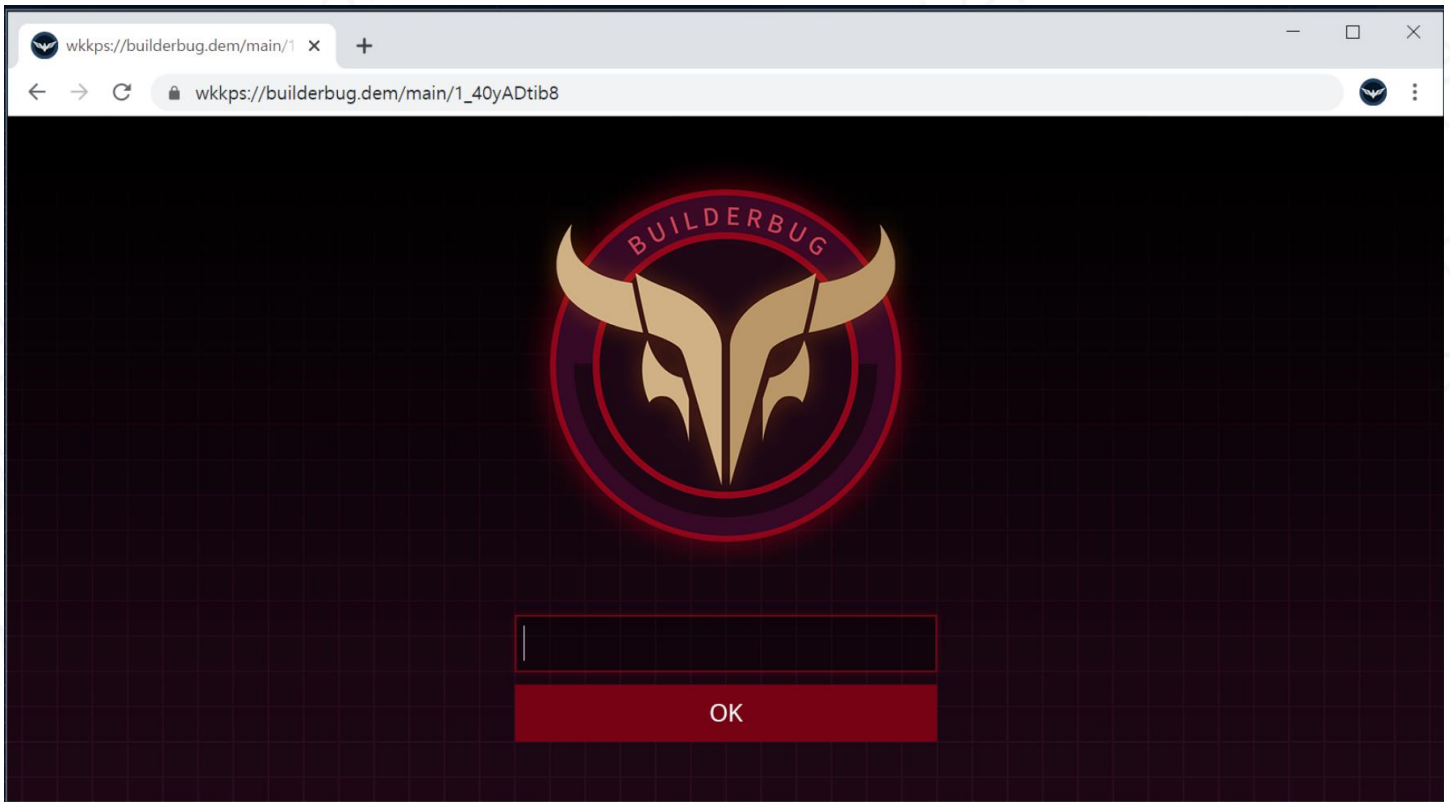


[빌더버그의 메시지 해독 작전]



시청에 폭발물을 설치한 범인을 찾기 위해 단서를 분석하던 중,
cctv 기록 파일 일부가 사라졌음을 알게 된 파이와 이선은 해당 시간대에 시청에 출입한
명단을 요청하여 분석했다. 해독한 직원 ID 리스트에서 수상한 URL 이 발견되는데...

오늘의 목표

빌더버그의 웹사이트에 접속하여 단서 수집하기

주요 함수

print, int, bin, range, append, join

> 학습 목표

- 스토리의 **사건**을 이해하며 미션을 해결하기 위해 필요한 데이터를 수집할 수 있습니다.
- 이진수를 표기하는 방식을 이해하고 문자열로 표현된 이진수 값을 숫자 데이터로 변환할 수 있습니다.

- 중첩 반복문의 구조를 알고 여러 조건에 따라 반복 수행하는 프로그램을 작성할 수 있습니다.
- 조건 제어문을 사용하여 참과 거짓 조건에 따라 어떤 일을 수행하는 프로그램을 만들 수 있습니다.
- 리스트의 요소들을 합하고 구분자가 추가된 문자열로 변환할 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)

- 이전 차시 스토리에 관해 이야기를 나눕니다.

[이전 스토리
이해]

〈이전 스토리〉

이선과 파이는 폭발물을 설치한 범인을 찾아내기 위해 시청 20 층 33 호의 cctv 기록 파일을 확인하던 도중, 저장된 기록 파일의 개수에 이상이 있음을 발견했다. 누군가 조작하여 사라진 파일 목록을 검색하여 범행이 일어났을 시간대를 알아냈고, 해당 시간대에 시청 로비에 출입한 명단을 확인하였다. 특정된 용의자들을 취조하기 위해 시청 직원 명부의 ID 를 해독하던 도중, 수상한 URL 을 발견한다.

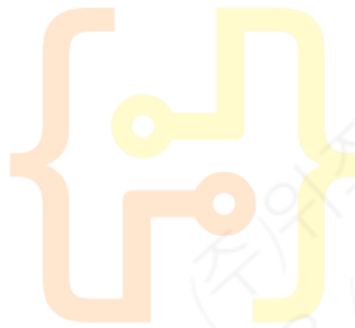
→ NIS 시스템에 접속하기 위한 ID 와 PW 를 학생에게 전달합니다.

- NIS 정보 요원이 되어 사건을 해결할 ID 를 부여합니다.
- ID : agent@nis.com / PW : python

히우코딩
HOW CODING

Mission1. 웹사이트에 접속하기.

- 미션의 목표를 설명합니다.



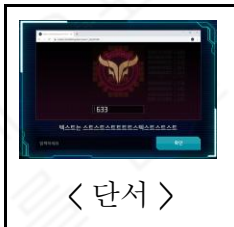
히우코딩
HOW CODING

[미션 목표]

- 1) 스토리를 이해하고 웹사이트 내에서 단서를 찾기 위해 무엇을 해야 하는지 알 수 있습니다.
- 2) 이진수 표기법을 이해하고 문자열로 표현된 이진수 값을 숫자 데이터로 변환할 수 있습니다.

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

[스토리 이해]



(시청 출입 명부를 조회했더니 출처를 알 수 없는 URL 이 발견되었다.)

(파이와 이선은 URL 에 접속하여 단서를 찾아보려 한다.)

"이선", "이 URL 주소가 빌더버그 조직과 관련이 있을까?"

"파이", "폭발물을 설치한 것이 빌더버그 조직의 소행임이 밝혀졌으니, 이 URL 을 이용해 범죄 계획을 꾸밌을지도 몰라."

"이선", "지금쯤 예정대로 폭발물이 터지지 않았으니 계획을 변경하고 있지 않을까?"

"파이", "이런 형태의 URL 은 보안 문제로 바로 접속이 불가능해."

"이선", "그럼 어떻게 해야 하지?"

"파이", "내가 우회할 방법을 찾아볼게."

"이선", "응, 고마워."

(파이가 IP 를 우회하여 웹 사이트 접속에 성공했다.)

"파이", "접속에 성공했어. 웹 사이트가 나오는군."

"이선", "접속 비밀번호를 입력하라는 화면이야. 패스워드가 뭘까?"

"파이", "여기 있는 그림들이 괜히 있는 것은 아닐거야."

"이선", "이런 형태의 표는 처음봐. 옆에 적힌 숫자들은 뭘지?"

"파이", "내가 한 번 봐볼게."

"이선", "네게 맡길게, 파이."

<<< 단서 확인하기 >>>

<단서>

웹 사이트에 접속하기 위한 패스워드를 알아내라.

			@	@	@	@				0001111000 → 120
		@					@			0010000100 → 132
	@							@		0100000010 → 258
@			@			@			@	1001001001 → 585
@									@	1000000001 → 513
@									@	1000000001 → 513
@			@	@	@	@			@	1001111001 → 633
	@							@		0100000010 → 258
		@					@			0010000100 → 132
			@	@	@	@				0001111000 → 120

→ 패스워드 입력 창과 함께 나타난 그림과 숫자들이 무엇을 의미하는지 이야기 나누어 봅니다.

(예시) 왼쪽편의 그림은 10*10 격자 칸 안에 문자로 채워져 있거나, 아무것도 없습니다. 그림 오른쪽의 0001111000 과 같은 수는 각 행의 문자와 공백 여부를 1 과 0 으로 나타낸 것입니다. 10 자리로 표현된 이진수를 십진수로 변환한 결과가 화살표 오른쪽의 값입니다. 변환 과정이 드러나 있는 수들의 이진수와 십진수 값이 일치하는지 확인한 다음, 동일한 방법으로 '?'로 표시된 행의 값을 알아낼 수 있습니다.

[학습 자료]

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

2 진수

- 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'

문자열→숫자 변환

- 문자열을 10 진수 정수로 변환
- 10 진수 외 진수 변환시 진법 표기
 - int('문자열')
 - ex) int('30')
 - int('0b101010', 2)
 - int('0x72', 16)

→ 이진수는 각 자릿수에 0 과 1 두개의 수를 이용하여 수를 표기합니다. 컴퓨터 내부에서는 여러 신호를 이진수로 표현하고 각 자리를 1 bit 로 다룸으로써 이진수는 컴퓨터 내 정보의 최소 단위가 됩니다. 이진수를 십진수로, 혹은 십진수를 이진수로 변환하여 같은 값을 달리 나타낼 수 있는데 이것을 진법 변환이라 합니다. 파이썬에서는 정수값을 입력할 때에 '4'와 같이 숫자만을 표기하면 십진수로 인식하게 됩니다. 입력하는 수가 이진수임을 나타내기 위해서는 이진수값의 앞자리에 'binary'를 의미하는 머리글자 '0b'를 추가하여 '0b0100'과 같이 작성해주면 됩니다.

→ 십진수와 이진수 이외에도 8 진수, 16 진수와 같은 진법이 존재합니다. 큰 수를 간결하게 표현하거나 색상 정보 표기 등 특수한 영역에서 쓰이는 진수들입니다. 이진수, 8 진수, 16 진수의 영문 이름의 앞글자를 딴 파이썬 내장함수 bin(), oct(), hex()가 존재합니다. 이들 함수를 이용하여 숫자를 해당 진수 형태의 문자열로 변환할 수 있습니다. 이번 차시에서는 이진수 문자열 변환을 배웁니다.

→ int()함수는 문자열을 십진수 정수로 변환하는 함수입니다. 십진수가 아닌 진법의 문자열을 int()함수를 이용하여 변환할 때에 변환할 값과 함께 그 값이 몇 진수인지를 표기해주어야 합니다. 이진수는 ('0b101010', 2), 16 진수는 ('0x72', 16)과 같이 진수 정보를 표기하여 함수에 입력해 줍니다.

- 사용할 API에 대해 설명합니다.

< API >

- 1) int('문자열') : 문자열을 십진수 정수로 변환하는 함수
- 2) print(값) : 값을 출력하는 함수

[코드 이해]

- 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 >

#첫 번째 행 정답 확인하기

```
first_row = '0b0001111000'
print(int(first_row, 2))
```

#비어있는 행 정답 구하기

```
key_value = '0b1001111001'
print(int(key_value, 2))
```

#정답 확인:

633

→ 주어진 data에 대해 이야기 나누어 봅니다.

(예시) 10*10 표의 각 칸에는 어떤 문자가 들어 있거나, 아무것도 없습니다. 표 오른쪽에는 표의 각 행을 나타낸 숫자들이 표현되어 있습니다. 아무것도 없는 칸은 0, 문자가 포함된 칸을 1로 해석하여 이진수를 대입해 보면 각 행의 문자 포함 여부를 이진수로 나타내었다는 것을 알 수 있습니다. 이진수를 한 번 더 십진수로 변환한 결과는 오른쪽 화살표의 값과 같습니다.

→ 비어있는 행의 값을 구하기 위해 어떻게 할지 이야기 나누어 봅시다.

(예시) 먼저 표의 정보로부터 이진수와 십진수로 표현된 값의 변환 결과가 옳은지 확인해봅니다. 첫 번째 행을 대상으로 변환 함수를 사용하여 이진수

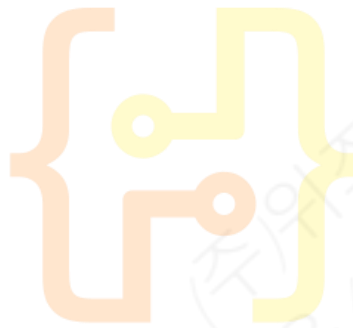
값을 입력하면 화살표 오른쪽의 십진수 정수 값이 반환되는지 확인합니다.
이진수 값 '0b0001111000'을 표기법을 지켜 입력하여 변수를 선언한 뒤,
int() 함수 내에 알맞게 입력합니다. 변환하려는 문자열이 이진수 값을
나타내는 2 를 콤마 뒤에 작성합니다.

→ 실행결과 창에 출력된 값을 정답창에 입력하여 확인하도록 합니다.

(예시) 출력된 값인 **633** 를 입력한 후 정답을 확인하도록 합니다.

→ 스스로 코딩하도록 유도합니다.

→ 미션 2 템플릿으로 이동합니다.



히우코딩

HOW CODING

Mission2. 웹사이트의 정체와 빌더버그의 지령 알아내기.

● 미션의 목표를 설명합니다

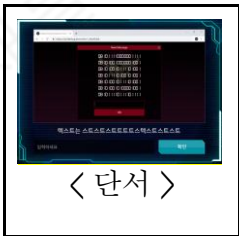
[미션 목표]

- 1) 스토리 속에서 단서를 확인하여 암호화된 지령을 해독합니다.
- 2) 중첩 반복문의 구조를 알고 여러 조건에 따라 반복 수행하는 프로그램을 만들 수 있습니다.
- 3) 조건 제어문을 사용하여 참과 거짓 조건에 따라 어떤 일을 수행하는 프로그램을 만들 수 있습니다.

[스토리 이해]

● 스토리의 내용을 이해합니다.

- 이전 미션에서 나온 프로그래밍 결과를 생각해보며 스토리를 확인합니다.



"이선", "접속하자마자 이런 화면이 뜨는데? 이 글은 뭐지?"

"파이", "읽어보니 조직 내 준수해야 할 원칙을 명시해놓은 글귀 같은데."

"이선", "이 아래 빈 입력 칸이 있어, 파이. 뭔가를 입력해야 하는걸까?"

"파이", "내 생각엔 그래. 마땅한 힌트가 될만한 것은 더 없어 보이는데."

"이선", "한 번 아무거나 입력해볼까?"

"파이", "해킹 사이트일지도 몰라. 위험하니 성급하게 조작해서는 안돼."

(쪽지가 도착했다.)

"폭발물 설치의 잘 끝냈겠지? 증거 인멸은 확실히 할 것. 다음 지령이다. 사흘 안에 완료하도록."

"파이", "뭐지? 쪽지가 도착했어."

"이선", "이건 시청에 폭발물을 설치했던 범인에게 내려온 지령이야."

"파이", "이 계정의 주인은 이 비밀 웹사이트를 통해 빌더버그의 지령을 전달받아 수행해왔군."

"이선", "파일에 다음 지시사항이 담겨있다고 했으니 파일을 열어보자."

"파이", "그래. 우리가 먼저 계획을 알아내 행동을 개시할 수 없도록 막아야 해."

"이선", "파이, 파일을 열어 봤는데 알 수 없는 리스트야. 이건 지시사항처럼 보이지 않는데."

"파이", "이런, 지령 역시 암호화 시켜놓았군. 이 숫자들과 관련이 있을 거야."

<<< 단서 확인하기 >>>

<단서>

쪽지에 암시된 지령의 내용을 밝혀라.

→ 사이트에서 도착한 쪽지로부터 지령 내용을 알아내기 위해 무엇을 해야 하는지 이야기 나누어 봅니다.

(예시) 빌더버그 조직의 직원에게 전달된 쪽지 안에는 다음 지령 내용이 있다고 했습니다. 그와 함께 이진수 리스트가 전달되었는데 이전 미션에서와 같이 암호화된 정보를 풀어내야 그 내용을 확인할 수 있습니다.

→ 지령 내용과 함께 전달된 이진수 리스트의 의미가 무엇인지 이야기 나누어 봅니다.

(예시) 이전 미션에서의 표 해석 값 힌트를 기억하여 그와 연계하여 추리해 봅니다. 힌트로 주어졌던 이진수 값 하나는 표에서 행 하나를 의미했었습니다. 문자로 채워지거나 그렇지 않은 각 칸의 상태를 1 과 0 로 옮기고, 이진수로 나타내어 해독하면 십진수 정수값이 됩니다. 주어진 이진수 리스트의 1 과 0 이 그러한 표의 상태를 의미하고 있다고 가정한 뒤, 이들 각각이 어떤 표의 상태를 나타내도록 출력하여 결과가 어떻게 되는지 확인해 봅니다.

[학습 자료]

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

2 진수

- 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'

숫자→2 진수 문자열 변환

- 숫자를 2 진수 형태의 문자열로 변환
 - bin(숫자)
 - ex) bin(42)

이중 반복문

- 같은 일을 반복하는 반복문 구조를 중첩시킨 반복문
- 반복문 안에 반복문이 들어있는 형태
- 반복 조건이 다른 데이터에 접근할 때 사용
- 외부 반복문의 1 번 실행마다 내부 반복문 실행 동작을 반복
- 구조 : for __ in:
for __ in:

for 변수 in 반복할 조건 1:

for 변수 in 반복할 조건 2:

수행할 문장

```
ex)
star = 10
for i in range(star):
    for j in range(i):
        print('*',end='',flush=True)
    print('\n')
```

조건문

- 조건의 참과 거짓을 판단하는 제어문
- 구조 : if - elif - else

if 조건 1:

수행할 문장 1

elif 조건 2:

수행할 문장 2

else:

수행할 문장 3

```
weather = 'rain'
```

```
if weather == 'rain':
```

```
    print('it's raining')
```

```
elif weather == 'snow':
```

```
    print('it's snowing')
```

```
elif weather == 'cloud':
```

```
    print('it's cloudy')
```

```
else:
```

```
    print('it's sunny')
```

print 줄바꿈 제거

- 파이썬 print 함수로 값 출력시 자동적으로 줄바꿈('wn') 출력
ex) for v in range(10):
 print('number:', v)
- 줄바꿈 없이 출력 : 출력할 값 뒤에 (, end = "", flush = True) 추가
ex) for i in range(10):
 print('*', end='', flush=True)

- 사용할 API 에 대해 설명합니다.

< API >

- 1) bin(숫자) : 숫자를 2 진수 문자열로 변환하는 함수

[코드 이해]

- 2) len('문자열') : 문자열의 길이 반환
- 3) range(값) : 연속된 정수를 만드는 함수
- 4) print(값) : 값을 출력하는 함수

→ bin 함수는 괄호 안에 숫자를 넣으면 그 값을 이진수 문자열로 변환하는 함수입니다.

- 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 : 모범답안에서 확인

#2 진수 리스트를 문자 행으로 나타내기

```
for i in range(len(num_list)):
    for j in bin(num_list[i]):
        if j == '1':
            print('###', end='', flush = True)
        elif j == '0':
            print(' ', end='', flush = True)
        else:
            print('\n')
    print('\n')
```

#정답 확인 :

18e0

→ 리스트의 이진수 값을 행으로 하여 표로 나타내는 프로그램을 작성하기 위해 어떻게 해야할지 이야기 나누어 봅니다.

(예시) 이진수 리스트의 1 과 0 을 표의 각 칸으로 간주하여 나타낼 때에 간주하여 출력합니다. 이전 미션의 힌트에서 이진수의 1 과 0 을 칸 내부 문자 존재 유무로 나타내었듯 표시하면 됩니다. 1 과 0 을 확연히 구분하기

위하여 어떤 문자를 사용해야 할지 이야기 나누어 보고 적절한 문자를 고릅니다. 0 은 아무것도 없음을 의미해야 하므로 ' '(공백)이 적절합니다. 1 은 '#'나 '@' 등 적당한 문자를 선택합니다. 검사해야 할 조건이 2 개 이상이므로 if elif else 조건문을 사용합니다. 반복할 조건 리스트 안에서 각 요소를 검사하여 값이 0 이라면 공백을, 1 이라면 문자를 출력하는 코드를 작성합니다.

→ 이진수의 0 과 1 값을 다른 문자로 대체하여 출력할 때, 각 행을 구분하여 하나의 표와 같이 나타내려면 어떻게 해야 하는지 이야기 나누어 봅니다.

(예시) 파이썬에서는 print 함수를 사용해 값을 출력할 때 자동으로 값의 끝에 줄바꿈이 추가되어 출력됩니다. 이것을 제거하기 위해서는 print 함수 안에 출력할 값과 함께, end = '', flush = True 를 추가적으로 작성하여 줍니다. 또한 실행 결과를 함께 보며 문자의 개수 하나씩 추가하여 코드를 수정합니다. 식별 가능한 상태가 될 때까지 2~3 개의 문자를 추가하여 출력하도록 합니다.

→ 실행결과 창에 출력된 값을 정답창에 입력하여 확인하도록 합니다.

(예시) 출력된 값이 나타내고 있는 정보가 18e0 임을 알아냅니다. 정답 확인하기 창에 직접 18e0 를 입력하여 확인하도록 합니다.

→ 스스로 코딩하도록 유도합니다.

→ 미션 3 템플릿으로 이동합니다.

HOW CODING

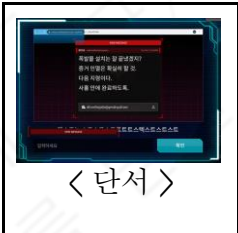
Mission3. 지령 내용 해독하기.

● 미션의 목표를 설명합니다

[미션 목표]

- 1) 스토리 속에서 단서를 확인하여 암호화된 지령을 올바르게 해독합니다.
- 2) 조건 제어문을 사용하여 참과 거짓 조건에 따라 어떤 일을 수행하는 프로그램을 만들 수 있습니다.
- 3) 리스트의 요소들을 합하고 구분자가 추가된 문자열로 변환할 수 있습니다.

[스토리 이해]



● 스토리의 내용을 이해합니다.

- 이전 미션에서 나온 프로그래밍 결과를 생각해보며 스토리를 확인합니다.

"이선", "번호를 입력했더니 파일이 열렸어."
"파이", "이런 글귀는 전에 봤는데."
"이선", "읽어 봤더니 여기 뭔가 이상한 점이 보이는데."
"파이", "나도 보고 있어. 텍스트가 부분적으로 손상된 것 같아."
"이선", "왜 그렇지 확인해봐줘, 파이."

<<< 단서 확인하기 >>>

<단서>

지령 정보에 담긴 비밀 메시지를 해독하라.

→ 지령 데이터를 보고 어떤 정보를 해독해야 하는지 이야기 나누어 봅니다.
(예시) 암호를 입력하여 열린 지령 데이터는 어떤 문서인데, 웹 페이지의 초기 화면에서 보았던 조직의 규율 문서와 동일한 내용입니다. 다만 앞부분을 읽어보면 원본 문서와 다른 점을 알아챌 수 있습니다. 내용이 유사하지만 원본 문서에서 몇 개의 문자가 빠져 있고 ‘-’로 대체되어 있습니다. 빠진 문자를 모두 조합하면 어떤 내용이 나올지 유추해 보고, 파이썬 프로그램으로 이를 해결할 방법에 대해 이야기 나누어 봅니다.

[학습 자료]

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

리스트→문자열 변환

- ‘구분자’.join(리스트)
- 리스트 요소들을 합하여 구분자가 추가된 문자열로 변환
ex) month = ['January', 'February', 'March', 'April', 'May', 'June']
halfyear = ','.join(month)
print(halfyear)

- 사용할 API 에 대해 설명합니다.

< API >

- 1) len(‘문자열’) : 문자열의 길이 반환
- 2) append(‘요소’) : 리스트 마지막에 요소 추가
- 3) join(리스트) : 리스트를 문자열로 변환하는 함수
- 4) range(값) : 연속된 정수를 만드는 함수
- 5) print(값) : 값을 출력하는 함수

[코드 이해]

- 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 : 모범답안에서 확인
#숨겨진 문자 찾아 해독하기
lines = []
for i in range(len(missing)):
    if missing[i] == '-':
        lines.append(paragraph[i])

print(''.join(lines))
```

#정답 확인 :

you have been doing well so far. but it was too easy. The real game is waiting. good luck.

→ 두 개의 데이터를 비교하여 어떻게 분석할지 이야기 나누어 봅니다.

(예시) 지령 문서에는 원본 문서에서 몇 개의 문자가 빠져 있고 ‘-’로 대체되어 있습니다. 두 문서는 문자열 변수에 저장되어 있습니다. 반복문을 통해 지령 문서 missing 의 ‘-’인 지점을 차례로 검색하여 조건 ‘-’과 일치하는 인덱스를 찾습니다. 찾은 인덱스는 원본 문자열 paragraph 의 문자에 대응하므로 paragraph 내의 해당 인덱스 값을 새로운 리스트 lines 에 저장합니다. 모든 검사가 끝나면 반복문을 빠져나옵니다.

→ 빠진 문자를 찾아 저장한 리스트를 하나의 문자열로 읽어낼 수 있도록 가공하려면 어떻게 해야 할까요?

(예시) 리스트에는 문자들이 ‘-’로 구분되어 저장되어 있기 때문에 읽고 해석하기 어렵습니다. 파이썬의 내장함수 join 을 사용하면 리스트의 모든 요소들을 합하여 구분자가 추가된 문자열로 변환해 줍니다. 구분자 없이 문자열로 합하길 원한다면 join 앞부분 구분자를 첨가하는 부분에 “(빈 작은따옴표)만을 입력해주면 됩니다.

→ 출력된 값을 결과 창에 입력하여 정답을 확인하도록 합니다.

(예시) 숨겨진 비밀 메시지는 **you have been doing well so far. but it was too easy. The real game is waiting. good luck.** 입니다. 정답을 확인합니다.

→ 스스로 코딩하도록 유도합니다.

→ 마무리 템플릿으로 이동합니다.

Summary.

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

2 진수

- 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'

문자열→숫자 변환

- 문자열을 10 진수 정수로 변환
- 10 진수 외 진수 변환시 진법 표기
 - int('문자열')
 - ex) int('30')
 - int('0b101010', 2)
 - int('0x72', 16)

숫자→2 진수 문자열 변환

- 숫자를 2 진수 형태의 문자열로 변환
 - bin(숫자)
 - ex) bin(42)

이중 반복문

- 같은 일을 반복하는 반복문 구조를 중첩시킨 반복문
- 반복문 안에 반복문이 들어있는 형태
- 반복 조건이 다른 데이터에 접근할 때 사용
- 외부 반복문의 1 번 실행마다 내부 반복문 실행 동작을 반복
- 구조 : for __ in:
for __ in:

```
for 변수 in 반복할 조건 1:
```

```
    for 변수 in 반복할 조건 2:
```

```
        수행할 문장
```

ex)

```
star = 10
```

```
for i in range(star):
```

```
    for j in range(i):
```

```
        print('*',end='',flush=True)
```

```
    print('\n')
```

조건문

- 조건의 참과 거짓을 판단하는 제어문
- 구조 : if - elif - else

```
if 조건 1:
```

```
    수행할 문장 1
```

```
elif 조건 2:
```

```
    수행할 문장 2
```

```
else:
```

```
    수행할 문장 3
```

```
weather = 'rain'
```

```
if weather == 'rain':
```

```
    print('it's raining')
```

```
elif weather == 'snow':
```

```
    print('it's snowing')
```

```
elif weather == 'cloud':
```

```
    print('it's cloudy')
```

```
else:
```

```
    print('it's sunny')
```


print 줄바꿈 제거

- 파이썬 print 함수로 값 출력시 자동적으로 줄바꿈('n') 출력
ex) for v in range(10):
 print('number:', v)
- 줄바꿈 없이 출력 : 출력할 값 뒤에 (, end = "", flush = True) 추가
ex) for i in range(10):
 print('*', end='', flush=True)

리스트→문자열 변환

- '구분자'.join(리스트)
- 리스트 요소들을 합하여 구분자가 추가된 문자열로 변환
ex) month = ['January', 'February', 'March', 'April', 'May', 'June']
 halfyear = ','.join(month)
 print(halfyear)

더 나아가기

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

< Mission 1 >

→ 문자가 채워진 표와 각 행에 대한 해석으로 주어진 웹 사이트 접속 패스워드 힌트의 그림이 전혀 다른 것이었고, 아래와 같이 주어졌다면, 어떻게 패스워드를 알아낼 수 있을까요?

@				@	10001 → 17
---	--	--	--	---	------------

	@		@		01010 → 10
@		@		@	? → ?
	@		@		01010 → 10
@				@	10001 → 17

(예시) 표의 각 행의 문자 포함 여부를 이진수 문자 0 과 1 로 구분하여 옮겨쓰고, 이를 십진수 정수로 해독했던 과정과 동일하게 이진수 문자열을 십진수로 변환하여 알아냅니다.

< Mission 2 >

→ 지령을 전달하는 쪽지에 이진수 리스트가 아닌, 십진수 값이 저장된 리스트가 주어지고 이진수 암호 해독과 같이 풀라는 힌트가 있었다면, 어떻게 다른 문자로 대체하여 화면에 나타낼 수 있을까요?

`decimal_list = [48143, 41993, 42473, 48425, 42473, 42249, 48623]`

(예시) 주어진 십진수를 이진수로 변환하여 리스트에 저장한 다음, 이진수 리스트를 이용하여 0 과 1 을 다른 문자로 대체하는 프로그램으로 출력하면 됩니다. 십진수를 이진수로 변환할 때에는 `bin()` 함수를 사용합니다.

< Mission 3 >

→ 원본과 다른 문단의 텍스트에서 빠진 글자를 찾은 결과 문자열을 공백으로 구분하여 전문적인 분석을 위해 전달해야 했다면, 결과 문자열 데이터를 어떻게 가공할 수 있을까요?

(예시) 리스트의 요소를 문자열로 변환하는 `join()` 함수에 구분자 ‘ ’(공백)을 추가한 다음 리스트의 문자 데이터를 문자열로 변환합니다.

평가 기준

평가 내용	1~5	강사 메모
학습		

사건 해결에 필요한 데이터를 스토리 속에서 파악할 수 있다.		
이진수를 표기하는 방식을 이해하고 문자열로 표현된 이진수 값을 숫자 데이터로 변환할 수 있다.		
중첩 반복문의 구조를 알고 여러 조건에 따라 반복 수행하는 프로그램을 작성할 수 있다.		
조건 제어문을 사용하여 참과 거짓 조건에 따라 어떤 일을 수행하는 프로그램 만들 수 있다.		
리스트의 요소들을 합하고 구분자가 추가된 문자열로 변환할 수 있습니다.		
태도		
어려운 점이 있어도 포기하지 않고 끝까지 해결하려고 노력하였다.		

모범 답안

코드	
Mission1	<pre>#첫 번째 행 정답 확인 first_row = '0b0001111000' print(int(first_row, 2)) #비어있는 행 정답 key_value = '0b1001111001' print(int(key_value, 2)) #정답 확인 : 633</pre>

Mission2	<pre> #data num_list = [0b1011110000001111, 0b1010010000001001, 0b1010010111101001, 0b1011110100101001, 0b1010010111101001, 0b1010010100001001, 0b1011110111101111] #2 진수 리스트를 문자 행으로 나타내기 for i in range(len(num_list)): for j in bin(num_list[i]): if j == '1': print('###', end='', flush = True) elif j == '0': print(' ', end='', flush = True) else: print('\n') print('\n') #정답 확인 : 18e0 </pre>
Mission3	<pre> #data paragraph = 'We are a form of warfare. We aim to achieve maximum consequential impact for where asymmetric happens. We attack where finite input allocation of resources exists. objectives should be clearly defined and work norms and means adopted by the organization are acceptable to the individual and groups. Each Person is responsible for completing the work. Organisation will appoint the job should be done. should be set up in such a way that every individual should be assigned a duty according to his skill and qualification. The person should continue the same work so that he specialises in his work. This helps in increasing production in the concern. The scope of authority and responsibility should be clearly defined. Every person should know his work with definiteness. If the duties are not clearly assigned. then it will not be possible to fix responsibility also. Everybody's responsibility will become nobody's responsibility. The principle states that top management should interfere only when something goes wrong. If the things are done as per plans then there is no need for the interference of top management. The management should leave routine things to be supervised by lower cadres. It is only in exceptional situations when attention of top easy management is drawn. The principle relieves top management of many botherations and routine things. </pre>

Principle of exception allows top management to concentrate on planning and policy formulation. Important time of management is not wasted on avoidable supervision. The responsibility of the superior does not decrease once he has delegated authority. A person can delegate authority and not responsibility. We will remain for the work even if it is delegated to the subordinate accountable work. So the responsibility of superior and subordinate remains absolute.'

missing = 'We are a form of warfare. We aim to achieve maximum consequential impact for where asymmetric happens. We attack where finite input allocation of resources exists. objectives should be clearly defined and work norms and means adopted by the organization are acceptable to the individual and groups. Each Person is responsible for completing the work. Organisation will appoint the job should be done. should be set up in such a way that every individual should be assigned a duty according to his skill and qualification. The person should continue the same work so that he specialises in his work. This helps in increasing production in the concern. The scope of authority and responsibility should be clearly defined. Every person should know his work with definiteness. If the duties are not clearly assigned then it will not be possible to fix responsibility also. Everybody's responsibility will become nobody's responsibility. The principle states that top management should interfere only when something goes wrong. If the things are done as per plans then there is no need for the interference of top management. The management should leave routine things to be supervised by lower cadre. It is only in exceptional situations when attention of top management is drawn. This principle believes top management of many botherations and routine things. Principle of exception allows top management to concentrate on planning and policy formulation. Important time of management is not wasted on avoidable supervision. The responsibility of the superior does not decrease once he has delegated authority. A person can delegate authority and not responsibility. We will remain for the work even if it is delegated to the subordinate accountable work. So the responsibility of superior and subordinate remains absolute.'

#숨겨진 문자 찾아 해독하기

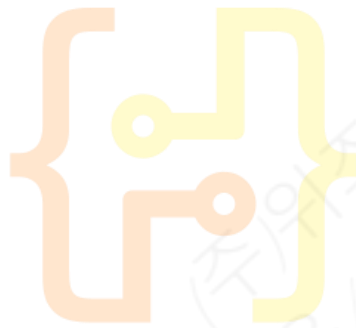
```
lines = []
for i in range(len(missing)):
    if missing[i] == '-':
        lines.append(paragraph[i])
```

	<pre>print(''.join(lines))</pre> <p>#정답 확인 : you have been doing well so far. but it was too easy. The real game is waiting. good luck.</p>																														
<div>더 나아가기</div> <div>Mission1</div>	<table><tr><td>@</td><td></td><td></td><td></td><td>@</td><td>10001 → 17</td></tr><tr><td></td><td>@</td><td></td><td>@</td><td></td><td>01010 → 10</td></tr><tr><td>@</td><td></td><td>@</td><td></td><td>@</td><td>? → ?</td></tr><tr><td></td><td>@</td><td></td><td>@</td><td></td><td>01010 → 10</td></tr><tr><td>@</td><td></td><td></td><td></td><td>@</td><td>10001 → 17</td></tr></table> <pre>key_value = '0b10101' print(int(key_value, 2))</pre> <p>#정답 확인 : 21</p>	@				@	10001 → 17		@		@		01010 → 10	@		@		@	? → ?		@		@		01010 → 10	@				@	10001 → 17
@				@	10001 → 17																										
	@		@		01010 → 10																										
@		@		@	? → ?																										
	@		@		01010 → 10																										
@				@	10001 → 17																										
<div>더 나아가기</div> <div>Mission2</div>	<pre>num_list = [] for i in range(len(decimal_list)): num_list.append(bin(decimal_list[i])) for i in range(len(num_list)): for j in num_list[i]: if j == '1': print('###', end='', flush = True) elif j == '0': print(' ', end='', flush = True) else: print('\n') print('\n')</pre> <p>#정답 확인 : 18e0</p>																														
<div>더 나아가기</div> <div>Mission3</div>	<pre>lines = [] for i in range(len(missing)): if missing[i] == '*':</pre>																														


```
lines.append(paragraph[i])  
  
print(' '.join(lines))
```

```
lines.append(paragraph[i])  
  
print(' '.join(lines))
```

#정답 확인 : you have been doing well so far. but it was too easy. The real game is waiting. good luck.



히우코딩

HOW CODING