

DART 빠른 참조

타입, 함수, 클래스, 미동기, null 안전성, 컬렉션

기본
Hello World <pre>void main() { print('Hello, Dart!'); }</pre>
변수 <pre>var name = 'Dart'; // type inferred String lang = 'Dart'; // explicit type final pi = 3.14; // runtime constant const max = 100; // compile-time constant</pre>
문자열 보간 <pre>var name = 'World'; print('Hello, \$name!'); print('1 + 1 = \${1 + 1}');</pre>
타입
내장 타입 <ul style="list-style-type: none">int 64비트 정수double 64비트 부동소수점num int와 double의 상위 타입String UTF-16 문자열bool true 또는 falseList 순서 있는 컬렉션 (배열)Set 순서 없는 고유 컬렉션Map 키-값 쌍dynamic 모든 타입의 정적 검사 비활성화void 반환값 없음
타입 검사 & 캐스팅 <pre>if (obj is String) print(obj.length); var s = obj as String; // cast print(obj.runtimeType); // runtime type</pre>
함수
함수 문법 <pre>int add(int a, int b) => a + b; void greet({required String name}) { print('Hello, \$name'); }</pre>
매개변수 <ul style="list-style-type: none">int f(int a) 필수 위치 매개변수int f([int a = 0]) 기본값이 있는 선택적 위치 매개변수f({required int a}) 필수 명명 매개변수f({int a = 0}) 기본값이 있는 선택적 명명 매개변수
클로저 & 테이아웃 <pre>var square = (int n) => n * n; [1, 2, 3].map((e) => e * 2); [1, 2, 3].forEach(print); // tearoff</pre>
제어 흐름
조건문 <pre>if (x > 0) { print('pos'); } else if (x == 0) { print('zero'); } else { print('neg'); } var result = x > 0 ? 'pos' : 'neg';</pre>
반복문 <pre>for (var i = 0; i < 5; i++) { } for (var item in list) { } while (x > 0) { x--; } do { x--; } while (x > 0);</pre>
Switch & 패턴 매칭 <pre>switch (color) { case 'red': print('R'); break; case 'blue': print('B'); break; default: print('?'); }</pre>
클래스
클래스 정의 <pre>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)); }</pre>
명명 & 팩토리 생성자 <pre>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']); }</pre>
상속 <pre>class Animal { void speak() {} } class Dog extends Animal { @override void speak() => print('Woof!'); }</pre>
믹스인 & 확장
믹스인 <pre>mixin Flyable { void fly() => print('Flying!'); } class Bird with Flyable {}</pre>
확장 메서드 <pre>extension StringX on String { String capitalize() => '\${this[0].toUpperCase()}\${substring(1)}'; } print('hello'.capitalize()); // Hello</pre>

추상 & 구현
<pre>abstract class Shape { double area(); } class Circle implements Shape { final double r; Circle(this.r); @override double area() => pi * r * r; }</pre>

Async/Await
Future <pre>Future<String> fetchData() async { var res = await http.get(url); return res.body; }</pre>

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

에러 처리
<pre>try { var data = await fetchData(); } on HttpException catch (e) { print('HTTP error: \$e'); } catch (e) { print('Error: \$e'); }</pre>

컬렉션
List 연산 <pre>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();</pre>
Map 연산 <pre>var m = {'a': 1, 'b': 2}; m['c'] = 3; m.containsKey('a'); // true m.entries.map((e) => '\${e.key}=\${e.value}');</pre>

스프레드 & 컬렉션 If/For
<pre>var all = {0, ..., nums}; var nav = 'Home', if (isAdmin) 'Admin'; var sq = [for (var i in nums) i * i];</pre>

Null 안전성
nullable 타입 <ul style="list-style-type: none">(int?) nullable int (null 가능)(int) non-nullable int (null 불가)
! null 단언 연산자
? null 인식 접근
?? null이면 기본값
??= null이면 할당
late 지연 초기화

Null 안전성 예시 <pre>String? name; // nullable int len = name?.length ?? 0; late final String title; // set before use name ??= 'default'; // assign if null</pre>

일반 패턴
값이 있는 Enum <pre>enum Color { red('FF0000'), green('00FF00'); final String hex; const Color(this.hex); }</pre>

레코드 & 구조 분해
<pre>(String, int) userInfo() => ('Alice', 30); var (name, age) = userInfo(); print('\$name is \$age');</pre>
봉인 클래스 <pre>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); }</pre>