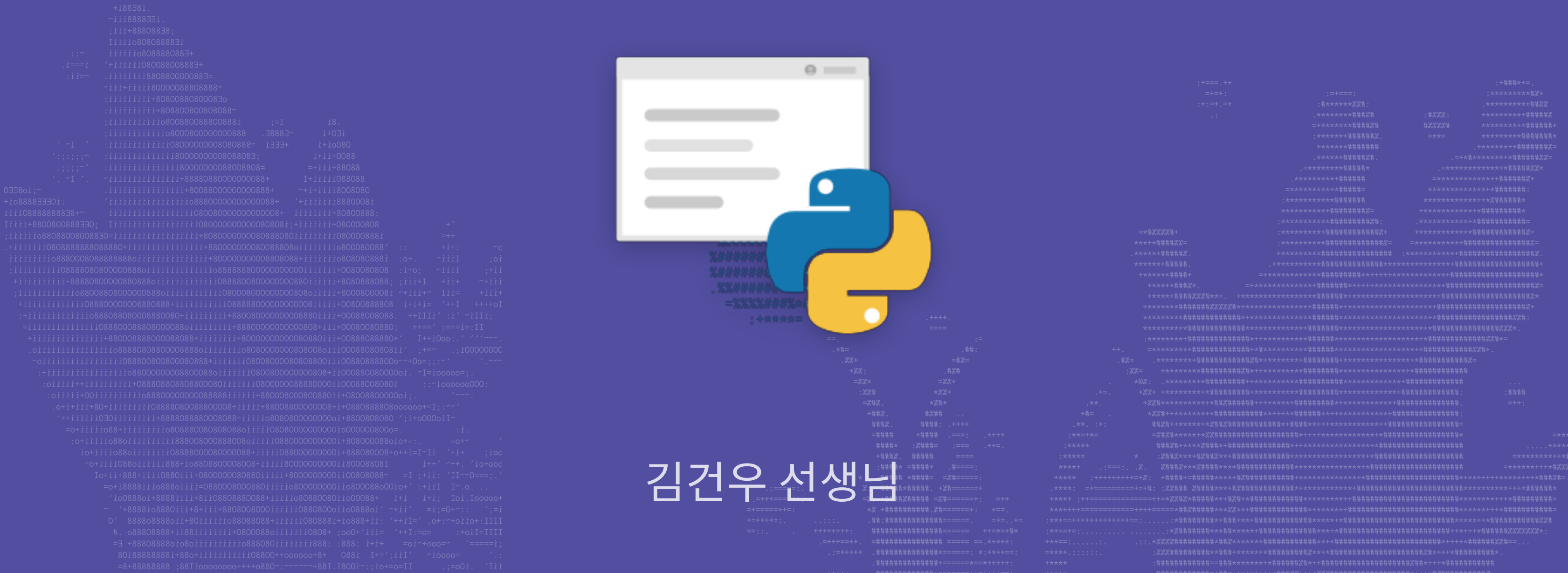


/\* elice \*/

# 본격! 프로그래밍

코드의 기본, 함수



# 목차

1. 함수의 기초
2. Python의 함수 100% 활용하기
3. 변수의 스코프

# 커리큘럼

1 ○

## 코드의 기본, 함수

코드의 불필요한 반복을 줄여주고, 더 이해하기 쉬운 코드를 만들어 주는 함수에 대해 배워봅니다.

2 ○

## 프로그래밍 세상의 설계도, 클래스 입문

보다 논리적인 프로그램을 설계할 수 있게 도와주는 클래스 개념에 대해 배우고, 직접 클래스를 설계해 봅니다.

# 커리큘럼

3 ○

## 클래스 더 알아보기

여러 클래스의 관계를 정의하는 상속 개념에 대해 배웁니다.  
데이터의 관계를 정의하며 프로그램의 큰 그림을 그려 봅니다.

4 ○

## 모듈과 패키지

다른 사람이 만든 프로그램을 내 프로그램에서 사용할 수 있게  
만들어 주는 모듈과 패키지에 대해 배웁니다.  
파이썬의 다양한 오픈 소스 패키지를 체험해 봅니다.

# 수강 대상



파이썬의 **기초 문법**은 배웠지만  
이제 무엇을 해야 할 지 모르겠는 분들



프로그래밍으로 **무엇을 할 수 있는지**  
감이 잘 잡히지 않는 분들



보다 **논리적이고 짜임새 있는** 프로그램을  
만들어 보고 싶은 분들

# 수강 목표

체계적인 프로그램을 설계해 볼 수 있습니다.

여러분 주변의 프로그램을 보는 시각이 달라집니다.

파이썬과 조금 더 친해집니다.

# 함수의 기초

# 함수란?

자주 사용하는 코드를 의미 단위로 묶어 둔 것



# 함수의 예시

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1
```

# 함수의 예시

```
# 1부터 6 사이의 숫자를 랜덤하게 리턴  
def roll_die():  
    return random.randint(1, 6)
```

# 함수의 예시

```
# seconds부터 0까지 1초마다 출력  
def count_down(seconds):  
    while seconds >= 0:  
        print(seconds)  
        time.sleep(1)
```

# 함수의 예시

```
# 1시간마다 현재 시각을 출력  
  
def clock_hourly():  
    while True:  
        print(datetime.now())  
        time.sleep(3600)
```

# 함수의 구성 요소

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
age = calculate_age(1988)
```

# 함수의 구성 요소

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
age = calculate_age(1988)
```

# 함수의 구성 요소

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
age = calculate_age(1988)
```

# 함수의 구성 요소

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
age = calculate_age(1988)
```



# return, 함수의 끝

# 태어난 연도를 받아 나이를 리턴하고 출력

```
def calculate_age(year_of_birth):  
    this_year = 2018  
    age = this_year - year_of_birth + 1  
    return age  
print(age)
```

# return, 함수의 끝

# 태어난 연도를 받아 나이를 리턴하고 출력

```
def calculate_age(year_of_birth):
```

```
    this_year = 2018
```

```
    age = this_year - year_of_birth + 1
```

```
    return age
```

```
    print(age) ← 실행되지 않음
```

# return, 함수의 끝

# 태어난 연도를 받아 나이를 리턴하고 출력

```
def calculate_age(year_of_birth):  
    this_year = 2018  
    age = this_year - year_of_birth + 1  
    print(age)  
    return age
```

# return vs. print

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
age = calculate_age(1988)
```

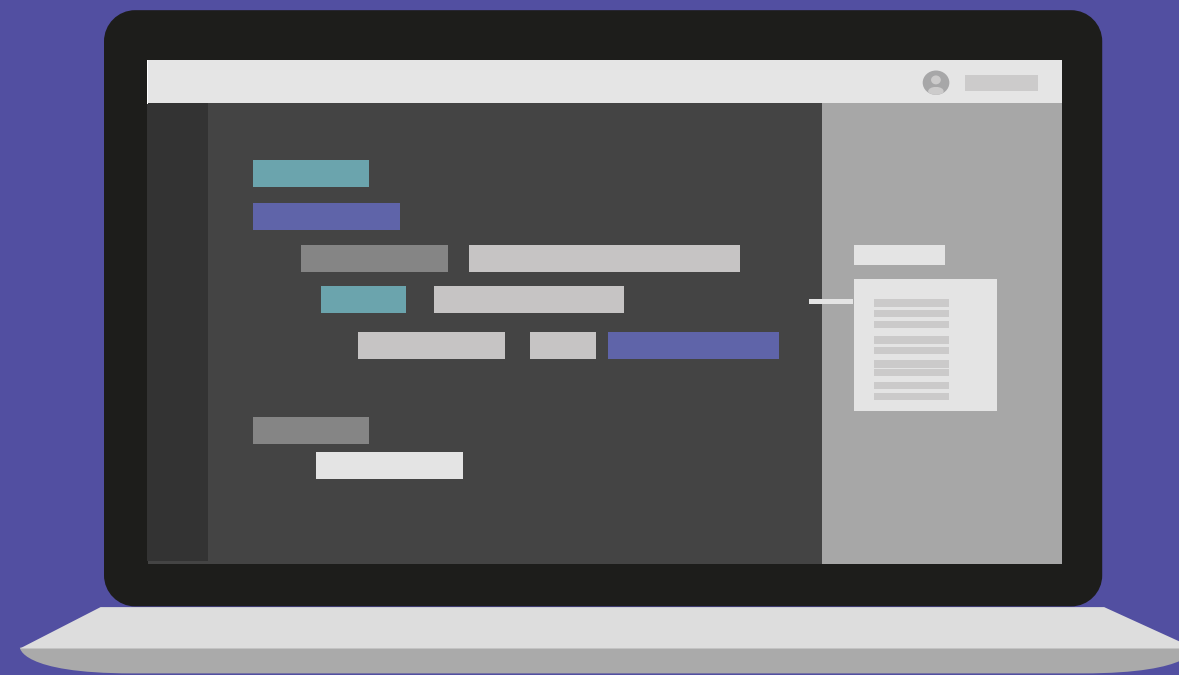
# return vs. print

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
print(calculate_age(1988))
```

# return vs. print

```
# 태어난 연도를 받아 나이를 리턴  
def calculate_age(year_of_birth):  
    this_year = 2018  
    print(this_year - year_of_birth + 1)  
  
age = calculate_age(1988)
```

# [실습 1] return vs. print



```
/* elice */
```

# Python 함수 100% 활용하기



# 매개변수가 많을 때

```
def sign_up(email, password, name, city,  
            gender, is_vip):
```

```
    ...
```

Seoul에 사는 Alice 님

```
sign_up("alice@elice.io", "c0derL@nd",  
        "Alice", "Seoul", "F", False)
```

# 매개변수가 많을 때

```
def sign_up(email, password, name, city,  
            gender, is_vip):
```

```
    ...
```

Alice에 사는 Seoul 님

```
sign_up("alice@elice.io", "c0derL@nd",  
        "Seoul", "Alice", "F", False)
```

# 매개변수가 많을 때

```
sign_up(email="alice@elice.io",  
        password="c0derL@nd",  
        city="Seoul", name="Alice",  
        gender="F", is_vip=False)
```

# 함수 디자인

어떤 커뮤니티의 90% 이상은 가입 시 일반 회원이다.  
그러나 소수의 회원들은 가입 시부터 VIP로 등록된다.

# 함수 디자인

```
def sign_up(email, password, name, city,  
            gender, is_vip):  
    ...  
  
alice = sign_up(..., is_vip=False)  
bob = sign_up(..., is_vip=False)  
james = sign_up(..., is_vip=False)  
mansour = sign_up(..., is_vip=True)
```

# 매개변수의 기본값

```
def sign_up(email, password, name, city,  
            gender, is_vip=False):  
    ...  
  
alice = sign_up(..., is_vip=False)  
bob = sign_up(..., is_vip=False)  
james = sign_up(..., is_vip=False)  
mansour = sign_up(..., is_vip=True)
```

# 매개변수의 기본값

```
def sign_up(email, password, name, city,  
            gender, is_vip=False):  
  
    ...  
  
alice = sign_up(...)  
bob = sign_up(...)  
james = sign_up(...)  
mansour = sign_up(..., is_vip=True)
```

# 파이썬의 내장 함수들

```
max(1, 3)          // 3
min(1, 3)           // 1
abs(-17)            // 17
round(18.8)         // 19
```



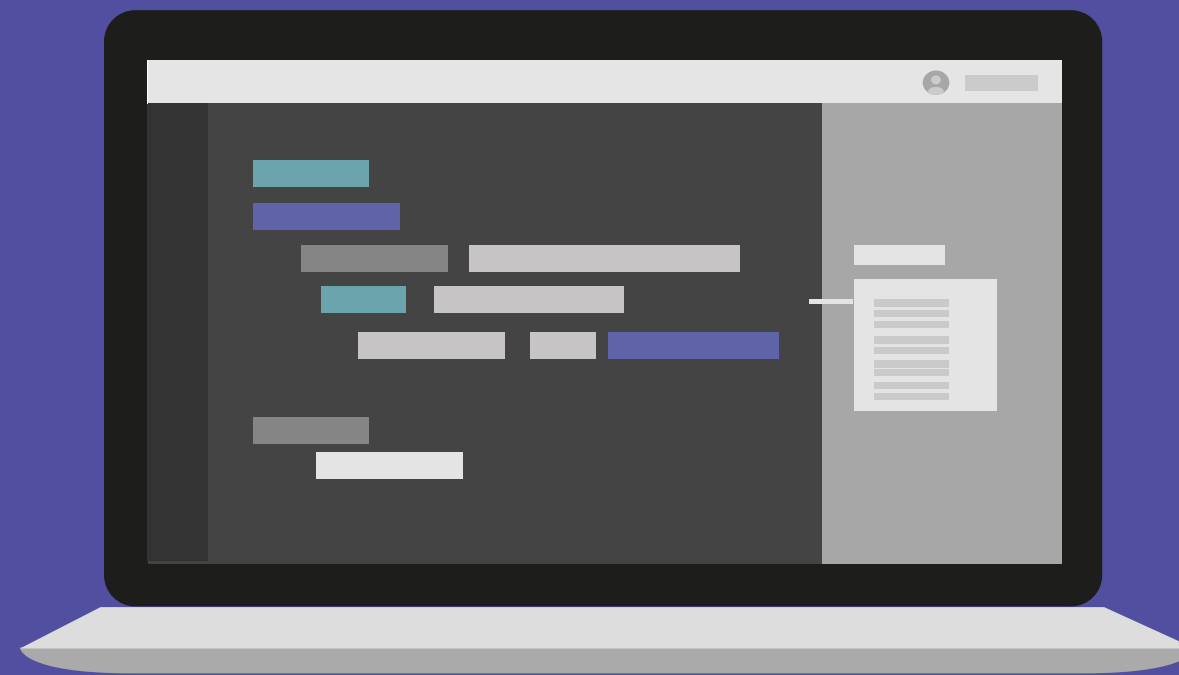
# 파이썬의 내장 함수들

```
len("Elice")      // 5
len([1, 3, 5])     // 3
str(37)            // "37"
int("9")           // 9
int("3.14")        // Error
```

# 파이썬의 내장 함수들

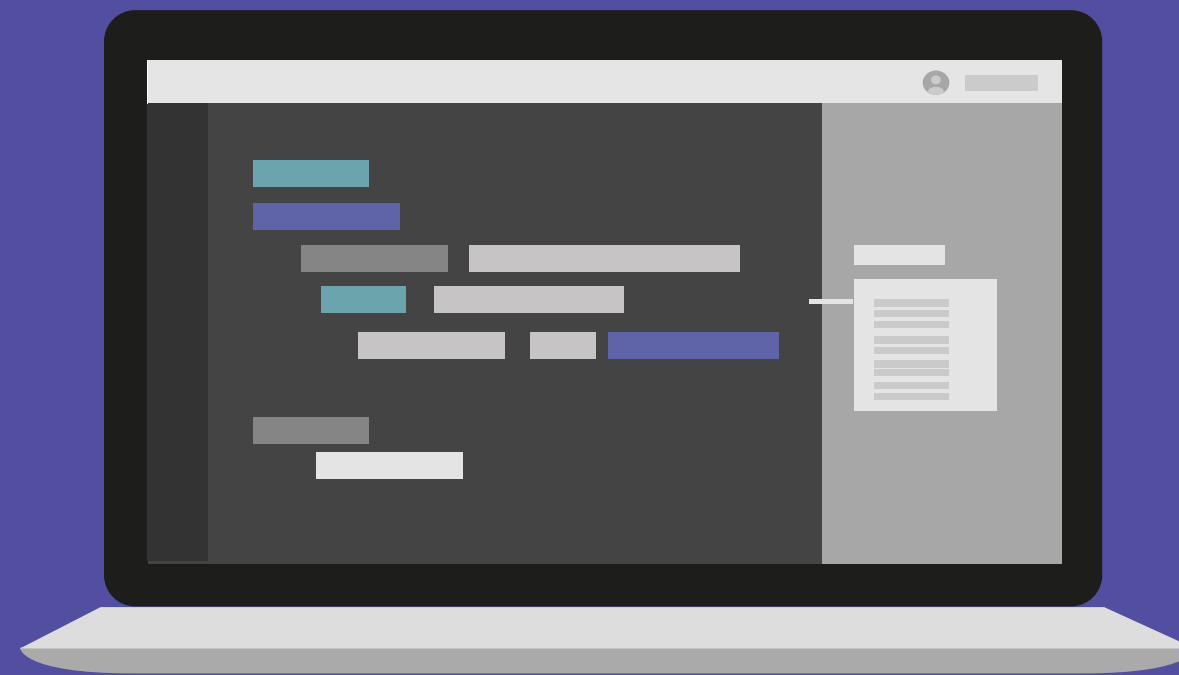
```
range(3)                // [0, 1, 2]
range(1, 4)              // [1, 2, 3]
reversed("Elice")        // "ecilE"
any([False, True, True]) // True
all([False, True, True]) // False
```

# [실습 2] 올바른 이름인가요?



`/* elice */`

# [실습 3] 올바른 이메일인가요?



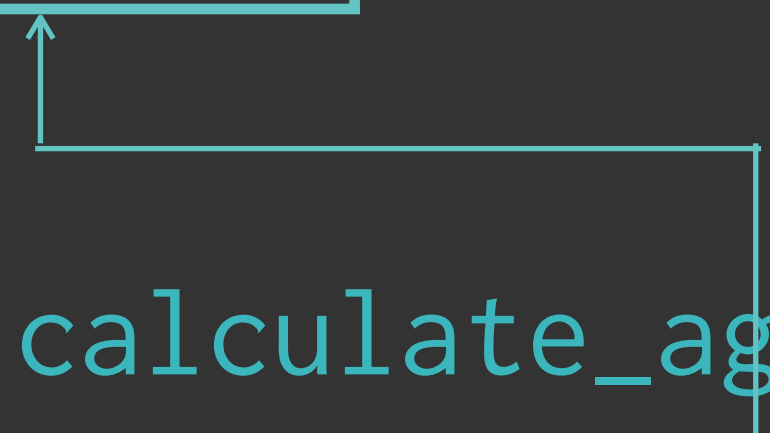
`/* elice */`

# 변수의 스코프

# 변수의 스코프

```
this_year = 2018
```

```
def calculate_age(year_of_birth):  
    return this_year - year_of_birth + 1
```



# 변수의 스코프

```
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1  
  
print(f"올해는 {this_year}년입니다.")
```

# 변수의 스코프

```
def calculate_age(year_of_birth):  
    this_year = 2018  
    return this_year - year_of_birth + 1
```

```
print(f"올해는 {this_year}년입니다.")
```

this\_year를 찾을 수 없습니다.

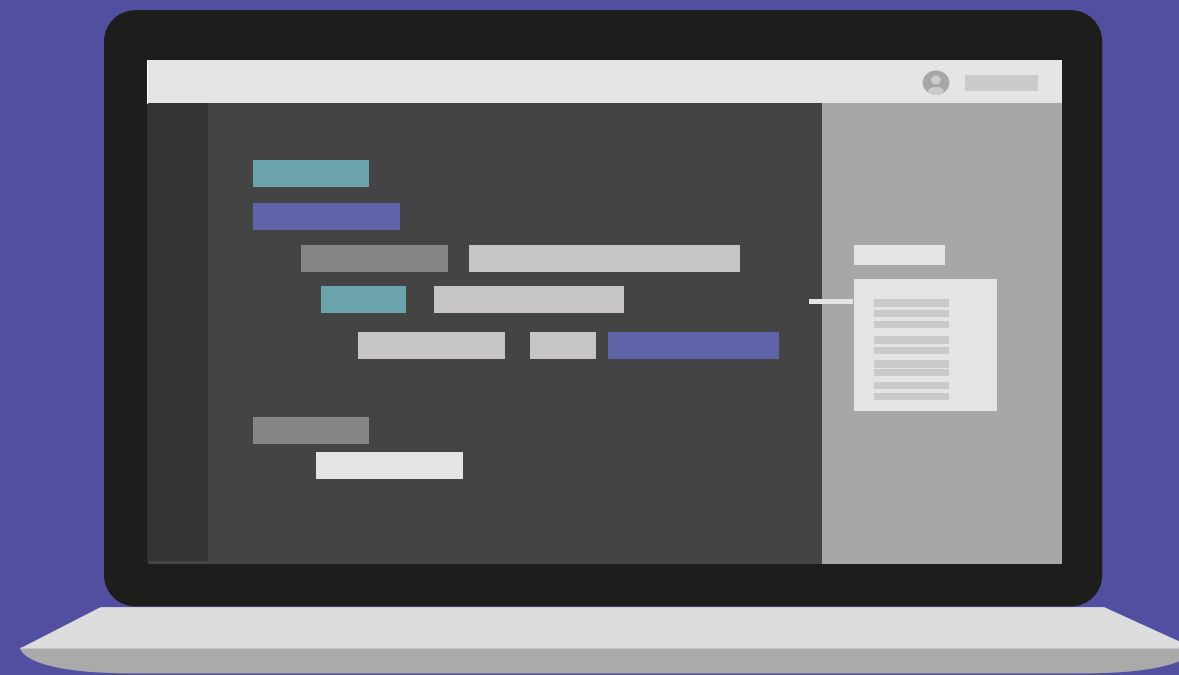


# 변수의 스코프

```
this_year = 2018
```

```
def calculate_age(year_of_birth):  
    this_year = 2000  
    return this_year - year_of_birth + 1
```

# [실습 4] 비밀번호 숨기기



```
/* elice */
```