

# জাভা প্রোগ্রামিংয়ে মেনু নিয়ে টুকিটাকি

মো: আবদুল কাদের

অ্যাপ্লিকেশন প্রোগ্রামে মেনু অপরিহার্য। এমএস ওয়ার্ড, এক্সেল, পাওয়ার পয়েন্ট থেকে শুরু করে কমবেশি সব অ্যাপ্লিকেশনে মেনু রয়েছে। মেনু নিয়ে কথা উঠলেই অবধারিতভাবে চলে আসে মেনুতে কি আইটেম আছে সে বিষয়ে। সাধারণত মেনুতে বিদ্যমান আইটেম এবং কাজের প্রকৃতির ওপর নির্ভর করেই মেনুর নামকরণ করা হয়। ধরা যাক, ফাইল মেনুর কথা। নতুন ফাইল খোলা, ফাইল সেভ করা, প্রিন্ট করা বা ফাইল বন্ধ করা প্রভৃতি কাজ ফাইল মেনু দিয়ে করা যায়। একেকটি কাজ করার জন্য একেকটি অপশন/আইটেম রয়েছে। এই আইটেমগুলোকে বলা হয় মেনু আইটেম। অনেক সময় মেনু আইটেম হিসেবে আরেকটি মেনুও থাকতে পারে। যেমন- ফাইল মেনুর ভেতর মেনু আইটেম হিসেবে Save As রয়েছে, যা আরেকটি মেনু। এই মেনুতে ক্লিক করলে আইটেমগুলো দৃশ্যমান হয়। একটি আইটেমকে বিভিন্ন প্রয়োজনে কাজে লাগাতে এর ভেতর আইটেম যোগ করা হয়। ফলে আগের আইটেমটিকে তৈরি করতে হয় মেনু হিসেবে, যাতে পরের আইটেমগুলো অন্তর্ভুক্ত থাকে।



চিত্র-১ : মেনু আইটেম ও মেনু

মেনুর আইটেমগুলোকে কাজ অনুযায়ী সহজে বোঝার জন্য আইটেমের সাথে ছবি যুক্ত করা যায়। তা ছাড়া মেনুর আইটেমগুলোকে কাজের ধরন অনুযায়ী ভাগ করা যায়। ফাইলের প্রাথমিক কাজ যেমন- ফাইল ওপেন, সেভ করার জন্য একটি ভাগে। আবার ফাইলের আউটপুট যেমন-প্রিন্ট বা পাবলিশ করার জন্য একই ভাগে এবং ফাইলটিকে বন্ধ করার জন্য আরেকটি ভাগে ভাগ করা যায়। প্রত্যেকটি ভাগকে আলাদাভাবে বোঝানোর জন্য একটি হরাইজন্টাল লাইন দিয়ে আইটেমগুলোকে আলাদা করা হয়ে থাকে (চিত্র-১)। আবার অনেক ক্ষেত্রে আইটেমকে কিবোর্ড দিয়ে সরাসরি কাজ করার জন্য আইটেমের পাশে কিবোর্ড শর্টকাটও দেয়া থাকে। ফলে ইউজারকে মেনুতে গিয়ে আইটেমে ক্লিক করার প্রয়োজন পড়ে না।



চিত্র-২ : কিবোর্ড শর্টকাট

এ লেখায় জাভা দিয়ে সহজভাবে মেনু তৈরির একটি প্রোগ্রাম দেখানো হয়েছে। এতে মেনু, মেনু আইটেম, মেনু আইটেম হিসেবে মেনু সংক্রান্ত বিষয়গুলো থাকবে।

প্রোগ্রামটি রান করার পদ্ধতি অন্যান্য জাভা প্রোগ্রামের মতোই। এজন্য অবশ্যই আপনার কমপিউটারে Jdk সফটওয়্যার ইনস্টল থাকতে হবে। এখানে সফটওয়্যারটির Jdk1.4 ভার্সন ব্যবহার করা হয়েছে এবং প্রোগ্রামগুলো D:\ড্রাইভের java ফোল্ডারে সেভ করা হবে।

নিম্নের এই প্রোগ্রামটি নোটপ্যাডে টাইপ করে MenuEx.java নামে সেভ করতে হবে।

```
import java.awt.*;
import javax.swing.*;
class MenuEx extends JFrame {
public MenuEx() {
super("Menu Example");
setSize(600, 100);
JMenu menu1, menu2, menu3, menu4,
menu5;
JMenuItem item1, item2, item3, item4,
item5, item6, item7, item8, item9, item10,
item11;
JMenuBar jb=new JMenuBar();
menu1=new JMenu("File");
item1=new JMenuItem("New");
item2=new JMenuItem("Open");
item3=new JMenuItem("Save");
item4=new JMenuItem("Exit");
menu2=new JMenu("Print");
item5=new JMenuItem("Quick Print");
item6=new JMenuItem("Print Preview");
menu1.add(item1);
menu1.add(item2);
menu1.add(item3);
menu2.add(item5);
menu2.add(item6);
menu1.add(menu2);
menu1.add(item4);
jb.add(menu1);
menu3=new JMenu("Edit");
item7=new JMenuItem("Cut");
item8=new JMenuItem("Paste");
menu3.add(item7);
menu3.add(item8);
jb.add(menu3);
menu4=new JMenu("Format");
item9=new JMenuItem("Font");
item10=new JMenuItem("WordArt");
menu4.add(item9);
menu4.add(item10);
jb.add(menu4);
menu5=new JMenu("Help");
item11=new JMenuItem("About");
menu5.add(item11);
jb.add(menu5);
getContentPane().add(jb);
}
public static void main(String args[]) {
FlowLayout fl=new
FlowLayout(FlowLayout.LEFT);
MenuEx jf=new MenuEx();
jf.getContentPane().setLayout(fl);
jf.setSize(400,300);
jf.show();
}
}
```

## প্রোগ্রাম রান করা

জাভার আগের প্রোগ্রামগুলোর মতো কমান্ড প্রম্পট ওপেন করে চিত্র-৩-এর মতো রান করতে হবে।



চিত্র-৩ : প্রোগ্রাম রান করার পদ্ধতি



চিত্র-৪ : প্রোগ্রাম রান করার পর আউটপুট

পরবর্তী পর্বগুলোতে জাভানির্ভর গুরুত্বপূর্ণ ও প্রয়োজনীয় প্রোগ্রামগুলো নিয়ে আলোচনা করা হবে

ফিডব্যাক : [balraith@gmail.com](mailto:balraith@gmail.com)

## পাইথনে হাতেখড়ি

(৭০ পৃষ্ঠার পর)

করে করতে পারেন।

```
>>> f = open('workfile', 'r')
```

```
>>> f.read()
```

'this is first line\nthis is second line\nthis is third line\n'

```
>>> f.read()
```

..

read() মেথড ব্যবহার করে পুরো ফাইল একসাথে রিড করা যায়। দ্বিতীয়বার একই কমান্ড দিলে ফাঁকা স্ট্রিং দেখাবে।

প্রতিটি লাইন আলাদা করে পড়তে চাইলে নিচের মতো করতে হবে।

```
>>> f = open('workfile', 'r')
```

```
>>> f.readline()
```

'this is first line\n'

```
>>> f.readline()
```

'this is second line\n'

```
>>> f.readline()
```

'this is third line\n'

```
>>> f.readline()
```

..

লুপের মাধ্যমেও এটা করতে পারবেন।

```
>>> for line in f:
```

```
print(line, end="")
```

this is first line

this is second line

this is third line

সব লাইনগুলোকে একটি লিস্ট রাখতে চাইলে list(f) বা f.readlines() ব্যবহার করেই তা করতে পারবেন।

যখন ফাইলের ওপর অপারেশন চালানো শেষ হবে, তখন f.close() ফাংশন ব্যবহার করতে হবে। তাহলে এটি সিস্টেমের যে রিসোর্সগুলো open()-এর মাধ্যমে ধরে রেখেছিল তা ছেড়ে দেবে। এ জন্য ভালো পদ্ধতি হচ্ছে with কিওয়ার্ড ব্যবহার করা। এর মাধ্যমে ফাইল ওপেন করলে কাজ শেষে নিজে থেকেই ফাইল ক্লোজ করে দেয়।

```
>>> with open('workfile', 'r') as f:
```

```
read_data = f.read()
```

```
>>> f.closed
```

True

ফিডব্যাক : [ahmadalsajid@gmail.com](mailto:ahmadalsajid@gmail.com)