Chapter 02.코틀린 기초 - 03.함수

함수

- 코틀린에서 함수를 선언하는 방법은 다음과 같다

```
fun 함수명(인자:타입, 인자:타입) : 반환 타입 {
  return 결과
}
```

- 함수 선언

```
fun sum(a: Int, b: Int) : Int {
  return a + b
}
```

- 함수 본문은 표현식이 될 수 있다

```
fun sum(a: Int, b: Int) : Int = a + b
```

- 표현식(expression), 구문(statement)의 차이는 표현식은 값을 만들어 낼 수 있고 문은 값을 만들어내지 않는다

- 반환 타입이 없어도 컴파일러가 분석하여 적절한 타입을 추론한다

```
fun sum(a: Int, b : Int) = a + b
```

- 중괄호가 있는 경우에는 생략이 불가능하다

```
fun sum(a: Int, b: Int) {
  return a + b // 컴파일 에러
}
```

- 반환형이 없는 경우 Unit이 반환 타입이 된다

```
fun printSum(a: Int, b: Int): Unit {
    println("$a 더하기 $b = ${a + b}")
}
```

- Unit은 생략 가능

```
fun printSum(a: Int, b: Int) {
    println("$a 더하기 $b = ${a + b}")
}
```

- 디폴트 파라미터

```
fun greeting(message:String = "안녕하세요!!") {
  println(message)
}

fun main () {
  greeting() // 안녕하세요!!

  greeting("Hello, World") // Hello, World
}
```

- 자바는 디폴트 파라미터 기능이 없다

```
class Message {

  public void greeting(String message) {
    if (message == null || message.length() == 0) {
      System.out.println("안녕하세요!!");
      return;
    }
    System.out.println(message);
  }

  public static void main(String[] args) {
    new Message().greeting(null);

    new Message().greeting("Hello, World");
  }
}
```

- 네임드 아규먼트

```
fun log(level: String = "INFO", message: String) {
    println("[$level]$message")
}

fun main() {
    log(message = "인포 로그")

    log(level = "DEBUG", "디버그 로그")

    log("WARN", "워닝 로그")

    log(level = "ERROR", message = "에러 로그")
}
```

- 자바는 가독성을 위해 설명 변수나 주석을 사용하는 경우도 있다. (IntelliJ에선 인자명을 표시해줌)

```
public class Logger {

    public void log(String level, String message) {
        if (level == null || level.length() == 0) {
            level = "INFO";
        }
```

```
        System.out.printf("[%s]%s%n", level, message);
    }

    public static void main(String[] args) {
        Logger logger = new Logger();

        logger.log("INFO", "인포 로그");

        logger.log(/* level = */"DEBUG", "디버그 로그");

        logger.log("WARN", "워닝 로그");

        String level = "ERROR"; // 설명 변수
        logger.log(level, "에러 로그");
    }
}
```

- 코틀린의 함수는 자바의 메서드보다 간결하게 작성이 가능하고 네임드 아규먼트나 디폴트 파라미터 기능은 굉장히 유용한 기능이다