1. Variable

void main() {

  String name = "Muhammad Rizky Raihan";

  print(name);

  var nama = "Rizky ";

  print(nama);

  final nama2 = "Raihan";

*/// variable tidak bisa diubah/dideklarasikan ulang*

  print(nama + nama2);

  final array1 = [1, 2, 3];

*/// isi variable bisa diubah tetapi tidak bisa dideklarasikan ulang*

  const array2 = [1, 2, 3];

*/// isi variable tidak bisa diubah dan variable tidak bisa dideklarasi ulang*

*//array1[0] = 10;*

*//array2[0] = 10;*

  print(array1);

  print(array2);

  late var value = getValue();

  print('variable sudah dibuat');

  print(value);

}

String getValue() {

  print('getValue() dipanggil');

  return 'Muhammad Rizky Raihan';

}

1. Number

void main(){

  int number1 = 10;

  double number2 = 10.5;

  print(number1);

  print(number2);

  num number = 10;

  print(number);

  number = 10.5;

  print(number);

*//  int & double tidak bisa diubah*

*// num bisa diubah valuenya menjadi int / double*

}

1. Boolean

void main() {

*// boolean*

  bool finish = false;

  print(finish);

  finish = true;

  print(finish);

*// end Boolean*

1. String

*// String*

  String firstName = 'Rizky';

  String lastName = "Raihan";

  print(firstName);

  print(lastName);

  print("$firstName $lastName");

  var fullName = '$firstName $lastName';

  print(fullName);

  var text = 'this is \'dart\' \$cool';

  print(text);

*/\**

*karakter \(backslash) di String bisa digunakan*

*untuk menekankan bahwa karakter setelahnya dianggap benar karakter tsb*

*\*/*

  var name1 = "$firstName $lastName";

*// kalau valuenya dalam bentuk variable ditambahkan dengan interpolation*

  var name2 = 'Rizky ' 'Raihan';

*/\*kalau valuenya tidak dalam bentuk variable tidak bisa ditambahkan*

*dengan spasi/tab*

*\*/*

  print(name1);

  print(name2);

*// Multiline String*

  var longString = '''

  this is long string

  multiline string

  learning dart

  ''';

  print(longString);

*// end String*

1. Dynamic Variable

*// Dynamic variable*

*/\**

*bisa menggunakan dynamic*

*bisa juga dengan membuat variable tanpa mendeklarasikannya*

*seperti -> var contoh;*

*\*/*

  dynamic hasil = 100;

  print(hasil);

  hasil = true;

  print(hasil);

  hasil = 'halo';

  print(hasil);

}

*// end Dynamic variable*

1. Konversi Tipe Data

void main() {

*// STRING, INT, DOUBLE*

  var inputString = '1000';

  var inputInt = int.parse(inputString); *// konversi dari string ke integer*

  var inputDouble = double.parse(inputString); *// konversi dari string ke double*

  print(inputString);

  print(inputInt);

  print(inputDouble);

  var intToDouble = inputInt.toDouble(); *// konversi dari int ke double*

  var doubleToInt = inputDouble.toInt(); *// konversi dari double ke int*

  print(intToDouble);

  print(doubleToInt);

  var intToString = inputInt.toString(); *// konversi dari int ke string*

  var doubleToString = inputDouble.toString(); *// konversi dari double ke string*

  print(intToString);

  print(doubleToString);

*// BOOLEAN*

  var input = 'true';

  var inputBoolean = input == 'true';

  var booleanToString = inputBoolean.toString();

  print(inputBoolean);

  print(booleanToString);

}

1. Operator Aritmatika

void main() {

*// Operator Aritmatika*

  var first = 10;

  var second = 3;

  print(first + second); *// output 13*

  print(first - second); *// output 7*

  print(first \* second); *// output 30*

  print(first / second); *// pembagian hasil double   // output 3.3333333333*

  print(first ~/ second); *// pembagian hasil int     // output 3*

  print(first % second); *// modulo/ sisa bagi        // output 1*

}

1. Operator Perbandingan

void main() {

  print(10 == 10); *// output true*

  print(10 != 10); *// output false*

  print(10 > 10); *// output false*

  print(10 < 10); *// output false*

  print(10 >= 10); *// output true*

  print(10 <= 10); *// output true*

  print('rizky' == 'rizky'); *// output true*

  print('rizky' != 'rizky'); *// output false*

}

1. Operator Penugasan

void main() {

  var tambah = 10;

  tambah += 10; *// --> tambah = tambah + 10*

  print(tambah);

  var kurang = 10;

  kurang -= 10; *// --> kurang = kurang - 10*

  print(kurang);

  var kali = 10;

  kali \*= 10; *// --> kali = kali \* 10*

  print(kali);

  double bagi = 10;

  bagi /= 10; *// --> bagi = bagi / 10*

  print(bagi);

  var sisaBagi = 10;

  sisaBagi ~/= 10; *// --> sisaBagi = sisaBagi ~/ 10*

  print(sisaBagi);

  var modulus = 10;

  modulus %= 10; *// --> modulus = modulus \* 10*

  print(modulus);

*// INCREMENT*

  var i = 1;

  var j = i++; *// j = i, i++*

  var k = ++i; *// k = (++i)*

  print(i);

  print(j);

  print(k);

*// DECREMENT*

  var l = 10;

  var m = l--; *// m = l, l--*

  var n = --l; *// n = (--l)*

  print(l);

  print(m);

  print(n);

}

1. Operator Logika

void main() {

*// Operator Logika*

  var nilaiAkhir = 80;

  var nilaiAbsen = 40;

  var hasilNilaiAkhir = nilaiAkhir >= 70;

  var hasilNilaiAbsen = nilaiAbsen >= 70;

  print(hasilNilaiAkhir); *// output true*

  print(hasilNilaiAbsen); *// output false*

*// var lulus = hasilNilaiAkhir && hasilNilaiAbsen;*

*// print(lulus);*

  var lulus = hasilNilaiAkhir || hasilNilaiAbsen;

  print(lulus); *// output true*

*/\**

*AND                           OR*

*true && true = true           true || true = true*

*true && false = false         true || false = true*

*false && true = false         false || true = true*

*false && false = false        false || false = false*

*KEBALIKAN (!)*

*true = false*

*false = true*

*\*/*

  print(!true); *// output false*

}

1. Operator Type Test

void main() {

*/\**

*as = Typecast, melakukan konversi tipedata secara paksa*

*is = true, jika object sesuai tipe data*

*is! = true, jika object tidak sesuai tipe data*

*\*/*

  dynamic variable = 100;

  var variableInt = variable as int;

  print(variable); *// output 100*

  print(variableInt); *// output 100*

  print(variable is int); *// output true*

  print(variable is String); *// output false*

  print(variable is bool); *// output false*

  print(variable is! int); *// output false*

  print(variable is! String); *// output true*

  print(variable is! bool); *// output true*

}

1. List

void main() {

*// create list int*

  List<int> listInt = [];

*// create list string*

  var listString = <String>[];

  print(listInt);

  print(listString);

  var nama = <String>['Muhammad', 'Rizky', 'Raihan'];

*// cara add list tradisional*

*// nama.add('Muhammad');*

*// nama.add('Rizky');*

*// nama.add('Raihan');*

  print(nama);

  print(nama.length);

  print(nama[0]);

  nama[1] = 'riski';

  print(nama[1]);

  nama.removeAt(2);

  print(nama);

}

1. Set

void main() {

*// di SET tidak bisa mengubah data dan menduplikat data*

  Set<int> numbers = {};

  var strings = <String>{};

  var doubles = <double>{};

  print(numbers);

  print(strings);

  print(doubles);

*// deklarasi secara langsung*

  var nama = <String>{'Muhammad', 'Rizky', 'Raihan'};

*//deklarasi tradisional*

*// nama.add('Muhammad');*

*// nama.add('Muhammad');*

*// nama.add('Rizky');*

*// nama.add('Rizky');*

*// nama.add('Raihan');*

  print(nama);

  print(nama.length);

  nama.remove('Rizky');

  nama.remove('Tidak Ada');

  print(nama);

  print(nama.length);

}

1. Map

void main() {

*// beberapa cara membuat map*

*// Map<String, String> map1 = {};*

*// var map2 = <String, String>{};*

*// manipulasi Map*

*// map.length => mendapatkan panjang map*

*// map[key] => mendapatkan data di map*

*// map[key] = value => mengubah data di map*

*// map.remove(key) => menghapus data di map*

  var nama = <String, String>{

    'first': 'Muhammad',

    'middle': 'Rizky', *// ===> deklarasi variable langsung*

    'last': 'Raihan'

  };

*// nama['first'] = 'Muhammad';*

*// nama['middle'] = 'Rizky';*

*// nama['last'] = 'Raihan';*

  print(nama);

  print(nama['first']);

  nama['middle'] = 'Iki';

  print(nama);

  nama.remove('last');

  print(nama);

}

1. Symbol

void main() {

  Symbol symbol1 = Symbol('Rizky Raihan'); *// kalau mau make spasi pake ini*

  var symbol2 = #belajar\_dart; *// kalau pake # gk bisa spasi*

  print(symbol1);

  print(symbol2);

}

1. If/Else

void main(){

  var nilai = 80;

  var absen = 80;

  if(nilai >= 80 && absen >= 80){

    print('Nilai anda A');

  } else if(nilai >= 70 && absen >= 70){

    print('Nilai Anda B');

  } else if(nilai >= 60 && absen >= 60) {

    print('Nilai Anda C');

  } else {

    print('Anda Tidak Lulus');

  }

}

1. Switch Case

void main() {

  var nilai = 'e';

  switch (nilai) {

    case 'A':

      print('Anda lulus dengan baik');

      break;

    case 'B':

    case 'C':

      print('Anda lulus');

      break;

    case 'D':

      print('Anda Tidak Lulus');

      break;

    default:

      print('Anda salah jurusan');

  }

}

1. Null

void main() {

  int number1;

  number1 = 10;

  int? number2;

  number2 = null;

  print(number1);

  print(number2);

}

1. Ternary Operator

void main(){

  var nilai = 75;

  var ucapan = nilai >= 70 ? 'Selamat Anda Lulus' : 'Silahkan Coba Lagi';

*// TANPA TERNARY*

*// String ucapan;*

*// if(nilai >= 70){*

*//   ucapan ='Selamat Anda lulus';*

*// }else{*

*//   ucapan = 'Silahkan coba lagi';*

*// }*

  print(ucapan);

}

1. Null Safety

void main() {

*// null check*

  int? age;

  if (age != null) {

    double ageDouble = age.toDouble();

    print(ageDouble);

  }

*// konversi nullable ke non nullable*

  String name = 'Rizky';

  String? nullableName = name;

  int? nullablePrice = null;

  if (nullablePrice != null) {

    int price = nullablePrice;

  }

*// default value*

  String? guest;

  String guestName = guest ?? 'Guest';

*// if(guest != null){*

*//   guestName = guest;*

*// }else{*

*//   guestName = 'Guest';*

*// }*

  print(guestName);

*// konversi secara paksa*

  int? nullableNumber;

  int nonNullableNumber = nullableNumber!; *// output error*

*// mengakses nullable member*

  int? dataInt;

  double? dataDouble = dataInt?.toDouble();

*// if(dataInt != null ){*

*//   dataDouble = dataInt.toDouble();*

*// }*

  print(dataDouble);

}

1. For Loop

void main() {

*//\* for(init statement;kondisi;post statement){*

*// // block perulangan*

*// }*

*// init statement, kondisi, post statement tidak wajib diisi, jika kondisi tidak diisi,*

*// berarti kondisi selalu bernilai true*

*/\**

*\*   for(;;){*

*\*   print('Perulangan Tanpa henti');*

*\*/*

  for (var counter = 1; counter <= 10; counter++) {

    print('Perulangan ke-$counter');

  }

}

1. While Loop

void main(){

  var counter = 1;

  while(counter<=10){

    print('Perulangan Ke-$counter');

    counter++;

  }

}

1. Do While Loop

void main(){

*// pada do while loop akan mengeksekusi perulangan minimal 1x*

  var counter = 100;

  do{                                 *// menjalankan perulangan*

    print('Perulangan ke-$counter');

    counter++;

  } while(counter<=10);               *// pengecekan perulangan*

}

1. Break And Continue

void main(){

*// break = digunakan untuk menghentikan perulangan.*

*// continue = digunakan untuk menghentikan perulangan saat ini,*

*//            lalu melanjutkan ke perulangan selanjutnya.*

  var counter = 1;

  while(true){

    print('Perulangan ke-$counter');

    counter++;

    if (counter > 10){

      break;

    }

  }

  for(var counter2 = 1; counter2 <= 100; counter2++){

    if(counter2 % 2 == 0){

      continue;

    }

    print('Perulangan ke-$counter2');

  }

}

1. For In

void main() {

  var names = <String>['Muhammad', 'Rizky', 'Raihan'];

*// tanpa for in*

*// for(var i = 0; i< names.length; i++){*

*//   print(names[i]);*

*// }*

  for (var value in names) {

    print(value);

  }

  var nameSet = <String>{'Muhammad', 'Rizky', 'Raihan'};

  for (var value in nameSet) {

    print(value);

  }

}

1. Function

void sayHello() {

  print('Hello World');

}

void main() {

  sayHello();

  sayHello();

}

1. Function Parameter

void sayHello(String firstName, String lastName) {

  print('Halo $firstName $lastName');

}

void main() {

  sayHello('Rizky', 'Raihan');

  sayHello('Raihan', 'Rizky');

}

1. Optional Parameter

void sayHello(String firstName, [String middleName = '', String lastName = '']){

  print('Hello $firstName $middleName $lastName');

}

void main(){

  sayHello('Muhammad');

  sayHello('Muhammad', 'Rizky');

  sayHello('Muhammad', 'Rizky', 'Raihan');

*// jika optional parameter tidak ingin nullable, kita harus menambahkan default value*

}

1. Named Parameter

void sayHello({ firstName = '', String lastName = ''}){ *// ='' default value*

*// required = kita harus mengisi sebuah parameter misal*

*// void sayHello({required firstName = '', String lastName = ''})*

  print('Hello $firstName $lastName');

}

void main(){

  sayHello();

  sayHello(firstName: 'Rizky');

  sayHello(lastName: 'Raihan');

  sayHello(lastName: 'Raihan' , firstName: 'Rizky');

*// sayHello('Rizky', 'Raihan'); output error*

}

1. Function Return Value

String sayHello(String name){

  return 'Hello $name';

}

int sum(List<int> numbers){

  var total = 0;

  for (var value in numbers){

    total += value;   *// total = value + value*

  }

  return total;

}

void main(){

  var data = sayHello('Rizky');

  print(data);

  var total = sum([10,10,10,10,10]);

  print(total);

  print(sum([5,5,5,5,5]));

}

1. Function Short Expression

*// int sum(int first, int second){*

*//   return first + second;*

*// }*

int sum(int first, int second) => first + second;

*// jika terdapat sebuah function yang hanya 1 baris, kita bisa menyingkatnya secara sederhana*

*// ketika membuat function short expression, kita tidak butuh kurung {} dan juga tidak butuh kata kunci return*

void main() {

  var total = sum(10, 10);

  print(total);

  print(sum(5, 5));

}

1. Inner Function

*// inner function yang dibuat di outer function hanya bisa diakses dari outer function saja, tidak bisa diakses dari luar outer function*

void main() {

  void sayHello() {

    print('Hello inner function');

    print('hello function inner');

  }

  sayHello();

  sayHello();

}

1. Higher Order Function

void sayHello(String name, String Function(String) filter) {

  var filteredName = filter(name);

  print('Hi $filteredName');

}

String filterBadWord(String name) {

  if (name == 'gila') {

    return '\*\*\*\*';

  } else {

    return name;

  }

}

void main() {

  sayHello('Eko', filterBadWord);

  sayHello('gila', filterBadWord);

}

1. Anonymous Function

*// anonymous function as parameter*

void sayHello(String name, String Function(String)filter){

  print('Hello ${filter(name)}');

}

void main(){

*// anonymous function as parameter*

  sayHello('Muhammad Rizky Raihan', (name) {

    return name.toUpperCase();

  });

*// short expression*

  sayHello('Muhammad Rizky Raihan', (name) => name.toLowerCase());

*// end of anonymous function as parameter*

*// anonymous function as variable*

  var upperFunction = (String name){

    return name.toUpperCase();

  };

  var lowerFunction = (String name) => name.toLowerCase();

  var result1 = upperFunction('rizky');

  print(result1);

  var result2 = lowerFunction('RAIHAN');

  print(result2);

*// end of anonymous function as variable*

}

1. Scope

*// scope = variable / function hanya bisa diakses didalam area dimana mereka dibuat*

*// co = variable dibuat di function, maka hanya bisa diakses dimethod tersebut.*

*//      jika dibuat didalam blok, maka hanya bisa diakses didalam blok tersebut.*

void main() {

  var name = 'Rizky';

  void sayHello() {

    var hello = 'Hello $name';

    print(hello);

  }

  sayHello();

  print(hello); *//ERROR, karena variable hello hanya bisa dipanggil ditempat ia dibuat.*

}

void contoh() {

*// sayHello(); error*

}

1. Closure

*// closure = kemampuan sebuah function / anonymous function berinteraksi dengan data-data disekitarnya dalam scope yang sama*

void main() {

  var counter = 0;

  void increment() {

    print('increment');

    counter++;

  }

  print(counter);

  increment();

  increment();

  print(counter);

}

1. Recursive Function

*// recursive function = function yang memanggil function dirinya sendiri.*

int factorialLoop(int value) {

*// faktorial biasa*

  var result = 1;

  for (var i = 1; i <= value; i++) {

    result \*= i; *// result = result \* i;*

  }

  return result;

}

int factorialRecursive(int value) {

  if (value == 1) {

    return 1;

  } else {

    return value \* factorialRecursive(value - 1);

  }

}

void loop(int value) {

  if (value == 0) {

    print('Selesai');

  } else {

    print('Perulangan ke-$value');

    loop(value - 1);

  }

}

void main() {

  print(factorialLoop(10));

  print(1 \* 2 \* 3 \* 4 \* 5 \* 6 \* 7 \* 8 \* 9 \* 10);

  print(factorialRecursive(10));

*//factorialRecursive(10) // fac(10) => 10 \* fac(9) => 10 \* 9 \* fac(8) =>*

  loop(10000);

}