

질문과 검색을 잘하는 법.

- 김성일

세미나 시작 전 질문 시간.

목차

파트 1 - 질문을 잘하는 방법

1. 질문을 할 때 가져야 할 마인드 셋
2. 좋은 질문을 하는 방법

파트 2 - 검색을 잘하는 방법

파트 1-1. 마인드셋부터 바꾸기

당신의 마인드셋부터 바꾸세요.

- 질문은 **부끄러운게 아닙니다.**
- 당신이 질문을 안하면, 아무도 당신이 뭘 모르는지 몰라요.
- 상대방이 당신에게 다가오는 것을 기대하지 마세요.
- 우리 동아리, 그리고 이 업계에서 제대로 된 질문을 했을 때, 대답을 안해주는 사람은 없을거예요.

파트 1-1. 질문을 할 때는 예의를 차려주세요.

- 당신의 질문을 받아주는 사람은 **시간이 넘쳐서** 질문을 받아주는게 아닙니다.
- 기본적인 예의는 지켜주세요.

파트 1-1. 질문의 결과를 지우지 마세요.

- 당신의 질문이 남에게 도움이 될 수 있습니다.
- 이러한 질문의 결과를 제발... **지우지 마세요**
- 지식의 공유를 피하지 마세요.
- 지식은 나누면서 증가하고, 스스로에게도 좋은 결과로 돌아옵니다.

파트 1-1. 나쁜 질문

아무리 생각해도 왜틀린지 모르겠습니다 저조모 살려주세요

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     char S[101];
6
7     scanf("%s", S);
8
9     for (int ascii = 97; ascii < 123; ascii++)
10    {
11        int count = 0;
12        for (int i = 0; S[i] != NULL; i++)
13        {
14            if (S[i] == ascii)
15            {
16                printf("%d", i);
17                count++;
18                break;
19            }
20        }
21        if (count == 0) printf("-1");
22    }
23    return 0;
24 }
```

1. 이 질문을 누군가가 보면 무슨 생각을

2. 이 질문의 문제점이 될까요?

파트 1-1. 이 파트가 끝나면 여러분은..



세모공비 studio

안녕하세요 초등학교 6학년 개발자입니다 그런 문제들은 구글링만 해봐도 넘쳐나는데 이렇게 막 질문을 올리지 마시고 기초부터 해보시는걸 추천드립니다 예를들어 책이나, 구글링, 유튜브를 통해서말이지요 카페는 요술램프가아니니 너무 카페에 의존하지 않으시면 좋겠습니다 아무리 초등학교 4학년이라고 해도 기초를 배우지않고 질문하는것을 무례한행동이라고 생각이됩니다 제생각에는 분제가 왜일어나는지 검색을 3일 4일이라도 해본다음에 해결이 되지 않는 문제들만 올리는것이 좋은것같습니다 만약에 이런식으로 계속 작업을하다보면 막상 어려운문제에 봉착했을때 그문제를 해결하지 못하게됩니다 그래도 힘 내시길 바랍니다 화이팅!

파트 1-2. 검색부터 하세요

- 질문을 하기 전에 검색부터 하세요.
- 당신과 같은 난관에 빠졌던 사람이 생각보다 **꽤나** 많습니다.
- 검색만 잘 한다면, 누군가에게 질문하기 전에 당신의 문제를 스스로 해결 할 수 있습니다.
- 스스로 검색하는 능력을 키우게 된다면, 그것도 당신이 가질 수 있는 하나의 기술이 됩니다.

검색을 어떻게 잘하냐고요? 오늘 검색의 기초를 알려드리겠습니다..

파트 1-2. 질문의 요점을 정리하세요.

- 질문을 받다보면, 본인이 뭘 모르는지도 모르는 사람도 많습니다.
- 본인이 처한 문제 상황은 **본인이 제일 잘 알고, 잘 알아야합니다.**
- 질문을 할 때는 질문의 요점을 **드과신으로 정리**해서 직무해주세요
- 추가적으로 자신이 무엇을 하고 포함해주면 좋습니다.

좋은 질문의 예시)

[Express.js 라우터 XXX 에러 문제]

제가.. **Express.js**로 **무엇**을 개발하고 있었는데요. **이번이 처음인 초보자**라 에러 메시지를 봐도 이해가 안됩니다. 현재 코드의 **7번째 라인**에 **A 에러**가 **B 메시지**와 **함께** 발생합니다. 도와주세요.

```
router.get('/accounts/:user', (req, res) => {  
  const user = req.params.user;  
  console.log('at ' + user);  
  const account = db[user];  
  if (!account) {  
    console.log('about to send 404 response');  
    → return res.status(404).send({message: 'user does not exist'});  
  }  
  return res.json(account);  
});
```

파트 1-2. 문제해결을 위한 시도를 구체적으로 정리하세요

- 혹시나 하는 말인데, **아무 시도도 안하고 질문부터 하는 사람**은 아니겠죠?
- 발생한 문제의 원인을 파악해나갔던 과정을 **요약해서 설명**해주세요.
- 문제를 통해 스스로가 그나마 파악한 **의심되는 부분**은 어디인지 정도도 알려주면 좋아요.
- 그리고.. 컴파일러가 알려주는 **에러 메세지**와 **스택 트레이스**는 당신에게 꽤나 친절한 힌트를 제공하고 있어요.

이러한 것들을 종합해서 당신이 시도했던 것은 무엇이고,

그 시도를 통해 얻은 현재 상태는 무엇이며 원하는 결과는 어떤 것인지 설명해주면 좋아요.

파트 1-2. 제발 코드만 딱 던져주세요!!

아무리 생각해도 왜틀린지 모르겠으니까 저 지모 살려주세요

- “저는 잘한 것 같은데.. 안돼요..” 같은 말을 하는 사람은 관심법을 쓸 줄 몰라요.
- 아까 말했듯이, 에러메시지와 현재 상황 알려주세요.
- 문제가 생겼을거라고 예상하는 코드
- 제발 전체 코드만 던져주고.. “왜 안돼요?”
- 본인 코드가 잘 작성한 코드라고 생각하지 마세요.
- **주석 없이 2달 뒤에 보면, 자신의 코드를 생각하나요?**
- 자신도 못 알아볼 (가능성이 높은) 코드를 작성하지 마세요. 그러니까! 주석 정도는 달아주세요.

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     char S[101];
6
7     scanf("%s", S);
8
9     for (int ascii = 97; ascii < 123; ascii++)
10    {
11        int count = 0;
12        for (int i = 0; S[i] != NULL; i++)
13        {
14            if (S[i] == ascii)
15            {
16                printf("%d", i);
17                count++;
18                break;
19            }
20        }
21        if (count == 0) printf("-1");
22    }
23    return 0;
24 }
```

==

]

파트 1-2. 질문을 보내기 전에 한 번 더 생각하세요.

- 방금까지 안내했던 사항을 따라가다보면, 현재 자신이 처한 문제와 상태는 파악했을까?
- 이렇게 파악한 부분을 따라서, 다시 한 번 정리하고 검색을 해보세요.
- 혼자서 해결 할 수 있는 문제로 변했을 수도 있어요.
- 그래도 해결이 안된다면, 이제 질문을 보내주세요.
- 한 번 더 정리하는 과정을 거치면 질문은 더 좋은 질문이 될 것이고, 스스로 해결을 할 수 있는 능력도 기를 수 있습니다.

이 때, 검색을 통해 스스로 해결하지 못했다면 **질문을 할 차례입니다.**

파트 1-2. 피드백을 주세요.

- 질문에 대한 답변을 받아서 해결했다고, 그게 끝이 아닙니다.
- 질문을 하고 답변을 받게 되었을 때, 그냥 고맙다고 하는 것도 사실 나쁘지는 않아요.
- 근데 그 답변을 통해, 더 나은 방법을 찾는 등의 서로간의 의견을 나누다보면 서로 건설적인 대화로 이루어질 수도 있어요.
- 코딩은 논술과도 같아서, 정답은 없어요.

파트 1-2. 해결한 질문을 스스로 정리하세요.

- 질문을 했다는 것을 창피해하지 마세요.
- 질문을 얻은 결과를 정리하고, 되도록 오픈된 공간에 공유하세요.
- 정리하면서 스스로의 문제가 무엇이었는지 다시 돌아 볼 수 있는 계기가 됩니다.

파트 1-2. 똑같은 질문을 하지 않도록 노력하세요.

- 질문하는 것은 나쁘지 않아요.
- 근데 똑같은 질문을 여러번 하는 건 나빠요.
- 똑같은 질문을 여러번 하는건 “애는 공부를 안하나? 왜 똑같은거 질문하지?”라는 생각을 심어줍니다.
- 근데 다른 사람이 저렇게 생각할까봐 질문안하는 사람이 더 나빠요.
 - 질문 안하고 넘기면, 더 큰 문제가 생깁니다.
 - 작은 문제를 큰 문제로 바꾸지 마세요. ASAP 하게 해결할 수 있는 순간에 해결하세요.
- 질문하고 얻은 결과를 정리하세요. 정리하고 본인의 지식으로 만들면 이런 문제도 안생기겠죠?

파트 1. 정리

- “질문”에 대한 마인드셋을 바꾸세요.
- 질문하기 전 검색부터 하세요.
- “좋은 질문”을 해주세요.
- “질문” - “답변”으로 끝내지 마세요. (피드백을 줄 수 있다면, 피드백을 주세요.)
- 질문과 답변의 결과를 정리해서, 똑같은 질문을 하지 않도록 노력하세요.

파트 1. 마무리

그래서 여기까지 질문 있나요?

파트 2. 검색을 하는 방법

- 이렇게 생각할 수도 있어요.
- 내가 검색하는 방법까지 배워야해요?
- 근데요. 대부분 검색하는 법도 잘 몰라요.



대부분의 사람들은..

검색하는 방법을 정.말.로. 잘

파트 2. 본인은 검색을 잘하고 있어요?

- 코딩하다 모르는 것이 생기면, 유사한 답을 찾으려고 검색을 하고 그 결과를 빠른 시간내로 찾는 편인가요?
- 모르는 것이 생겼을 때, 검색해서 정보를 얻는 일련의 행위가 자연스럽게 몸에 베어 있어요?

여기서 **Yes** 라고 대답을 못한다면.. 다음 슬라이드부터 집중하세요.

파트 2. 추천하는 검색 사이트 순서

1. Google
2. Stack Overflow
3. Github
4. Official Docs
5. **생활 코딩**

위 순서는 개인적 경험 기반으로, 내가 생각했을 때 최대한 양질의 결과를 얻기 쉬운 순서로 나열한거예요.

5까지 다 해봤는데, 안된다? → **질문을 할 순서예요.**

질문하려는 것에 대해 잘 모르거나, 개발 초심자라면 → **1,2 정도까지만 해보고 나서, 질문으로 넘어와도 괜찮다고 생각합니다.**

파트 2. 구글은 신이에요.

- 네이버, 다음? 검색엔진으로 사용하지 마세요.
- 개발 관련 질문과 정보들의 양과 질은 국내 사이트 <<<<< 구글 입니다.
- 개발을 할 때 사용하는 언어들은 English가 Base입니다.

파트 2. 영문으로 검색하세요.

- 혹시 영어 할 줄 아세요? 그래도 서강대 학생인데...
- 개발과 관련된 정보는 대부분 영어 자료가 질이 높고 양도 많
- 한국어로 검색을 하면, **뉴앙스** 혹은 **번역어 선택**에 따라 답변을 얻기 힘들어요.



추가 팁 : 영문으로 검색해도, 구글 언어를 한국어로 해놓으면 한국 문서들이 우선적으로 나오는 경향이 있습니다. 검색할 때 언어를 영어로 설정하는 것도 좋은 방법입니다.

파트 2. 구글에서 지원해주는 기능을 사용하세요.

- 검색시 **site:stackoverflow.com** 과 같은 것을 적어 놓으면 **stackoverflow**에서 해당하는 문서들만 보여줍니다.
- 이걸 알고 계셨나요?
 - 구글링 팁 : <https://makemoneyskills.com/googling-31-tips/>

파트 2. 검색어에 특수문자가 포함된 경우

- 예를 들어, “kotlin ? meaning” 같은 검색어를 검색 할 때, 검색 엔진이나 사이트 내의 검색에서 “?” 같은 특수 문자는 검색어에서 제외되는 경우가 많아요.
 - c++ => cpp, || => or, && => and, & => ampersand, ₩ => blackslash
- 이러한 문제가 생길 때, “kotlin question mark meaning” 과 같은 방식의 검색어를 사용하는 것이 더욱 좋은 결과를 보여줘요.

이거 꿀팁이에요.



kotlin ? meaning

🔍 전체 🖼️ 이미지 📺 동영상 📰 뉴스 📍 지도 ⋮ 더보기

검색결과 약 27,300,000개 (0.42초)

도움말: [한국어 검색결과만 검색합니다.](#) 환경설정에서 검색 언어를 지정할 수 있습니다

Kotlin is a **static type, object-oriented programing** (OOP) language with the Java virtual machine, Java libraries and Android. Kotlin as the less verbose language provides briefer and less redundant code compiled into JavaScript or an LLVM encoder.

<https://whatis.techtarget.com/definition/Kotlin>

[What is Kotlin? - Definition from WhatIs.com](#)

[https://stackoverflow.com/questions/what-is-the-kotlin...](https://stackoverflow.com/questions/what-is-the-kotlin-double-bang-operator)

[What is the Kotlin double-bang \(!!\) operator? - Stack Overflow](#)

2017. 5. 29. — conversion to a non-nullable type (T), !! will throw Null value is null . It is documented here along with Kotlin means ...

답변 3개 · 인기 답변: This is unsafe nullable type (T?) conversion to a non-nullable type (T) ...

What does ?. mean in Kotlin when used on the left side of an assignment? ... 2018

What does -> and ? operators mean in Kotlin? - Stack Overflow 2020

What is the meaning of Type! in Kotlin? [duplicate] - Stack Overflow 2018

what does "::" mean in kotlin? - Stack Overflow 2018

[stackoverflow.com](#) 검색결과 더보기



kotlin question mark meaning



🔍 전체 🖼️ 이미지 📺 동영상 📰 뉴스 📍 지도 ⋮ 더보기

도구

검색결과 약 575,000개 (0.44초)

결과의 차이가 느껴지시나요?

도움말: [한국어 검색결과만 검색합니다.](#) 환경설정에서 검색 언어를 지정할 수 있습니다.

[https://kotlinlang.org/Concepts/Null safety](https://kotlinlang.org/Concepts/Null%20safety)

[Null safety | Kotlin](#)

2021. 7. 8. — Note that this only works where b is immutable (meaning it is a local variable that is not modified between the check and its usage or it is ...

[https://askinlot.com/what-does-question-mark-mean...](https://askinlot.com/what-does-question-mark-mean-in-kotlin)

[What does question mark mean in Kotlin? - AskingLot.com](#)

2020. 4. 10. — What does question mark mean in Kotlin? ... For Kotlin types, it means that null value is allowed to be represented by this type. In practice it ...

[https://stackoverflow.com/questions/what-does-do-i...](https://stackoverflow.com/questions/what-does-do-i-mean-by-kotlin-question-mark)

[What does ?: do in Kotlin? \(Elvis Operator\) - Stack Overflow](#)

2018. 1. 17. — I find the Wikipedia definition pretty accurate: ... The Elvis Operator is represented by a question mark followed by a colon: ?: and it can ...

답변 10개 · 인기 답변: TL;DR: If the resulting object reference [first operand] is not null, it is returned ...

파트 2. 검색 결과 중에 잘 선택하기

- 프로그래밍 언어 및 프레임워크 같은 것들을 시시각각 변합니다.
- 시대상에 따라 코딩스타일도 변하고, 언어에서 사용할 수 있는 기능도 변합니다.
- **구글은 검색 결과가 언제 만들어진 문서인지도 알려줍니다. 이걸 적극 활용하세요.**
- **근데 그 기준이 뭐냐구요?**
- 아래와 같은 기준으로 **휴리스틱하게 적당히 골라주세요.**
 - 라이브러리, 프레임워크 (수개월)
 - 각종 표준, 프로그래밍 언어, 운영체제 (수개월~수년)
 - 각종 프로세스, 설계 방법론 등 소프트웨어 공학에서 다루는 주제들 (수년~수십년)
 - 알고리즘, 자료구조, 프로그래밍 언어론, 운영체제론 등 주로 전산학에서 다루는 주제들 (수십년)

파트 2. 검색을 통해 얻은 결과는 정리하세요

- 질문 쪽에서도 말했지만.. 제발 알게 된 결과는 정리하세요.
- “해결 했다고” 생각하는 문제는, 보통 다음번에 만났을 때, 따로 정리해놓거나 내 지식으로 만들어두지 않으면 한 번 더 마주했을 때 바로 해결하지 못하는 경우가 많아요.
- 이러한 불상사를 방지하기 위해 개인 노트에더라도 제발 정리해놓으세요.
→ 오픈 된 블로그에 정리해두거나 하면 더 좋아요. **지식은 나눌수록 커져요.**

파트 2. StackOverflow

- 네이버 지식IN : 개발자 버전!
- 일단 구글 검색에서 얻은 결과는 StackOverflow가 일단 압도적이고 대부분일거예요.
- 구글 검색에서도 해결이 안되면?
- StackOverflow에서 들어가서 검색을하세요.
- 그리고 똑같이 **검색 툴이 제공해주는 기능**을 사용하면 더 좋습니다.

파트 2. Github 및 Docs

- 사용하고 있는 프레임워크, 프로젝트 등이 Github에 올라와 있다면, **Issue** 혹은 **Pull Request**를 확인해보세요.
- 비슷한 의문 혹은 **질문을 가진 사람이 올린 글**이 있을 수도 있어요.
- 만약 **Docs가 잘 정리된 언어나 프레임워크라면, 그것을 잘 활용**하자.
- 현재 오류가 나는 부분에 관련된 내용을 Docs에 검색하고 활용법을 보다보면 내가 뭘 잘못했는지 확인 할 수 있어요.
- 추가적으로 기본적으로 제공해주는 **예제나 tutorial 같은 것**을 보는 것도 꽤 도움이 됩니다.

파트 2. 생활코딩

- 한국의 StackOverflow와 유사한 역할을 하고 있는 곳이라고 생각하는 곳이에요.
- 생활코딩 그룹에서 페이스북에서 제공하는 “**이 그룹 검색**” 기능을 통해 검색어를 잘 설정해서 검색하세요.
- 여기서도 또한 영문으로 먼저 검색하고, 2차적으로 한국어를 사용하세요.
- **한국 그룹인데 왜 영문 먼저냐구요? 코딩언어는 영어거든요.**

마무리

- 검색과 질문은 잡기술이 아니에요.
- 개발자로서 커리어를 잡은 당신에게는 평생 동반해야 할 중요한 기술이에요.
- 질문과 검색을 이렇게까지 해야하냐구요?
- 네.
- 이런 과정을 누군가 알려주는 사람 찾기 힘들어요. (진짜로.)
- 이런 과정을 스스로 체득하고 얻어가면 도움이 꽤 될거예요.
- 평소에 검색하고 질문하는 습관을 들이세요 제발.

References

- <https://boxnwhis.kr/2020/09/27/ir-for-developers.html>
- <https://brunch.co.kr/@codestates/4>
- https://jbee.io/essay/good_questionor/
- <https://velog.io/@kshired/Tips-개발자가-질문과-검색을-잘하는-법>

마지막 글은 제가 발표한 것을 글로 작성한 버전이에요.

다시 복기하고 싶으시다면, 읽어봐도 좋아요.

끝났으니, 다시 질문 시간.