নবম-দশম শ্রেণী

ডিজিটাল প্রযুক্তি

সমস্যা সমাধানে প্রোগ্রামিং

ক্লাস পরিচালনায়

মোঃ রনি সেখ

রোলঃ STICT00136

পদের নামঃ সিনিয়র শিক্ষক (আইসিটি)

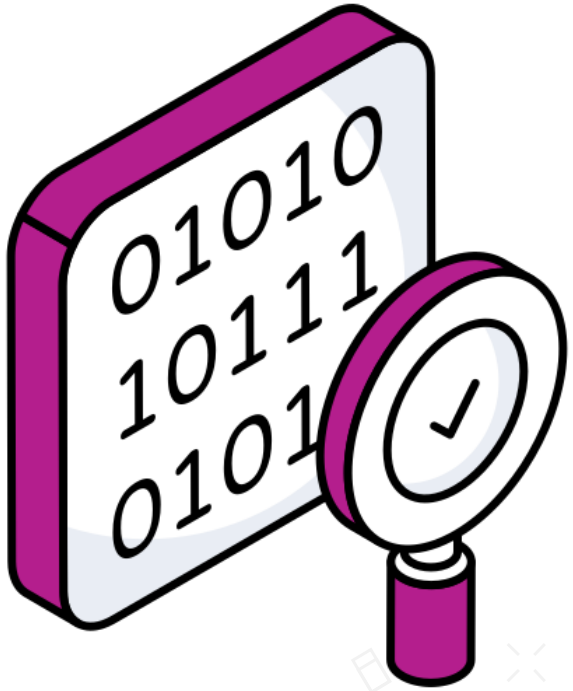
বীরশ্রেষ্ঠ মুন্সী আব্দুর রউফ পাবলিক কলেজ, বিজিবি সদর দপ্তর, পিলখানা, ঢাকা ।

বিষয়বস্তু

- প্রোগ্রাম ডিজাইনের সূচনা
  - বাইনারী সংখ্যা পদ্ধতি
  - মেশিন ভাষা
  - অনুবাদক (কম্পাইলার, ইন্টারপ্রেটার)
  - পাইথন
- ভ্যারিয়েবল এবং ডেটা টাইপ
- ইনপুট
- পাইথনে শর্তের ব্যবহার
  - if, elif, else
- পাইথনে লুপের ব্যবহার
  - While loop

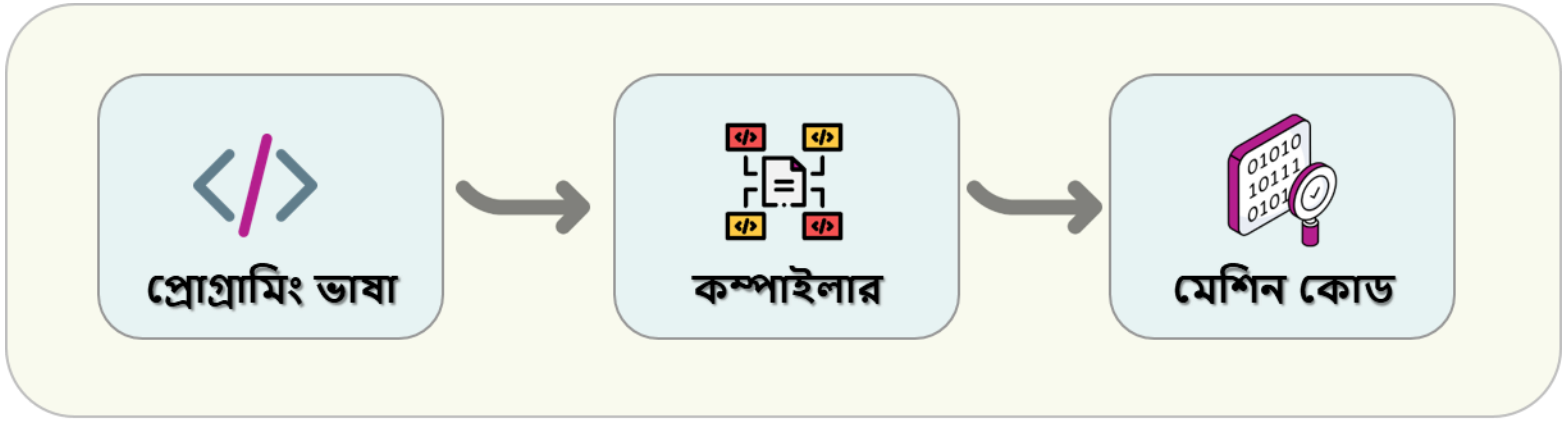
প্রোগ্রাম ডিজাইনের সূচনা

- **বাইনারী সংখ্যা পদ্ধতিঃ** যে সংখ্যা পদ্ধতিতে ০ এবং ১ অর্থাৎ দুটি মাত্র অংক ব্যবহার করা হয়, তাকে **বাইনারী সংখ্যা পদ্ধতি** বলে।
- **মেশিন ভাষাঃ** মূলত ০ এবং ১ এর সমন্বয়ে গঠিত ভাষাই হল মেশিন ভাষা।
- **অনুবাদকঃ** যেসকল প্রোগ্রামের মাধ্যমে উৎস প্রোগ্রাম/সোর্স কোডকে মেশিন/অবজেক্ট কোডে রুপান্তর করা হয়, তাকে অনুবাদক বলে।
- **অনুবাদক দুই ধরনেরঃ**
  - ✓ কম্পাইলার
  - ✓ ইন্টারপ্রেটার



প্রোগ্রাম ডিজাইনের সূচনা

- **কম্পাইলারঃ** কম্পাইলার হল একধরনের প্রোগ্রাম/সফটওয়্যার যা উচ্চ স্তরের প্রোগ্রামিং ভাষায় লেখা **সোর্স কোডকে মেশিন ভাষায় বা বাইনারী কোডে** অনুবাদ করে।

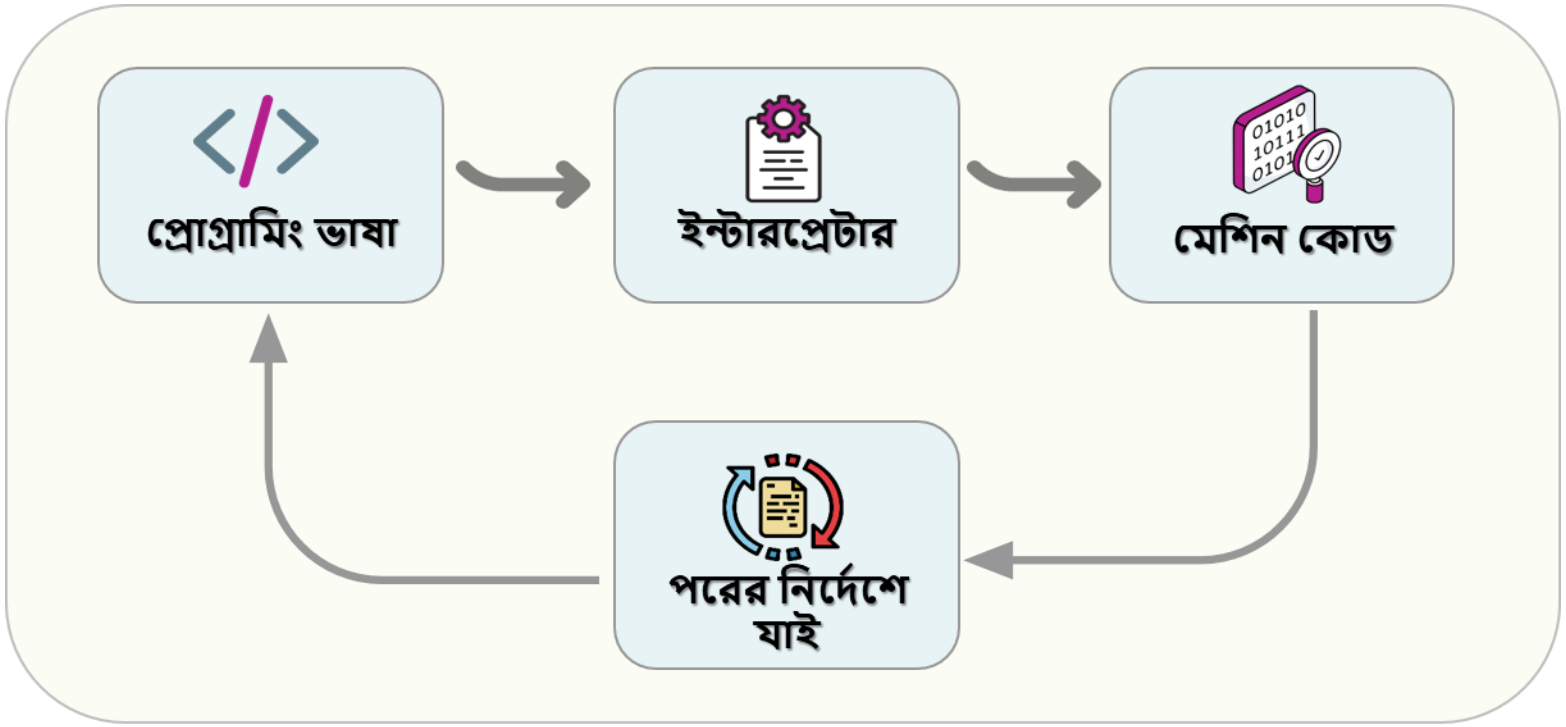


চিত্র ১: কম্পাইলার

- ✓ কম্পাইলার **সম্পূর্ণ প্রোগ্রামকে** একসাথে অনুবাদ করে।
- ✓ কম্পাইলার প্রোগ্রামের **সবগুলো ভুল** একসাথে প্রদর্শন করে।

প্রোগ্রাম ডিজাইনের সূচনা

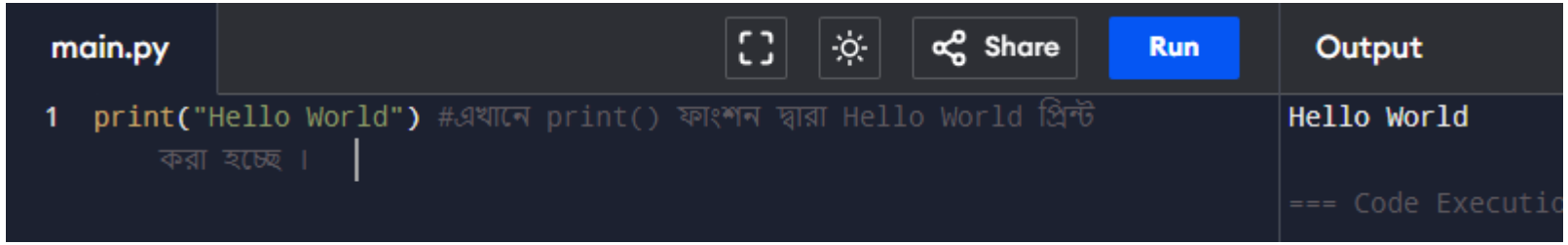
- **ইন্টারপ্রেটারঃ** ইন্টারপ্রেটার হল একধরনের প্রোগ্রাম/সফটওয়্যার যা উচ্চ স্তরের প্রোগ্রামিং ভাষায় লেখা **সোর্স কোডকে** এক লাইন করে পড়ে এবং তাৎক্ষণিকভাবে **মেশিন ভাষায় বা বাইনারি কোডে** অনুবাদ করে।



চিত্র ২: ইন্টারপ্রেটার

প্রোগ্রাম ডিজাইনের সূচনা

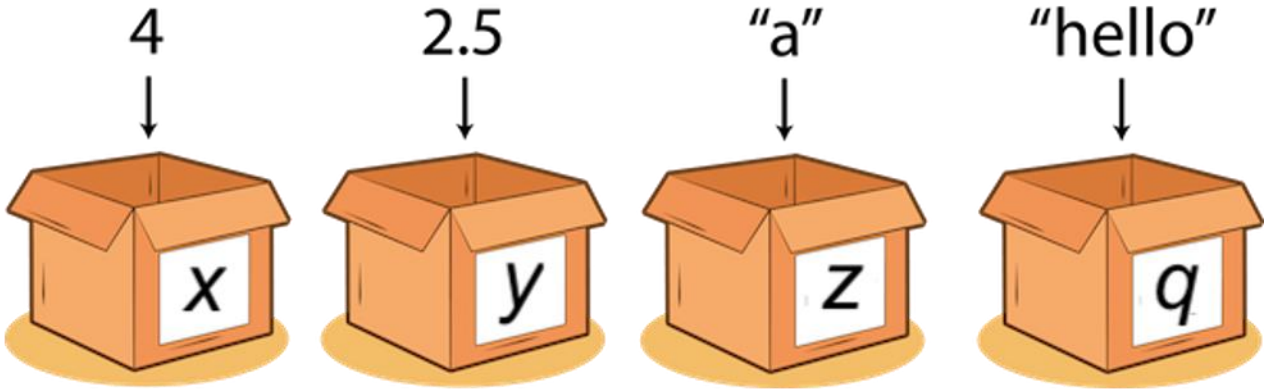
- **পাইথনঃ** পাইথন হল একটি **অবজেক্ট ওরিয়েন্টেড** উচ্চ স্তরের প্রোগ্রামিং ভাষা।
  - ✓ ১৯৯১ সালে নেদারল্যান্ডের কম্পিউটার বিজ্ঞানী **"গুইডো ভ্যান রোসাম"** এই প্রোগ্রামিং ভাষা তৈরি করেন।
  - ✓ ব্রিটিশ কমেডি শো **"মন্টি পাইথন'স ফ্লাইং সার্কাস"** থেকে এই ভাষার নামকরণ করা হয়।
  - ✓ **১ম পাইথন প্রোগ্রামঃ**



গুইডো ভ্যান রোসাম

ভ্যারিয়েবল

- **ভ্যারিয়েবল** হল একটি বাক্সের মত যার মধ্যে নির্দিষ্ট একটি তথ্য জমা রাখা যায়।
- **ভ্যারিয়েবল** পরিবর্তনশীল
- প্রোগ্রামিং করার সময় যখন ডেটা সংরক্ষণ করার প্রয়োজন হয় তখন **মেমোরি স্পেস** দরকার হয়। এই মেমোরি স্পেসকেই প্রোগামিং এর ভাষায় **ভ্যারিয়েবল** বলে।
- ভ্যারিয়েবলের মধ্যে কোন ভ্যালু জমা রাখার জন্য **একটি সমান** (=) চিহ্ন ব্যবহার করা হয়।



চিত্র – ৩: ভ্যারিয়েবল

ভ্যারিয়েবল

- **ভ্যারিয়েবল নামকরনের শর্তঃ**
  - ✓ ভ্যারিয়েবলের নাম সবসময় **একটিমাত্র** শব্দে হবে। যেমনঃ **number = 7**
  - ✓ দুটি শব্দের মধ্যে **স্পেস** বাদ দিয়ে ভেরিয়েবলের নামকরণ করা যায় **যেমনঃ MyNumber = 7**
  - ✓ দুটি শব্দের মধ্যে **আন্ডারস্কোর** ব্যবহার করেও ভ্যারিয়েবলের নামকরণ করা যায়। **যেমনঃ My_Number =7**
  - ✓ ভ্যারিয়েবলের নামের **প্রথম অক্ষর (0-9)** বা **কোন প্রতীক যেমন** (*) ব্যবহার করা যাবে না।
  - ✓ পাইথন একটি **Case Sensitive** প্রোগ্রামিং ভাষা। যেমনঃ **My_variable** এবং **my_variable** দুটি ভিন্ন ভ্যারিয়েবল হিসেবে বিবেচিত হবে।

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

ভ্যারিয়েবল

- **ভ্যারিয়েবল ব্যবহার করে পাইথন প্রোগ্রামঃ**

main.py

Share Run

```
number = 7
print(number)

```

Output: 7

ডেটা টাইপ

- **int ডেটাটাইপঃ**
  - ✓ ভ্যারিয়েবলে আমরা পূর্ণসংখা জমা রাখতে পারি ।
  - ✓ পূর্ণসংখ্যাকে ইংরেজিতে **integer number** বলে । তাই ভ্যারিয়েবলে পূর্ণসংখ্যা জমা রাখলে তার ডেটাটাইপ কে **int** বলা হয় ।
  - ✓ **যেমনঃ** a = 5

- **float ডেটাটাইপঃ**
  - ✓ ভ্যারিয়েবলে আমরা **দশমিক সংখ্যা বা ভগ্নাংশ** জমা রাখতে পারি ।
  - ✓ এমন সংখ্যাকে ইংরেজিতে **floating number** বলে । তাই ভ্যারিয়েবলে **দশমিক সংখ্যা** জমা রাখলে তার ডেটাটাইপ কে **float** বলা হয় ।
  - ✓ **যেমনঃ** a = 5.03

ডেটা টাইপ

- **str ডেটাটাইপঃ**
  - ✓ ভ্যারিয়েবলে যদি কোন টেক্সট বা অক্ষর জাতীয় তথ্য জমা রাখতে চাই, তাহলে সেটিকে স্ট্রিং বলা হয়।
  - ✓ এ ধরনের ডেটা টাইপকে **str** বলা হয় যা **string** এর সংক্ষিপ্ত রূপ।
  - ✓ যেমনঃ a = ‘c’ এবং b=‘this is a string’

- **bool ডেটাটাইপঃ**
  - ✓ ভ্যারিয়েবলে **সত্য বা মিথ্যা** তথ্য জমা রাখার জন্য এই ডেটাটাইপ ব্যবহার করা হয়।
  - ✓ **যেমনঃ a = True**

ইনপুট

- **পাইথনে ইনপুট পদ্ধতি**
  - ✓ প্রোগ্রামে ব্যবহারকারীর কাছ থেকে **ইনপুট** নেয়ার জন্য পাইথনে **input()** ফাংশন ব্যবহার করা হয় ।

Input function

Variable name

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

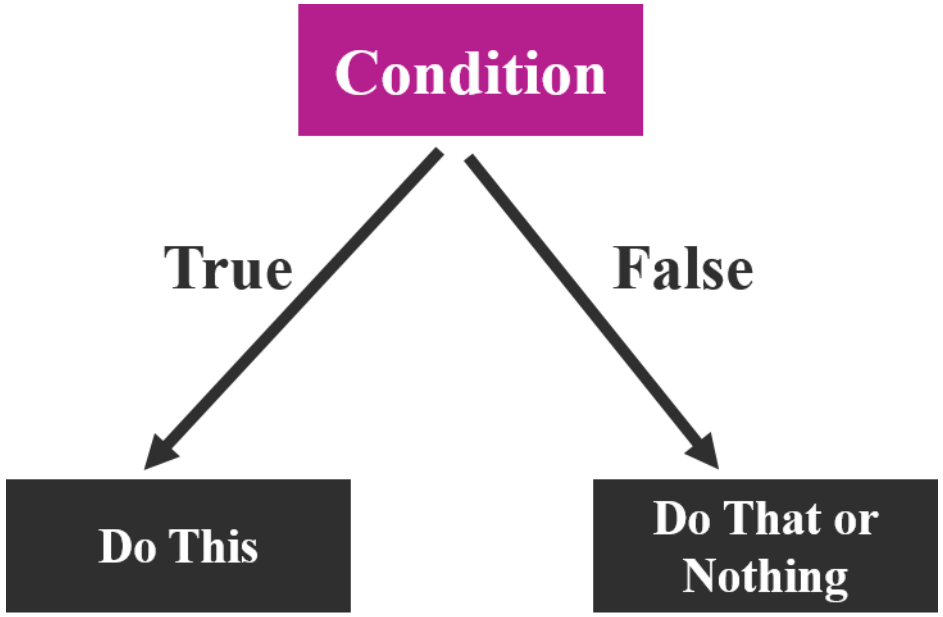
print() ফাংশন ব্যবহার করে ফলাফল প্রিন্ট করা হচ্ছে ।

  - ✓ ইনপুট নেয়ার সময় নির্দিষ্ট **কমান্ড/নির্দেশনা** দিতে -

```
my_input = input('Enter your name:')
```

পাইথনে শর্তের ব্যবহার

- শর্তসাপেক্ষে কোন স্টেটমেন্ট সম্পাদনের জন্য **কন্ডিশনাল কন্ট্রোল স্টেটমেন্ট** ব্যবহার করা হয়।
- **if , elif, else**



main.py Share Run

```python
if condition1:
    # condition1 সত্য হলে এই কোড ব্লক রান হবে
    statement1
elif condition2:
    # condition2 সত্য হলে এই কোড ব্লক রান হবে
    statement2
elif condition3:
    # condition3 সত্য হলে এই কোড ব্লক রান হবে
    statement3
else:
    # উপরের কোন condition-ই সত্য না হলে এই কোড ব্লক রান হবে
    statement_else
```

পাইথনে শর্তের ব্যবহার

- সমস্যা - ১

main.py

```python
# কাস্টমার পছন্দ
order = input("আপনার পছন্দের খাবারের নাম লিখুন: ").lower()

# মেনু অনুযায়ী সিদ্ধান্ত
if order == "pizza":
    print("আপনার পিজা প্রস্তুত হচ্ছে!")
elif order == "burger":
    print("বার্গার আসছে, একটু অপেক্ষা করুন!")
elif order == "pasta":
    print("আপনার জন্য পাস্তা প্রস্তুত করা হচ্ছে!")
else:
    print("দুঃখিত, এই খাবার আমাদের মেনুতে নেই।")
```

Output

```
আপনার পছন্দের খাবারের নাম লিখুন: pAstA
আপনার জন্য পাস্তা প্রস্তুত করা হচ্ছে!

=== Code Execution Successful ===
```

পাইথনে লুপের ব্যবহার

- পূনরাবৃত্তিমূলক কাজ সহজে এবং অল্প কোড লিখে সম্পন্ন করার জন্য **লুপ** ব্যবহার করা হয়।
- পাইথনে **২ ধরনের লুপ** রয়েছে।
  - ✓ **for লুপ**
  - ✓ **while লুপ**

- **while loop**

main.py

```
while condition:
    # কোড ব্লক যা condition সত্য থাকলে চলবে
    statement
```

পাইথনে লুপের ব্যবহার

- সমস্যা - ১

main.py

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

Output

```
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!

=== Code Execution Successful ===
```

initialization condition increment

main.py

```
counter = 0

while counter < 5:
    print("Hello, World!")
    counter += 1  # কাউন্টার বাড়ানো হচ্ছে
```

Output

```
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!

=== Code Execution Successful ===
```

পাইথনে শর্তের ব্যবহার

- সমস্যা - ২

main.py

```python
# সঠিক পাসওয়ার্ড সেট করা
correct_password = "python123"

# ব্যবহারকারীকে পাসওয়ার্ড দিতে বলা হচ্ছে
user_password = input("পাসওয়ার্ড দিন: ")

# যতক্ষণ না সঠিক পাসওয়ার্ড দেওয়া হচ্ছে, ততক্ষণ লুপ চলবে
while user_password != correct_password:
    print("পাসওয়ার্ড ভুল হয়েছে! আবার চেষ্টা করুন।")
    user_password = input("পাসওয়ার্ড দিন: ")

print("অভিনন্দন! আপনি সঠিক পাসওয়ার্ড দিয়েছেন।")
```

Output

```
পাসওয়ার্ড দিন: 123
পাসওয়ার্ড ভুল হয়েছে! আবার চেষ্টা করুন।
পাসওয়ার্ড দিন: python123
অভিনন্দন! আপনি সঠিক পাসওয়ার্ড দিয়েছেন।

=== Code Execution Successful ===
```

ধন্যবাদ