

[Intelsat 인공위성 해킹]



NIS 상황실에 빌더버그 조직이 헤이그 도시를 파괴한다는 첩보가 도착했다.
이선과 파이는 빌더버그가 사용하는 인공위성 Intelsat 을 찾아냈고
그들의 비밀 정보를 확인하기 위해 인공위성 해킹을 시도한다.
그러던 중 인공위성이 궤도를 이탈하기 시작해 시간은 얼마 남지 않았는데...

오늘의 목표

빌더버그의 인공위성 접속 비밀번호 알아내기

주요 함수

print, int

➤ 학습 목표

- 스토리의 3 요소인 **인물, 사건, 배경**에 대해 이해하며 스토리를 파악할 수 있습니다.
- **파이썬** 프로그래밍 언어가 무엇인지 알 수 있습니다.

- 연산자 기호에 대해 알고 사용하여 연산할 수 있습니다.
- 변수에 대한 개념을 이해하여 변수를 만들어 산술연산을 할 수 있습니다.
- 자료형인 **int, float** 형에 대해 알고 자료형을 알맞게 변환할 수 있습니다.
- 주요 함수를 사용하여 프로그래밍하는 방법을 배우고 활용하여 사건을 해결할 수 있습니다.

✓ 수업 전 체크하기

- ‘수업 환경 테스트’ 를 눌러 수업 환경을 체크합니다.
- 학생과 반갑게 인사를 나누고 선생님을 소개합니다.
- 학생의 프로그래밍 경험 여부를 파악합니다.
- 학생이 툴 사용 방법을 이해하고 있는지 파악합니다.

✓ 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)

Intro.

- 학습 자료에 정리된 파이썬 프로그래밍 언어에 관해 이야기를 나눕니다.



[학습 자료]

파이썬 프로그래밍 언어는 무엇이고 어떤 특징이 있는지 함께 이야기 나누어 봅시다.

→ 파이썬이란?

- 1990 년, 귀도 반 로섬이 개발한 언어
- 대화 기능의 인터프리터 언어
- 간단하고 쉬운 문법
- 풍부한 라이브러리
- 오픈 소스, 무료
- 웹, 응용 SW, 데이터분석 등 다양한 분야에 활용

- 스토리의 내용을 간략하게 설명하고 스토리의 3 요소에 관해 이야기 합니다.

[스토리 이해]



< 파이 >



< 이선 >

<전체 스토리 내용>

형사 ‘이선’과 정보 요원 ‘파이’가 빌더버그 조직이 일으키는 사건을 해결하기 위해 단서들을 찾습니다. 단서들을 이용해 파이썬으로 프로그래밍하여 사건을 빠르게 해결해 나가는 스토리입니다.

<에필로그 - 무료체험 수업>

빌더버그 조직이 헤이그 도시를 파괴한다는 첩보가 NIS 상황실에 접수되어 이선과 파이가 단서를 찾는다.

첫 번째 단서는 그들이 사용하는 인공위성은 1,000~2,000hz 주파수이다.

두 번째 단서는 그들의 인공위성 이름은 알파벳 ‘I’가 들어있다.

세 번째 단서는 그들의 인공위성 이름은 8 개의 문자로 되어있다.

이 단서들을 통해 빌더버그 조직이 사용하는 인공위성이 ‘Intelsat’ 이라는 것을 알아냈다.

<차시 스토리>



< 빌더버그 >

이선과 파이는 찾아낸 빌더버그의 인공위성 'Intelsat' 을 해킹하여 그들의 비밀 정보를 파악하려고 한다. 비밀번호를 알아야 접속이 가능해, 시간이 필요하지만 그들의 인공위성이 궤도를 이탈하고 있어 시간이 많지 않다.

→ 스토리 3 요소 : 인물, 사건, 배경

→ 스토리 3 요소에 관해 이야기를 나누어 봅니다.

(예시) 인물은 이야기 사건의 발전을 일으키는 스토리의 주요한 요소입니다. 사건은 갈등으로 유발된 것으로 원인과 결과입니다. 그리고 배경은 이야기가 펼쳐지는 환경, 시간과 공간으로 인물을 탄생시키고 사건을 유발하게 시키는 바탕이 됩니다. 그 바탕에 어울리는 인물과 사건이 존재하게 됩니다. 스토리 3 요소를 잘 생각하며 스토리를 봅시다.

→ 에필로그(무료체험 수업)와 차시 스토리에 대해 이야기 합니다.

(예시) 스토리의 3 요소는 다음과 같습니다.

1) 인물

- 이선 : 현장에서 단서를 찾는 형사
- 파이 : 프로그래밍으로 단서를 찾는 정보 요원
- 빌더버그 조직 : 시스템을 자유자재로 컨트롤하여 전세계를 지배하려는 비밀 조직

2) 사건 : 빌더버그 조직이 네덜란드의 도시, 헤이그를 파괴한다는 첩보가 있어 이선과 파이가 그들의 인공위성인 'Intelsat' 을 찾아내 비밀 정보를 파악하려고 합니다. 하지만 인공위성은 접속 비밀번호가 있어야 해킹할 수 있습니다. 이선과 파이는 비밀번호를 알아내기 위해 단서를 찾습니다.

3) 배경 : 현대, 네덜란드, 국가 정보 보안 상황실에서 일어나는 사건입니다.

→ 스토리의 내용을 이해하고 사건을 해결하기 위해 파이썬 프로그래밍 언어를 사용합니다.

→ 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



하우코딩

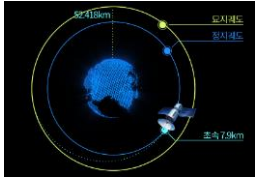
HOW CODING

Mission1. Intelsat 에 접속할 수 있는 시간 구하기

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



[스토리 이해]



< 단서 >

(빌더버그의 인공위성 Intelsat 에 접속하기 위해 시도 중이다.)

(이선은 ITU 에서 온 메일을 파이에게 전달하는데..)

"이선", "파이! ITU 에서 메일이 왔는데..."

"이선", "Intelsat 인공위성이 계속 궤도를 이탈하고 있다는군."

"파이", "흠... 연료가 떨어져 가나 보네."

"파이", "그럼 접속 가능한 시간이 얼마 남지 않았다는 거야!"

"파이", "Intelsat 은 지금 고도 **35,786km** 의 정지궤도에서 돌고 있는데,"

"파이", "이대로 라면 곧 표지궤도로 들어가 버리겠어."

"이선", "표지궤도가 뭐야?"

"파이", "할 일을 다 한 인공위성의 무덤인 셈이야."

"파이", "표지궤도에 들어가면 그 안에 있는 정보를 영원히 확인하지 못할 거야."

"

"이선", "파이! Intelsat 이 표지궤도로 들어가기까지 시간은 얼마나 남은 거야?"

"파이", "잠깐, 계산해볼게."

<<< Intelsat 인공위성이 표지궤도 진입까지 남은 시간 >>>

<단서>

1) 표지궤도의 고도는 **52,418km** 이다.

2) Intelsat 인공위성은 초속 **7.9km** 의 속도로 이동 한다.

→ 파이와 이선의 현재 상황에 관해 이야기를 나누어 봅니다.

(예시) 빌더버그 조직의 인공위성 Intelsat 에 접속해서 비밀 정보를 알아내야 하는데 시간이 많지 않습니다.

→ 시간이 많지 않은 이유에 대해 학생에게 질문합니다.

(예시) Intelsat 이 연료가 부족해서 정지궤도를 이탈해 표지궤도로 가고 있습니다. 표지궤도로 도착하면 접속이 불가능하여 그 안의 비밀 정보를 볼 수가 없습니다.

[학습 자료]

→ 현재 들고 있는 정지궤도에서 출발하여 묘지궤도로 진입하는데 남은 시간을 어떻게 알 수 있을지 학생에게 질문합니다.

(예시) 파이가 정지궤도의 고도가 35,786km 이라고 했고 찾은 단서에서는 묘지궤도의 고도가 52,418km 라고 했으니 이동해야 하는 최소 거리는 $52,418 - 35,786 = 16,632 \text{ km}$ 가 나옵니다. 인공위성이 궤도를 따라 돌면서 어떤 상황에 따라 조건이 달라져 이동하기 때문에 최소 거리를 이동할 때 걸리는 시간을 파악해야 합니다. 그리고 남은 거리를 Intelsat 의 속력인 7.9km/s 로 이동할 때 얼마나 걸리는지 알아야 합니다.

→ 이러한 복잡한 계산을 파이썬 프로그래밍 언어로 가능하다는 것을 설명해 줍니다.

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

이동거리, 이동시간, 속력의 관계

- 이동거리 : 물체가 이동한 총 거리
- 이동시간 : 물체가 이동하는데 걸린 시간
- 속력 : 물체의 빠르기, 1 초동안 이동거리(m/s)
- 이동시간 = 이동거리 / 속력

파이썬 연산자 기호

+, -, *, /	덧셈, 뺄셈, 곱셈, 나눗셈
%	나머지
//	몫
**	제곱

변수

- 데이터를 저장하는 공간

- 데이터 또는 계산의 결과를 저장
- 변수 이름 = 값
값을 변수에 저장할 때 '=' 기호 사용
ex) answer = 100

→ 이동거리, 이동시간, 속력과의 관계에 관해 설명해 줍니다.

(예시) 정지궤도에서 묘지궤도로 이동하는 Intelsat 의 경로(거리)가 더 길어진다면 또는 속력이 더 느려진다면 Intelsat 이 묘지궤도에 진입까지의 남은 시간이 많아져 해킹할 수 있는 시간도 길어집니다.

→ 파이썬 프로그래밍 언어에서 산술 연산을 할 때 사용하는 연산자 기호에 관해 설명해 줍니다.

(예시) 파이썬의 연산자는 일반적으로 사용하는 연산자 기호와 같은 뜻으로 사용되는 기호도 있지만, '/' 몫을 구하는 연산 기호처럼 파이썬에서만 정의하는 새로운 기호도 있습니다. 더하기 '+', 빼기 '-', 곱하기 '*', 나누기 '/' 와 같은 연산자들은 일상생활에서 많이 쓰고 익숙한 연산자입니다. 이에 추가로 나머지 '%', 몫 '//', 제곱 '**' 과 같은 연산자를 제공하여 더욱더 쉽게 연산을 할 수 있도록 합니다.

→ 변수에 대한 개념을 설명해 줍니다.

(예시) 변수는 데이터를 저장하는 공간으로 데이터 또는 계산의 결과를 저장합니다. 변수 이름을 만들고 그 안에 값을 저장할 때는 '=' 기호를 사용합니다. 변수에 값을 저장하는 것을 '값을 할당한다' 라고도 표현합니다.

- 사용할 API 에 대해 설명합니다.

< API >

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

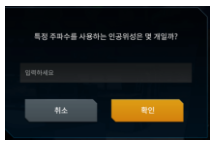
→ print 함수는 () 안에 값을 넣으면 그 값을 실행결과 창에 출력해 줍니다.

- 괄호 안에 숫자와 변수명은 그대로 넣고 문자는 작은 따옴표(') 또는 큰 따옴표(")로 묶어 줍니다.

[코드 이해]



< 입력 버튼 >



< 정답 입력 >

- 학생의 이름 또는 숫자를 결과 창에 출력해보며 연습합니다.
→ int 함수는 () 안에 값을 넣으면 그 값을 정수로 변환해 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 >

#Intelsat 의 이동 거리 구하기

#첫 번째 방법 : 연산자 기호 사용

#Intelsat 의 이동 거리 = 묘지궤도 - 정지궤도

```
52418 - 35786
print(52418 - 35786, "km")
```

#두 번째 방법 : 변수사용하여 연산

```
stop_high = 35786
death_high = 52418

print(death_high - stop_high, "km")
```

#결과값 확인 : 16632 km

#Intelsat 이 묘지궤도 진입까지 남은 시간 구하기

```
#시간 = 거리 / 속력
#Intelsat 속력 : 초속 7.9km
#먼저 구한 결과값을 distance 변수에 저장
```

```
distance = death_high - stop_high
time_s = distance / 7.9
time_m = time_s / 60
print(time_m, "분")
```

#결과값 확인 : 35.088607 분

#시간을 정수 단위로 변환

```
print( int(time_m) )
```

#정답 확인 : 35

→ 학생에게 print("안녕하세요") 코드를 코드 창에 구현하고 실행 버튼을 클릭하여 실행결과 창에 출력되는 내용을 확인하도록 합니다.

→ 실행 결과 창에 '안녕하세요' 가 출력되지 않는다면 해당 오류를 수정하도록 설명합니다.

→ Intelsat 이 이동하는 거리를 구하는 연산을 할 때 '52418 - 35786' 만 작성했을 시에 실행결과 창에 결과값이 나오게 하기 위해서 어떻게 해야 하는지 설명합니다.

(예시) 결과 창에 무언가를 출력하기 위해서는 print 함수를 사용합니다.

→ 실행결과 창에 'km' 단위를 함께 출력하는 방법을 설명합니다.

(예시) print 출력 함수 괄호 안에 여러 개의 결과값을 함께 출력할 때는 ',' 를 추가하고 숫자 일 때는 숫자 그대로, 문자일 때는 그 문자를 ' , " ' 로 묶어 줍니다.

→ 궤도의 고도를 변수에 저장한 후에 연산하는 방법을 설명합니다.

→ 정지궤도와 묘지궤도와의 거리를 계산하여 distance 라는 변수에 값을 저장하여 이동 시간을 구하는 공식에 대입하여 계산하는 방법을 설명합니다.

(예시) ‘이동 시간 = 이동 거리 / 속도’ 이라는 공식에 대입합니다. 이동 거리는 distance 변수에 저장된 값으로 묘지궤도의 고도에서 정지궤도의 고도를 뺀 값입니다. 그리고 속력은 Intelsat 이 1 초에 7.9km 를 이동하는 값을 대입합니다.

$\text{distance} / 7.9\text{km} = 2105.316455$ 초 가 나옵니다.

→ 시간 단위를 변환해 줍니다.

(예시) 2105.316455 초를 time_s 라는 변수에 넣어 시간 단위를 분으로 변환해주기 위해 time_s 를 60 으로 나누어 줍니다. (1 분은 60 초입니다.) 분으로 변환한 시간은 time_m 변수에 저장하면 35.088607 분 입니다.

→ 실수를 정수로 변환해 줍니다.

(예시) time_m 변수에 저장된 실수 35.088607 를 int(값) 함수를 사용하여 값을 정수로 변환해 줍니다.

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

(예시) Intelsat 인공위성을 해킹하여 비밀 정보를 파악할 수 있는 시간은 35 분이 남았습니다. 그래서 **35** 를 입력합니다.

→ 스스로 코딩하도록 유도합니다.

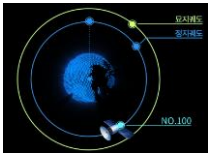
→ 미션 2 템플릿으로 이동합니다.

Mission2. Intelsat 접속 비밀번호 알아내기

● 스토리의 내용을 이해합니다.

[스토리 이해]

- 이전 미션에서 나온 프로그래밍 결과를 생각해보며 스토리를 확인합니다.



< 단서 >

"이선", "35 분이 지나기 전에 접속 해야해."

"파이", "비밀번호를 반복해서 확인할 수 있는 프로그램을 만들어볼게."

"이선", "잠깐, 빌더버그 조직에 잠복해 있던 스파이에게서 연락이 왔어."

(스파이가 메시지를 보내왔다.)

"이선", "파이! 인공위성에 접속하는 비밀번호는 해당 궤도의 둘레와
관련이 있대."

"이선", "Intelsat 의 궤도 둘레를 알아야겠는데."

"파이", "간단하게 해결할 수 있겠네. "

"파이", "정지궤도의 고도에 지구의 반지름을 더해서 두배하고 파이를
곱해주면 돼."

"이선", "무슨 말인지 어려운데...파이? 파이는 뭐야. 너 이름이잖아..."

"파이", "장난칠 시간 없어. 파이는 둘레를 구하는 공식에 필요한 수치야."

"파이", "그리고 지구 반지름은 6,371km 이고.."

"파이", "내가 바로 계산해볼게."

(파이는 빠른 손놀림으로 키보드를 치기 시작한다.)

<<< Intelsat 인공위성의 접속 비밀번호 찾기 >>>

<단서>

1) 비밀번호는 정지궤도 둘레를 인공위성 번호로 나눈 숫자이다.

→ 파이가 비밀번호를 알아내는 프로그램을 만들려고 했지만 다른 도움을 통해 더 빠르게 단서를 찾을 수 있었는지 이야기 나누어 봅니다.

(예시) 파이가 프로그래밍하여 단서를 찾아내기 전에 빌더버그 조직에 숨어 지내는 스파이가 전달해준 단서를 통해 빠르게 사건을 해결할 수 있을 것으로 생각합니다.

→ 파이는 찾은 단서를 어떤 프로그래밍으로 해결할 수 있을지 이야기 나누어 봅니다.

(예시) 정지궤도 둘레를 알아내는 연산을 파이썬 프로그래밍으로 합니다. 둘레를 구하여 인공위성 번호로 나누어 Intelsat 인공위성 접속 비밀번호를 알아낼 수 있습니다.

[학습 자료]

- 학습하게 될 개념에 대해 해당 코드를 구현하면서 적절하게 설명합니다.

원의 둘레

- 원의 테두리의 길이
- 원의 둘레 = 원의 반지름 * 2 * 원주율
- 원주율 = 3.14 (정해진 값)

변수 이름 사용 규칙

- 문자, 숫자, '_' 만 포함
- 문자 또는 '_'로만 시작
- 예약어와 같은 이름 불가능
ex) False, True, and, import, return, while, for 등
- 대문자와 소문자 구별

수치 데이터 자료형

- 정수형 : int 형, 개수나 순서에 사용
ex) 나이, 연도, 월, 일 등

- 실수형 : float 형, 일반적인 계산에 사용

ex) 길이, 무게, 온도, 부피 등

→ 원의 둘레는 테두리의 길이를 나타내는 것으로 구하는 방법을 설명해 줍니다.

(예시) 원의 둘레는 원의 반지름을 두 번 더한 값에 원주율을 곱하면 구할 수 있습니다. 원주율은 둘레의 길이를 구할 때 주어지는 수치로 '3.14' 입니다.

→ 변수의 이름을 지정해주는 규칙에 대해 설명해 줍니다.

(예시) 변수의 이름을 지정할 때는 꼭 지켜야 하는 규칙이 있습니다. 문자, 숫자, 특수 기호는 '_' 만 포함될 수 있고 문자는 영문과 한글 모두 가능하며 영문의 경우에는 대문자와 소문자를 구별합니다. 변수 이름의 시작은 숫자 제외하고 문자, '_' 기호 모두 가능하지만 파이썬 프로그래밍 언어에서 사용하는 함수이름, 클래스 이름 등 예약어에 대해서는 사용할 수 없습니다. 변수의 이름을 지정해주는 규칙 외에 프로그래밍을 하는 사람들이 서로의 코드를 잘 이해할 수 있도록 적절한 이름을 지정해주는 것도 중요합니다.

→ 파이썬 수치 데이터 자료형에 대해 설명해 줍니다.

(예시) 파이썬에서 사용하는 데이터 자료형에는 'int, float, string, list, tuple, set 등 다양한 종류가 있습니다. 이번 차시에서는 수치 데이터 자료형인 int, float 형에 대해 설명합니다.

- 사용할 API에 대해 설명합니다.

< API >

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

→ print 함수는 () 안에 값을 넣으면 그 값을 실행결과 창에 출력해 줍니다.

- 괄호 안에 숫자와 변수명은 그대로 넣고 문자는 작은 따옴표(') 또는 큰 따옴표(")로 묶어 줍니다.
- 학생의 이름 또는 숫자를 결과 창에 출력해보며 연습합니다.

→ int 함수는 () 안에 값을 넣으면 그 값을 정수로 변환해 줍니다.

[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 >

#Intelsat 인공위성의 접속 비밀번호

#첫 번째 방법 : 연산자 기호 사용

```
password = (stop_high + 6371) * 2 * 3.14 / 100
password = int(password)
print(password)
```

#두 번째 방법 : 변수사용하여 연산

```
intelsat_num = 100
pi = 3.14
earth_radius = 6371

password = (stop_high + earth_radius) * 2 * pi / intelsat_num
password = int(password)
print(password)
```

#정답 확인 : 2647

→ 단서를 확인하여 Intelsat 인공위성 접속 비밀번호를 알아내기 위한 방법에 대해 이야기 합니다.

(예시) 단서에서 비밀번호는 정지궤도 둘레를 인공위성 번호로 나눈 숫자라고 했습니다. 정지궤도의 둘레를 먼저 구하도록 합니다. 정지궤도는 원이기 때문에 원의 둘레를 구하는 공식에 대입하여 구하도록 합니다.

→ 정지궤도의 둘레 구하기 위해서 필요한 데이터를 학생과 함께 파악하도록 합니다.

(예시) 정지궤도의 둘레를 구하기 위해서는 반지름의 길이를 알아야 합니다. 현재 스토리 속에 나온 정지궤도의 고도인 **35,786km** 에 지구의 반지름인 **6,371km** 를 더해야 정지궤도의 반지름을 구할 수 있습니다. 반지름을 구한 후 두 번 더한 값에 원주율인 3.14 를 곱하여 정지궤도의 둘레 '264745.96km' 를 구합니다.

→ 비밀번호를 구하기 위해서 인공위성의 번호를 어떻게 알 수 있을지 이야기 나눕니다.

(예시) 인공위성의 번호는 단서 이미지 속에서 찾을 수 있습니다. Intelsat 의 번호는 no. 100 입니다.

→ Intelsat 에 접속하기 위한 비밀번호를 구하도록 합니다.

(예시) 정지궤도의 둘레인 '264745.96' 에 인공위성 번호인 '100' 으로 나누면 **2647.4596** 이고 이 결과를 int(값) 함수를 사용하여 값을 정수로 변환하여 비밀번호 2647 을 구합니다.

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

(예시) Intelsat 인공위성을 접속하기 위한 비밀번호인 **2647** 을 입력합니다.

→ 스스로 코딩하도록 유도합니다.

→ 마무리 템플릿으로 이동합니다.

HOW CODING

Summary.

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

파이썬 연산자 기호

- +, -, *, /, %, //, **

변수

- 변수 이름 = 값

값을 변수에 저장할 때 '=' 기호 사용

- 변수 이름 사용 규칙

파이썬 수치 데이터 자료형

- 정수형 : int 형, 개수나 순서에 사용
- 실수형 : float 형, 일반적인 계산에 사용

API

- print(값) : 출력 함수
- int(값) : 값을 정수로 변환하는 함수

더 나아가기

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

< Mission 1 >

→ 대기의 마찰이 심해져 Intelsat 인공위성의 속력이 초당 6.5km 로 느려졌다면 해킹할 수 있는 시간은 얼마나 남게 될까요?

(예시) Intelsat 인공위성의 속력이 늦어졌다면 묘지궤도에 늦게 도착하게 됩니다. 그 만큼 해킹을 시도할 수 있는 시간이 많아 지는 것입니다. '이동 시간 = 이동 거리 / 속력' 의 공식을 통해 계산합니다.

< Mission 2 >

→ 원의 둘레를 구할 때 필요한 수치인 원주율은 보통 3.14 라고 알고 사용하지만 더 정확한 수로 나타내면 3.1415926535 입니다. 이 원주율을 사용하면 비밀번호는 어떻게 될까요?

(예시) 소수 두번째 자리까지 끊어서 계산할 때보다 1 큰수가 나오게 됩니다.

평가 기준

평가 내용	1~5	강사 메모
학습		
인물, 사건, 배경을 기준으로 이번 차시 스토리를 설명할 수 있다.		
파이썬 프로그래밍 언어에서 사용하는 연산자 기호를 알고 연산할 수 있다.		
변수의 개념을 알고 적절한 이름을 지정하여 사용할 수 있다.		
데이터 자료형 중 int, float 형에 대해 알고 적절한 형태로 변환시킬 수 있다.		
주요 함수를 사용하여 프로그래밍하고 결과 값을 찾아 미션을 해결할 수 있다.		
태도		
어려운 점이 있어도 포기하지 않고 끝까지 해결하려고 노력하였다.		

모범 답안

코드	
Mission1	<p>#Intelsat 의 이동 거리 구하기</p> <p>#첫 번째 방법 : 연산자 기호 사용</p> <p>#Intelsat 의 이동 거리 = 묘지궤도 - 정지궤도</p> <p>52418 - 35786</p> <p>print(52418 - 35786, "km")</p>

	<p>#두 번째 방법 : 변수사용하여 연산</p> <pre> stop_high = 35786 death_high = 52418 print(death_high - stop_high, "km") #결과값 확인 : 16632 km #Intelsat 이 표지궤도 진입까지 남은 시간 구하기 #시간 = 거리 / 속력 #Intelsat 속력 : 초속 7.9km #먼저 구한 결과값을 distance 변수에 저장 distance = death_high - stop_high time_s = distance / 7.9 time_m = time_s / 60 print(time_m, "분") #결과값 확인 : 35.088607 분 #시간을 정수 단위로 변환 print(int(time_m)) #정답 확인 : 35 </pre>
Mission2	<p>#Intelsat 인공위성의 접속 비밀번호</p> <p>#첫 번째 방법 : 연산자 기호 사용</p> <pre> password = (stop_high + 6371) * 2 * 3.14 / 100 </pre>

	<pre>password = int(password) print(password) #두 번째 방법 : 변수사용하여 연산 intelsat_num = 100 pi = 3.14 earth_radius = 6371 password = (stop_high + earth_radius) * 2 * pi / intelsat_num password = int(password) print(password) #정답 확인 : 2647</pre>
더 나아가기 Mission1	<pre>#Intelsat 의 이동 거리 구하기 distance = stop_high - death_high time_s = distance / 6.5 time_m = time_s / 60 print(time_m, "분") #정답 : 42</pre>
더 나아가기 Mission2	<pre>#Intelsat 인공위성의 접속 비밀번호 intelsat_num = 100 pi2 = 3.1415926535 earth_radius = 6371 password = (stop_high + earth_radius) * 2 * pi2 / intelsat_num password = int(password) print(password) #정답 확인 : 2648</pre>