম ইনপুট দিয়ে আমাদের প্রোগ্রামে সেই ইনপুট অনুযায়ী কি কি সমস্যা তৈরি হয় সেটি খুঁজে বের করা দরকার।

চলো আমাদের প্রোগ্রামে আমরা ৪-৫ টি ভিন্ন রকম ইনপুট দেই এবং কিছু ভুল ইনপুটও দেই। দিয়ে যাচাই করি এক্ষেত্রে কি কি সমস্যা হচ্ছে।

ক্রম	ভুল ইনপুট	প্রোগ্রামে এর প্রভাব

এবারে একটি কর্মপরিকল্পনা তৈরি করা দরকার যেন আমরা এই সম্ভাব্য ত্রুটিগুলো দূর করে আমাদের প্রোগ্রাম আরও নির্ভুল করতে পারি। তাই দলের সবাই মিলে আলোচনা করি কীভাবে এই সম্ভাব্য ত্রুটি ও তার ঝুঁকি আমরা সমাধান করতে পারি।

আমাদের কর্মপরিকল্পনার মূল অংশ নিচে লিখে ফেলি -

সম্ভাব্য ঝুঁকি নিরূপণে আমাদের কর্মপরিকল্পনা -
প্রোগ্রামে যা পরিবর্তন আনা প্রয়োজন -

এবারে সেই অনুযায়ী আমাদের প্রোগ্রাম পরিবর্তন করে নতুন প্রোগ্রাম কম্পিউটারে রান করি ও নিচে লিখে ফেলি -

সেশন-১১: ভিন্ন দলের প্রোগ্রাম যাচাই

এবারে আমরা নিজেদের দলের প্রোগ্রাম অন্য আরেকটি দলের সাথে বিনিময় করব। যেই দলের প্রোগ্রাম আমরা পেলাম, তা ওই দলের নির্ধারিত বাস্তব সমস্যা সমাধানে সক্ষম কি না সেটি যাচাই করি।

পাশাপাশি এই বিষয়ে একটি সংক্ষিপ্ত প্রতিবেদন তৈরি করে ফেলি নিচের ছকগুলো পূরণ করে -

যেই দলের প্রোগ্রাম যাচাই করেছি তাদের নির্ধারিত সমস্যা -
সমাধানে দলটির ডিজাইন করা প্রোগ্রাম -
এই প্রোগ্রামে ব্যবহৃত বিভিন্ন ইনপুটের তালিকা -
প্রোগ্রামটি থেকে বিভিন্ন ধাপে প্রাপ্ত আউটপুট -

প্রোগ্রামটিতে ইনপুটের ভিন্নতায় কোনো ত্রুটি তৈরি হলে সেটি শনাক্ত করি -
সেই ত্রুটি মোকাবেলায় আমাদের দলের পক্ষ থেকে পরামর্শ -

পুরো প্রতিবেদন তৈরি হয়ে গেলে ওই দলকে প্রতিবেদনটি বুঝিয়ে দেই এবং নিজেদের দলের প্রতিবেদন গ্রহণ করি।

তাহলে আমরা এই অভিজ্ঞতায় প্রোগ্রাম ডিজাইন করা সম্পর্কে ধারণা পেলাম, একটি সমস্যা সমাধানে বিভিন্ন ইনপুট দিয়ে গাণিতিক অপারেশন ব্যবহার করে প্রোগ্রাম ডিজাইন করা সম্পর্কেও জানতে পারলাম। আবার সমস্যাটি সমাধানে নানারকম ইনপুটের জন্য সম্ভাব্য আউটপুট ও সেক্ষেত্রে সম্ভাব্য ত্রুটি ঝুঁকি নির্ণয় করে সেটি সমাধান করার উপায় নিয়েও আমরা কাজ করলাম। এভাবেই আমরা প্রোগ্রামিং ব্যবহার করে বিভিন্ন সমস্যাকে সমাধান করতে পারি নিজেদের প্রয়োজনে।