**JOBSHEET 3**

**PEMROGRAMAN MOBILE**



**SERLI PUTRI MAHARANI**

**TI-3E / 26**

**2141720023**

**PROGRAM STUDI D – IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

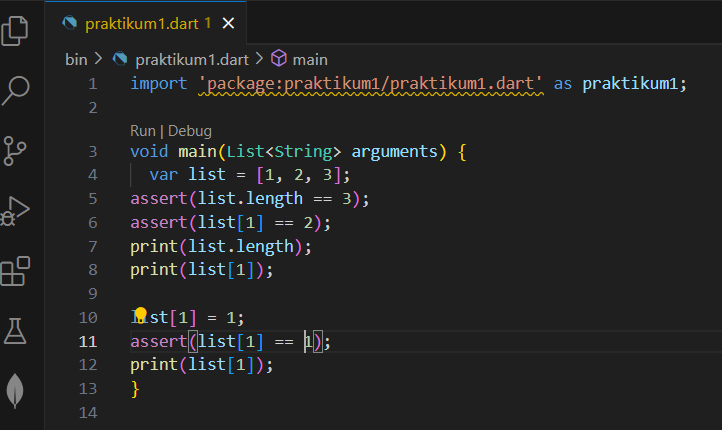
**2023**

**PRAKTIKUM 1**

**Langkah 1**

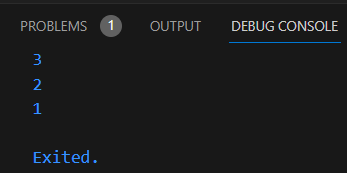
Ketik atau salin kode program berikut ke dalam void main().





**Langkah 2**

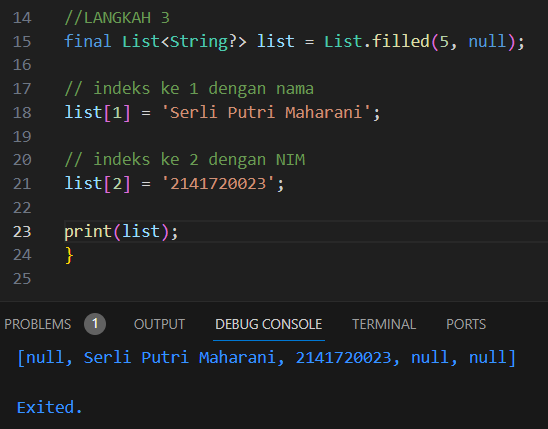
Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan!



**Langkah 3**

Ubah kode pada langkah 1 menjadi variabel final yang mempunyai index = 5 dengan default value = null. Isilah nama dan NIM Anda pada elemen index ke-1 dan ke-2. Lalu print dan capture hasilnya.

Apa yang terjadi ? Jika terjadi error, silakan perbaiki

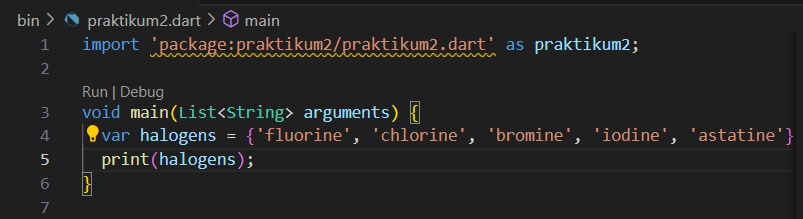


**PRAKTIKUM 2**

**Langkah 1**

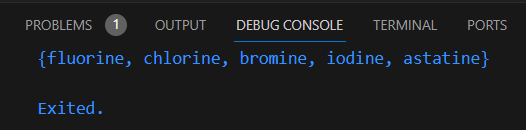
Ketik atau salin kode program berikut ke dalam fungsi main().





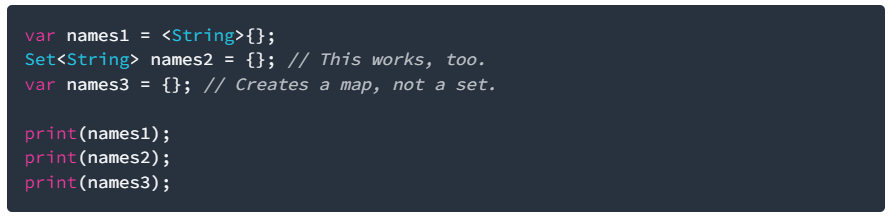
**Langkah 2**

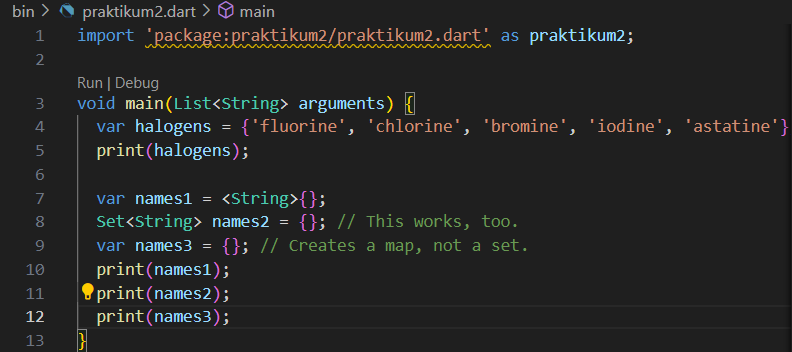
Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

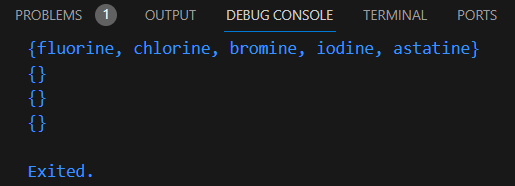


**Langkah 3**

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

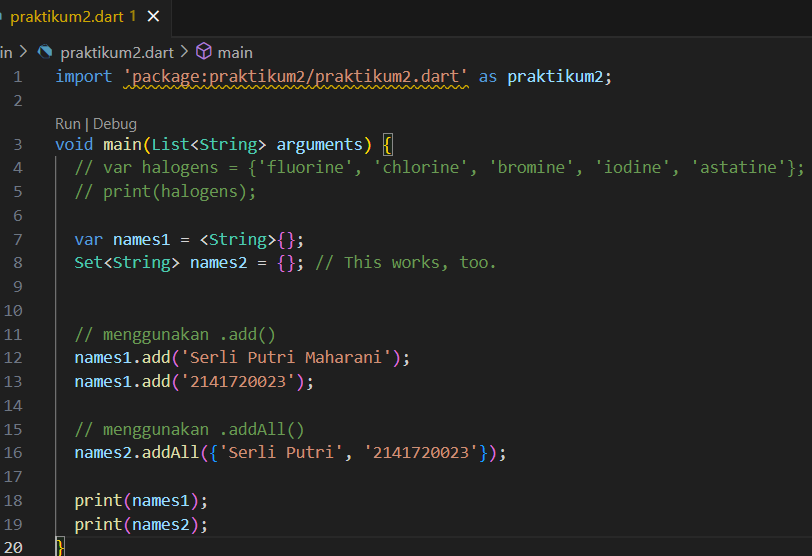
Apa yang terjadi ? Jika terjadi error, silakan perbaiki namun tetap menggunakan ketiga variabel tersebut.

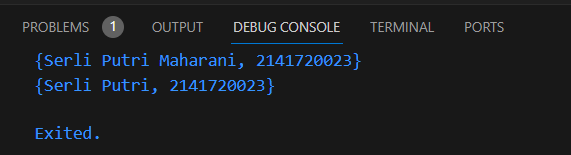




Tambahkan elemen nama dan NIM Anda pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu .add() dan .addAll(). Untuk variabel Map dihapus, nanti kita coba di praktikum selanjutnya.

Dokumentasikan code dan hasil di console, lalu buat laporannya.

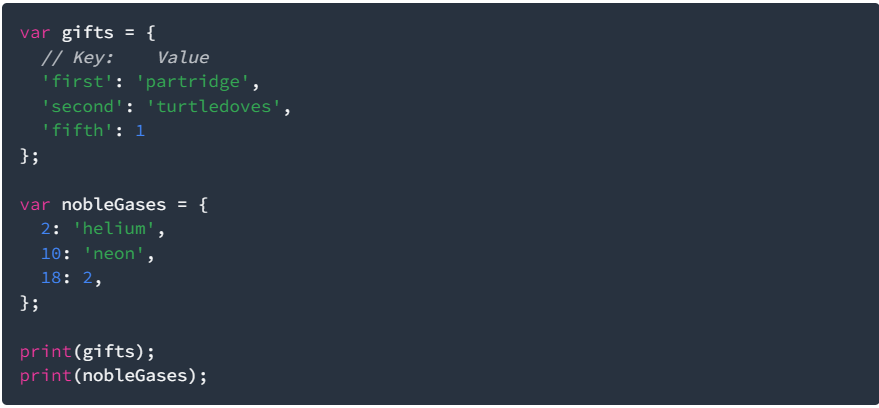




**PRAKTIKUM 3**

**Langkah 1**

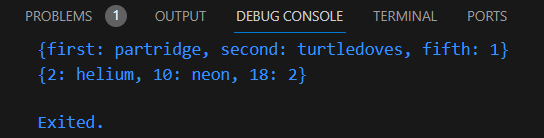
Ketik atau salin kode program berikut ke dalam fungsi main().





**Langkah 2**

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

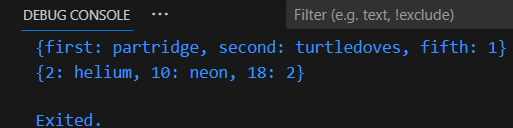


**Langkah 3**

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

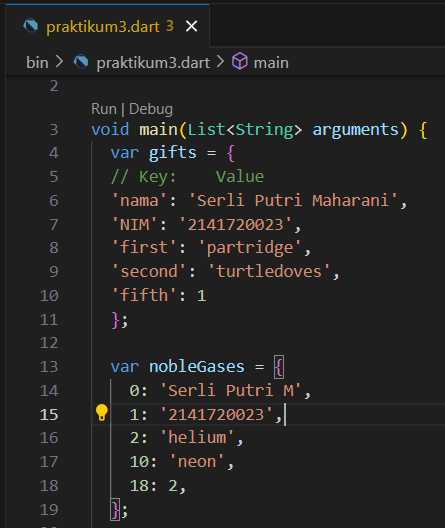


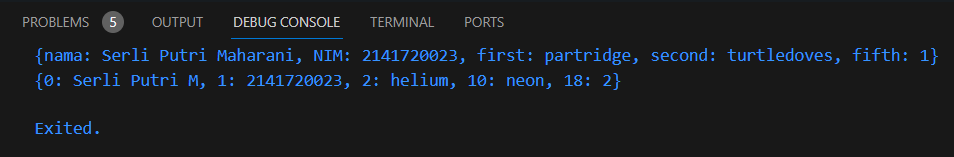




Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Tambahkan elemen nama dan NIM Anda pada tiap variabel di atas (gifts, nobleGases, mhs1, dan mhs2). Dokumentasikan hasilnya dan buat laporannya!

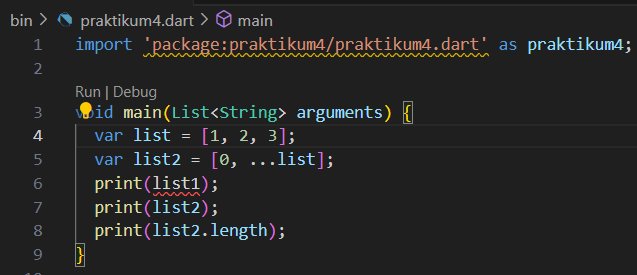


**PRAKTIKUM 4**

**Langkah 1**

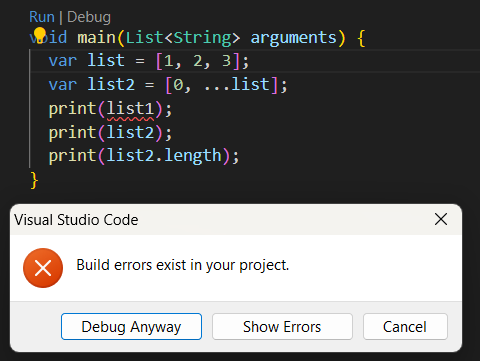
Ketik atau salin kode program berikut ke dalam fungsi main().

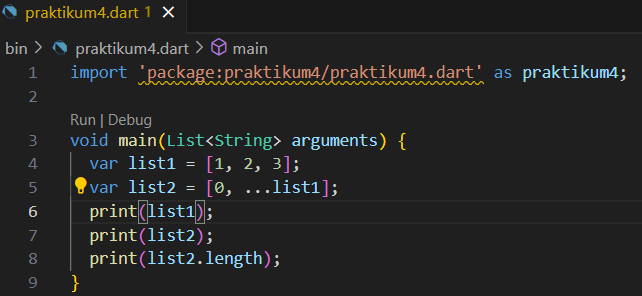


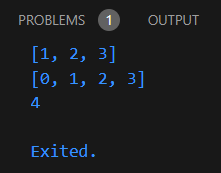


**Langkah 2**

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.



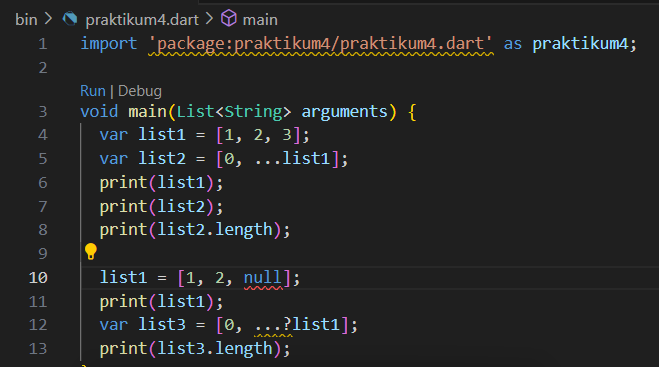




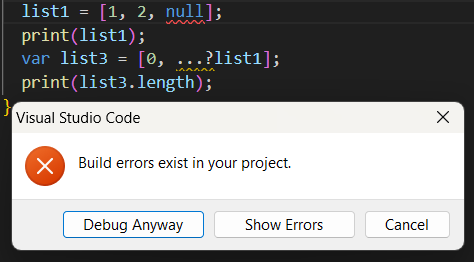
**Langkah 3**

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.





Apa yang terjadi ? Jika terjadi error, silakan perbaiki.



Tambahkan variabel list berisi NIM Anda menggunakan Spread Operators. Dokumentasikan hasilnya dan buat laporannya!

**Langkah 4**

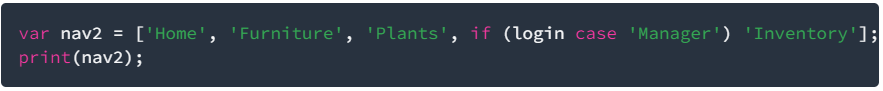
Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.



Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel promoActive ketika true dan false.

**Langkah 5**

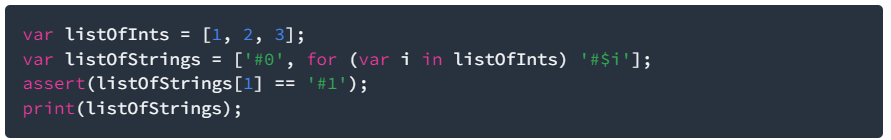
Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.



Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel login mempunyai kondisi lain.

**Langkah 6**

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.



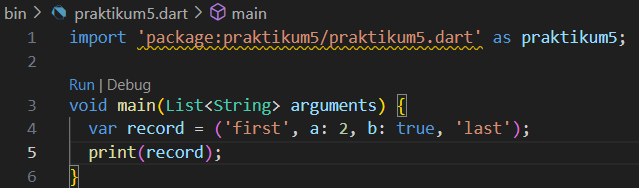
Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Jelaskan manfaat **Collection For** dan dokumentasikan hasilnya.

**PRAKTIKUM 5**

**Langkah 1**

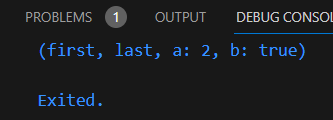
Ketik atau salin kode program berikut ke dalam fungsi main().





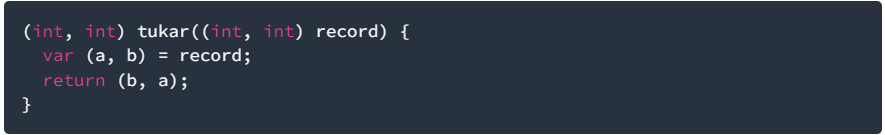
**Langkah 2**

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

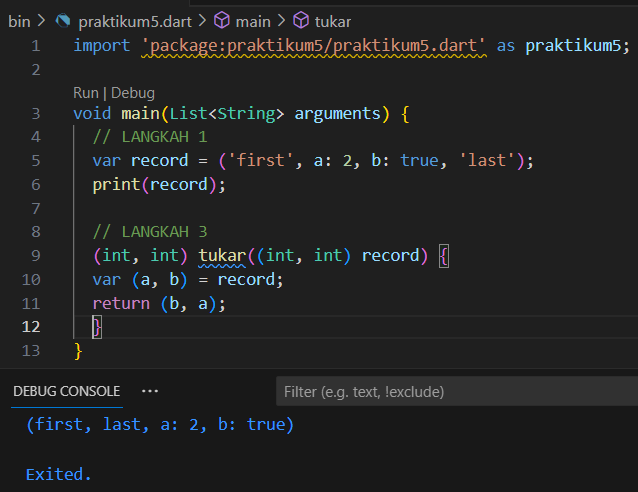


**Langkah 3**

Tambahkan kode program berikut di luar scope void main(), lalu coba eksekusi (Run) kode Anda.

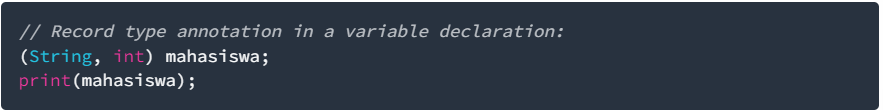


Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gunakan fungsi tukar() di dalam main() sehingga tampak jelas proses pertukaran value field di dalam Records.

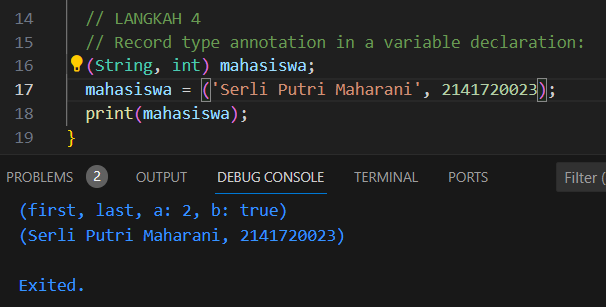


**Langkah 4**

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

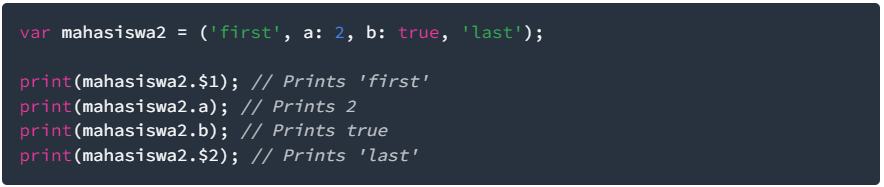


Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Inisialisasi field nama dan NIM Anda pada variabel record mahasiswa di atas. Dokumentasikan hasilnya dan buat laporannya!

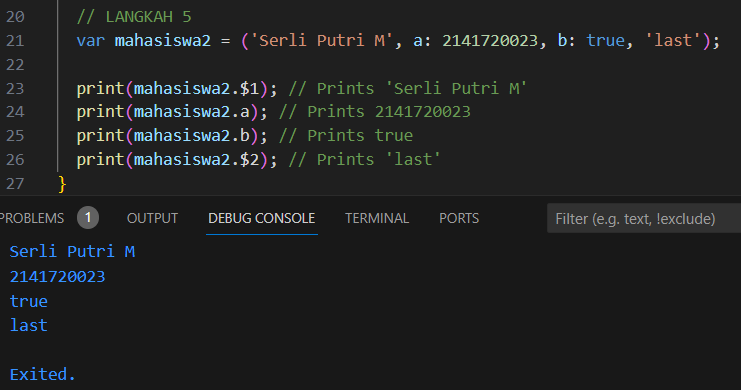


**Langkah 5**

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.



Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gantilah salah satu isi record dengan nama dan NIM Anda, lalu dokumentasikan hasilnya dan buat laporannya!



**TUGAS PRAKTIKUM**

1. Silakan selesaikan Praktikum 1 sampai 5, lalu dokumentasikan berupa screenshot hasil pekerjaan Anda beserta penjelasannya!
2. Jelaskan yang dimaksud Functions dalam bahasa Dart!

* Functions dalam Bahasa Dart adalah blok bangunan dari kode yang dapat dibaca, dipelihara, dan dapat digunakan Kembali. Fungsi adalah sekumpulan pernyataan untuk melakukan tugas tertentu dan mengatur program ke adlam blok kode logis. Setelah didefinisikan, fungsi dapat dipanggil utnuk mengakses kode dan membuat kode dapat digunakan Kembali. Deklarasi fungsi berisi nama fungsi, tipe nilai kembalian fungsi, dan parameter-parameter yang dibutuhkan oleh fungsi tersebut.

1. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!

* Parameter posisi: parameter yang diberikan berdasarkan urutan posisi saat memanggil fungsi.

Contoh:

void ucapan(String nama, String ucap) {

print('$ucap, $nama!');

}

ucapan('Serli', 'Hello'); // Output: Hello, Serli!

* Parameter opsional: parameter yang tidak wajib diisi saat memanggil fungsi.

Contoh:

void ucapan(String nama, [String ucap]) {

if (ucap == null) {

ucap = 'Hello';

}

print('$ucap, $nama!');

}

ucapan(Serli'); // Output: Hello, Serli!

ucapan('Serli', 'Hi'); // Output: Hi, Serli!

* Parameter Bernama: parameter yang diberikan dengan menyebutkan namanya saat memanggil fungsi.

Contoh:

void ucap({String nama, String ucapan}) {

if (ucapan == null) {

ucapan = 'Hello';

}

print('$ucapan, $nama!');

}

ucap(nama: 'Serli'); // Output: Hello, Serli!

ucap(nama: 'Serli', ucapan: 'Hi'); // Output: Hi, Serli!

* Parameter nilai default: parameter yang memiliki nilai default jika tidak diisi saat memanggil fungsi.

Contoh:

void ucap(String nama, {String ucapan = 'Hello'}) {

print('$ucapan, $nama!');

}

ucap('Serli'); // Output: Hello, Serli!

ucap('Serli', ucapan: 'Hi'); // Output: Hi, Serli!

* Parameter rest: parameter yang dapat menerima jumlah argument yang tidak terbatas saat memanggil fungsi.

Contoh:

void sum(int a, int b, [int c, int d, int e]) {

int result = a + b;

if (c != null) {

result += c;

}

if (d != null) {

result += d;

}

if (e != null) {

result += e;

}

print(result);

}

sum(1, 2); // Output: 3

sum(1, 2, 3); // Output: 6

sum(1, 2, 3, 4); // Output: 10

sum(1, 2, 3, 4, 5); // Output: 15

1. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!

* Functions sebagai first-class objects: fungsi yang diperlakukan seperti objek lainnya dalam Bahasa pemrograman. Dalam konsep ini, fungsi dapat disimpan dalam variabel, dijadikan parameter pada fungsi lain, dan dijadikan nilai kembalian dari fungsi lain. Konsep ini memungkinkan penggunaaan fungsi secara fleksibel dan dapat meningkatkan efisiensi kode.

Contoh:

int add(int n1, int n2) {

return n1 + n2;

}

// fungsi sebagai variabel

var calc1 = add;

// fungsi sebagai parameter

int calculate(int Function(int, int) operation, int n1, int n2) {

return operation(n1, n2);

}

var result = calculate(add, 4, 2);

print(result); // Output: 6

1. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!

* Anonymous functions atau lambda functions adalah fungsi yang tidak memiliki nama dan biasanya digunakan untuk melakukan operasi sederhana yang tidak memerlukan definisi fungsi yang lengkap. Fungsi ini dapat didefinisikan secara langsung di dalam kode atau sebagai parameter pada fungsi lain.

Contoh:

void main() {

var add = (int a, int b) => a + b;

var subtract = (int a, int b) => a - b;

var result1 = add(4, 2);

var result2 = subtract(4, 2);

print(result1); // Output: 6

print(result2); // Output: 2

}

1. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!

* Lexical scope:
* Menentukan bagian program mana yang dapat menggunakan dan mengubah bagian lainnya
* Membuat sumber daya yang dideklarasikan (variabel, fungsi, dan objek) hanya tersedia untuk fungsi di mana mereka dideklarasikan untuk setiap fungsi anak yang bersarang di dalamnya
* Sumber daya yang dideklarassikan dalam fungsi anak yang bersarang biasanya tidak tersedia untuk induknya
* Lexical scope ditentukan pada saat menulis kode

Contoh:

void main() {

String outerVariable = "Saya di luar!";

void innerFunction() {

String innerVariable = "Saya di dalam!";

print(outerVariable); // mencetak "Saya di luar!"

}

// print(innerVariable); // ReferenceError: innerVariable is not defined

innerFunction();

}

* Lexical closures:
* Adalah fungsi yang memiliki akses ke variabel dalam konteks di mana mereka di mana mereka didefinisikan, bukan di mana mereka di panggil
* Dibuat ketika sebuah fungsi di definisikan di dalam fungsi lain, dan fungsi dalam memiliki akses ke variabel luar fungsi dalam
* Memungkinkan untuk privasi data dan enkapsulasi dalam Dart
* Konsep yang lebih lanjut dan sering menjadi topik wawancara teknis

Contoh:

void outerFunction() {

String outerVariable = "Saya di luar!";

void innerFunction() {

String innerVariable = "Saya di dalam!";

print(outerVariable); // mencetak "Saya di luar!"

}

return innerFunction;

}

void main() {

Function closure = outerFunction();

closure(); // mencetak "Saya di luar!"

}

1. Jelaskan dengan contoh cara membuat return multiple value di Functions!

* Menggunakan list: dengan mengembalikan sebuah list yang berisi beberapa nilai

Contoh:

List<int> getMinMax(List<int> numbers) {

int min = numbers[0];

int max = numbers[0];

for (int i = 1; i < numbers.length; i++) {

if (numbers[i] < min) {

min = numbers[i];

}

if (numbers[i] > max) {

max = numbers[i];

}

}

return [min, max];

}

void main() {

List<int> numbers = [5, 2, 8, 1, 9, 3];

List<int> result = getMinMax(numbers);

print('Minimum value: ${result[0]}');

print('Maximum value: ${result[1]}');

}

* Menggunakan map: mengembangkan sebuah map yang berisi beberapa nilai

Contoh:

Map<String, dynamic> getPerson() {

String name = 'John Doe';

int age = 30;

String address = '123 Main St';

return {'name': name, 'age': age, 'address': address};

}

void main() {

Map<String, dynamic> person = getPerson();

print('Name: ${person['name']}');

print('Age: ${person['age']}');

print('Address: ${person['address']}');

}

* Menggunakan record: mengembalikan sebuah record yang berisi beberapa nilai

Contoh:

record Person(String name, int age, String address) {}

Person getPerson() {

String name = 'John Doe';

int age = 30;

String address = '123 Main St';

return Person(name, age, address);

}

void main() {

Person person = getPerson();

print('Name: ${person.name}');

print('Age: ${person.age}');

print('Address: ${person.address}');

}

1. Kumpulkan berupa link commit repo GitHub pada tautan yang telah disediakan di grup Telegram!