

Dart Tham Khảo Nhanh

Kiểu dữ liệu, hàm, class, async, null safety, collection

Cơ Bản

Hello World

```
void main() {
  print('Hello, Dart!');
}
```

Biến

```
var name = 'Dart'; // suy luận kiểu
String lang = 'Dart'; // kiểu tường minh
final pi = 3.14; // hằng lúc runtime
const max = 100; // hằng lúc biên dịch
```

Nội Suy Chuỗi

```
var name = 'World';
print('Hello, $name!');
print('1 + 1 = ${1 + 1}');
```

Kiểu Dữ Liệu

Kiểu Built-in

int	Số nguyên 64-bit
double	Dấu phẩy động 64-bit
num	Supertype của int và double
String	Chuỗi UTF-16
bool	true hoặc false
List	Collection có thứ tự (mảng)
Set	Collection không thứ tự, duy nhất
Map	Cặp key-value
dynamic	Bất kỳ kiểu nào, tắt kiểm tra tĩnh
void	Không có giá trị trả về

Kiểm Tra & Ép Kiểu

```
if (obj is String) print(obj.length);
var s = obj as String; // ép kiểu
print(obj.runtimeType); // kiểu runtime
```

Hàm

Cú Pháp Hàm

```
int add(int a, int b) => a + b;
void greet({required String name}) {
  print('Hello, $name');
}
```

Tham Số

int f(int a)	Tham số vị trí bắt buộc
int f([int a = 0])	Tham số vị trí tùy chọn với mặc định
f({required int a})	Tham số named bắt buộc
f({int a = 0})	Tham số named tùy chọn với mặc định

Closure & Tearoff

```
var square = (int n) => n * n;
[1, 2, 3].map((e) => e * 2);
[1, 2, 3].forEach(print); // tearoff
```

Luồng Điều Khiển

Điều Kiện

```
if (x > 0) { print('dương'); }
else if (x == 0) { print('không'); }
else { print('âm'); }
var result = x > 0 ? 'dương' : 'âm';
```

Vòng Lặp

```
for (var i = 0; i < 5; i++) { }
for (var item in list) { }
while (x > 0) { x--; }
do { x--; } while (x > 0);
```

Switch & Pattern Matching

```
switch (color) {
  case 'red': print('R'); break;
  case 'blue': print('B'); break;
  default: print('?');
}
```

Classes

Định Nghĩa Class

```
class Point {
  final double x, y;
  Point(this.x, this.y);
  double distanceTo(Point p) =>
    sqrt(pow(x - p.x, 2) + pow(y - p.y, 2));
}
```

Named & Factory Constructors

```
class Point {
  double x, y;
  Point(this.x, this.y);
  Point.origin() : x = 0, y = 0;
  factory Point.fromJson(Map j) =>
    Point(j['x'], j['y']);
}
```

Kế Thừa

```
class Animal { void speak() {} }
class Dog extends Animal {
  @override
  void speak() => print('Woof');
}
```

Mixins & Extensions

Mixins

```
mixin Flyable {
  void fly() => print('Đang bay');
}
class Bird with Flyable {}
```

Extension Methods

```
extension StringX on String {
  String capitalize() =>
    '${this[0].toUpperCase()}${substring(1)}';
}
print('hello'.capitalize()); // Hello
```

Abstract & Implements

```
abstract class Shape {
  double area();
}
class Circle implements Shape {
  final double r;
  Circle(this.r);
  @override double area() => pi * r * r;
}
```

Async/Await

Futures

```
Future<String> fetchData() async {
  var res = await http.get(uri);
  return res.body;
}
```

Streams

```
Stream<int> count(int n) async* {
  for (var i = 0; i < n; i++) {
    yield i;
  }
}
```

Xử Lý Lỗi

```
try {
  var data = await fetchData();
} on HttpException catch (e) {
  print('HTTP error: $e');
} catch (e) {
  print('Error: $e');
}
```

Collections

Thao Tác List

```
var nums = [1, 2, 3];
nums.add(4);
nums.where((n) => n > 2); // [3, 4]
nums.map((n) => n * 2); // [2,4,6,8]
var sorted = nums..sort();
```

Thao Tác Map

```
var m = {'a': 1, 'b': 2};
m['c'] = 3;
m.containsKey('a'); // true
m.entries.map((e) => '${e.key}=${e.value}');
```

Spread & Collection If/For

```
var all = [0, ...nums];
var nav = ['Home', if (isAdmin) 'Admin'];
var sq = [for (var i in nums) i * i];
```

Null Safety

Kiểu Nullable

int?	int nullable (có thể null)
int!	int không nullable (không bao giờ null)
!	Toán tử khẳng định null
?.	Truy cập null-aware
??	Mặc định nếu null
??=	Gán nếu null
late	Khởi tạo trì hoãn

Ví Dụ Null Safety

```
String? name; // nullable
int len = name?.length ?? 0;
late final String title; // đặt trước khi dùng
name ??= 'default'; // gán nếu null
```

Dart Tham Khảo Nhanh

Các Pattern Thường Gặp

Enum Có Giá Trị

```
enum Color {  
  red('FF0000'), green('00FF00');  
  final String hex;  
  const Color(this.hex);  
}
```

Records & Destructuring

```
(String, int) userInfo() => ('Alice', 30);  
var (name, age) = userInfo();  
print('$name is $age');
```

Sealed Classes

```
sealed class Shape {}  
class Circle extends Shape { final double r; Circle(this.r); }  
class Rect extends Shape { final double w, h; Rect(this.w,  
this.h); }
```