

Variables

```
void main() {  
  print('hello world');  
}
```

- Dart에서 세미콜론(;) 필수
 - autofomatter 없음
 - cascade operator 라는 끝내주는 기능 때문임
- 코드에 반드시 main을 넣어줘야 함

변수 만들기

var

```
void main() {  
  var name = '재형';  
  name = '재형아님';  
}
```

- 변수 수정가능
- 같은 타입이어야 가능
- 관습적으로 함수나 메소드 내부에 지역 변수 선언할 때 사용

명시적으로 타입 지정해주기

```
void main() {  
  String name = '재형';  
}
```

- class에서 변수나 property를 선언할 때에는 타입 지정

dynamic

- 변수에 타입이 없음

- 되도록 안쓰는 게 좋음

```
void main() {
  var name; // == dynamic name;
  name = '재형';
  name = 12;
  name = false;

  if(name is String){
    // 이 안에서는 name 변수가 String 타입으로 취급
  }
  if(name is int){
    // 이 안에서는 name 변수가 int 타입으로 취급
  }
}
```

null safety

- 개발자가 null 값을 참조할 수 없도록 하는 것
- 변수 뒤 ? 표시
- dart 변수는 기본적으로 nullable이 아니다

```
// name 변수에 string 아니면 null이 들어가도록 하려면
void main() {
  String? name = 'jason';
  name = null;

  // 물음표를 붙여주면 name의 값에 null을 넣어줄 수 있다.

  name?.isNotEmpty; // 1
  if(name != null){ // 2
    name.isNotEmpty;
  }
  // 1과 2는 name이 null이 아닌 경우 isNotEmpty 실행되는 같은 의미를 가진 코드
}
```

final

```
void main() {
  final name = 'jason';
  name = 'nico'; // (X)
}

// const와 같음
// 변수 선언하면 값을 바꿀 수 없음
```

late

- 변수를 먼저 만들고 값을 나중에 넣음

```
void main() {  
  late final String name;  
  // 변수 선언을 먼저하고 api 요청을 통해 받은 데이터를 나중에 넣어주는 등으로 활용 가능  
  // data fetching 에 유용  
  name = 'jason';  
  name = '12'; // X  
}
```

const

- JS의 const와 다르다
- 값을 변경 할 수 없는 점이 final과 같지만
- const는 컴파일 시 값이 결정됨
- final log2 = DateTime.now();
 - 실행 시에 시간이 결정되므로 사용 가능
- const log3 = DateTime.now();
 - 컴파일 시의 시간을 담을 수 없으므로 사용 불가. 실행은 지금이 아니라 내일도 할 수 있음