Dart

 Flutter 개발을 위한 언어

Variables

분류	메모
var/type	• 재할당 가능 • 보통 var 권장(지역변수 등에서) • type 은 class의 property에 권장
final	• 재할당 불가능
dynamic	• 자주 쓰는 것 권장x • 다양한 타입 지정 가능 • if(___ is type) 하면 해당 타입의 메서드 사용 가능.
const	• 컴파일 할 때 값을 알고 있는 것 • api키 등. • 수정 불가능(재할당) • final과의 차이 : final은 런타임 중에 만들어 질 수 있다.

- 사용자가 앱을 사용하는 중에 할당 가능
- const는 앱스토에 올리기 전에 컴파일러 인식해야

nullsafety

- 기본적인 dart변수는 Non-nullable
- null이 되기 원한다면? => '?'추가.

```
String? Name = 'ddd'
```

- If(name.isEmpty) ⇒ name?.~~

late

- 아직 어떤 값이 올지 몰라.
- 나중에 정의하기로 함.
- 그대신 정의하기 전에 사용하려고 하면 경고해줌.
- API가져올 때 일등공신

📌 요약**:** final / dynamic / const / nullsafety / late

Data Types

분류	메모
Basic data types (all 객체)	• String "", '' • bool

- num
    1. int
    2. double

→ 우클릭 go to type definition으로 class 확인가능

Lists

- 사용법(대괄호)
    1. var numbers = [1, 2, 3, 4];
    2. List<int> numbers = [1, 2, 3, 4];
- collection if, collection for 사용가능
    - collection if ：조건에 따라 요소 추가 가능

ex> [1, 2, 3, 4, if(true) 5]

→ 마지막 요소 뒤에 , 붙여주면 예쁘게 세로 정렬

String Interpolation

- text에 변수를 추가하는 방법

ex> 작따, 큰따 상관 없음

```
var name = ‘경희’;
```

```
var age = 20;
```

```
var greeting = ‘My name is $name. Age is ${age+1}.’;
```

ex> function에 사용되는 경우

```
String sayHello(String name, int age, String country){
return 'Helloi $name, you are $age, and you are from $country'
}
```

⇒ 모두 text로 만들어준당

- 주의점, 안에 I’m 같이 따옴포 쓸 땐 앞에 escape 기호 써주기

Collection For

- 사용법

```
var oldFriends = ['nico', 'lynn',];
var newFriends = [
  'lewis',
  'ralph',
  'darren',
  for(var friend in oldFriends) '★ $friend',
];
```

Maps

(JS,TS의 object)

- key : value로 이루어짐
  1. key와 value 내용에 따라 컴파일러가 type 정해줌.

     ```
     var students = {key : value};
     ```

  2. key와 value type 지정 가능

     ```
     Map<int, bool> student ={ key : value };
     ```

     - type은 List일 수도 있다.(복잡한 것들…)
     - 반대로 List의 타입이 Map일 수도

- 그러나 이런 경우엔 class를 더 많이 쓴다.

- 모든 아이템이 unique
- 중괄호/ 대괄호 사용으로 Set / List 구분 가능
  - var numbers = { }; ⇒ Set
  - var numbers = [ ]; ⇒ List
- 혹은 type 지정 가능

```
Set<int> numbers = {}
```

Sets

📌 **요약:** Basic types / List / String interpolation / Collection For / Map / Set

---

Function(1)

소주제

기본 형태

메모

```
ReturnType F_Name(P_Type p_name){
    print("Hello $name nice to meet you");
}
```

fat arrow syntax

- 바로 return만 하는 함수의 경우 화살표 함수 가능

```
String sayHello(String name){
  return "Hello $name nice to meet you"
}
```

⇒

```
String sayHello(String name) =>
"Hello $name nice to meet you";
```

```
num plus(num a, num b) => a+b;
```

=

Named Parameters

- function 일반 parameter ***다수*** 인 경우
  - *(String name, int age, String country …)*
    - Positional Parameter라고 부름

⇒ 파라미터 순서 기억하기 어려움

⇒ 순서 관계 없이 쓰는 법 : Named Parameter

1. 파라미터를 중괄호로 감싼다.

```
String sayHello(
 { String name, int age, String country }
){…}
```

2. 사용 시 순서에 관계 없이 *key:value* 형태로 사용

```
sayHello(name : 'nico', country : 'cuba', age : 19);
```

- Dart = **null** safty
  - 사용자가 parameter를 보내지 않는다면?

    ⇒ default 지정 가능 or Modifier 주기

- default value 지정하기

```
String sayHello( {
  String name = '안뇽',
  int age = 20,
  String country = 'Korea',
 } ){…}
```

→ default주기 싫다면?

- 파라미터 앞 modifier 지정

  ‘required’ ⇒ 값없으면 안 돌아감

```
required String name = '안뇽',
```

- 대괄호로 감싼 parameter는 nullable, optional.

Optional Positinal Parameters

⇒ 잘 안 써

```
String sayHello(
  String name = '안뇽',
  int age = 20,
  [String? country = 'Korea'],
) => 'Hello $name, you are $age and from $country'


void main(){
  var result = sayHello('nico', 20);
  print(result); // Korea나오긴 함
}
```

QQ Operator

(?? , ?=)

- null값을 보내게 하고 싶다면?

```
// String?하면?
// 가능은 하지만 복잡.

String capitalizeName(String? name) {
  if(name != null){
    return name.toUppercase();
  }
  return 'ANON';
}
void main(){
  capitalizeName('nico');
  capitalizeName(null);

}
```

- 삼항연산자 사용 가능

- 더 짧은 방법 : QQ Operator

```
String capitalizeName(String? name) =>
    name?.toUppercase() ?? 'ANON';
```

```
// name이 null일 것도 고려해서 ?
// left ?? right => left가 null이면 right출력


void main(){
  capitalizeName('nico');
  capitalizeName(null);

}
```

📌 요약: **fat arrow syntax / Named Parameters**

Class

소주제

메모

- ...
- ...

📌 요약:
