在Dart中，你可以导入一个库来使用它所提供的功能。

库的使用可以使代码的重用性得到提高，并且可以更好的组合代码。

Dart中任何一个dart文件都是一个库，即使你没有用关键字library声明

**1.1. 库的导入**

import语句用来导入一个库，后面跟一个字符串形式的Uri来指定表示要引用的库，语法如下：

import '库所在的uri';

**常见的库URI有三种不同的形式**

* 来自dart标准版，比如dart:io、dart:html、dart:math、dart:core(但是这个可以省略)

//dart:前缀表示Dart的标准库，如dart:io、dart:html、dart:math

import 'dart:io';

* 使用相对路径导入的库，通常指自己项目中定义的其他dart文件

//当然，你也可以用相对路径或绝对路径的dart文件来引用

import 'lib/student/student.dart';

* Pub包管理工具管理的一些库，包括自己的配置以及一些第三方的库，通常使用前缀package

//Pub包管理系统中有很多功能强大、实用的库，可以使用前缀 package:

import 'package:flutter/material.dart';

**库文件中内容的显示和隐藏**

如果希望只导入库中某些内容，或者刻意隐藏库里面某些内容，可以使用show和hide关键字

* **show关键字：**可以显示某个成员（屏蔽其他）
* **hide关键字：**可以隐藏某个成员（显示其他）

import 'lib/student/student.dart' show Student, Person;

import 'lib/student/student.dart' hide Person;

**库中内容和当前文件中的名字冲突**

当各个库有命名冲突的时候，可以使用as关键字来使用命名空间

import 'lib/student/student.dart' as Stu;

Stu.Student s = new Stu.Student();

**1.2. 库的定义**

**library关键字**

通常在定义库时，我们可以使用library关键字给库起一个名字。

但目前我发现，库的名字并不影响导入，因为import语句用的是字符串URI

library math;

**part关键字**

在之前我们使用student.dart作为演练的时候，只是将该文件作为一个库。

在开发中，如果一个库文件太大，将所有内容保存到一个文件夹是不太合理的，我们有可能希望将这个库进行拆分，这个时候就可以使用part关键字了

mathUtils.dart文件

part of "utils.dart";

int sum(int num1, int num2) {

return num1 + num2;

}

dateUtils.dart文件

part of "utils.dart";

String dateFormat(DateTime date) {

return "2020-12-12";

}

utils.dart文件

part "mathUtils.dart";

part "dateUtils.dart";

test\_libary.dart文件

import "lib/utils.dart";

main(List<String> args) {

print(sum(10, 20));

print(dateFormat(DateTime.now()));

}

**export关键字**

官方不推荐使用part关键字，那如果库非常大，如何进行管理呢？

* 将每一个dart文件作为库文件，使用export关键字在某个库文件中单独导入

mathUtils.dart文件

int sum(int num1, int num2) {

return num1 + num2;

}

dateUtils.dart文件

String dateFormat(DateTime date) {

return "2020-12-12";

}

utils.dart文件

library utils;

export "mathUtils.dart";

export "dateUtils.dart";

test\_libary.dart文件

import "lib/utils.dart";

main(List<String> args) {

print(sum(10, 20));

print(dateFormat(DateTime.now()));

}