**Dart code test**

Introduction to this course [00:44](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=44s) - Dartpad.dev & mathematical [05:20](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=320s) - Variables [14:21](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=861s) - const & final [18:48](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=1128s) - Choose good names! [20:53](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=1253s) - Let's install dart and vscode [28:52](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=1732s) - dynamic type [34:42](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2082s) - Type conversion [36:28](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2188s) - Casting [38:26](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2306s) - String [40:52](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2452s) - String concatenation [41:46](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2506s) - String interpolation [43:36](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2616s) - Multiline string [46:18](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=2778s) - Let's review what we learn (1) [57:00](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=3420s) - Practice (1) [57:28](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=3448s) - bool & Relational operators [01:13:45](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=4425s) - Control flow - if [01:21:36](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=4896s) - Let's review what we learn (2) [1:24:54](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=5094s) - Conditional operator [1:28:08](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=5288s) - switch statement [1:33:05](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=5585s) - Dart 3 & switch statement! [1:36:41](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=5801s) - Example with switch statement [1:39:35](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=5975s) - Enums [1:44:28](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=6268s) - Practice (2) [1:44:34](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=6274s) - Loops! : while & do while [1:55:00](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=6900s) - Loops! : for [02:02:31](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=7351s) - Let's review what we learn (3) [02:07:29](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=7649s) - Functions [02:29:18](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=8958s) - Classes [02:37:50](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=9470s) - Encapsulation [02:51:09](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=10269s) - Null safety [03:06:49](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=11209s) - Constructor [03:15:42](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=11742s) - Named constructors [03:18:48](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=11928s) - Constructor forwarding [03:20:34](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=12034s) - final & const again! [03:27:05](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=12425s) - Practice (3) [03:27:10](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=12430s) - late keyword [03:31:27](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=12687s) - static keyword [03:35:42](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=12942s) - Lists [03:44:50](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=13490s) - for each in iterable [03:46:36](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=13596s) - Sets [03:51:23](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=13883s) - Maps [03:56:39](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=14199s) - Practice (4) [03:56:44](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=14204s) - Inheritance [04:09:05](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=14945s) - is keyword [04:10:10](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=15010s) - Abstract classes [04:16:00](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=15360s) - Interfaces [04:23:00](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=15780s) - Mixins [04:26:45](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=16005s) - Libraries & Packages [04:35:38](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=16538s) - What is an API? [04:42:41](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=16961s) - Example with API [04:51:25](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=17485s) - Futures [04:56:22](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=17782s) - Functions as a parameter! [05:04:40](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=18280s) - Futures Explained [05:08:51](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=18531s) - async & await keyword [05:14:03](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=18843s) - Errors & Exceptions [05:20:56](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=19256s) - Practice (5) [05:21:00](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=19260s) - Let's Start Social Media Project !!! [05:33:46](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=20026s) - Get Users Input! [05:39:24](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=20364s) - Login & Signup [05:44:57](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=20697s) - Factory constructor [05:49:53](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=20993s) - Inheritance in our project [05:52:35](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=21155s) - Exceptions handling in the project [05:54:26](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=21266s) - Add menu to project [06:16:22](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=22582s) - Streams [06:21:05](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=22865s) - Add Chatroom! [06:55:53](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=24953s) - Add Styles to the project! [07:04:39](https://www.youtube.com/watch?v=c-ydMfJNfyY&t=25479s) - Dart in Server! (dart\_frog)

void main() {

print(5%2);

}

//1

Variable

01

void main() {

int myValue = 1;

print(myValue);

}

02

void main() {

int myValue = 1;

double mydouble =1.25;

print(myValue);

print(mydouble);

} // 1 1.25

03

void main() {

var noType=1;

print(noType);

}//1

04

void main() {

int num1=5;

int num2=8;

print(num1 + num2);

}

05

void main() {

int num1=5;

int num2=8;

int module1 = 5%8;

print(num1 + num2);

print(num1 - num2);

print(num1 / num2);

print(num1 ~/ num2);

print(num1 \* num2);

print(num1 % num2);

print(module1);

}

13

-3

0.625

0

40

5

5

Constants ; const & final

Const working compile time

01

void main() {

final int age=20;

age =22;

}

The final variable 'age' can only be set once.

02

void main() {

final int age=20;

int age1=30;

age1=31;

const name ='sumith';

// const

var serverUrl= 'https://google.com';

serverUrl ='';

final date =DateTime.now();

// date =DateTime.now();

// const cant re assign value

// final use cant re assign value

}

Choose good variable name

void main() {

int studentScore= 20;

}// use lover camalcase

Java script is dynamic type language

void main() {

  dynamic age = 20;

  age = 22;

}

Chack type of vaiable

void main() {

  var age;

  age = 20;

  age = "hello";

  print(age is! int);

}

False/true

void main() {

  double age;

  age = 20;

  int age2 = age.toInt();

  print(age2);

}

void main() {

num age;

  age = 20;

  int age2 = age as int;

  print(age2);

}

void main() {

  num age =20;

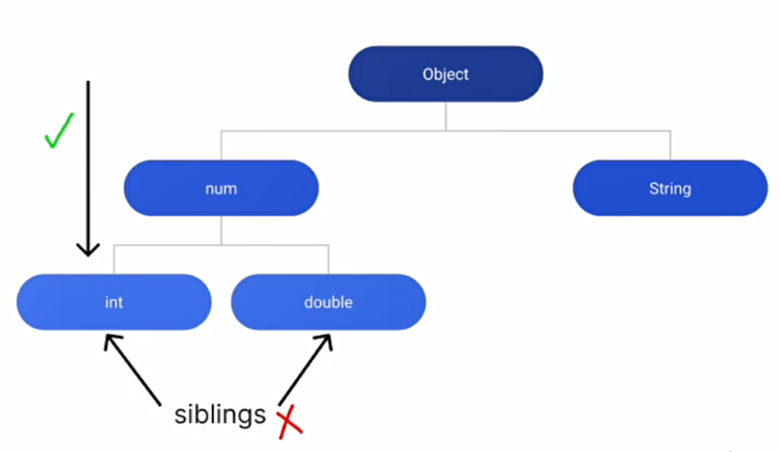
  var  age2 = age as double;

  print(age2);

}

Unhandled exception:

type 'int' is not a subtype of type 'double' in type cast



String

void main() {

  String a = 'a';

  print(a.codeUnitAt(0) );

}

97

Character number

void main() {

  String a = 'abcd';

  print(a.codeUnits);

}

[97, 98, 99, 100]

String concatenation

void main() {

  var name = 'magi';

  var lastName = 'sumith';

  print(name+ ' ' + lastName);

}

void main() {

  var name = 'magi';

  var lastName = 'sumith';

  String nameAndlastName = name + ' ' + lastName;

  print(nameAndlastName);

}

void main() {

  var name = 'magi';

  var lastName = 'sumith';

  //String nameAndlastName = name + ' ' + lastName;

  print('$name $lastName');

}

void main() {

  var name = 'magi';

  var lastName = 'sumith';

  //String nameAndlastName = name + ' ' + lastName;

  print('$name $lastName and age is ${20}');

}

void main() {

  var name = 'magi';

  var lastName = 'sumith';

  //String nameAndlastName = name + ' ' + lastName;

  print('$name $lastName and age is ${(2+2)}');

}

void main() {

  var multiLineString = 'Hello\n My Name Is sumith \nand im 25  \n:)';

   print(multiLineString);

}

Use \n

void main() {

  var multiLineString = 'hhdhdhdhdhdh'

      +' '+'dgddhdhdhdhdhdh';

  print(multiLineString);

}

+’’+

void main() {

  const minutsCount = 60;

  var userName = 'sumith indika';

  var followerCounter = 100;

  var followingCounter = 200;

  var watchHour = 30.7;

  var bio = '''i m sumith indika

   my age is 20 ..)''';

  var hours = watchHour.toInt();

  var minutes = watchHour - hours;

  minutes = minutes \* minutsCount;

  print('My UserName is $userName');

  print('my follower : is $followerCounter count');

  print('my following : is $followingCounter count');

  print('my watc Time is  : $hours h : ${minutes.toInt()} min');

  print('my bio  is : $bio');

}

void main() {

  const minutsCount = 60;

  const mCount = 10000;

  var userName = 'sumith indika';

  var followerCounter = 11900;

  //var followingCounter = 200;

  var watchHour = 30.7;

  var bio = '''i m sumith indika

   my age is 20 ..)''';

  var hours = watchHour.toInt();

  var minutes = watchHour - hours;

  minutes = minutes \* minutsCount;

  //var thousend = followerCounter / 1000;

  // kk

  var thousend = followerCounter.toInt();

  var thk = thousend / 1000;

  var kValue = thk.toInt();

//k rest

  var getRestvalue = followerCounter % 1000;

  // mm

  var mValue = followerCounter / mCount;

// 0M 1K 500

  print('My UserName is $userName');

  print(

      'my followers : is ${mValue.toInt()}M : $kValue K : $getRestvalue counts');

  //print('my following : is $followingCounter count');

  print('my watc Time is  : $hours h : ${minutes.toInt()} min');

  print('my bio  is : $bio');

}

Boolean

void main() {

  var MyFollowerCoun = 100;

  var newFollowerCount = 99;

  newFollowerCount++;

   //newFollowerCount = newFollowerCount + 1;

  var result = !(MyFollowerCoun == newFollowerCount);

  print(result);

  print(newFollowerCount);

}

void main() {

  var MyFollowerCoun = 200;

  var newFollowerCount = 99;

  newFollowerCount++;

   //newFollowerCount = newFollowerCount + 1;

  var result = (MyFollowerCoun > newFollowerCount);

  print(result);

  print(newFollowerCount);

}

void main() {

  var MyFollowerCoun = 100;

  var newFollowerCount = 100;

  var myfollowerlikes = 10;

  var newfollowerlikes = 9;

 // newFollowerCount++;

  //newFollowerCount = newFollowerCount + 1;

  var result = !(MyFollowerCoun == newFollowerCount);

  print(result);

  //print(newFollowerCount);

}

..

void main() {

  var MyFollowerCoun = 100;

  var newFollowerCount = 100;

  var myfollowerlikesCount = 10;

  var newfollowerlikesCount = 9;

  // newFollowerCount++;

  //newFollowerCount = newFollowerCount + 1;

  var result = (MyFollowerCoun > newFollowerCount &&

      myfollowerlikesCount > newfollowerlikesCount && 2 == 2);

  print(result);

  //print(newFollowerCount);

}

void main() {

  var MyFollowerCoun = 100;

  var newFollowerCount = 100;

  var myfollowerlikesCount = 10;

  var newfollowerlikesCount = 9;

  // newFollowerCount++;

  //newFollowerCount = newFollowerCount + 1;

  var result = (MyFollowerCoun > newFollowerCount ||

      myfollowerlikesCount > newfollowerlikesCount || 2 == 2);

  print(result);

  //print(newFollowerCount);

}

|| or akak hari

void main() {

  var MyFollowerCoun = 100;

  var newFollowerCount = 100;

  var myfollowerlikesCount = 10;

  var newfollowerlikesCount = 9;

  // newFollowerCount++;

  //newFollowerCount = newFollowerCount + 1;

// false&& true  || true

  var result = (MyFollowerCoun > newFollowerCount &&

     ( myfollowerlikesCount > newfollowerlikesCount || 2 == 2));

  print(result);

  // use () change pracident

  //print(newFollowerCount);

}

If else statement

void main() {

  int likes = 99;

  String uName = 'sumith';

  if (likes >= 100) {

    print("good engagement ");

  } else {

    print("Not good engagement ");

  }

}

void main() {

  int likes = 99;

  String uName = 'jhone';

  if (uName == 'sumith') {

    print("yes you are sumith");

  } else {

    print("Not your sumith ");

  }

}

void main() {

  var likes = 500;

  if (likes >= 1000) {

    print("very nice engagement ");

  }else if(likes < 1000 && likes >= 700){

    print("good engagement ");

  }else if(likes < 700 && likes >=300){

 print("ok engagement ");

  }else {

     print("bad  engagement ");

  }

}

void main() {

  var like = 99;

  String message =(like >= 100) ?

  "GOOD engagement": "bad engagement" ;

  print(message);

}

? and question : else

  var name = 'sumith';

  switch (name) {

    case 'sumith':

      print('You are sumith');

      break;

    case 'jhone':

      print('You are jhone');

      break;

    default:

      print('Unknow ');

break;

  }

  int number = 1;

  switch (number) {

    case 1:

      print('one');

      break;

    case 2:

      print('two');

      break;

    case 3:

      print('three');

      break;

    case 4:

      print('fore');

      break;

    case 5:

      print('five');

      break;

    case 6:

      print('six');

      break;

    default:

      print(' $number is Unknow ');

      break;

  }

  String  SocialMedia = 'facebook';

  switch (SocialMedia) {

    case 'facebook':

      print(' $SocialMedia is  blue');

      break;

    case 'whatUp':

      print('green');

      break;

    case 'youtube':

      print('red');

      break;

    case 'imo':

      print('indigo');

      break;

    default:

      print(' $SocialMedia is Unknow ');

      break;

  }

Enum

Is a class can use call object name

enum SocialMedia {

  facebook,

  whatup,

  youtube,

  imo,

}

enum Color{

  green,red,blue,indigo,

}

void main() {

 // double like = 500;

  // if (like >= 1000) {

  //   print('very good engagement');

  // } else if (like < 1000 && like <= 700) {

  //   print(' good engagement');

  // } else if (like < 700 && like >= 300) {

  //   print('ok engagement');

  // } else {

  //   print('bad engagement');

  // }

  // String  SocialMedia = 'facebook';

  SocialMedia socialMedia = SocialMedia.facebook;

  switch (socialMedia) {

    case SocialMedia.facebook:

      print(Color.blue);

      break;

    case SocialMedia.whatup:

      print(Color.green);

      break;

    case SocialMedia.youtube:

      print(Color.red);

      break;

    case SocialMedia.imo:

      print(Color.indigo);

      break;

    default:

      print(' $SocialMedia is Unknow ');

      break;

  }

}

Loop while do while

void main() {

  int sum = 0;

  while (sum < 5) {

    print("always!");

    sum++;

  }

}

void main() {

  bool ihavePost = true;

  int countOfpost = 5;

  // int sum = 0;

  while (ihavePost) {

    print("you have another post");

    countOfpost--;

    //countOfpost = countOfpost - 1;

    if (countOfpost == 0) {

      ihavePost = false;

    }

  }

}

void main() {

  //bool ihavePost = true;

  int countOfpost = 5;

  // int sum = 0;

  while (countOfpost > 0) {

    print("you have another post");

    countOfpost--;

  }

}

void main() {

  //bool ihavePost = true;

  int countOfpost = 5;

  // int sum = 0;

  while (countOfpost > 0) {

    //condition check here

    print("you have another post");

    countOfpost--;

    if (countOfpost == 3) {

 print('break!!!');

      break;

    }

  }

For loop

void main() {

  int countOfpost = 5;

 for (int i = 0; i < countOfpost; i++) {

    print("you have another post");

  }

}

void main() {

  int countOfpost = 5;

  for (int i = 0; i < countOfpost; i++) {

    // print("you have another post");

    if (i % 2 == 0) {

       print('ok');

      continue;

    }

    print(i);

    //

  }

}

ok

1 ok 3 ok

  var postcounter = 10;

  for (var i = 1; i <= postcounter; i++) {

    print('postnumber $i');

  }

………………….

enum Content {

  Post,

  reel,

  store,

}

void main() {

  var postcounter = 10;

  var reelCount = 5;

  var storeCount = 7;

  var selectedContent = Content.reel;

  switch (selectedContent) {

    case Content.Post:

    for (var i = 1; i <= postcounter; i++) {

    print('postCount $i');

  }

      break;

        case Content.reel:

    for (var i = 1; i <=  reelCount; i++) {

    print('reelnumber $i');

  }

      break;

        case Content.store:

    for (var i = 1; i <= storeCount; i++) {

    print('storenumber $i');

  }

      break;

  }

}

Function

void main() {

  var tweet1 = 'Tweet1';

  var tweet2 = 'Tweet2';

  var tweet3 = 'Tweet3';

  var tweet4 = 'Tweet4';

  var tweet5 = 'Tweet5';

  bool reFrash = true;

// home page

  print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  if(reFrash){

      print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  }

}

void main() {

  var result = getmyTweet('sumith', 'indika', 'ABC');

  print(result);

}

String getmyTweet(String username, String name, String lastName) {

  var tweet1 = 'Tweet1 for  $username';

  var tweet2 = 'Tweet2 for  $username';

  var tweet3 = 'Tweet3 for  $username';

  var tweet4 = 'Tweet4 for  $username';

  var tweet5 = 'Tweet5 for  $username';

  print('Your other name $name $lastName');

  print('this is your tweet:');

  print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  return 'this is return testing';

}

void main() {

  var result = getmyTweet('sumith');

  print(result);

}

String getmyTweet(String username,[String ? name, String ? lastName] ) {

  var tweet1 = 'Tweet1 for  $username';

  var tweet2 = 'Tweet2 for  $username';

  var tweet3 = 'Tweet3 for  $username';

  var tweet4 = 'Tweet4 for  $username';

  var tweet5 = 'Tweet5 for  $username';

  print('Your other name $name $lastName');

  print('this is your tweet:');

  print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  return 'this is return testing';

}

Use null mandatory this variable

void main() {

  var result = getmyTweet(username: 'madi',name: 'mari',lastName: 'miya');

  print(result);

}

String getmyTweet(

  {

    String? username,

    String?  name,

    String?  lastName ,

  }) {

  var tweet1 = 'Tweet1 for  $username';

  var tweet2 = 'Tweet2 for  $username';

  var tweet3 = 'Tweet3 for  $username';

  var tweet4 = 'Tweet4 for  $username';

  var tweet5 = 'Tweet5 for  $username';

  print('Your other name $name $lastName');

  print('this is your tweet:');

  print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  return 'this is return testing';

}

void main() {

  var result = getmyTweet('joni', name: 'mari',lastName: 'miya');

  print(result);

}

String getmyTweet(

  String? username,

  {

    String?  name,

    String?  lastName ,

  }) {

  var tweet1 = 'Tweet1 for  $username';

  var tweet2 = 'Tweet2 for  $username';

  var tweet3 = 'Tweet3 for  $username';

  var tweet4 = 'Tweet4 for  $username';

  var tweet5 = 'Tweet5 for  $username';

  print('Your other name $name $lastName');

  print('this is your tweet:');

  print(tweet1);

  print(tweet2);

  print(tweet3);

  print(tweet4);

  print(tweet5);

  return 'this is return testing';

}

void main() {

  var result = welcome(name: 'gayan');

  print(result);

}

String welcome({required String name}) =>

  "Hello hi $name";

Use => and not want use return and {}

Class custom type

void main() {

  String text = 'hello';

  var newpost = Post();

  newpost.title = 'change title';

  newpost.likes = 1;

  newpost.incrementLikes();

  print(newpost.likes);

}

class Post {

  var title = 'newPost';

  var likes = 0;

  void incrementLikes() {

    likes++;

  }

}

2

void main() {

  String text = 'hello';

  var newpost = Post();

  newpost.title = 'change title';

  newpost.likes = 1;

  newpost.incrementLikes();

  newpost.incrementLikes();

  print(newpost.likes);

}

class Post {

  var title = 'newPost';

  var likes = 0;

  void incrementLikes() {

    likes++;

    print('Your likes is : $likes');

  }

}

class Post {

  var \_title = 'new Post'; //  \_ private can access in main.dart file

  var \_likes = 0;

  set like(int value) {

    if (value > 1) {

      print("its not a allow ");

      return;

    }

    \_likes = \_likes+ value;

  }

  int get like => \_likes;

  void incrementLikes() {

    \_likes++;

  }

  int getLikes() {

    return \_likes;

  }

}

..

class Post {

  var \_title = 'new Post'; //  \_ private can access in main.dart file

  var \_likes = 1000; //1k

  set newLikes(int value) {

    if (value > 0) {

      print("its not a allow .. ");

      return;

    }

    \_likes = \_likes + value;

  }

  String get like {

    var count = \_likes / 1000;

    return "${count.toInt()} K";

  }

  void incrementLikes() {

    \_likes++;

  }

  int getLikes() {

    return \_likes;

  }

}

import 'Post.dart';

void main() {

  var newpost = Post();

  // newpost.incrementLikes();

  // newpost.newLikes = 10;

  print(newpost.like);

}

1k

Nulable

class Account {

  int ? money = null;

}

import 'Account.dart';

void main() {

  var account = Account();

  print(account.money);

}

Null

void main() {

  var account = Account();

  print(account.money\* 2);

}

NoSuchMethodError: The method '\*' was called on null.

void main() {

  var account = Account();

  String? newName;

  if(newName!= null){

    print(newName.length);

  }

}

import 'Account.dart';

void main() {

  Account? account;

  String? newName;

  int accountNamelength = account?.name?.length?? 0;

  //int len = newName?.length ?? 0; // IF NULL USE  0 NOT null use newName.length

  if (newName != null) {

    int len = newName.length;

  }

}

…………..

class Account {

  int? \_money = null;

  String? name = null;

  int get money {

    return \_money ?? 0;

  }

}

void main() {

  Account? account;

  String? newName;

  int accountNamelength = account?.name?.length ?? 0;

  //int len = newName?.length ?? 0; // IF NULL USE  0 NOT null use newName.length

  if (newName == null) {

    return;

  }

    int l = newName.length;

}

Constructors

class Account {

var name = 'sumith';

  var lastname = 'indika';

}

import 'Account.dart';

void main() {

  Account account = Account();

  print('full name : ${account.name} ${account.lastname}');

}

…………

class Account {

  var name = '';

  var lastname = '';

  Account(String name, String lastname) {

    this.name = name;

    this.lastname = lastname;

  }

}

void main() {

  Account account = Account('sumith', 'indika');

  print('full name : ${account.name} ${account.lastname}');

}

…………..

class Account {

  var name = '';

  var lastname = '';

  Account(this.name, this.lastname);

}

void main() {

  var account = Account('sumith', 'indika');

  print('full name : ${account.name} ${account.lastname}');

}

Same result

class Account {

  var name = '';

  var lastname = '';

  Account({required this.name, required this.lastname});

}

void main() {

  var account = Account(name: 'madii', lastname:'MARAAA');

  print('full name : ${account.name} ${account.lastname}');

}

……

class Account {

  final String name;

  final String  lastname;

  Account({required this.name,required this.lastname});

}

void main() {

  var account = Account(name: 'madii', lastname:'MARAAA');

  print('full name : ${account.name} ${account.lastname}');

}

…………

class Account {

  final String name;

  final String lastname;

  Account({required this.name, required this.lastname});

  Account.defaultAccount()

  :name ='sumith' ,lastname ='indika';

}

void main() {

  var account = Account.defaultAccount();

  print('full name : ${account.name} ${account.lastname}');

}

……….

class Account {

  final String name;

  final String lastname;

  Account(String this.name, String this.lastname);

  //: this.name =name ,this.lastname =lastname;

  Account.defaultAccount()

      : name = 'sumith',

        lastname = 'indika';

}

void main() {

  var account = Account('madi', 'mira');

  print('full name : ${account.name} ${account.lastname}');

}

….

3.19.47

class Account {

  final String name;

  final String lastname;

  Account({required this.name, required this.lastname});

  //: this.name =name ,this.lastname =lastname;

  Account.defaultAccount() : this(name: 'aumith', lastname: 'indika');

 // Account.jhoneAccount();

}

void main() {

  var account = Account('madi', 'mira');

  print('full name : ${account.name} ${account.lastname}');

}

Const & final & constructor

Const variable assign compile time final runtime

Const must assign Constancevalue

void main() {

  const name = 'sumith';

  const number = 1234;

 const addButtonPost = Button('add');

 const addButtonVideo = Button('add');

  const addButtonProfile = Button('add');

   const addButton1 = Button('add');

}

class Button {

 final String title ;

 const Button(this.title);

}

……………

class Account {

  final String name;

  final String lastname;

late final String fullname;

  Account(this.name, this.lastname) {

    fullname = '$name $lastname ${lengthYourfullname()}';

  }

  int lengthYourfullname() {

    return name.length + lastname.length;

  }

}

void main() {

  const name = 'sumith';

  const number = 1234;

  const addButtonPost = Button('add');

  const addButtonVideo = Button('add');

  const addButtonProfile = Button('add');

  const addButton1 = Button('add');

  var account = Account("sumith", 'indika').fullname;

  print(account);

}

Late final :full name

Lays initialization

Static keyword

class Config {

  // static

  static final String baseUrl = 'https://google.com';

 static String getUrl() {

  return 'https://google.com';

}

}

void main() {

  var url = Config.getUrl();

}

List

Lot of item

void main() {

  List<String> posts = ['post1', 'post2', 'post3'];

  print(posts);

}

,…..

void main() {

  var posts = <String>[

    'post number1',

    'post2',

    'post3',

  ];

  posts[1] = 'post number2';

  print(posts);

}

void main() {

  var posts = <String>[

    'post number1',

    'post2',

    'post3',

  ];

  posts[1] = 'post number2';

  posts.add('post 4');

  print(posts);

}

void main() {

  var posts = <String>[

    'post number1',

    'post2',

    'post3',

  ];

  // posts[1] = 'post number2';

  // posts.add('post 4');

  posts.insert(2, 'post my 2.5');

  print(posts);

}

……………

void main() {

  var posts = <String>[

    'post1',

    'post2',

    'post3',

  ];

  // posts[1] = 'post number2';

  //posts.add('post 4');

  //posts.insert(2, 'post my 2.5');

  //posts.removeAt(1);

 // print(posts);

  for (var i = 0; i < posts.length; i++) {

    print(posts[i]);

  }

}

]

……

void main() {

  var posts = <String>[

    'post1',

    'post2',

    'post3',

  ];

  for (var post in posts) {

     print(post);

  }

}

**List**

Ordered

Duplicate allow

Speed depend on size of item

**Set**

Order not matter

Duplicate ignored

Very fast fetch item

void main() {

  var listPosts = <String>[

    'post number1',

    'post2',

    'post3',

  ];

 Set<String> setPosts = {

    'post1',

    'post2',

    'post3 add in',

  };

  print(listPosts.contains('post number1'));

  print(setPosts.contains('post2'));

}

,………

 Set<String> setPosts = {

    'post1',

    'post2',

    'post3 add in',

  };

  //print(listPosts.contains('post number1'));

  print(setPosts);

}

….

  Map<String, String> mappost ={

    'madi' : 'post1',

    'sumith':'post2',

    'indika':'post3',

  };

  print(mappost);

….

  Map<String, String> mappost ={

    'madi' : 'post1',

    'sumith':'post2',

    'indika':'post3',

  };

  print(mappost['madi']);

..

  Map<String, String> mappost = {

    'madi': 'post1',

    'sumith': 'post2',

    'indika': 'post3',

  };

  mappost.remove('madi');

  print(mappost);

….

  Map<String, String> mappost = {

    'madi': 'post1',

    'sumith': 'post2',

    'indika': 'post3',

  };

  mappost.forEach((key, value) {

    print('key:$key , value: $value');

  });

Key and value use forEach

**Inheritance**

**Overriding feature father of content**

Child inherited for father

class Content {

  String name = 'content1';

  @override

  String toString() {

    return name;

  }

}

void main() {

  var content = Content();

  print(content);

}

………………

class Content extends Object {

  final String title;

  //default

  Content(this.title);

  void publish() {

    print('publishing content');

  }

}

class Post extends Content {

  Post(String title) : super(title);

  @override

  void publish() {

    // TODO: implement publish

    print('pubishing post');

  }

}

class Reel extends Content {

  Reel(String title) : super(title);

@override

  void publish() {

    // TODO: implement publish

     print('pubishing reel');

  }

}

class Story extends Content {

  Story(String title) : super(title);

  @override

  void publish() {

    // TODO: implement publish

    print('pubishing store');

  }

}

void main() {

  var post = Post('post 1');

  post.publish();

  var reel = Reel('reel 1');

  reel.publish();

  var story = Story('story 1');

   story.publish();

}

……….

void main() {

  var post = Post('post 1');

  post.publish();

  print(post is Content);  //is check

}

True

**Abstract class**

abstract class Content extends Object {

  final String title;

  //default

  Content(this.title);

  void publish();

}

class Post extends Content implements Shareable {

  Post(String title) : super(title);

  @override

  void publish() {

    print('publish post');

  }

  @override

  void share() {

    print('post sharing 1');

  }

}

abstract class Shareable {

  void share();

}

class Reel extends Content implements Shareable {

  Reel(String title) : super(title);

  @override

  void publish() {

    print('publish reel');

  }

  @override

  void share() {

    print('post is sharing ');

  }

}

class Story extends Content {

  Story(String title) : super(title);

  @override

  void publish() {

    //

    print('publish store');

  }

}

…………….

class Profile implements Shareable {

  @override

  void share() {

    print('share profile');

  }

}

void main() {

  Shareable profile = Profile();

  Shareable post = Post('post');

  print(profile is Shareable);

  List<Shareable> list = [

    Profile(),

    Post('post 1'),

  ];

  for (var ShareableItem in list) {

    ShareableItem.share();

  }

  profile.share();

  post.share();

}

**Mixing**

Interface is abstract

Mixing not abstract code

mixin like {

  int count = 0;

  void incriment() {

    count--;

  }

  void decriment() {

    count++;

  }

}

class Profile with like implements Shareable {

  @override

  void share() {

    print('share profile');

  }

}

void main() {

  var profile = Profile();

  var post = Post('post 1');

  post.incriment();

  profile.incriment();

package & libraries

pub.dev

void main() {

  var number1 = 1000;

  var number2 = 100;

  if (number1 > number2) {

    print(number1);

  } else if (number1 < number2) {

    print(number2);

  } else {

    print("equal");

  }

}

Dart pub get

void main() {

  var response =

      http.get(Uri.parse('https://coffee.alexflipnote.dev/random.json'));

  print(response);

}

Instance of 'Future<Response>'

void main() {

  var response =

      http.get(Uri.parse('https://coffee.alexflipnote.dev/random.json'));

  response.then((value) {

    print(value);

  });

  var StringData = 'file : something';

  print(response); //debug yellow compile error

  var function = (String data) {

    print('print in function body $data');

  };

  function('madi is print');

  //nameOfFunction();

}

void nameOfFunction(Function functionParameter) {}

Instance of 'Future<Response>'

print in function body madi is print

Instance of 'Response'

void main() {

  getCoffee(((value) {

  }));

}

void doPrint(dynamic value) {

  print(value);

}

void getCoffee(Function (dynamic value) callback) {

  var response =

      http.get(Uri.parse('https://coffee.alexflipnote.dev/random.json'));

  response.then( callback,

  ).onError((error, stackTrace) {

    print('Errror');

  });

}

This is Call back

void main() {

 print(getCoffee());

  var StringData = 'file : something ';

}

Future<String> getCoffee() async{

  var response =

    await  http.get(Uri.parse('https://coffee.alexflipnote.dev/random.json'));

  return response.toString();

}

Instance of 'Future<String>'

Feuture string not a string use await and async

Error handle

void main() async {

  //print(await getCoffee());

  var StringData = 'file : something ';

  try {

    print(10 ~/ 0);

  } catch (e) {

    if (e is UnsupportedError) {

      print('YOUR NUMBER INCORRECT');

    }

  }

  finally{

    print('done');

  }

}

void main() async {

  print(await getCoffee());

  var StringData = 'file : something ';

}

Future<String> getCoffee() async {

  try {

    var response =

        await http.get(Uri.parse('https://gflipnote.dev/random.json'));

    return response.toString();

  } catch (e) {

    print(e);

    return "you have a error in futere request";

  }

}

Api