02

연산자와 제어문

Dart

연산자 알아보기

나누기 연산자 - ~/

- 다트에서 나누기 연산자는 /와 ~/
- ~/ 연산자는 나누기 결과를 정수로 반환

• 나누기 연산자

```
main() {
  int a = 8;
  print('a / 5 = ${a / 5}');
  print('a ~/ 5 = ${a ~/ 5}');
}
```

▶ 실행 결과

```
a / 5 = 1.6
a ~/ 5 = 1
```

연산자 알아보기

타입 확인과 변환 – is, as

- is 연산자는 타입을 확인해 true나 false로 알려 주고, as 연산자는 타입을 바꿔 줍니다.

• 타입 확인과 캐스팅

```
class User {
  void some() {
    print("User...some()...");
  }
}

main() {
  Object obj = User();
  // obj.some();      // 오류

  if (obj is User) {   // 타입 확인, 자동 형 변환
    obj.some();
  }

  Object obj1 = User();
  (obj1 as User).some();   // 명시적 형 변환
}
```

연산자 알아보기

반복해서 접근하기 - ..., ?..

- .. 혹은 ?.. 연산자는 같은 객체를 반복해서 접근할 때 편리하게 사용할 수 있는 캐스케이드 연산자

• User 클래스 선언

```
class User{
  String? name;
  int? age;

  some() {
    print('name: $name, age: $age');
  }
}
```

• 객체 생성과 멤버 접근

```
var user = User();
user.name = 'kkang';
user.age = 10;
user.some();
```

• 캐스케이드 연산자 사용 예

```
User()
  ..name = 'kkang'
  ..age = 30
  ..some();
```

실행 흐름 제어하기

for 반복문에서 in 연산자

- for 문에 범위 연산자인 in을 사용하면 오른쪽에 명시한 컬렉션 타입의 데이터 개수만큼 반복해서 실행

• for 문 사용 예

```
main() {
  var list = [10, 20, 30];
  for(var i = 0; i < list.length; i++) {
    print(list[i]);
  }
}
```

▶ 실행 결과

```
10
20
30
```

• in 연산자로 간소화한 반복문

```
main() {
  var list = [10, 20, 30];
  for(var x in list) {
    print(x);
  }
}
```

실행 흐름 제어하기

예외 던지기와 예외 처리

- 예외를 던지는 throw 문
  - Exception 클래스 이외에 다른 객체 가능

• 예외 던지기

```
some() {
  throw Exception('my exception');
}
```

• 문자열 던지기

```
some() {
  throw 'my exception';
}
```

• 사용자 정의 객체 던지기

```
class User{}
some() {
  throw User();
}
```

실행 흐름 제어하기

예외 던지기와 예외 처리

- try~on~finally 예외 처리
  - try 문에 작성한 코드에서 예외가 발생하면 on 문이 실행
  - finally 문에는 예외와 상관없이 무조건 실행할 코드를 작성

• try~on~finally 예외 처리

```
some() {
  throw FormatException('my exception');
}
main(List<String> args) {
  try {
    print('step1....');
    some();
    print('step2....');
  } on FormatException {
    print('step3....');
  } on Exception {
    print('step4....');
  } finally {
    print('step5....');
  }
  print('step6....');
}
```

▶실행 결과

```
step1....
step3....
step5....
step6....
```

실행 흐름 제어하기

예외 던지기와 예외 처리

• 예외 객체 가져오기

```
some() {
  throw FormatException('my exception');
}
main(List<String> args) {
  try {
    print('step1....');
    some();
    print('step2....');
  } on FormatException catch(e) {
    print('step3....$e');
  } on Exception catch(e) {
    print('step4....$e');
  } finally {
    print('step5....');
  }
  print('step6....');
}
```

• try~catch 예외 처리

```
try {
  some();
} catch(e) {
  print('catch....$e');
}
```

▶실행 결과

```
step1....
step3....FormatException: my exception
step5....
step6....
```

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare