

# □ জুবায়ের ইনস্টিটিউট অব টেকনোলজি

খিরাটি, কাপাসিয়া, গাজীপুর

## ফ্লাটার কোর্স সিলেবাস

ক্লাস নম্বর	বিষয়বস্তু
ক্লাস-১	Dart Programming Introduction, Data types, Variables
ক্লাস-২	Dart Loops
ক্লাস-৩	Conditional Logics
ক্লাস-৪	Function (Parameter, Return type)
ক্লাস-৫	Flutter Widget Introduction, Platform specific widgets, Layout widgets, Platform independent / basic widgets
ক্লাস-৬	Layout Widget in detail.
ক্লাস-৭	Create a basic one page App Part-1
ক্লাস-৮	Create a basic one page App Part-2
ক্লাস-৯	Asset, Image, Http, Network image
ক্লাস-১০	
ক্লাস-১১	
ক্লাস-১২	
ক্লাস-১৩	
ক্লাস-১৪	
ক্লাস-১৫	
ক্লাস-১৬	
ক্লাস-১৬	
ক্লাস-১৭	
ক্লাস-১৮	

ক্লাস-২০	
ক্লাস-২১	
ক্লাস-২২	
ক্লাস-২৩	
ক্লাস-২৪	
ক্লাস-২৫	
ক্লাস-২৬	
ক্লাস-২৭	
ক্লাস-২৮	
ক্লাস-২৯	
ক্লাস-৩০	
ক্লাস-৩১	
ক্লাস-৩২	
ক্লাস-৩৩	
ক্লাস-৩৪	
ক্লাস-৩৫	
ক্লাস-৩৬	
ক্লাস-৩৭	
ক্লাস-৩৮	
ক্লাস-৩৯	
ক্লাস-৪০	
ক্লাস-৪১	
ক্লাস-৪২	
ক্লাস-৪৩	
ক্লাস-৪৪	

# Flutter বাংলা টিউটোরিয়াল

মোঃ আবির হোসেন

## Flutter কি?

Flutter হলো গুগলের একটি ওপেনসোর্স সফটওয়্যার ডেভেলপমেন্ট ফ্রেমওয়ার্ক যা দিয়ে সহজেই অ্যান্ড্রয়েড, আইওএস, ওয়েব, ডেস্কটপ ইত্যাদি অপারেটিং সিস্টেমের জন্য অ্যাপ্লিকেশন ডিজাইন করা যায়।

Dart প্রোগ্রামিং ল্যাংগুয়েজ ব্যবহার করে ফ্লাটারে অ্যাপ্লিকেশন ডেভেলপ করা হয়।

## কম্পিউটারে Flutter ইনস্টল করব কিভাবে?

Flutter উইন্ডোজ, ম্যাকওএস, লিনাক্স ইত্যাদি অপারেটিং সিস্টেমে ইনস্টল করা যায়। এখানে উইন্ডোজ অপারেটিং সিস্টেমে কিভাবে ইনস্টল করতে হবে সেটা ব্যখ্যা করা হলো:

<https://docs.flutter.dev/get-started/install/windows>

এই লিংক থেকে ফ্লাটার ডাউনলোড ও ইনস্টল করার সম্পূর্ণ প্রক্রিয়া জানা যাবে।

## Flutter এর বৈশিষ্ট্য:

Flutter ফ্রেমওয়ার্ক ডেভেলপারদের নিম্নোক্ত সুবিধাগুলো দেয় –

- এটি একটি আধুনিক এবং সক্রিয় ফ্রেমওয়ার্ক
- এতে Dart প্রোগ্রামিং ল্যাংগুয়েজ ব্যবহৃত হয় যা সহজেই শেখা যায়।
- দ্রুত এবং অল্প সময়ে ডেভেলপ করা যায়।
- সুন্দর এবং মসৃণ User Interface ডেভেলপ করা যায়।
- অসংখ্য Widget ব্যবহার করার সুবিধা।
- একই UI বিভিন্ন প্ল্যাটফর্ম সাপোর্ট করে।

## Flutter এর সুবিধা:

Flutter এ অসংখ্য কাস্টমাইজেবল সুন্দর সুন্দর widget আছে যা দিয়ে হাই পারফরম্যান্স অ্যাপ তৈরী করা যায়। নিজের ইচ্ছেমত যে কোন সুবিধা অ্যাপারে মধ্যে ব্যবহার করা যায়। আরো কিছু সুবিধা আছে, যা হলো-

- Dart এর বিশাল সংখ্যক সফটওয়্যার প্যাকেজ রিপোজিটরি আছে যা দিয়ে অ্যাপ এর কার্যক্ষমতা অনেক বাড়ানো যায়।
- ডেভেলপার একই কোড বিভিন্ন প্ল্যাটফর্মে ব্যবহার করতে পারে। যেমন, অ্যান্ড্রয়েড, আইওএস, ওয়েব ইত্যাদি।
- Flutter এ খুব অল্প টেস্টিং এর প্রয়োজন হয় কারন এটা একই কোড ব্যবহার করে।
- Flutter এর সহজবোধ্যতা খুব দ্রুত অ্যাপ ডেভেলপ করতে সাহায্য করে। আর অনেক বেশী কাস্টমাইজ করা যায়, আর সেটা Flutter কে আরো বেশী শক্তিশালী করে তোলে।
- Flutter ব্যবহারে widget এবং layout তৈরীতে ডেভেলপারদের পূর্ণ নিয়ন্ত্রণ থাকে।
- Flutter অসংখ্য গুরুত্বপূর্ণ Developer tools ব্যবহার করার সুবিধা দেয় এবং Hot Reload নামে Flutter এর অসাধারণ একটি ফিচার আছে যা দ্রুত সময়ে কোড রান করতে সাহায্য করে।

### **Flutter এর অসুবিধাঃ**

অনেক সুবিধার মধ্যে অল্প কিছু অসুবিধাও ফ্লাটারে আছে। সেগুলো হলো-

- যেহেতু Flutter এ Dart নামের নতুন একটি প্রোগ্রামিং ল্যাংগুয়েজ ব্যবহার হয় তাই যাদের আগে থেকেই অন্য কোন প্রোগ্রামিং ল্যাংগুয়েজ শেখা আছে তাদের নতুন করে Dart প্রোগ্রামিং ল্যাংগুয়েজ শিখতে হবে। তবে এটা অন্যান্য অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিংয়ের মতই এবং Dart শেখা খুবই সহজ।
- আধুনিক ফ্রেমওয়ার্কগুলো লজিক এবং ইউজার ইন্টারফেস আলাদা রাখার চেষ্টা করে কিন্তু Flutter-এ লজিক এবং ইউজার ইন্টারফেস একইসাথে কোড করা হয়। এই সমস্যা আমরা স্মার্ট কোডিং এবং হাই লেভেল মডিউল ব্যবহার করে লজিক এবং ইউজার ইন্টারফেস আলাদাভাবে কোড করে সমাধান করতে পারি।
- Flutter হলো আরেকটি নতুন মোবাইল অ্যাপ্লিকেশন ডেভেলপমেন্ট ফ্রেমওয়ার্ক। ডেভেলপাররা প্রায়ই সঠিক ডেভেলপমেন্ট প্ল্যাটফর্ম নির্ণয় করতে সিদ্ধান্তহীনতায় ভুগে।

### **উইন্ডোজ অপারেটিং সিস্টেমে কিভাবে Flutter ইনস্টল করব?**

এখন আমরা শিখব কিভাবে উইন্ডোজ অপারেটিং সিস্টেমে Flutter ইনস্টল করা হয়।

ধাপ-১ঃ-

শুরুতেই <https://docs.flutter.dev/get-started/install/windows> এই লিংক থেকে Flutter এর আপডেট ভার্সন ডাউনলোড করে নেব। এপ্রিল, ২০২৩-এ ফ্লাটারের সর্বশেষ ভার্সন হলো 3.7.10। এবং ডাউনলোড ফাইলের নাম flutter\_windows\_3.7.10-stable.zip। এই সময়ের পরবর্তীতে ফাইলের ভার্সন ও নাম পরিবর্তন হতে পারে।

ধাপ-২ঃ-

C ড্রাইভের ভিতর src নামে একটা ফোল্ডার তৈরী করি। এরপর ডাউনলোড করা Zip ফাইলটি src ফোল্ডারে Unzip করি। তাহলে Flutter এর লোকেশন হবে C:\src\flutter\ এমন।

ধাপ-৩ঃ

এখন আমাদের উইন্ডোজের System path আপডেট করতে হবে। এর জন্য উইন্ডোজের Start বাটনে ক্লিক করে env লিখে সার্চ করি। তাহলে Edit the system environment variables

Step 2 – Unzip the zip archive in a folder, say C:\flutter\

Step 3 – Update the system path to include flutter bin directory.

Step 4 – Flutter provides a tool, flutter doctor to check that all the requirement of flutter development is met.

flutter doctor

Step 5 – Running the above command will analyze the system and show its report as shown below –

Doctor summary (to see all details, run flutter doctor -v):

[√] Flutter (Channel stable, v1.2.1, on Microsoft Windows [Version

10.0.17134.706], locale en-US)

[√] Android toolchain - develop for Android devices (Android SDK version

28.0.3)

[√] Android Studio (version 3.2)

[√] VS Code, 64-bit edition (version 1.29.1)

[!] Connected device

! No devices available

! Doctor found issues in 1 category.

The report says that all development tools are available but the device is not connected. We can fix this by connecting an android device through USB or starting an android emulator.

Step 6 – Install the latest Android SDK, if reported by flutter doctor

Step 7 – Install the latest Android Studio, if reported by flutter doctor

Step 8 – Start an android emulator or connect a real android device to the system.

Step 9 – Install Flutter and Dart plugin for Android Studio. It provides startup template to create new Flutter application, an option to run and debug Flutter application in the Android studio itself, etc.,

- Open Android Studio.

- Click File → Settings → Plugins.
- Select the Flutter plugin and click Install.
- Click Yes when prompted to install the Dart plugin.
- Restart Android studio.

## Dart (ডার্ট)

Dart হলো Google এর ডেভেলপ করা একটি প্রোগ্রামিং ল্যাংগুয়েজ যা ওয়েব ব্রাউজার এবং মোবাইল ডিভাইস সহ একাধিক প্ল্যাটফর্মের অ্যাপ্লিকেশন তৈরী করতে ব্যবহার করা যায়। Flutter ফ্রেমওয়ার্কের প্রধান প্রোগ্রামিং ল্যাংগুয়েজ হিসেবে Dart ব্যবহার করা হয়।

## ভেরিয়েবল (Variable)

Dart প্রোগ্রামে তথ্য (Data) সংরক্ষণের জন্য ভেরিয়েবল ব্যবহার করা হয়। ভেরিয়েবল হচ্ছে একটা মেমোরী লোকেশন (Memory Location) এর নাম যেখানে একটা মান (Value) সংরক্ষিত থাকে। ভেরিয়েবলের মান আমরা প্রোগ্রাম রান করার সময় প্রয়োজনে পরিবর্তন করতে পারি।

### **ভেরিয়েবল ডিক্লেয়ার করাঃ**

Dart প্রোগ্রামে আমরা var কিওয়ার্ড (keyword) ব্যবহার করে ভেরিয়েব ডিক্লেয়ার করতে পারি। এক্ষেত্রে আমরা যেই ভেরিয়েবল ডিক্লেয়ার করব সেটা কি ধরনের ভেরিয়েবল (variable type) তা কম্পাইলার নিজে নিজে বুঝে নেয়। যেমনঃ

```
var name = 'John';
```

```
var age = 30;
```

উপরের উদাহরনে name এবং age দুইটি ভেরিয়েবল যাদের মান (value) দেয়া আছে যথাক্রমে 'John' এবং 30। ভেরিয়েব দুইটি ডিক্লেয়ার করার সময় আমরা var কিওয়ার্ড ব্যবহার করেছি। এক্ষেত্রে Dart নিজে থেকেই name কে String এবং age কে int হিসেবে ধরে নেবে।

আবার আমরা ভেরিয়েবল ডিক্লেয়ারের সময় নির্দিষ্টভাবে ভেরিয়েব টাইপ (variable type) বলে দিতে পারি। যেমনঃ

```
String name = 'John';
```

```
int age = 30;
```

উপরের উদাহরণে name এবং age ভেরিয়েবল দুটি ডিক্লেয়ার করার সময় এদের টাইপ স্পষ্টভাবে বলে দেয়া আছে।

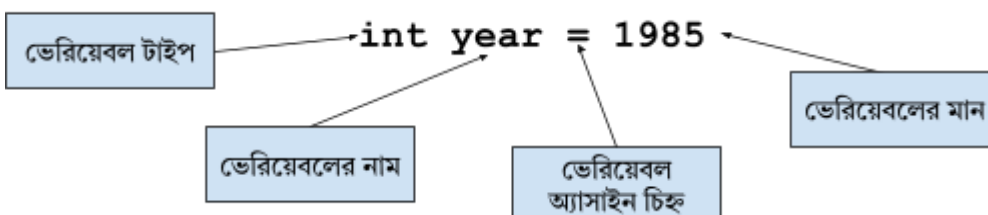
### ভেরিয়েবল টাইপঃ

Dart এ মূলত চারটি প্রাথমিক ভেরিয়েবল টাইপ রয়েছে। এগুলো হলো, int, double, String, bool. পূর্ণসংখ্যার জন্য int ব্যবহার করা হয়। যেমনঃ int age = 21। দশমিক সংখ্যার জন্য double ব্যবহার করা হয়। যেমনঃ double price = 80.99। কোন কিছুর নাম যা অক্ষর ব্যবহার করে লিখা হয় তার জন্য String ব্যবহার করা হয়। যেমনঃ String name = 'Kamal Hossain'। সত্য অথবা মিথ্যা এমন মানের ক্ষেত্রে bool ব্যবহার করা হয়। যেমনঃ bool isFriday = true।

//List, Map

### ভেরিয়েবল ডিক্লেয়ার করার নিয়মঃ

ভেরিয়েবল ডিক্লেয়ার করতে প্রথমে আমাদের লিখতে হবে ভেরিয়েব টাইপ, তারপর ভেরিয়েবলের নাম, তারপর সমান চিহ্ন, এরপর ভেরিয়েবলের মান। নিচের উদাহরন লক্ষ্য করিঃ





## ভেরিয়েবল নাম লেখার নিয়মঃ (Variable Naming)

Dart প্রোগ্রামিংয়ের ভেরিয়েবল নামের জন্য আমরা ইংরেজী লেটার, নাম্বার এবং আন্ডারস্কোর ব্যবহার করতে পারি। কিন্তু ভেরিয়েবল নামের শুরু অবশ্যই ইংরেজী লেটার অথবা আন্ডারস্কোর দিয়ে করতে হবে। নাম্বার দিয়ে ভেরিয়েবলের নাম লিখা শুরু করা যাবে না। ভেরিয়েবলের নামের মধ্যে স্পেস অথবা স্পেশাল ক্যারেক্টার (@#\$\$%^&\* ইত্যাদি) ব্যবহার করা যাবে না। নিচে ভেরিয়েব নামের কিছু সঠিক এবং ভুল উদাহরণ দেয়া হলোঃ

birthYear	সঠিক
number10	সঠিক
5sky	ভুল
_firstValue	সঠিক
School Name	ভুল
today\$	ভুল

## ভেরিয়েবল নাম জায়গাঃ (Variable Scope)

Variable Scope হলো যেখান থেকে আমরা ভেরিয়েবল অ্যাক্সেস করতে পারি বা ব্যবহার করতে পারি। Dart প্রোগ্রামিংয়ে ফাইলের শুরুতে (টপ লেভেল), ফাংশনের ভিতরে অথবা ক্লাসের ভিতরে ভেরিয়েবল ডিক্লেয়ার করা যায়। ফাংশন এবং ক্লাসের ভিতরে ভেরিয়েব ডিক্লেয়ার করলে শুধু ফাংশন এবং ক্লাসের ভিতর থেকেই ভেরিয়েবল অ্যাক্সেস করা যাবে।

উদাহরণঃ

```
// declaring variables at top-level
var name = 'John';
int age = 30;
void main() {
    // declaring variables within a function
    var message = 'Hello, Dart!';
```

```
print(name); // output: John
print(age); // output: 30
print(message); // output: Hello, Dart!
}output: Hello, Dart! }
```

উপরের উদাহরণে টপল লেভেল ভেরিয়েবল ডিক্লেয়ার করা হয়েছে var এবং int কিওয়ার্ড ব্যবহার করে। main ফাংশনেও var কিওয়ার্ড ব্যবহার করা হয়েছে। name এবং age ভেরিয়েবল দুইটি main ফাংশন থেকে অ্যাক্সেস করা যাবে, কারন ভেরিয়েব দুইটি টপ লেভেলে ডিক্লেয়ার করা হয়েছে। messege ভেরিয়েবলটি শুধুমাত্র main ফাংশন থেকে অ্যাক্সেস করা যাবে। কার এটি main ফাংশনে ডিক্লেয়ার করা হয়েছে।

### সাংরাংশঃ

Dart সহ সকল প্রোগ্রামিং ল্যাংগুয়েজের জন্য Variable অত্যাবশ্যকীয় অংশ। Dart ল্যাংগুয়েজে সাধারণত var কিওয়ার্ড ব্যবহার করা হয় যখন ভেরিয়েবল টাইপ অনুমান করা যায় কিন্তু স্পষ্ট নয়। ভেরিয়েবল ব্যবহারে দক্ষ হতে উদাহরণের প্রোগ্রামটি প্রাকটিস করুন।

## Dart List (ডার্ট লিস্ট)

**Dart** ল্যাংগুয়েজে একাধিক **Data** একটা জায়গায় সংরক্ষণ করতে **List** ব্যবহার করা হয়। যেমন ধরি, একটা বক্সে ২০ টি আম আছে। এখানে আমরা যদি আমগুলোকে Data ধরি তাহলে বক্সটি হলো List। List এর ভিতর যে কোন ধরনের Data রাখা যায়। যেমন, **number, String, Object** ইত্যাদি। **List** এর সাইজ ডাইনামিক। অর্থাৎ List এর আইটেম সংখ্যা দরকার অনুযায়ী বাড়ানো, কমানো কিংবা Modify করা যায়।

### List কিভাবে ব্যবহার করবো?:

একটি খালি লিস্ট এভাবে তৈরী করা যায়ঃ

```
List<DataType> listName = [];
```

### উদাহরণঃ

```
List<int> numbers = [];
```

Value সহ List তৈরী করা যায় এভাবেঃ

```
List<DataType> listName = [value1, value2, ...];
```

### For example:

```
List<String> names = ['Alice', 'Bob', 'Charlie'];
```

### List কিভাবে Access করা যায়?

List এর আইটেমে ক্রম অনুসারে List এর আইটেম Access করা যায়।

মনে রাখতে হবে List এর আইটেমের ক্রম সংখ্যা 0 থেকে শুরু হয়।

### যেমনঃ

```
List<int> numbers = [10, 20, 30];
```

### উপরের List এর দ্বিতীয় আইটেম(20) এভাবে Access করা যায়:

```
var secondNumber = numbers[1];
```

### Lists মডিফাই করাঃ

List মডিফাই বলতে লিস্টের মধ্যে যেকোন ধরনের পরিবর্তনকে বুঝায়। এটা হতে পারে লিস্টের ভিতর নতুন আইটেম যোগ করা, কোন আইটেম ডিলিট করা।

লিস্টের ভিতর নতুন আইটেম যোগ করতে add() মেথড ব্যবহার করতে হয়। এই মেথড ব্যবহার করলে নতুন আইটেমটি লিস্টের একদম শেষে যোগ হবে।

লিস্টের ভিতর নির্দিষ্ট জায়গায় আইটেম যোগ করতে insert() মেথড ব্যবহার করা হয়।

লিস্ট থেকে কোন আইটেম মুছে ফেলতে remove() মেথড ব্যবহার করা হয়।

যেমনঃ

```
List<String> names = ['Alice', 'Bob'];
```

এই লিস্টের শেষে যদি 'Charlie' এই আইটেমটি যোগ করতে চাই তাহলে লিখতে হবে-

```
names.add('Charlie');
```

এখন যদি 'Dave' এই আইটেমটি ২য় স্থানে যোগ করতে চাই তাহলে লিখতে হবে-

```
names.insert(1, 'Dave');
```

'Bob' এই আইটেমটি যদি লিস্ট থেকে মুছে ফেলতে চাই-

```
names.remove('Bob');
```

### List এর Operation এবং Manipulation:

লিস্টের মধ্যে বিভিন্ন **Operation** করা যায়:

একটা List আরেকটা List এর সাথে যোগ করা যায় + (প্লাস) অপারেটর ব্যবহার করে।

List এর সাইজ অর্থাৎ লিস্টের মধ্যে কতগুলো আইটেম আছে সেটা জানতে **length** প্রোপার্টি ব্যবহার করা হয়।

List এর একটা অংশ নিয়ে নতুন লিস্ট তৈরী করতে **sublist()** মেথড ব্যবহার করা হয়।

যেমনঃ

```
List<int> list1 = [1, 2, 3];
```

```
List<int> list2 = [4, 5, 6];
```

উপরের দুইটা List একসাথে যোগ করতে (Concatenate) -

```
var combinedList = list1 + list2;
```

লিস্টের সাইজ জানতে -

```
var size = names.length;
```

List এর একটা অংশ নিয়ে নতুন লিস্ট তৈরী করতে:

```
var sublist = names.sublist(1, 3);
```

List iteration (লিস্ট ইটারেশন):

লিস্ট ইটারেশন বলতে লিস্টের সব আইটেমগুলো একসাথে একটার পর একটা নিয়ে কাজ করা বুঝায়।

বিভিন্ন পদ্ধতিতে লিস্ট ইটারেশন করা যায়। যেমনঃ for লুপ, for-in লুপ, forEach() মেথড।

উদাহরণঃ

```
List<String> names = ['Alice', 'Bob', 'Charlie'];
```

for লুপ ব্যবহার করে :

```
for (var i = 0; i < names.length; i++)  
{  
    // names[i] তম আইটেম নিয়ে যে কাজ করতে চাই।  
}
```

For-in লুপ ব্যবহার করে :

```
for (var name in names)  
{ // name আইটেমটি নিয়ে যে কাজ করতে চাই }
```

forEach() মেথড ব্যবহার করেঃ

```
names.forEach((name)
```

```
{
```

```
// name আইটেমটি নিয়ে যে কাজ করতে চাই  
  
});
```

### List Searching এবং Sorting:

আমরা লিস্টের ভিতর যে কোন নির্দিষ্ট আইটেম সার্চ করতে পারি এবং বিভিন্নভাবে সাজাতে পারিঃ

একটা আইটেম লিস্টের ভিতর আছে কি নেই সেটা চেক করতে contains() মেথড ব্যবহার করতে হয়।

একটা আইটেম কোন কততম জায়গায় আছে সেটা জানতে indexOf() মেথড ব্যবহার করা হয়।

যেমন:

```
List<String> names = ['Alice', 'Bob', 'Charlie'];
```

এই লিস্টে 'Bob' আইটেমটি আছে কি না সেটা জানতে-

```
var exists = names.contains('Bob');
```

'Charlie' আইটেমটি কততম পজিশনে (index) আছে সেটা জানতে-

```
var index = names.indexOf('Charlie');
```

## Firestore Database (ফায়ারবেইস ফায়ারস্টোর ডাটাবেইস)

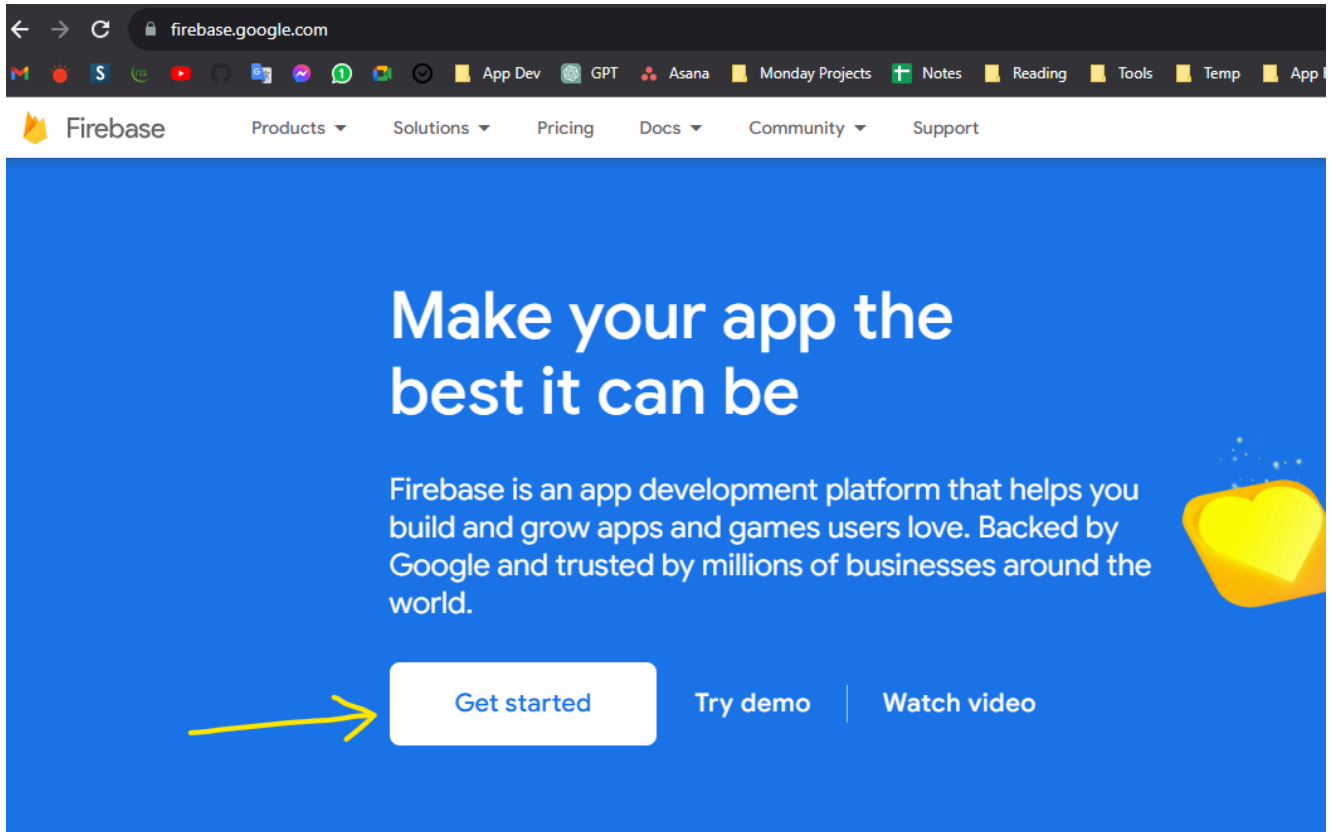
### 1. Flutter প্রজেক্ট তৈরী:

Firestore ডাটাবেইস ব্যবহার করতে শুরুতে আমাদের Flutter প্রজেক্ট তৈরী করতে হবে।  
যদি আগে থেকেই প্রজেক্ট তৈরী থাকে তাহলে নতুন করে প্রজেক্ট তৈরী করার প্রয়োজন নেই।

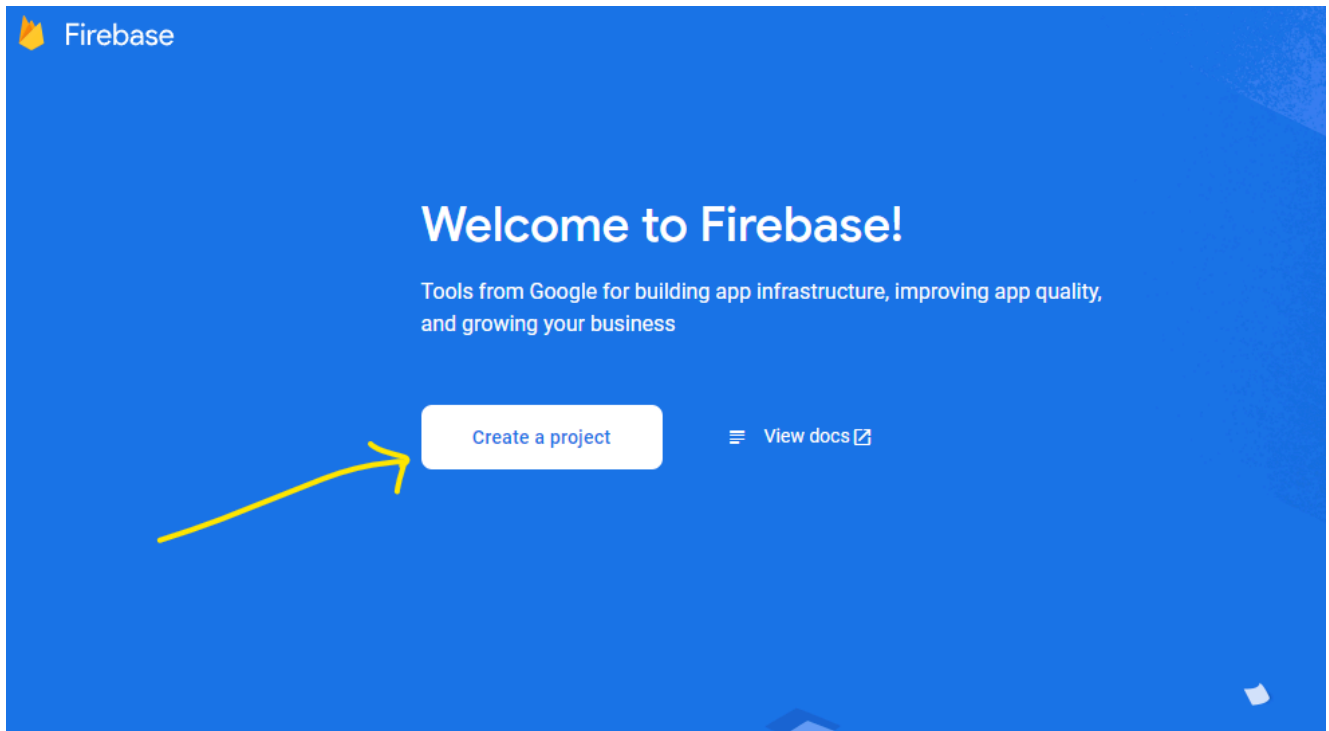
### 2. Firebase অ্যাকাউন্ট তৈরী:

ফায়ারবেইস ডাটাবেইস ব্যবহার করতে আমার ফায়ারবেইস এর ওয়েবসাইটে একাউন্ট করে নিতে হবে।  
যদি আগে থেকে অ্যাকাউন্ট থাকে তাহলে নতুন করে একাউন্ট করার প্রয়োজন নেই।

শুরুতে <https://firebase.google.com/> এই ওয়েবসাইটে যাবো। যেহেতু ফায়ারবেইস গুগলের একটি সার্ভিস, তাই গুগল বা জিমেইল একাউন্ট থাকলে সেটাই আমরা ফায়ারবেইস অ্যাকাউন্ট হিসেবে ব্যবহার করতে পারবো।



উপরের লিংকে গেলে এমন একটি পেজ দেখতে পাবো। চিহ্নিত বাটনে (Get started) ক্লিক করলে  
নিচের মত একটি পেজ আসবে।



এখান থেকে চিহ্নিত বাটনে ক্লিক করবো। তাহলে নিচের মত একটি পেজ আসবে।

✕ Create a project (Step 1 of 3)

Let's start with a name for your project <sup>?</sup>

Enter your project name

my-awesome-project-id

☐ I accept the [Firebase terms](#)

☐ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Continue


এখানে আমরা আমাদের প্রজেক্ট এর নাম দিয়ে Continue বাটনে ক্লিক করবো।



## Let's start with a name for your project <sup>?</sup>

Project name

MyApp

 myapp-e7bca

☒ I accept the [Firebase terms](#) <sup>?</sup>

☒ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Continue

## Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

✕ A/B testing <sup>?</sup>

✕ Crash-free users <sup>?</sup>

✕ User segmentation & targeting across Firebase products <sup>?</sup>

✕ Event-based Cloud Functions triggers <sup>?</sup>

✕ Free unlimited reporting <sup>?</sup>



Enable Google Analytics for this project  
Recommended

Previous

Create project

পরের পেজে উপরের চিহ্নিত বাটন

3. Setup Firebase Project
4. Create Firestore DB
5. Create Collection
6. Add Data to Collection
7. Install firestore Packages in Flutter Project
8. Initialize Firebase in Main Function
- 9.

## **Dart Programming Problems**

### **Basic Level Problems**

1. Write a Dart program that prints "Hello, world!" to the console.
2. Write a Dart program that takes two numbers as input and prints their sum.
3. Write a Dart program that takes a name as input and prints a greeting message to the console.
4. Write a Dart program that takes a string as input and prints its length.
5. Write a Dart program that takes a list of numbers as input and prints their sum.
6. Write a Dart program that takes a list of strings as input and prints the longest string.
7. Write a Dart program that takes a string as input and prints its reverse.
8. Write a Dart program that takes a list of integers as input and prints the largest and smallest numbers.
9. Write a Dart program that takes a list of numbers as input and prints the average.

10. Write a Dart program that takes a string as input and counts the number of vowels in the string.

### **Intermediate Level Problems**

11. Write a Dart program that checks if a given number is prime or not.
12. Write a Dart program that takes a list of numbers as input and checks if it is sorted in ascending order.
13. Write a Dart program that takes a list of integers as input and prints the second largest number.
14. Write a Dart program that takes a list of integers as input and checks if it contains duplicates.
15. Write a Dart program that takes a list of strings as input and checks if all the strings have the same length.
16. Write a Dart program that takes a list of integers as input and removes all the even numbers.
17. Write a Dart program that takes two lists of integers as input and returns a new list containing the common elements.
18. Write a Dart program that takes a string as input and checks if it is a palindrome.
19. Write a Dart program that takes a list of integers as input and returns a new list containing the unique elements.
20. Write a Dart program that takes a list of strings as input and returns a new list containing the strings in alphabetical order.
21. Write a Dart program that takes a list of integers as input and returns the maximum difference between any two elements in the list.
22. Write a Dart program that takes a list of integers as input and returns the maximum product that can be obtained by selecting a non-empty subsequence of the list.

23. Write a Dart program that takes a list of integers as input and returns the length of the longest consecutive subsequence of the list.
24. Write a Dart program that takes a list of integers as input and returns the k-th largest element in the list.
25. Write a Dart program that takes a list of integers as input and returns the frequency of each element in the list.
26. Write a Dart program that takes a list of integers as input and returns the median of the list.
27. Write a Dart program that takes a list of integers as input and returns the mode of the list.
28. Write a Dart program that takes a list of integers as input and returns the standard deviation of the list.
29. Write a Dart program that takes a list of integers as input and returns the variance of the list.
30. Write a Dart program that takes a list of integers as input and returns the range of the list.

### **Advanced Level Problems**

31. Write a Dart program that simulates a simple calculator that can perform addition, subtraction, multiplication, and division.
32. Write a Dart program that takes a list of integers as input and finds the two elements that add up to a given target number.
33. Write a Dart program that takes a list of strings as input and groups them by their length.
34. Write a Dart program that takes a list of integers as input and returns the maximum sum that can be obtained by selecting a non-empty subsequence of the list.

35. Write a Dart program that takes a list of integers as input and returns the longest increasing subsequence of the list.
36. Write a Dart program that takes a list of integers as input and returns the longest common subsequence of the list.
37. Write a Dart program that takes a string as input and finds the longest repeated substring in the string.
38. Write a Dart program that takes a list of integers as input and sorts it using the quicksort algorithm.
39. Write a Dart program that takes a list of integers as input and sorts it using the mergesort algorithm.
40. Write a Dart program that takes a list of integers as input and sorts it using the heapsort algorithm.
41. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with length less than or equal to k.
42. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no adjacent elements.
43. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no three adjacent elements.
44. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no four adjacent elements.
45. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no five adjacent elements.
46. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no six adjacent elements.
47. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no seven adjacent elements.
48. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no eight adjacent elements.

49. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no more than nine adjacent elements.
50. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no more than ten adjacent elements.
51. Write a Dart program that takes a list of integers as input and returns the maximum number of non-overlapping intervals that can be selected from the list, where each interval has a maximum size of  $k$ .
52. Write a Dart program that takes a list of integers as input and returns the minimum number of moves needed to move all the elements to the same position, where each move consists of selecting a pair of adjacent elements and swapping them.
53. Write a Dart program that takes a list of integers as input and returns the minimum number of moves needed to move all the elements to the same position, where each move consists of selecting a pair of elements and swapping them.
54. Write a Dart program that takes a list of integers as input and returns the length of the longest common subsequence of three or more lists.
55. Write a Dart program that takes a list of integers as input and returns the maximum length of a contiguous subarray that appears in at least two of the lists.
56. Write a Dart program that takes a list of integers as input and returns the minimum number of moves needed to move all the elements to the same position, where each move consists of selecting a pair of adjacent elements and swapping them, and the cost of each move is proportional to the absolute difference between the two elements being swapped.
57. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no consecutive elements, where the elements are in a circular arrangement.

58. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no three consecutive elements, where the elements are in a circular arrangement.
59. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no four consecutive elements, where the elements are in a circular arrangement.
60. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray within the list. A subarray is defined as a contiguous subset of the list.
61. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no five consecutive elements, where the elements are in a circular arrangement.
62. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no six consecutive elements, where the elements are in a circular arrangement.
63. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no seven consecutive elements, where the elements are in a circular arrangement.
64. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no eight consecutive elements, where the elements are in a circular arrangement.
65. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no nine consecutive elements, where the elements are in a circular arrangement.
66. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no ten consecutive elements, where the elements are in a circular arrangement.

67. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no eleven consecutive elements, where the elements are in a circular arrangement.
68. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no twelve consecutive elements, where the elements are in a circular arrangement.
69. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no thirteen consecutive elements, where the elements are in a circular arrangement.
70. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no fourteen consecutive elements, where the elements are in a circular arrangement.
71. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no fifteen consecutive elements, where the elements are in a circular arrangement.
72. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no sixteen consecutive elements, where the elements are in a circular arrangement.
73. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no seventeen consecutive elements, where the elements are in a circular arrangement.
74. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no eighteen consecutive elements, where the elements are in a circular arrangement.
75. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no nineteen consecutive elements, where the elements are in a circular arrangement.



76. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no twenty consecutive elements, where the elements are in a circular arrangement.
77. Write a Dart program that takes a list of integers as input and returns the length of the longest palindromic subsequence of the list.
78. Write a Dart program that takes a list of integers as input and returns the maximum number of non-overlapping intervals that can be selected from the list, where each interval has a maximum size of  $k$  and no two intervals overlap by more than  $d$ .
79. Write a Dart program that takes a list of integers as input and returns the maximum length of a contiguous subarray that appears in at least three of the lists.
80. Write a Dart program that takes a list of integers as input and returns the minimum number of moves needed to move all the elements.

### **Expert Level Problems**

81. Write a Dart program that implements a binary search algorithm for a sorted list of integers.
82. Write a Dart program that implements a breadth-first search algorithm for a graph represented as an adjacency list.
83. Write a Dart program that implements a depth-first search algorithm for a graph represented as an adjacency matrix.

84. Write a Dart program that implements Dijkstra's algorithm for finding the shortest path in a weighted graph. The program should take as input the number of vertices and edges in the graph, as well as the starting and ending vertices. The program should output the shortest path between the two vertices.
85. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no adjacent elements, where the elements are in a circular arrangement.
86. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no three adjacent elements, where the elements are in a circular arrangement.
87. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no four adjacent elements, where the elements are in a circular arrangement.
88. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no five adjacent elements, where the elements are in a circular arrangement.
89. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no six adjacent elements, where the elements are in a circular arrangement.
90. Write a Dart program to implement a deep copy function for a binary tree data structure. The program should take as input a binary tree and should output a new binary tree that is a deep copy of the original tree, meaning that all of the nodes in the new tree should be distinct from the nodes in the original tree.
91. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no seven adjacent elements, where the elements are in a circular arrangement.

92. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no eight adjacent elements, where the elements are in a circular arrangement.
93. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no nine adjacent elements, where the elements are in a circular arrangement.
94. Write a Dart program that takes a list of integers as input and returns the maximum sum of any subarray with no ten adjacent elements, where the elements are in a circular arrangement.
95. Write a Dart program that takes a string as input and returns the length of the longest palindromic substring.
96. Write a Dart program that takes a string as input and returns the minimum number of cuts needed to partition the string into a set of palindromic substrings.
97. Write a Dart program that takes two strings as input and returns the longest common subsequence of the two strings.
98. Write a Dart program that takes two strings as input and returns the minimum number of operations needed to convert one string to the other, where the operations include insertion, deletion, and substitution of a single character.
99. Write a Dart program that takes a list of integers as input and returns the length of the longest increasing subsequence of the list.
100. Write a Dart program that takes a list of integers as input and returns the length of the longest decreasing subsequence of the list.