



# Dart Chapter 2

## <basic> Dart의 자료형

- String = 문자열 (' ', " ")
- Boolean = True/False
- Int = 정수형
- double = 실수(소수점 존재)
- num = Could be int or double

→ 대부분 Object, Class 로 구성되어 있음(객체지향적)

→ 실제로 자료형 안에 무엇이 들어있는지 알 수 있다.

→ 변수 지정 이후 `.` 을 통해 해당 type이 가지는 모든 `method` 를 확인해볼 수 있다.

## List

이미 int라고 지정된 list에 String을 추가할 수는 없음

`var` 로 선언하는 것이 항상 권장됨

리스트 마지막에 `.` 붙이는 거 잊지말기!!

리스트 마지막 요소를 넣고 `.` 를 써주면 알아서 multiline으로 바꿔준다.(중요)

```
var numbers = [1, 2, 3];
List<int> numbers = [1, 2, 3];

numbers.add(4);
numbers.first; //return first one
numbers.last; // return last one
numbers.clear; // clear list
```

- collection if / collection for을 지원한다.

### **collection if**

if로 존재할 수도, 안할 수도 있는 요소를 가지고 list를 생성할 수 있다.

```
var giveMeFive = true;

var numbers = [1,
                2,
                3,
                if (giveMeFive) 5,]; // giveMeFive == True라면
```

## String Interpolation

→ text에 변수를 추가하는 방법

text에서 내가 변수를 삽입하고 싶은 부분에 `$variable` 를 넣는다.

text 안에서 변수를 계산하고 싶을 때는 `${variable + 2}` 같이 중괄호를 써준다.

```
var name = 'name';
var greeting = 'hello everyone, my name is ';

// I want to attach name to greeting
// then do like this

// 변수가 이미 존재할 때.
var greeting = 'hello everyone, my name is $name, nice to meet you!';

// 변수를 계산할 때.
var age = 10;
var greeting = 'hello everyone, my name is $name and I am ${age + 2}!';
```

## Collection for

```
var oldFriends = ['A', 'B'];
var newFriends = ['C',
                  'D',
                  'E',]

// Now append oldFriends list to newFriends
var newFriends = ['C',
                  'D',
                  'E',
                  for (var friend in oldFriends) "$friend",]
// list에 있는 요소를 하나씩 가져와서 넣어준다.
```

## Maps

파이썬의 dictionary와 유사한 형태

```
// 자료형을 한 번 주목해서 봐보자
// compiler가 알아서 type을 지정해주었을텐데 과연 어떤 type이 되어있을까?

Type: Map<String, Object>
-> key, value로 구성된 Map을 만들었는데 key는 String이고 value는 Object이다.
```

Q. What is Object?

A. Everything comes from Object in Dart. Object basically can be any type.  
-> 어떠한 type이든 될 수 있다. 따라서 value에는 내가 원하는 어느 것이든 올 수 있다.  
-> 혹은 value의 type을 모두 String으로 바꾼다.

```
var player = {
```

```
'name': 'nico',
'xp': 19.99,
'superpower': false,};
```

- Create Empty Map
  - var을 통해 compiler가 지정해주기를 기다리지 않고 내가 원하는 type을 명시해줄 수도 있다.

```
// type Map<key's type, value's type>
Map<int, bool> player = {
    1: true,
    2: false,
    3: true,
};
```

- 원하는 어떤 key든 value든 명시해줄 수 있다.

```
Map<List<int>, bool> player = {
    [1, 2, 3, 5]: true, };

// 이런 형태를 추천하진 않음. 이렇게 할거면 class 쓰기
List<Map<String, Object>> players = [ {
    'name': 'nico',
    'xp': 19.99,
    'superpower': false},
    {
    'name': 'nico2',
    'xp': 19.9999,
    'superpower': false},] //이런 식으로 여러 개의 맵을 넣어 리스트 생성도 가능
```

- Map의 method와 property
  - addall, isEmpty, addEntries, contain, update..

## Sets

*Sets on Dart very east to create.*

```
var numbers = {1, 2, 3, 4};
Set<int> numbers = {1, 2, 3, 4};
```

### *Difference between Set and List*

- Set = 집합, 중복 불허, 중괄호 사용 { }

```
var numbers = {1, 2, 3, 4};
numbers.add(1);
numbers.add(1);
numbers.add(1);
```

이렇게 1을 여러 번 추가해도 나중에 출력하면 unique한 값이 출력됨  
(이미 존재하기 때문에 추가되지 않음)

- List = 리스트, 중복 허용, 대괄호 사용 [ ]