



# Dart Chapter 1

## Basic

- **We must use ; end of code.**
- 모든 기능이 동작하려면 main 함수에 넣어야 한다.

## 변수 만들기

1. 변수의 타입을 굳이 지정해줄 필요가 없다.

```
var name = 'name'
```

알아서 변수의 type을 추측해준다.

update 가능하며 그러나 기존 type과 동일한 type으로 update 해야 한다.

혹은 명시적으로 변수의 타입을 지정해준다. `String name = 'name'`

| `String name = 'name' = var name = 'name'`

- `var name = 'name'`  
→ 함수나 메소드 내부에 **지역 변수**를 선언할 때 주로 사용(**언제나 권장사항**)
- `String name = 'name'`  
→ class에서 변수나 property를 선언할 때 type을 지정해준다.

## Null safty

| **어떤 변수, 혹은 데이터가 null이 될 수 있음을 명시하는 것이다.**

명시한다면 Dart가 도와줄거야~ 개발자가 null 값을 참조하지 않도록 해준다.

**만약 코드에서 null 값을 참조했다면 runtime error 발생**

Dart는 null을 참조할 수 없게 safty를 마련해놓음

`null` = 부재

기본적으로 모든 변수는 **non-nullable**이다.

null을 해주기 위해서는 “?”를 붙여준다.

```
// string으로 고정되는 것이 아니라 null값도 될 수 있게 하고 싶다?
// -> 변수 타입 뒤에 ? 붙이기
// 기존에는 string으로 지정되면 string type으로만 update 가능했었다.
String? nico = 'nico';
nico = null;
```

- 변수를 사용하기 전에 null값인지 check해야 한다.

```
// nico가 null이 아니면 isEmpty 속성을 달라고 요청
nico?.isEmpty
```

## Dynamic type

**여러가지** 타입을 가질 수 있는 변수에 지정해준다.

사용하는 것이 추천되진 않음, 그러나 종종 유용

- 변수가 어떤 타입일지 알기 어려운 경우에 사용한다.(flutter, json에서 유용)

`var name;` 이런 식으로 무언가를 지정해주지 않으면 **dynamic**으로 지정된다.

**Dynamic은 정말 필요할 때 써야해!**

```
// 아래의 코드가 모두 가능하게 됨
var name; //or dynamic name
name = 'name'; //string
name = 12; //double
name = true; //boolean

// dynamic이라고 선언했을 때
dynamic name;

// dynamic이 type 체크 이후 모든 type에 관한 속성을 제공해준다.
if (name is String){
  ~String에 관한 여러 속성 등장~
  ~ex) isEmpty()
}

if (name is int){
  ~int에 관한 여러 속성 등장~
  ~ex) isOdd, isEven
}
```

## Final Variables

이 두 가지는 모두 선언 했을 때 나중에 수정 가능한 변수들이 생성된다.

```
String name = 'name' = var name = 'name'
```

**Question.** 한 번 정의된 변수를 다시 바꿀 수 없게 하려면?

→ **final 사용**, `final name = 'name'`, 이 변수는 수정 불가하다! (== const)

## late

`late` = **Modifier after final and var**, final과 var 앞에 붙을 수 있는 수식어

`late final String name` : 초기 데이터 없이 변수를 선언할 수 있다.

- late를 사용하면 먼저 변수를 만들고 나중에 데이터를 넣을 수 있게 해준다.
- Prevent mistakes.
  - If try to access late variable, Dart notice you can't access it until late variable has data
  - 만약 late 변수에 값이 할당 되지 않는다면 Dart는 late 변수에 관한 접근을 막는다
  - like null safty

```
// data없이 변수 선언
late final String name;

// do something, go to api(API 요청으로 데이터를 받고 그 데이터를 나중에 변수에 넣는다.)
name = 'name';
```

## Time line

1. Make late variables
2. Request to API (Hit the API)
3. Data comes back
4. Data on late variables

## Constant Variables

`const` on Dart. Not same with `const` of JavaScript and TypeScript

`const` of JavaScript and TypeScript are more simliar with Dart's `final`

`const` → *Make compile-time constant*

`const` 는 compile-time에 “알고 있어야” 하는 값이다.

Hard coding. When you compile, compiler must know data of const variables

```
//operate same with final -> can't modify
const name = 'name';
```

## Recap(review) - Make Variables

- 변경 가능 한 변수 - `var`
  - 주로 var을 사용하는 것이 권장됨
    - 지역 변수를 선언할 때 var 권장

```
var name = 'name' or String name = 'name' //Can modify if keep data type
```

- 변수에 값을 딱 한 번 할당 - `final`

```
final name = 'name' //값을 재할당하지 못함
```

- 어떤 데이터가 들어올지 모를 때 - `dynamic`
  - 조심스럽게 사용해야 하는 type
  - 변수를 사용하기 전에 type을 체크하는 것이 필요하다.

```
dynamic name; // it can be string, int, double etc..
```

- compile할 때 알고 있어야 할 변수(상수) 선언 - `const`
  - 수정이 불가함
  - `const` VS `final`
    - `final` can be created by the runtime → 앱 실행 중에 `final` 변수는 생성될 수 있다.

- 사용자가 앱을 동작하면서 변수를 만들 수 있다라는 것.
- const는 위와 같은 내용이 불가.

```
// api key 같이 앱스토어에 올리기 전에 알고 있어야할 변수
const api_key = 'jdhfkjahdskjfhui'
```

- null 값을 참조할 수 없게 막는 것 - `null safety`
  - 기본적으로 dart의 변수들은 non-nullable, null이 될 수 없다.
  - 그러나 null은 유용함
    - 어떤 변수가 null 될 수도 있다고 알려주기 위해 “?”를 붙여준다

```
String? name = 'name' ;
name = null; (허용)

//dart's checking
name?.isEmpty;

//or

if (name != null){
    name.isEmpty();
}
```

- 미리 변수를 생성하고 나중에 data를 할당 - `late`
  - final, var, string 등 앞에 붙이기
  - 어떤 type의 변수가 아직 여기에 올지 몰라!
  - **API에서 data를 가져올 때 유용하다.**

```
late final String name;
//나중에 data를 할당해 주어야 변수를 사용할 수 있음
```