

02

# 변수와 타입

Dart

# 데이터 타입

- 모든 변수는 객체

• int 타입 사용 예

```
int? no = 10;

main() {
  bool? data = no?.isEven;
  no = null;
  Object? obj = no;
}
```

## 다트의 타입 클래스

표 4-1 다트에서 제공하는 데이터 타입 클래스

라이브러리	타입 클래스	데이터
dart:core library	bool	true, false
	double	실수
	int	정수
	num	숫자(double과 int의 상위 클래스)
	String	문자열
dart:typed_data	ByteData	바이트

# 데이터 타입

---

## 문자열 표현하기

- 문자열 데이터를 다룰 때는 String 클래스
- 작은따옴표('hello')나 큰따옴표("hello"), 또는 삼중 따옴표("""hello""")

### • 문자열 표현법

```
main() {  
    String data1 = 'hello';  
    String data2 = "world";  
    String data3 = """  
        hello  
        world  
    """;  
    String data4 = """"  
        hello  
        world  
    """;  
}
```

# 데이터 타입

## 문자열 표현하기

- 두 문자열이 같은지 비교할 때는 다음처럼 = 연산자
- 문자열에 동적인 결과를 포함하는 것을 **문자열 템플릿**

### • 문자열 비교

```
main() {  
    String str1 = 'hello';  
    String str2 = 'hello';  
  
    print(str1 == str2);  
}
```

▶ 실행 결과

true

### • 문자열 템플릿

```
main() {  
    int no = 10;  
    String name = 'kkang';  
  
    String myFun() {  
        return 'kim';  
    }  
  
    print('no : $no, name : $name, 10 + 20 : ${10 + 20}, myFun() : ${myFun()}');  
}
```

▶ 실행 결과

no : 10, name : kkang, 10 + 20 : 30, myFun() : kim

# 데이터 타입

## 형 변환하기

- 변수가 모두 객체이므로 **형 변환** type casting도 자동으로 이뤄지지 않습니다.

### • 타입 오류

```
main() {  
    int n1 = 10;  
    double d1 = 10.0;  
  
    double d2 = n1; // 오류  
    int n2 = d1;    // 오류  
}
```

### • int와 double 형 변환

```
main() {  
    int n1 = 10;  
    double d1 = 10.0;  
  
    double d2 = n1.toDouble(); // 성공  
    int n2 = d1.toInt();       // 성공  
}
```

### • int와 double 형 변환

```
main() {  
    int n1 = 10;  
    double d1 = 10.0;  
  
    double d2 = n1.toDouble(); // 성공  
    int n2 = d1.toInt();       // 성공  
}
```

# 상수 변수 - const, final

## 컴파일 타임 상수 변수 - const

- 컴파일 타임 상수 변수는 const 예약어로 선언
- const 변수는 톱 레벨이나 함수 내에 지역 변수로 선언할 수 있지만, 클래스에 선언할 때는 static 변수로만 선언

• 초깃값 없이 const 변수 선언 오류

```
const String data1; // 오류

class User {
    static const String data2; // 오류

    void some() {
        const String data3; // 오류
    }
}
```

• 상수 변수값 바꾸기 오류

```
const String data1 = 'hello';

class User {
    static const String data2 = 'hello';

    void some() {
        const String data3 = 'hello';

        data1 = 'world'; // 오류
        data2 = 'world'; // 오류
        data3 = 'world'; // 오류
    }
}
```

# 상수 변수 - const, final

## 런타임 상수 변수 - final

- const와 마찬가지로 값을 바꿀 수 없지만, 초기값을 대입하는 시점이 꼭 선언문이 아닐 수도 있습니다.

```
• final 변수

final int no1; // 초기값이나 이후에 값을 대입하지 않아서 오류

class MyClass {
    final int no2;
    MyClass(this.no2);

    void some() {
        final no3;
        no3 = 10;
        no3 = 20; // 값을 바꿀 수 없어서 오류
    }
}
```

이후에 값을 대입하므로 초기화  
하지 않아도 정상

# 상수 변수 - const, final

## 상수 변수와 문자열 템플릿

- const 예약어로 선언한 String 타입 상수 변수에 문자열 템플릿으로 값을 대입할 때는 템플릿 내부에도 컴파일 타임 상수를 사용

```
• 문자열 템플릿에 상수 변수 사용

main() {
    String s1 = 'hello';
    const String s2 = 'world';
    final String s3 = 'helloworld';

    String s4 = '$s1, $s2';
    const String s5 = '$s2';
    const String s6 = '$s1, $s2, $s3'; // 오류
    final String s7 = '$s1, $s2, $s3';
}
```



# var와 dynamic 타입

## 타입 유추 - var

- 컴파일러가 해당 변수에 대입하는 값을 해석해 타입을 추측

### • 초깃값의 타입으로 선언

```
main() {  
  var no = 10;  
  no = 20;  
  no = 'hello'; // 오류  
}
```

### • dynamic 타입으로 선언

```
main() {  
  var no2;  
  no2 = 10;  
  no2 = 'hello';  
  no2 = true;  
}
```

var로 선언하면서 초기화하지 않았으므로 여러 타입 대입 가능

# var와 dynamic 타입

---

## 모든 타입 지원 - dynamic

- 변수에 타입 대신 dynamic을 사용하면 모든 타입의 데이터를 대입

• dynamic 타입으로 선언

```
main() {  
    dynamic data = 10;  
    data = 'hello';  
    data = true;  
}
```



# 감사합니다

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



윌리엄 셰익스피어  
William Shakespeare