Dart 리스트와 맵 다루기

데이터 구조의 기초

학습목표

- List와 Map의 기본 개념과 생성 방법 이해
- 데이터 구조의 수정 및 조작 방법 숙달
- 복합 데이터 타입의 처리와 출력 방법 습득
- List와 Map을 활용한 실전 예제 학습

기본 개념 요약

- List는 순서가 있는 객체 컬렉션으로 인덱스를 통해 접근
- Map은 키-값 쌍을 저장하는 연관 배열 구조
- 두 구조 모두 동적으로 크기 조절 가능
- 다양한 데이터 타입 저장 지원
- List와 Map은 Dart에서 가장 널리 사용되는 컬렉션 타입

List 기본 사용법

- List 선언 시 대괄호 `[]` 사용
- 0부터 시작하는 인덱스로 요소 접근
- `add()` 메서드로 새 요소 추가
- `forEach()` 메서드로 모든 요소 순회 가능
- List는 중복된 값을 허용

list_basic.dart

```dart
void main() {
  // String 타입의 리스트 생성
  List<String> months = ['January', 'February', 'March'];

  // 리스트에 요소 추가
  months.add('April');

  // 리스트의 모든 요소 출력
  months.forEach(print);

  // 인덱스를 사용하여 요소에 접근
  print(months[0]); // January 출력

  // 리스트의 길이 확인
  print(months.length); // 4 출력
}
```

List 심화 사용법

- `insert()` 메서드로 특정 위치에 요소 삽입
- `remove()` 메서드로 요소 제거
- `map()` 메서드로 각 요소 변환
- `where()` 메서드로 조건에 맞는 요소 필터링

list_advanced.dart

```dart
void main() {
  List<int> numbers = [1, 2, 3, 4, 5];

  // 특정 인덱스에 값 삽입
  numbers.insert(2, 10); // [1, 2, 10, 3, 4, 5]

  // 값 제거
  numbers.remove(4); // [1, 2, 10, 3, 5]

  // 조건에 따라 필터링
  List<int> evenNumbers = numbers.where((number) => number % 2 == 0).toList();
  print(evenNumbers); // [2, 10]

  // 각 요소를 제곱하여 새로운 리스트 생성
  List<int> squaredNumbers = numbers.map((number) => number * number).toList();
  print(squaredNumbers); // [1, 4, 100, 9, 25]
}
```

list_composite.dart

```dart
void main() {
  // 영화 정보를 담는 Map 생성
  Map<String, dynamic> movie1 = {
    "title": "Star Wars",
    "year": 1977
  };

  Map<String, dynamic> movie2 = {
    "title": "Empire Strikes Back",
    "year": 1980
  };

  // Map을 요소로 가지는 List 생성
  List<Map<String, dynamic>> movies = [movie1, movie2];

  // 첫 번째 영화의 제목 출력
  print(movies[0]['title']); // Star Wars 출력

  // 리스트에 새로운 영화 추가
  movies.add({
    "title": "Return of the Jedi",
    "year": 1983
  });

  // 모든 영화 제목 출력
  movies.forEach((movie) => print(movie['title']));
}
```

Map 기본 사용법

- Map 선언 시 중괄호 `{}` 사용
- 키와 값은 모든 타입 가능
- 키는 중복될 수 없음 (중복 시 덮어씀)
- `forEach()` 메서드로 모든 키-값 쌍 순회 가능

map_basic.dart

```dart
void main() {
  // 정수 키와 문자열 값을 가지는 Map 생성
  Map<int, String> months = {
    1: 'January',
    2: 'February',
    3: 'March'
  };

  // 새로운 키-값 쌍 추가
  months[4] = 'April';

  // Map의 모든 키-값 쌍 출력
  months.forEach((key, value) {
    print('$key: $value');
  });

  // 특정 키에 해당하는 값 출력
  print(months[1]); // January 출력
}
```

Map 심화 사용법

- `containsKey()` 메서드로 특정 키 존재 확인
- `containsValue()` 메서드로 특정 값 존재 확인
- `remove()` 메서드로 키-값 쌍 제거
- `map()` 메서드로 각 값을 변환

map_advanced.dart

```dart
void main() {
  Map<String, int> scores = {
    'Alice': 90,
    'Bob': 80,
    'Charlie': 95
  };

  // 키 존재 여부 확인
  print(scores.containsKey('Alice')); // true 출력

  // 값 존재 여부 확인
  print(scores.containsValue(80)); // true 출력

  // 키-값 쌍 제거
  scores.remove('Bob');
  print(scores); // {Alice: 90, Charlie: 95}

  // 점수를 5점씩 증가시킨 새로운 Map 생성
  Map<String, int> increasedScores = scores.map((key, value) => MapEntry(key, value + 5));
  print(increasedScores); // {Alice: 95, Charlie: 100}
}
```

map_validation.dart

```dart
void main() {
  Map<String, dynamic> data = {
    'name': 'John',
    'age': 30,
    'city': 'Seoul'
  };

  // 키 존재 여부 확인
  if (data.containsKey('name')) {
    print('Name exists: ${data['name']}');
  }

  // 값 존재 여부 확인
  if (data.containsValue('Seoul')) {
    print('Seoul found in values');
  }
}
```

map_complex.dart

```dart
import 'dart:convert'; // JSON 인코딩/디코딩을 위한 import

void main() {
  // JSON 형식의 사용자 데이터
  Map<String, dynamic> user = {
    'id': 1,
    'info': {
      'name': 'Alice',
      'contacts': [
        {'type': 'email', 'value': 'alice@email.com'},
        {'type': 'phone', 'value': '123-456-7890'}
      ]
    }
  };

  // Map을 JSON 문자열로 변환
  String jsonString = jsonEncode(user);
  print(jsonString);

  // JSON 문자열을 Map으로 파싱
  Map<String, dynamic> parsed = jsonDecode(jsonString);
  print(parsed['info']['contacts'][0]['value']); // alice@email.com 출력
}
```

List와 Map 활용 예제: 간단한 쇼핑 카트

- 상품명, 가격, 수량을 저장하는 Map을 List에 담아 쇼핑 카트 구현
- 총 가격 계산 및 상품 목록 출력 기능 구현

shopping_cart.dart

```dart
void main() {
  // 쇼핑 카트 List 생성
  List<Map<String, dynamic>> cart = [
    {'name': 'Apple', 'price': 1000, 'quantity': 3},
    {'name': 'Banana', 'price': 500, 'quantity': 5},
    {'name': 'Orange', 'price': 700, 'quantity': 2},
  ];

  // 총 가격 계산
  int totalPrice = 0;
  cart.forEach((item) {
    totalPrice += item['price'] * item['quantity'];
  });
  print('Total price: $totalPrice'); // Total price: 8900

  // 상품 목록 출력
  print('Shopping Cart:');
  cart.forEach((item) {
    print('${item['name']}: ${item['quantity']} x ${item['price']}');
  });
}
```

요약

- Dart의 List와 Map은 유연하고 강력한 데이터 구조 제공
- 타입 안정성과 동적 확장성을 모두 지원하는 현대적 컬렉션
- JSON 처리와 복잡한 데이터 구조 구현에 적합
- List와 Map을 조합하여 다양한 데이터 처리 가능

END