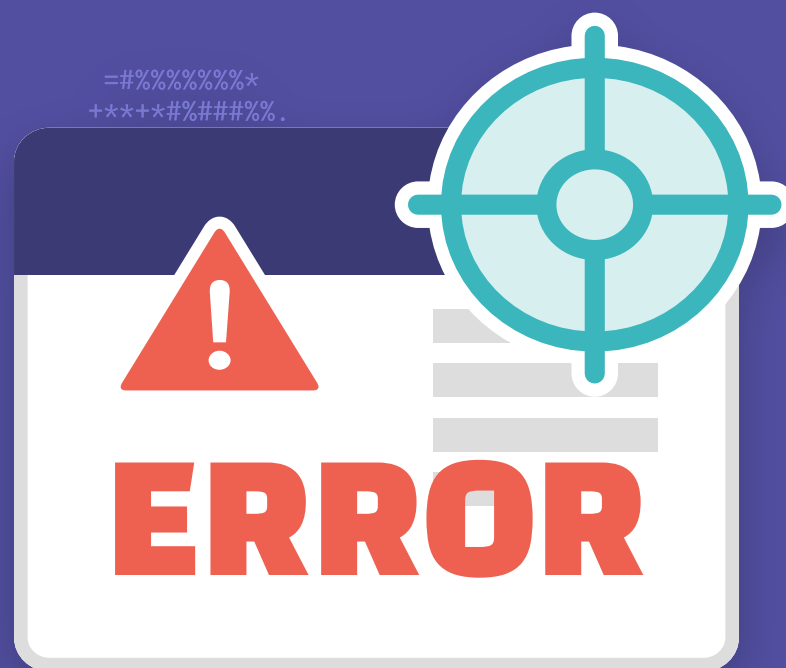


```
/* elice */
```

도전! 디버깅 입문

에러를 만나도 당황하지 않는 법



김건우 선생님

[illegible]

일반적인 코딩 수업은...

어떻게 하면 **좋은 프로그램**을 만들까?

좋은 코드를 짜려면 어떻게 해야 할까?

도전! 디버깅 입문 수업은...

어떻게 하면 실수를 덜 할 수 있을까?

틀린 코드를 어떻게 고쳐야 할까?

에러를 만났을 때

```
def greeting(your_name):  
    print("Hello, " + yourname + "!")  
  
greeting("Donald Trump")  
# Hello, Donald Trump!
```

에러를 만났을 때

```
Traceback (most recent call last):  
  File "main.py", line 4, in <module>  
    greeting("Donald Trump")  
  File "main.py", line 2, in greeting  
    print("Hello", yourname + "!")  
NameError: name 'yourname' is not defined
```

에러를 만났을 때

분명 선생님이랑 똑같이 짰는데 **이상한 에러**가 떠요...
왜 제 코드만 안 돌아갈까요 ㅠㅠ



함수에 오타가 있네요!
코드의 2번째 줄을 확인해 보세요 :)

수강 목표

버그를 만났을 때 **당황**하지 않게 됩니다.

버그를 **해결**하는 다양한 방법들을 알게 됩니다.

안전한 코드를 작성하는 법을 배웁니다.

수강 대상



에러를 보기만 해도 안절부절 못 하시는 분들



파이썬을 쓸 줄은 알지만 **혼자 코딩**하는 것이 두려우신 분들



안전하고 **사용자를 고려한** 프로그램을 만들고 싶으신 분들

커리큘럼

1 ○

왜 디버깅이 필요할까요?

프로그래머와 사용자가 겪을 수 있는 다양한 상황을 이해합니다.
디버깅이 무엇인지, 디버깅이 왜 중요한지 배웁니다.

2 ○

에러 읽고 대처하기

코딩을 하면서 만나는 다양한 에러 코드를 이해하고,
이에 대처하는 방법을 배워 봅니다.
에러를 마주했을 때의 공포감을 극복합니다.

커리큘럼

3 ○

나의 첫 테스트 코드

코드가 바뀌어도 올바르게 동작할 수 있도록 도와 주는
테스트 코드를 작성해 봅니다.

4 ○

실전 디버깅!

여러 가지 실전 문제를 풀어 보며,
다양한 버그를 찾아내고 안전한 코드를 설계하는 능력을 기릅니다.

목차

1. 버그와 예외
2. 디버깅이란?
3. 디버깅이 필요한 이유

버그와 예외

버그 (Bug)

코드가 원하는 대로 동작하지 않는 모든 경우

- 프로그램이 **죽는** (크래시가 발생하는) 경우
- 예상한 속도보다 훨씬 **느리게** 동작하는 경우
- **잘못된** 결과를 출력하는 경우
- 코드 설계 상 **일어나면 안 되는** 상황이 일어난 경우

예외 (Exception)

일반적이지 않은 코드의 동작

- 사용자가 프로그램을 **강제 종료**하는 경우
- **분수의 분모에 0**이 들어가는 경우
- 열고자 하는 **파일이 존재하지 않는** 경우
- **인터넷이 불안정한** 경우

예외 처리 (Exception handling)



예외 처리 (Exception handling)



파일을 불러오는 중에 **인터넷이 끊겼다면?**

예외 처리 (Exception handling)



파일을 불러오는 중에 인터넷이 끊겼다면?

→ 처리되지 않은 예외는 **버그**를 만들 수 있음!

예외 처리 (Exception handling)

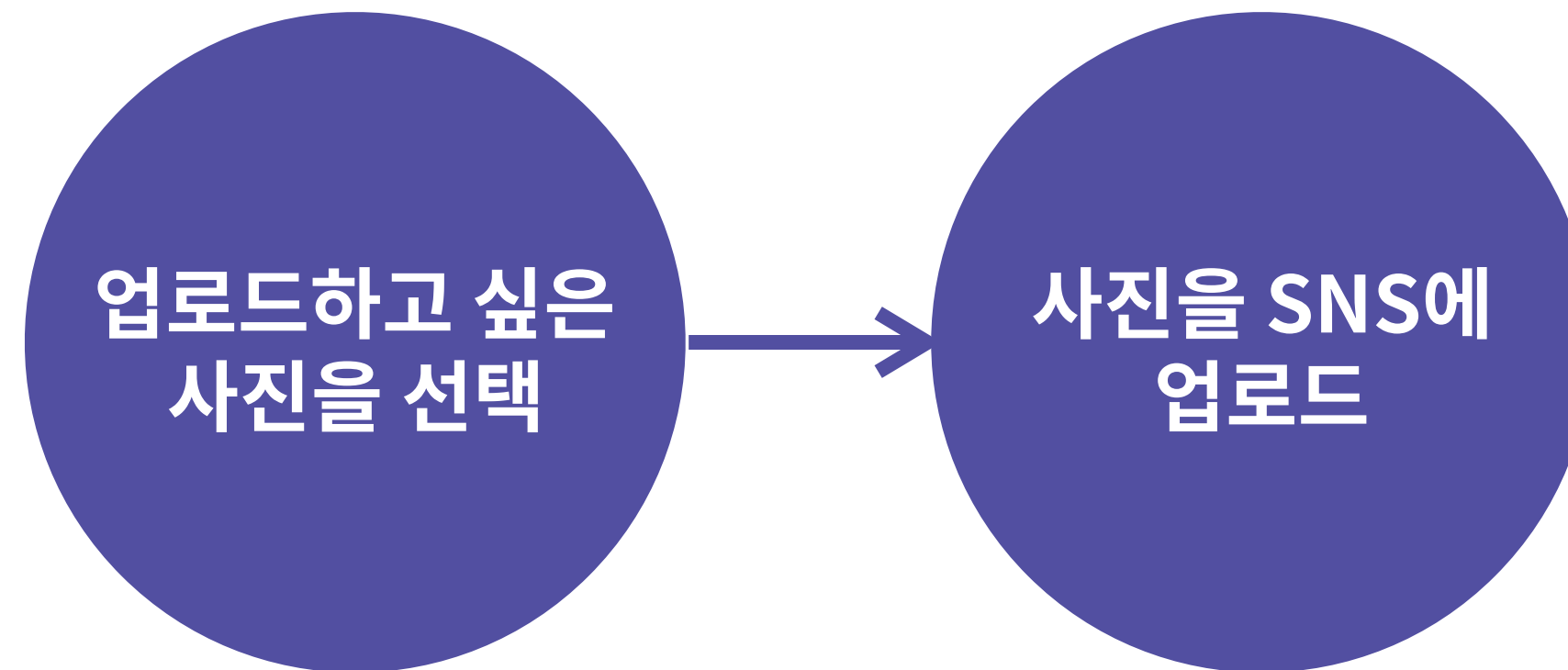


사진 업로드 중에 배터리가 없어 **휴대폰이 꺼졌다면?**

예외 처리 (Exception handling)

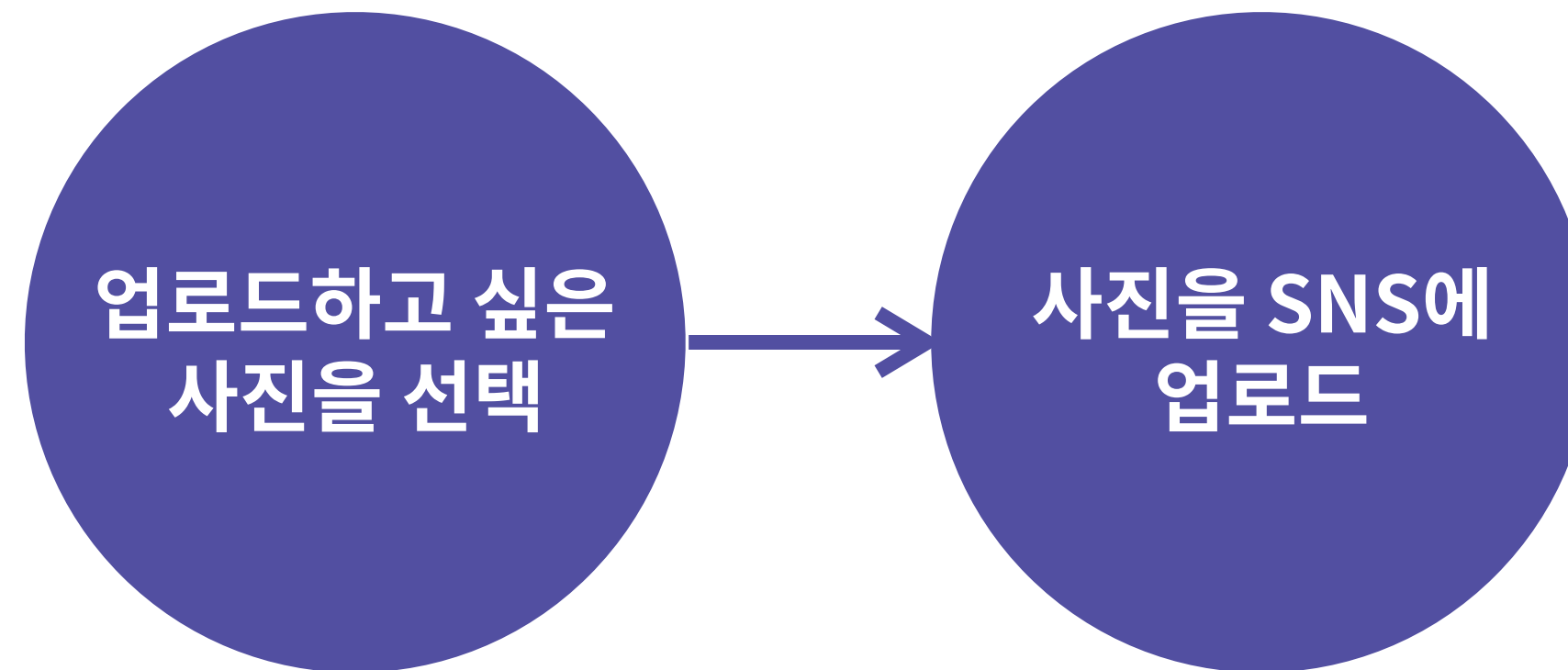


사진 업로드 중에 배터리가 없어 휴대폰이 꺼졌다면?

→ 개발자는 사용자의 다양한 상황을 **예측**할 수 있어야 함

디버깅이란?

디버깅이란?

사용/테스트 중 찾아낸 버그를 없애 나가는 과정

- 버그의 **원인**을 찾고
- 발생한 버그를 **해결**하고
- 비슷한 버그의 **재발을 방지**하는 것

버그의 원인 찾기

- 어떤 **상황**에서, 어떤 조작을 했을 때 일어나는가?
- 얼마나 잦은 **빈도**로 발생하는가?
- 에러 **메시지**가 뜬다면 내용이 무엇인가?

발생한 버그 해결하기

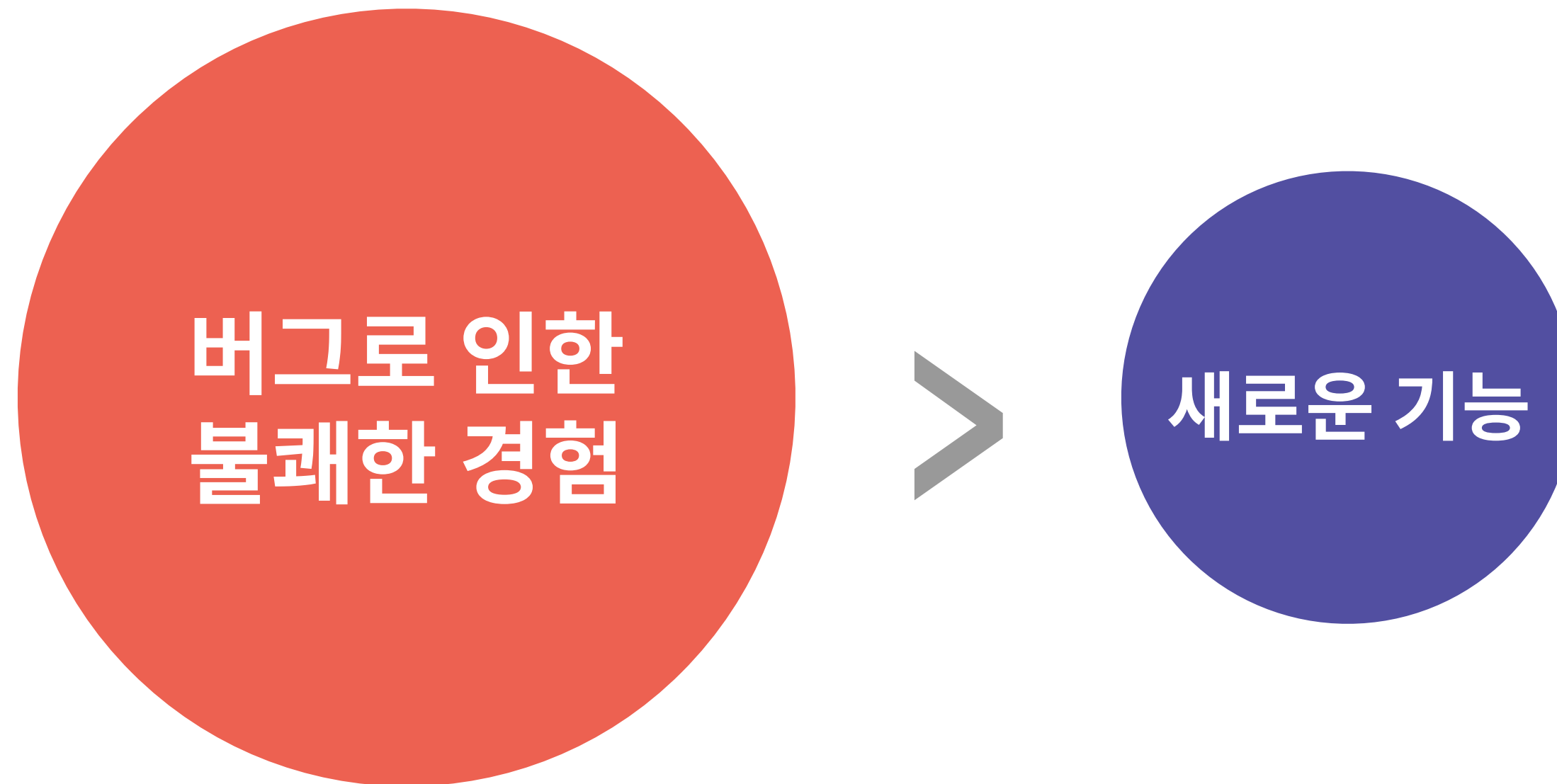
- 버그가 발생한 **원인**을 해결하려면 어떻게 해야 하는가?
- **새로운 코드**가 일으킬 새로운 버그는 없는가?

버그 재발 방지하기

- 버그를 일으킨 코드가 **다른 곳**에서 사용되는가?
- **비슷한 설계**를 이용한 다른 코드는 없는가?
- 고친 코드의 **약점**은 없는가?
- 코드를 지속적으로 **검증**할 방법은 무엇인가?

디버깅이 중요한 이유

사용자 경험과 직결



처음부터 완벽한 코드는 없다

코드가 잘 작동하지 않을 때 어... 왜 안 되지?

코드가 한 번에 작동할 때 어... 왜 벌써 되지?

빠르게 문제를 파악/해결하는 것이 능력!

실패는 성공의 어머니

많은 버그를 경험할 수록,
더 **안전한 코드**를 설계하는 능력이 생긴다!

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice