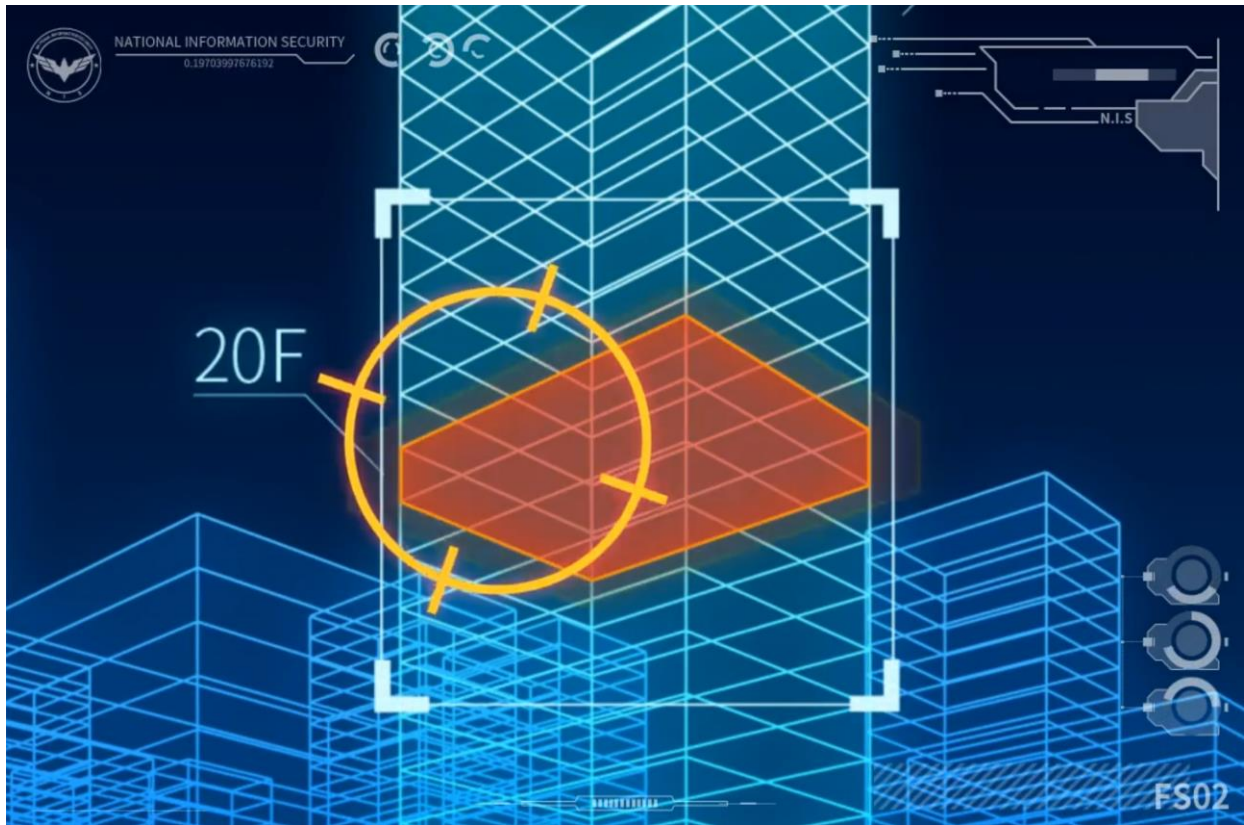


## [ 시청에 설치된 폭발물 수색 작전 ]



빌더버그 조직의 타깃인 시청 건물 20층 어딘가에 폭발물이 설치되어 있다는 사실을 알아냈다. 하지만 건물에 보관된 중요한 기밀자료를 보존하기 위해서는 폭발물의 위치를 알아내 폭파를 막아내야만 하는데...

오늘의 목표

시청에 설치된 폭발물 수색하기

주요 함수

print, range, append, sum, int

### ➤ 학습 목표

- 스토리의 **사건**을 이해하며 미션을 해결하기 위해 필요한 데이터를 수집할 수 있습니다.
- 반복 제어문을 사용하여 리스트 요소의 합계를 구하는 프로그램을 만들 수 있습니다.

- 항목의 값을 서로 비교하는 비교 연산자에 대해 알고 적절하게 사용할 수 있습니다.
- 2 차원 리스트의 형태를 알고 각 리스트의 값에 접근하여 요소를 추출할 수 있습니다.
- 주요 함수를 배우고 활용해 데이터를 분석함으로써 사건을 해결할 수 있습니다.

#### ✓ 수업 전 체크하기

- '수업 환경 테스트' 를 눌러 수업 환경을 체크합니다.
- 학생과 반갑게 인사를 나누고 전 시간 스토리에 대해 이야기를 나눕니다.

#### ✓ 수업 중 체크하기

- 스토리의 내용을 잘 이해했는지 적절한 질문을 통해 지속적으로 확인합니다.
- 코드를 입력할 때, 대소문자를 정확히 입력하였는지, 따옴표와 괄호의 짝이 잘 맞는지 확인하도록 합니다.
- 띄어쓰기가 잘 되어 있는지 확인합니다.

#### ✓ 수업 후 체크하기

- 스토리를 잘 이해했는지 확인합니다.
- 파이썬 프로그래밍 개념을 잘 이해했는지 확인합니다.
- 화면의 '수업 종료' 버튼을 누른 뒤 피드백을 작성합니다. (하단 피드백 예시 참고)

HOW CODING

## Intro.

- 이전 차시 스토리에 관해 이야기를 나눕니다.

[ 이전 스토리  
이해 ]

〈이전 스토리〉

이선과 파이는 총 40 층으로 이루어진 시청 건물에서 각 층의 용도를 파악하여 빌더버그 조직이 타깃으로 삼을만한 층을 추리했다. 시청 건물은 일반자료실, 기밀자료실, 회의실, 사무실의 네 가지 용도로 사용하고 있었는데, 이 중에서 중요한 자료가 보관된 기밀자료실을 타깃으로 삼았고 기밀자료실로 사용하고 있는 층에서도 위험물 감지 센서가 작동하지 않았던 20 층에 폭발물을 설치한 사실을 알아냈다.

→ NIS 시스템에 접속하기 위한 ID 와 PW 를 학생에게 전달합니다.

- NIS 정보 요원이 되어 사건을 해결할 ID 를 부여합니다.
- ID : [agent@nis.com](mailto:agent@nis.com) / PW : python

히우코딩  
HOW CODING

Mission1. 20 층 전체 호수의 총면적 확인하기.

- 미션의 목표를 설명합니다.



## [ 미션 목표 ]

- 1) 스토리 속에서 단서를 확인하여 폭발물 수색 가능성 여부를 판단할 수 있습니다.
- 2) 반복 제어문을 사용하여 같은 일을 반복하는 프로그램을 만들 수 있습니다.
- 3) 리스트 내 요소의 합계를 구하는 프로그램을 만들 수 있습니다.

## [ 스토리 이해 ]

- 스토리의 내용을 이해합니다. 학생 스스로 클릭하면서 스토리를 읽도록 합니다.



( 빌더버그 조직의 타깃인 시청 건물에는 다행히 사람이 없다. )

( 하지만 건물에 보관된 기밀자료 모두를 훑기엔 시간이 부족하다. )

( 기밀자료를 보존하기 위해 시청 폭파를 막아야만 한다. )

“파이”, “이선! 시청 건물 20 층에 폭발물이 있어.”

“이선”, “응, 그래서 NIS 상황실에서 폭발물 감지 드론을 현장에 보낸대.”

“파이”, “그래, 드론에는 화학물질을 감지하는 센서가 있어서”

“파이”, “드론에서 보내 준 화학물질 농도 데이터를 분석하면 폭발물을 찾을 수 있을 거야.”

“이선”, “근데 문제가 있어. 파이!”

“이선”, “NIS 에서 사용하는 드론은 한 번 비행으로 면적 300 평 이하까지만 감지가 가능하대.”

“파이”, “잠깐, 시청은 한 층 당 면적이 꽤 넓은 것으로 알고 있는데.”

“이선”, “그래? 20 층 전체 호수의 면적이 얼마나 되는지 먼저 확인해야겠군.”

“파이”, “맞아. 시청에 요청해서 내부 도면 좀 확보해줘.”

<<< 단서 확인하기 >>>

<단서>

20 층 전체 호수 총면적을 구해라.

### [ 학습 자료 ]

→ 시청 건물 20 층에 설치된 폭발물을 찾기 위한 방법이 무엇인지 이야기 나누어 봅시다.

(예시) 폭발물이 설치된 20 층으로 폭발물 감지 드론을 보내려고 합니다. 드론에는 화학물질을 감지하는 센서가 있어 곳곳을 수색하며 보내주는 데이터를 분석해 일정 기준 농도 이상인 위치를 찾아 폭발물이 어디 있는지 알려줍니다.

→ 파이와 이선은 20 층 전체 호수의 면적을 파악하려고 했던 이유는 무엇인지 이야기 나누어 봅시다.

(예시) NIS 에서 보유한 폭발물 감지 드론은 한 번의 비행으로 면적 300 평 이하까지만 수색을 할 수가 있습니다. 그래서 우선 20 층 전체 호수의 총면적을 구해 보고 수색 가능 여부를 판단하려고 하는 것 같습니다.

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

#### # 반복문

- 같은 일을 반복할 때 사용하는 제어문
- 순서가 있는 자료형에서만 사용
- 구조 : for \_\_ in

for 변수 in 반복할 조건:

수행할 문장

ex) numbers = [1, 2, 3]

for v in numbers:

print(v)

#### # 리스트 합계

- 리스트 요소의 합계
- 요소들을 변수에 반복적으로 더해 저장

for 변수 1 in 리스트:

변수 2 = 변수 2 + 변수 1

ex) numbers = [1, 2, 3, 4, 5, 6, 7]

adding = 0

for i in numbers:

adding = adding+i

→ 리스트를 반복문으로 사용하면 각 요소의 값을 변수에 대입하여 어떤 수행을 할 수 있게 해줍니다. 그리고 리스트의 요소 개수는 반복할 횟수를 정해줍니다.

→ 리스트 요소의 합계를 구하고자 할 때는 첫 번째 요소값을 변수 1에 대입하여 합계를 저장할 변수 2에 더한 값을 다시 합계를 저장할 변수 2에 대입해 줍니다. 이 때, 합계를 저장할 변수 2에 초기값 0을 미리 대입합니다.

## [ 코드 이해 ]

- 사용할 API에 대해 설명합니다.

### < API >

1) print(값) : 값을 출력하는 함수

- 학생이 스스로 코드를 구현하여 정답을 찾아 확인하도록 합니다.
- 스토리 속에서 해결해야 할 미션에 대해 질문하며 해당 코드를 학생 스스로 구현하도록 설명해줍니다. 코드는 제공하지 않습니다.

### < CODE >

#data

#호수별 면적리스트

room\_area =

[24,24,35,32,8,27,23,35,19,19,18,22,23,19,22,22,22,18,22,30,30,23,22,30,30,23,23,28,25,20,23,30,18]

**#20 층 전체 호수 총면적 구하기**

```
area_sum=0
for i in room_area:
    area_sum = area_sum + i
print(area_sum)
```

#정답 확인:

789

→ 주어진 data 에 대해 이야기 나누어 봅니다.

(예시) 20 층의 전체 호수에 대한 면적 평수를 나타낸 리스트 입니다. 이 리스트를 len 함수를 사용하면 총 33 개의 호수인지 알아낼 수 있습니다.

→ 전체 호수의 총면적을 구하기 위해서 어떻게 할지 이야기 나누어 봅니다.

(예시) 총면적을 구하기 위해서는 주어진 리스트에 있는 요소값을 모두 더하면 됩니다. for 반복문을 사용하여 in 뒤에 반복할 횟수가 될 room\_area 리스트를 넣어줍니다. 첫 번째 요소값부터 추출하여 면적합계를 저장할 area\_sum 변수에 더하고 대입해주는 것을 마지막 요소값까지 반복합니다. 이 때 area\_sum 변수는 for 반복문 시작 전 라인에서 0 으로 초기화 해줍니다.

→ 실행결과 창에 출력된 값을 정답창에 입력하여 확인하도록 합니다.

(예시) area\_sum 변수에 저장된 값인 **789** 를 입력한 후 정답을 확인하도록 합니다.

→ 스스로 코딩하도록 유도합니다.

→ 미션 2 템플릿으로 이동합니다.



## Mission2. 감지할 호수 선별하기.

## ● 미션의 목표를 설명합니다

## [ 미션 목표 ]

- 1) 스토리 속에서 단서를 확인하여 감지할 호수를 선별할 수 있는 조건을 알아낼 수 있습니다.
- 2) 반복 제어문을 사용하여 같은 일을 반복하는 프로그램을 만들 수 있습니다.
- 3) 조건 제어문을 사용하여 참과 거짓 조건에 따라 어떤 일을 수행하는 프로그램을 만들 수 있습니다.

## ● 스토리의 내용을 이해합니다.

## [ 스토리 이해 ]

- 이전 미션에서 나온 프로그래밍 결과를 생각해보며 스토리를 확인합니다.



"이선", "총면적이 789 평이라고?"

"이선", "NIS 상황실에서 보내려는 드론이 감지하기에는 너무 넓어."

"파이", "맞아. 드론이 꼭 감지해야 하는 곳을 선별해보자."

"파이", "이선! 내가 도면을 보니까 무슨 표시가 있던데. 무슨 표시야?"

"이선", "아! 나도 도면 받으면서 물어봤는데, 면적에 따라 다르게 운영되는 표시라고 했어."

"파이", "그럼 이 표시대로 운영하는 기준을 찾아내면 드론이 감지해야 할 면적을 좀 줄일 수 있을 것 같아."

<<< 단서 확인하기 >>>

<단서>

감지해야 할 호수를 모두 찾아라.

→ 폭발물 감지 드론이 필수적으로 감지해야 할 호수를 찾을 방법에 대해 이야기 나누어 봅니다.

## [ 학습 자료 ]

(예시) 20 층 전체 호수의 총면적은 789 평으로 폭발물 감지 드론이 탐지할 수 있는 면적인 300 평보다는 훨씬 넓은 면적입니다. 그래서 필수적으로 탐지해야할 호수를 선별하여 총면적이 300 평 이하가 되도록 해야 하는데, 이 때 시청에서 보내온 20 층 도면을 자세히 봐야 합니다. 도면 안에는 전체 33 개의 호수가 있고 호수마다 사람 아이콘이 있는 곳과 없는 곳으로 구분되어 있는 것을 발견하게 됩니다. 이 아이콘 표시는 이선이 말한 것과 같이 다르게 운영이 되고 있는 기준이라고 볼 수 있습니다. 사람 아이콘이 있는 호수는 항상 사람이 상주하여 운영되고 사람 아이콘이 없는 호수는 무인으로 운영되고 있는 자료실이라고 생각할 수 있습니다. 그리고 그 안에는 숫자가 써 있는데 그 숫자는 미션 1 에서 주어졌던 해당 호수의 면적을 나타냅니다. 사람 아이콘 표시의 유무에 따라 호수의 면적이 어떻게 다른지 유추해보고 실제로 사람 아이콘이 있는 호수의 최대, 최소 면적을 찾아보고 사람 아이콘이 없는 호수의 최대, 최소 면적을 찾아봐서 그 값을 비교해 봅니다. 학생이 자연스럽게 23 평보다 작은 평수의 호수를 선별할 수 있도록 유도합니다.

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

## # 비교 연산자

- 숫자, 문자열 등 두 항목을 서로 비교하는 연산자

파이썬 연산자 기호	의미
$x == y$	x 와 y 가 같다
$x != y$	x 와 y 는 같지 않다
$x < y$	x 는 y 보다 작다
$x <= y$	x 와 y 보다 작거나 같다

$x > y$	x 는 y 보다 크다
$x \geq y$	x 는 y 보다 크거나 같다

- 결과는 True 또는 False 반환

ex) `1 != 'one'`

`30 > 70`

`'Alpha' <= 'Avocado'`

### # 조건문

- 조건의 참과 거짓을 판단하는 제어문
- 구조 : if

if 조건:

수행할 문장

ex) `rain = True`

`if rain:`

`print("it's raining")`

- 사용할 API 에 대해 설명합니다.

### < API >

- 1) `range(값)` : 연속된 정수를 만드는 함수
- 2) `append('요소')` : 리스트 마지막에 요소 추가
- 3) `sum('항목')` : 반복 가능한 항목의 합계를 구하는 함수
- 4) `print(값)` : 값을 출력하는 함수

→ `range` 함수는 () 안에 값을 넣으면 그 값보다 1 작은 연속된 정수를 만들어 줍니다.

→ `append` 함수는 () 안에 넣은 요소를 해당 리스트 마지막에 추가해 줍니다.

[ 코드 이해 ]

→ sum 함수는 () 안에 넣은 항목의 합계를 구해 줍니다.

- 학생이 스스로 코드를 구현하여 정답을 찾아 확인하도록 합니다.
- 스토리 속에서 해결해야 할 미션에 대해 질문하며 해당 코드를 학생 스스로 구현하도록 설명해줍니다. 코드는 제공하지 않습니다.

#### < CODE >

```
invest_list = []
invest_area = []
for i in range(len(room_area)):
    if room_area[i] < 23:
        invest_list.append(i+1)
        invest_area.append(room_area[i])
print(sum(invest_area))
#선별한 호수 총면적 : 271

print(invest_list)

#정답 확인 :
[5, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 23, 30, 33]
```

→ 드론이 필수적으로 탐지해야할 호수를 선별하기 위한 프로그램을 작성하기 위해 어떻게 해야할지 이야기 나누어 봅니다.

(예시) 20 층 내의 전체 호수 면적 리스트인 room\_area 의 모든 요소값을 확인하는 것을 반복합니다. 각 호수의 면적을 i 라는 변수에 대입하여 해당 면적이 23 평보다 작은 면적을 가진 호수만을 조사할 호수 리스트인 invest\_list 에 추가하도록 합니다. 이 때 구하고자 하는 것은 어떤 호수들을 감지해야하는지 알아야 하기 때문에 invest\_list 에는 해당 호수를 추가하도록 합니다. 하지만 우리가 선별한 호수 전체의 면적이 드론이 감지할 수 있는 면적인 300 평 이하가 맞는지도 확인해야 하므로 선별한

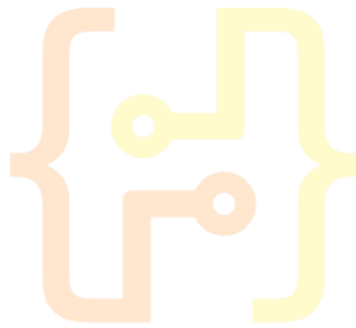
호수의 면적을 저장할 `invest_area` 리스트를 만들고 여기에 해당 면적을 추가하여 총면적을 확인해 300 평 이하인지 확인해 봅니다.

→ 실행결과 창에 출력된 값을 정답창에 입력하여 확인하도록 합니다.

(예시) `invest_list` 변수에 저장하여 출력한 후 그대로 입력하여 정답을 확인하도록 합니다.

→ 스스로 코딩하도록 유도합니다.

→ 미션 3 템플릿으로 이동합니다.



히우코딩  
HOW CODING

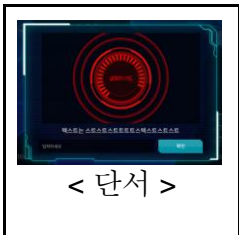
## Mission3. 폭발물이 설치된 호수 찾기.

## ● 미션의 목표를 설명합니다

## [ 미션 목표 ]

- 1) 데이터의 구조를 파악할 수 있습니다.
- 2) 0 과 1 로만 나타내는 2 진수에 대해 알 수 있습니다.
- 3) 조건의 참과 거짓을 판단하는 비교연산자를 사용할 수 있습니다.

## [ 스토리 이해 ]



## ● 스토리의 내용을 이해합니다.

- 이전 미션에서 나온 프로그래밍 결과를 생각해보며 스토리를 확인합니다.

"이선", "드론이 각 호수를 돌아다니며 데이터를 보내고 있대."

"파이", "응 나도 보고 있어."

"파이", "드론이 보내온 데이터를 분석프로그램에 돌려서 기준 농도 62 를 초과하는 곳을 찾아야 해."

"이선", "알았어. 기다릴게."

<<< 단서 확인하기 >>>

<단서>

화학물질 농도가 높은 호수를 찾아라.

→ 드론이 보내주는 데이터에서 어떤 곳이 폭발물이 설치되었을 가능성이 있는지 이야기 나누어 봅니다.

(예시) 드론은 화학물질 농도를 감지하고 있는데 보내온 데이터를 분석하였을 때 화학물질 기준 농도 62 보다 높은 곳을 찾으면 폭발물이 설치되어 있을 가능성이 큼니다.

## [ 학습 자료 ]

## ● 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

## # 2 진수란?

- 이진수, binary number
- 컴퓨터가 이해할 수 있는 0 과 1 만으로 나타내는 수

십진수	0	1	2	3	4	5
이진수	0000	0001	0010	0011	0100	0101

- 표기 : 0b 이진수값  
ex) 0b10100, 0b111001

## [ 코드 이해 ]

- 사용할 API 에 대해 설명합니다.

## &lt; API &gt;

- 1) int(값) : 값을 정수로 변환하는 함수
- 2) print(값) : 값을 출력하는 함수

- 학생이 스스로 코드를 구현하여 정답을 찾아 확인하도록 합니다.
- 스토리 속에서 해결해야 할 미션에 대해 질문하며 해당 코드를 학생 스스로 구현하도록 설명해줍니다. 코드는 제공하지 않습니다.

## &lt; CODE &gt;

```
#data
dron_sensing =
[[5,0b110000],[9,0b011101],[10,0b100100],[11,0b111100],[12,0b11110
1],[14,0b10110],[15,0b100101],[16,0b110000],[17,0b110011],[18,0b10
0100],[19,0b111100],[23,0b111101],[30,0b111101],[33,0b111111]]

for room, value in dron_sensing:
    if int(value) > 62:
        print(room)

#정답 확인 :
```

33

→ 드론이 보내온 데이터를 어떻게 분석할지 이야기 나누어 봅니다.

(예시) 드론이 감지한 호수의 화학물질 농도를 각 데이터로 보내오는데 이 데이터는 리스트가 리스트 안에 들어 있는 형태인 2 차원 리스트 입니다. 내부 리스트는 [호수, 농도] 로 되어 있는데 농도를 나타내는 요소는 2 진수 표기법으로 되어 있습니다. 드론이 보내는 센싱데이터는 컴퓨터가 빨리 분석하여 처리할 수 있도록 2 진수로 화학물질의 농도를 보내옵니다. 이 농도를 정수(10 진수)로 변환하였을 때 기준 농도가 62 보다 큰 곳을 찾으면 폭발물이 설치된 호수를 알 수 있습니다.

→ 결과 창에 출력된 값을 입력 버튼을 누르고 정답을 확인하도록 합니다.

(예시) 폭발물을 설치한 호수는 **33** 입니다. 정답을 확인합니다.

→ 스스로 코딩하도록 유도합니다.

→ 마무리 템플릿으로 이동합니다.

## Summary.

- 이번 차시의 미션을 해결하면서 배웠던 프로그래밍 개념과 파이썬 API 를 복습합니다.

### # 반복문

- 같은 일을 반복할 때 사용하는 제어문
- 순서가 있는 자료형에서만 사용
- 구조 : for \_\_ in

for 변수 in 반복할 조건:

수행할 문장

ex) numbers = [1, 2, 3]



```
for v in numbers:
    print(v)
```

### # 리스트 합계

- 리스트 요소의 합계
- 요소들을 변수에 반복적으로 더해 저장

```
for 변수 1 in 리스트:
    변수 2 = 변수 2 + 변수 1

ex) numbers = [1, 2, 3, 4, 5, 6, 7]
adding = 0
for i in numbers:
    adding = adding+i
```

### # 비교 연산자

- 숫자, 문자열 등 두 항목을 서로 비교하는 연산자

파이썬 연산자 기호	의미
$x == y$	x 와 y 가 같다
$x != y$	x 와 y 는 같지 않다
$x < y$	x 는 y 보다 작다
$x <= y$	x 와 y 보다 작거나 같다
$x > y$	x 는 y 보다 크다
$x >= y$	x 는 y 보다 크거나 같다

- 결과는 True 또는 False 반환

ex) `1 != 'one'`

`30 > 70`

`'Alpha' <= 'Avocado'`

## # 조건문

- 조건의 참과 거짓을 판단하는 제어문
- 구조 : if

if 조건:

수행할 문장

ex) `rain = True`

if rain:

`print("it's raining")`

## # 2 진수란?

- 이진수, binary number
- 컴퓨터가 이해할 수 있는 0 과 1 만으로 나타내는 수

십진수	0	1	2	3	4	5
이진수	0000	0001	0010	0011	0100	0101

- 표기 : 0b 이진수값

ex) `0b10100`, `0b111001`

## # API

- `range(값)` : 연속된 정수를 만드는 함수
- `append('요소')` : 리스트 마지막에 요소 추가
- `sum('항목')` : 반복 가능한 항목의 합계를 구하는 함수

- print(값) : 값을 출력하는 함수
- int(값) : 값을 정수로 변환해주는 함수

## 더 나아가기

- 학습한 코드를 응용하도록 추가 질문을 합니다. 기존 코드에서 수정 또는 처음부터 다시 구현해 보도록 합니다.

### < Mission 1 >

→ 20 층의 전체 호수에 대한 면적 평수를 나타낸 리스트중, 30 평 미만인 호수들의 전체 면적을 구해야 한다면 어떻게 할 수 있을까요?

(예시) 면적이 30 이하라는 조건에 충족하는 값만을 더할 수 있도록 for 반복문 아래에 if 조건문을 추가해주면 원하는 범위 내의 값의 합을 구할 수 있습니다.

### < Mission 2 >

→ 20 층에서 드론이 감지해야 할 호수로 선별된 리스트 중, 전체 호수의 평균보다 큰 호수에는 추가적으로 드론을 더 투입하기로 했다면, 탐지할 호수의 평수는 어떻게 알아낼 수 있을까요?

(예시) sum 으로 구한 전체 호수의 면적 값을 len 로 구한 값으로 나누면 리스트의 평균값을 구할 수 있습니다. 알아낸 리스트의 평균값을 기준으로 이를 초과하는 방의 면적을 구하는 코드를 작성하여 풀 수 있습니다.

### < Mission 3 >

→ 드론이 보내온 각 호수의 데이터를 분석하여 화학물질 농도가 낮은 순서부터 출력하려면 어떻게 해야 할까요?

(예시) 드론이 보내온 데이터를 리스트 안에 리스트가 있는 2 차원 리스트 형태로 데이터를 10 진수로 변환하는 int 함수를 사용하여 모든 요소를 변환한 후에 정렬 함수인 sort 를 사용하여 작성하도록 합니다.

## 평가 기준

평가 내용	1~5	강사 메모
<b>학습</b>		
사건 해결에 필요한 데이터를 스토리 속에서 파악할 수 있다.		
반복 제어문을 사용하여 리스트 요소의 합계를 구할 수 있습니다.		
항목의 값을 서로 비교하는 비교 연산자에 대해 알고 적절하게 사용할 수 있습니다.		
2 차원 리스트의 형태를 알고 각 리스트의 값에 접근할 수 있습니다.		
주요 함수를 사용하여 프로그래밍하고 결과 값을 찾아 미션을 해결할 수 있다.		
<b>태도</b>		
어려운 점이 있어도 포기하지 않고 끝까지 해결하려고 노력하였다.		

## 모범 답안

코드	
Mission1	#data

	<p><b>#호수별 면적리스트(총 33 호)</b></p> <pre>room_area = [24,24,35,32,8,27,23,35,19,19,18,22,23,19,22,22,22,18,22,30,30,23,22,30,30,23,23, 28,25,20,23,30,18]</pre> <p><b>#20 층 전체 호수 총면적 구하기</b></p> <pre>area_sum=0 for i in room_area:     area_sum = area_sum + i print(area_sum)</pre> <p><b>#정답 확인 : 789</b></p>
<b>Mission2</b>	<pre>invest_list = [] invest_area = [] for i in range(len(room_area)):     if room_area[i] &lt; 23:         invest_list.append(i+1)         invest_area.append(room_area[i]) print(sum(invest_area)) #선별한 호수 총면적 : 271  print(invest_list)</pre> <p><b>#정답 확인 : [5, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 23, 30, 33]</b></p>
<b>Mission3</b>	<p><b>#data</b></p> <pre>dron_sensing = [[5,0b110000],[9,0b011101],[10,0b100100],[11,0b111100],[12,0b111101],[14,0b101 10],[15,0b100101],[16,0b110000],[17,0b110011],[18,0b100100],[19,0b111100],[23, 0b111101],[30,0b111101],[33,0b111111]]</pre> <pre>for room, value in dron_sensing:     if int(value) &gt; 62:</pre>

	<pre>print(room)</pre> <p>#정답 확인 : 33</p>
<p>더 나아가기 Mission1</p>	<pre>area_sum2=0 for i in room_area:     if i &lt; 30:         area_sum2 = area_sum2 + i print(area_sum2)</pre> <p>#정답 확인 : 537</p>
<p>더 나아가기 Mission2</p>	<pre>extra_area = [] for i in range(len(invest_list)):     if invest_list[i] &gt; int(sum(invest_area)/len(invest_area)):         extra_area.append(invest_list[i]) print(extra_area)</pre> <p>#정답 확인 : [23, 30, 33]</p>
<p>더 나아가기 Mission3</p>	<pre>sort_list = [] for room, value in dron_sensing:     sort_list.append(int(value))     sort_list.sort() print(sort_list)</pre> <p>#정답 확인 : [22, 29, 36, 36, 37, 48, 48, 51, 60, 60, 61, 61, 61, 63]</p>