

# Functions

- void 라는 것은 아무 의미가 없다는 뜻 ?

```
// 함수 옆에 리턴 할 값의 타입을 줘야 함.
String sayHello(String name) {
    return "Hello $name nice to meet you!";
}

String sayHello(String name) =>
    "Hello $name nice to meet you!";
// function의 구조가 return 한 줄 뿐이라면 화살표로 중괄호 없이 작성 가

void main() {
    print(sayHello('nico'));
}

// Hello nico nice to meet you!
```

## named parameter

```
String sayHello(String name, int age, String country) {
    return "Hello $name, you are $age, and you come from $country";
}

void main() {
    print(sayHello('nico', 12, 'korea'));
}
```

- 함수가 여러 파라미터를 필요할 때 위의 코드처럼 넣어주면 된다.
- 하지만 사용자가 각각의 자리에 무엇을 의미하는 지 알 수 없다.
- 이럴 때 named argument를 사용하면 된다.

```
void main() {
    print(sayHello(
        // named argument
        age: 12,
        country: 'cuba',
        name: 'nico'
    ));
}
```

- 이런 식으로 적어주고 함수에도 추가적인 작업이 필요하다.

```
String sayHello({
  String name = 'anon',
  int age = 10,
  String country = 'korea',
}) {
  return "Hello $name, you are $age, and you come from $country";
}

String sayHello({
  required String name,
  required int age,
  required String country
}) {
  return "Hello $name, you are $age, and you come from $country";
}
```

- 함수의 파라미터들을 중괄호로 묶어준다
- 그렇게 된다면 오류가 뜨는데 이것은 dart의 null safety 때문이다
- 위처럼 default값을 주거나
- required modifier를 통해 반드시 값이 필요하다는 것을 알린다.

## Optional Positional Parameters

```
String sayHello(String name, int age, String country) =>
  "Hello $name, you are $age years old from $country";

void main() {
  print(sayHello('nico', 2, 'cuba')); // positional parameter 방식으로 함수 호출
}
```

- positional parameter를 사용하면서 parameter를 optional하게 가져가려면 어떻게 해야할까?
- 위 코드에서 country를 optional하게 바꿔보겠다.

```
String sayHello(
  String name,
  int age,
  [String? country = 'cuba'] // 대괄호, 타입 뒤 ?, default값 지정
) =>
  "Hello $name, you are $age years old from $country";

void main() {
  print(sayHello('nico', 2)); // 마지막 argument를 보내지 않고도 함수를 호출
}
```

```
// Hello nico, you are 2 years old from cuba
```

- 대괄호로 감싸주고
- ?를 통해 null이 될 수도 있다는 표시를 해주고
- default value를 준다

## QQ Operator

- 소문자를 입력하면 대문자로 바꿔주는 함수를 만드려고 한다.
- 그런데 사용자가 null 값도 입력 할 수 있도록 하고싶다.

```
String capitalizeName(String? name) => name.toUpperCase();

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

- null 입력을 위해 name의 타입에 ?를 넣어주면 toUpperCase에서 오류가 난다.
  - null일지도 모르는 값에 해당 함수를 호출 할 수 없기 때문

```
// null이 아닌 경우와 null인 경우를 나눠서 리턴 해주면 된다.
String capitalizeName(String? name) {
  if (name != null) {
    return name.toUpperCase();
  }
  return 'ANON';
}

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

- 위의 코드를 간략하게 작성 할 수 있다.

```
String capitalizeName(String? name) =>
  name != null ? name.toUpperCase() : 'ANON';

void main() {
  capitalizeName('nico');
}
```

```
capitalizedName(null);
}
```

- QQ operator를 사용하여 더 짧게 사용 할 수 있다.

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

// QQ operator란?
// left ?? right
// left가 null이 아니면 left를 출력
// left가 null이면 right를 출력

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

- 추가적인 것

```
void main() {
    String? name;           // null일지도 모르는 name이라는 변수
    name ??= 'nico';         // ??= 는 name이 null이면 nico라는 문자를 넣어주라는
    print(name);             // name이 원래 null이어서 nico라는 문자가 들어감
    name = null;             // 다시 name을 null로 바꿔주고
    name ??= 'another';      // 해당 코드가 실행되면 name에 another가 들어간다
    print(name);
}

// nico
// another
```

## Typedef

- 자료형이 헛갈릴 때 도움이 될 alias을 만드는 방법

```
List<int> reverseListOfNumbers(List<int> list) {
    var reversed = list.reversed;
    return reversed.toList();
}

void main() {
    reverseListOfNumbers([1,3,4,5]);
}
```

```
// typedef로 list integer를 선언하여 사용 할 수 있다.
// 어떻게 활용 될 지는 감이 안온다

typedef ListOfInts = List<int>;

ListOfInts reverseListOfNumbers(ListOfInts list) {
    var reversed = list.reversed;
    return reversed.toList();
}

void main() {
    reverseListOfNumbers([1,3,4,5]);
}
```

- map을 typedef로 만드는 법
- 사용자가 함수에게 주는 data보다 더 구체적으로 구조를 만들고 싶음

```
typedef UserInfo = Map<String, String>;

String sayHi(UserInfo userInfo) {
    return "Hi ${userInfo['name']}";
}

void main() {
    print(sayHi({"name":"nico"}));
}
```