

02

Null Safety

Dart

널 포인트 예외 관리하기

널 안전성이란?

- 널 안전성null safety란 널 포인트 예외NPE: null point exception를 프로그램을 실행하기 전 코드를 작성하는 시점에 점검하는 것을 의미
- NPE 발생 가능성을 컴파일러가 미리 점검해 널에 안전한 코드를 작성

널 포인트 예외 관리하기

널 허용과 널 불허

- 변수를 선언할 때 널 허용(Nullable)과 널 불허(NonNull)로 구분
- 닥트 언어에서 변수는 기본으로 널 불허로 선언
- 널 허용으로 선언하려면 타입 뒤에 물음표 ?를 추가

• 널 불허와 널 허용 변수 선언

```
int a1 = 10;  
int? a2 = 10; 널 허용 변수 선언
```

• 널 대입 차이

```
int a1 = 10;  
int? a2 = 10;  
  
testFun() {  
    a1 = null; // 오류  
    a2 = null;  
}
```

• 모든 타입에 적용되는 널 허용 설정

```
String str1 = null; // 오류  
String? str2 = null;  
  
class User{ }  
User user1 = null; // 오류  
User? user2 = null;
```

널 포인트 예외 관리하기

널 불허 변수의 초기화

- 변수를 선언하면서 초기값을 주지 않으면 자동으로 널로 초기화

• 함수에서 널 불허 지역 변수 초기화

```
int a1; // 오류

class User {
    int a1; // 오류
}

testFun() {
    int a1; // 성공
    a1 = null; // 오류
```

널 불허 변수를 초기화하지 않아서 오류 발생

널 불허 변수에 널을 대입해서 오류 발생

• 초기화하지 않은 널 불허 변수 오류 발생

```
int a1; // 오류
int? a2;
```

• 값을 대입하지 않고 사용한 예

```
testFun() {
    int a1;
    print(a1 + 10); // 오류
}
```

• 값을 대입하고 사용한 예

```
testFun() {
    int a1;
    a1 = 10;
    print(a1 + 10); // 성공
}
```

널 포인트 예외 관리하기

var 타입의 널 안전성

- var로 선언한 변수는 널 허용 여부도 대입하는 값에 따라 컴파일러가 자동으로 결정

• var 타입 변수에 물음표를 붙이면 오류

```
main() {  
    var a1 = 10;  
    var a2 = null;  
    var a3;  
    var? a4 = null; // 오류  
}
```

• var 타입 변수의 널 허용 여부

```
var a1 = 10;    // int  
var a2 = null;  // dynamic  
var a3;         // dynamic  
  
testFun() {  
    a1 = 20;  
    a1 = null;  // 오류  
  
    a2 = 20;  
    a2 = "hello";  
    a2 = null;  
  
    a3 = 20;  
    a3 = "hello";  
    a3 = null;  
}
```

널 포인트 예외 관리하기

var 타입의 널 안전성

• var 타입 변수에 타입이 정의된 변수 대입하기

```
int no1 = 10; // 널 불허
int? no2;     // 널 허용

var a1 = no1; // int로 결정
var a2 = no2; // int?로 결정

testFun() {
    a1 = 20;
    a1 = null; // 널 불허 변수에 널을 대입해서 오류

    a2 = 20;
    a2 = "hello"; // int? 타입에 문자열을 대입해서 오류
    a2 = null;
}
```

널 포인트 예외 관리하기

dynamic 타입의 널 안전성

- dynamic 타입은 모든 타입의 데이터를 대입할 수 있으므로 널을 허용하는 Nullable도 포함

```
• dynamic 타입의 널 허용

dynamic a1 = 10;
dynamic a2;
dynamic? a3;

testFun() {
    a1 = null;
    a2 = null;
    a3 = null;
}
```

널 포인트 예외 관리하기

널 안전성과 형 변환

- Nullable은 NonNull의 상위 타입

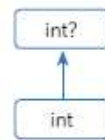


그림 4-2 널 허용과 널 불허 관계

• 자동 형 변환

```
int a1 = 10;
int? a2 = 10;

main() {
    a1 = a2; // 오류
    a2 = a1; // 성공
}
```


널 포인트 예외 관리하기

널 안전성과 형 변환

- 다트에서 명시적 형 변환 연산자는 as

• 명시적 형 변환

```
int a1 = 10;  
int? a2 = 20;  
  
main() {  
  a1 = a2 as int;  
  print("a1: $a1, a2: $a2");  
}
```

▶ 실행 결과

a1: 20, a2: 20

널 포인터 예외 관리하기

초기화를 미루는 late 연산자

- late는 초기화를 미루는 연산자

• 초기화 미루기

```
int a1;      // 컴파일 오류  
late int a2; // 성공
```

• 값을 대입하고 사용하기

```
late int a2; // 성공  
  
main() {  
    // print('${a2 + 10}'); // 주석을 해제하면 실행 오류  
    a2 = 10;  
    print('${a2 + 10}'); // 성공  
}
```

▶ 실행 결과

a2: 20

널 안전성 연산자

널인지 점검할 때 - ! 연산자

- 변수 이름 뒤에 ! 연산자를 추가하면 이 변수값이 널일 때 런타임 오류가 발생

• 함수 호출문에 ! 연산자 사용

```
int? some(arg) {  
    if ( arg == 10 ) {  
        return 0;  
    } else {  
        return null;  
    }  
}  
  
main() {  
    int a = some(10)!;  
    print('a : $a');    // a : 0  
    int b = some(20)!;  // some() 함수가 널을 반환하므로 런타임 오류  
    print('b : $b');  
}
```

• 널인지 점검

```
int? a1 = 20;  
  
main() {  
    a1!;  
    a1 = null;  
    a1!;    // 런타임 오류  
}
```

널 안전성 연산자

멤버에 접근할 때 - ?. , ?[] 연산자

- 널 허용 객체나 리스트의 멤버에 접근할 때는 ?.나 ?[] 연산자를 사용

• 널 리스트에 인덱스로 접근할 때 null 반환하기

```
main() {  
    List<int>? list = [10, 20, 30];  
    print('list[0] : ${list?[0]}');  
    list = null;  
    print('list[0] : ${list?[0]}');  
}
```

▶ 실행 결과

```
list[0] : 10  
list[0] : null
```

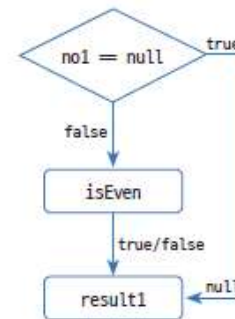


그림 4-3 ?. 연산 로직

• 널 허용 객체의 멤버에 접근하기

```
String? str = "hello";  
  
main() {  
    str.isEmpty; // 오류  
}
```

• 널 객체의 멤버에 접근할 때 null 반환하기

```
main() {  
    int? no1 = 10;  
    bool? result1 = no1?.isEven;  
    print('result 1 : $result1');  
  
    no1 = null;  
    bool? result2 = no1?.isEven;  
    print('result 2 : $result2');  
}
```

▶ 실행 결과

```
true  
null
```

널 안전성 연산자

값을 대입할 때 - ??= 연산자

- 널 허용 변수에 널이 아닌 값만 대입하고 싶다면 ??= 연산자를 사용

• 널일 때 대입하지 않기

```
main() {  
    int? data3;  
    data3 ??= 10;  
    print('data3 : $data3');  
    data3 ??= null;  
    print('data3 : $data3');  
}
```

▶ 실행 결과

data3 : 10
data3 : 10

널 안전성 연산자

값을 대체할 때 - ?? 연산자

- 널 허용 변수가 널일 때 대체할 값을 지정하고 싶다면 ?? 연산자를 사용

• 널일 때 값 대체하기

```
main() {  
    String? data4 = 'hello';  
    String? result = data4 ?? 'world';  
    print('result : $result');  
  
    data4 = null;  
    result = data4 ?? 'world';  
    print('result : $result');  
}
```

▶ 실행 결과

result : hello
result : world



감사합니다

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



윌리엄 셰익스피어
William Shakespeare