

[코틀린 입문] 1주차 주간 질문

파일을 업로드하고 설문지를 제출하면 Google 계정과 연결된 이름과 사진이 기록됩니다. yenarue@gmail.com님이 아니신가요? [계정 전환](#)

* 필수항목

이메일 주소 *

yenarue@gmail.com

제출자명 *

김예나

참여 월/요일/시간대 (ex. 10월/화요/10시) *

온라인 스터디가 없는 스터디의 경우 월만 적어주세요! (ex. 11월)

4월/일요/21시

1. Kotlin의 디자인 철학을 나타내는 4가지 단어와 각 단어의 의미에 대해 이해하신 내용을 설명해주세요. *

- 간결성 (Concise) : boilerplate 코드를 줄인다.
- 안전성 (Safe) : Exception 발생을 줄여 런타임 에러를 줄인다. 런타임 에러가 발생할 가능성이 있는 로직은 문법적인 제약을 걸음으로서 컴파일 에러를 발생시켜 발견하기 힘든 런타임 에러를 지양하도록 한다. (Null-Safety, Smart casts 등...)
- 상호운용성 (Interoperability) : JVM 환경에서 동작가능하게 함으로서 Java와의 호환성을 유지하여 기존 Java 프로젝트에서 Kotlin 코드를 상호 운용할 수 있도록 한다. 이 철학은 타 플랫폼으로도 확장되어 JavaScript 에서도 운용되도록 구현되고 있다.
- 툴 친화성 (Tool-friendly) : 코틀린 언어를 사용하는데에 필요한 도구를 최대한 지원한다. 특히 JetBrains에서 개발된 만큼 JetBrains IDE에서 최상의 도구들을 파워풀하게 지원하고 있다.

응답을 수정 중입니다. 이 페이지의 URL을 공유하면 다른 사용자도 내 응답을 수정할 수 있습니다.

빈 설문지 열기

2. Basic/From Java to Kotlin 단원에서 배운 내용 중 가장 흥미로웠던 부분 1가지를 골라 조금 더 공부해보고 이해하신 내용을 적어주세요. *

Kotlin이 JVM 환경 뿐만이아니라 브라우저 환경(엄밀히는 JavaScript)에서 까지도 상호 운용이 가능하다는 점이 흥미로웠습니다. 관련하여 짧게 알아본 결과, 브라우저 환경에서는 100% 대체가능한 것 같지는 않았습니다. 현업에서 Node.js로 서버를 구현하고 있는 터라 언어 통일이 가능할지도 모른다는 생각이 들었는데 조금 아쉽게 되었으나 $\pi.\pi$ 점점 호환성을 높여갈 것이라 믿으려 합니다 :-)

3. Kotlin의 When이 Java의 Switch에 비해 강력한 이유는 무엇일까요? *

- case, break 문이 줄어들어 간결성이 높아져 가독성이 좋아졌다.
- Java의 Switch와는 달리 Kotlin의 When은 구문(statement)이 아닌 표현식(expression)이므로 변수나 함수에 직접 대입이 가능하다
- Java Switch의 case 에 대응하는 Kotlin When의 branch에는 case보다 좀 더 넓은 의미의 조건문을 작성할 수 있다. value만을 담았던 case와는 달리 branch는 조건문 자체를 담을 수 있다.
- `is` 와 함께 사용하면 Smart Cast가 이루어져 안전성이 더 높다
- argument 없이 쓰일 수 있기 때문에 switch보다 좀 더 사용범위가 넓다.

4. 예외 처리 (Exceptions)에 대해 Java와 Kotlin의 가장 큰 차이점은 무엇일까요? *

> No difference between checked and unchecked exceptions.

Java는 'Checked Exception' 정책을 도입해 `throw`가 명시되어있는 함수를 호출하면 반드시 `try-catch` 구문으로 감싸주도록 했었다. 로직의 Happy path 이외의 것들은 Checked Exception으로 처리하려는 시도가 있었지만 empty catch 구문이 생기거나 흐름을 읽기 어려워 지는 경우가 많았다. 결국 '프로그램 흐름을 예외로 제어하면 안된다'는 인식과 'NullObject' 패턴 등의 `try-catch` 구문을 피하기 위한 다양한 대안점이 떠오르면서 배보다 배꼽이 커지는 상황이 발생하였다. 결국 Java의 Checked Exception은 실패한 것이라는 인식이 강해졌다.

그에 대한 영향을 받아 Kotlin에서는 기본적으로 Checked Exception을 따로 검사하지 않는다. 그래서 따로 `try-catch` 구문을 작성해주지 않아도 정상적으로 컴파일된다. @Throws를 이용해 예외발생시 예외를 올려줄 것이라고 지정하여도 외부에서 해당 함수를 호출했을 때 `try-catch` 구문이 없어도 컴파일 에러가 발생하지 않는다. (물론 그 함수를 Java에서 부르면 컴파일 에러가 발생한다.) 그에 비해 Java에서는 에러를 던지면 반드시 try-catch로 감싸거나 위로 올려주어야 한다. 이것이 Checked Exception을 강제했던 Java와의 큰 차이점이라고 생각한다.

5. 확장함수 (Extensions)를 이용해 만들어보고 싶은 유용한 유틸리티 함수 아이디어를 생각해주세요. 함수의 기능, Receiver, function이름, argument(있는 경우), return type 까지 적어주시면 됩니다. (ex: 이 문장이 ?가 들어간 문장인지 확인하는 함수. `fun String.isQuestion(): Boolean`) *

입력된 문자열이 이메일 형식인지 확인하는 함수. `fun String.isEmail(): Boolean`

6. Control Structures/Extensions 단원에서 배운 내용 중 가장 흥미로웠던 부분 1가지를 골라 조금 더 공부해보고 이해하신 내용을 적어주세요. *

확장 함수가 굉장히 흥미로웠습니다. 단원 이름이 Extension Function이 아닌 Extensions 인 것으로 보아 변수도 확장 가능하지 않을까 싶어 공부해보니 Extension property 라는 개념이 있는 것을 발견하게 되었습니다. 정확하게는 '변수'가 생겨나는 것은 아니고 getter와 setter가 생성되는 것이었습니다. getter, setter는 Extension Function과 마찬가지로 확장 클래스의 static 메소드로 변환됩니다. 인스턴스에 종속되는 변수가 생겨나는 것은 아니고 정적 변수라는 점을 주의해야 할 것 같습니다.

7. Edu Tools 실습 예제 Mastermind에 대한 답변 코드를 풀고 파일 또는 코드 캡처를 업로드해주세요. (모두 풀지 못하셨더라도 푸신 부분까지 공유 부탁드립니다 :))

 evaluateGuess - 김...

토론시간에 주로 다뤄졌으면 하는 문제(중복체크가능) *

- ☒ 문제 1
- ☐ 문제 2
- ☐ 문제 3
- ☒ 문제 4
- ☒ 문제 5
- ☐ 문제 6
- ☒ 문제 7
- ☐ 기타:

2주차에는 각자 관심있는 주제를 나누어 조금 더 깊게 공부해보고 서로에게 공유하는 방식으로 진행하고자 합니다. 아래 주제 중 더 공부해보고 싶은 주제를 골라주세요. *

중복 선택 가능하며 모든 참가자분들의 선택 분포에 따라 1주차 온라인 스터디 때 배분할 계획입니다. 시간이 되시는 분은 여러개를 진행하시는 것도 무방하며 반드시 최소 1개 이상은 담당하셔야 합니다.

- ☐ Nullability 파트 : Null과 NPE에 대한 이론적 개념, Null Operator 등에 대해 조금 더 깊게 공부해보고 공부한 내용을 공유합니다. (관련하여 <https://medium.com/@kbm1378/코틀린-입문-스터디-7-nullability-77d92220aad2> 참고)
- ☒ Functional Programming 파트 : Lambda, 함수형 패러다임이 무엇인지에 대해 조금 더 깊게 공부해보고 공부한 내용을 공유합니다. (관련하여 <https://medium.com/@kbm1378/코틀린-입문-스터디-8-functional-programming-b794381c186c> 참고)
- ☐ 실습 과제 파트 : Collection Operator를 활용하는 실습 과제를 풀어보고 더 좋은 풀이법을 고민하여 공유합니다. (관련하여 <https://medium.com/@kbm1378/코틀린-입문-스터디-9-실습-mastermind-in-a-functional-style-nice-string-taxi-park-3ce80530c212> 참고)

응답을 수정 중입니다. 이 페이지의 URL을 공유하면 다른 사용자도 내 응답을 수정할 수 있습니다.

빈 설문지 열기

이번주에 학습하시면서 이해가 안가셨거나 궁금하신 질문들을 모두 적어주세요 *

- Kotlin스러운 코드란 어떤 코드일까요? 스터디 진행자님께서 학습하실때 도움이 되셨던 베스트 프랙티스 프로젝트나 코드리딩을 해볼만한 프로젝트가 있다면 추천 부탁드립니다.?!
이번 스터디에서 열심히 익혀서 꼭 Kotlin스러운 코드를 짜보고 싶습니다!

+ 아직 Kotlin에 대해서 자세히 아는 바가 없어서인지 질문도 없는 느낌입니다ㅠ.ㅠ ㅎㅎ...
스터디 시간에 이야기 나누다보면 잘못 이해한 점들이나 궁금한 점들이 더 나타나지 않을까 싶습니다!

답장 사본을 내가 제공한 주소로 전송합니다.

제출

Google 설문지를 통해 비밀번호를 제출하지 마세요.

reCAPTCHA

[개인정보처리방침](#) [이용약관](#)

이 설문지는 Studypie 내부에서 생성되었습니다. [악용사례 신고](#) - [서비스 약관](#)

Google 설문지

응답을 수정 중입니다. 이 페이지의 URL을 공유하면 다른 사용자도 내 응답을 수정할 수 있습니다.

[빈 설문지 열기](#)