

2장 재귀와 귀납적 사고 실습

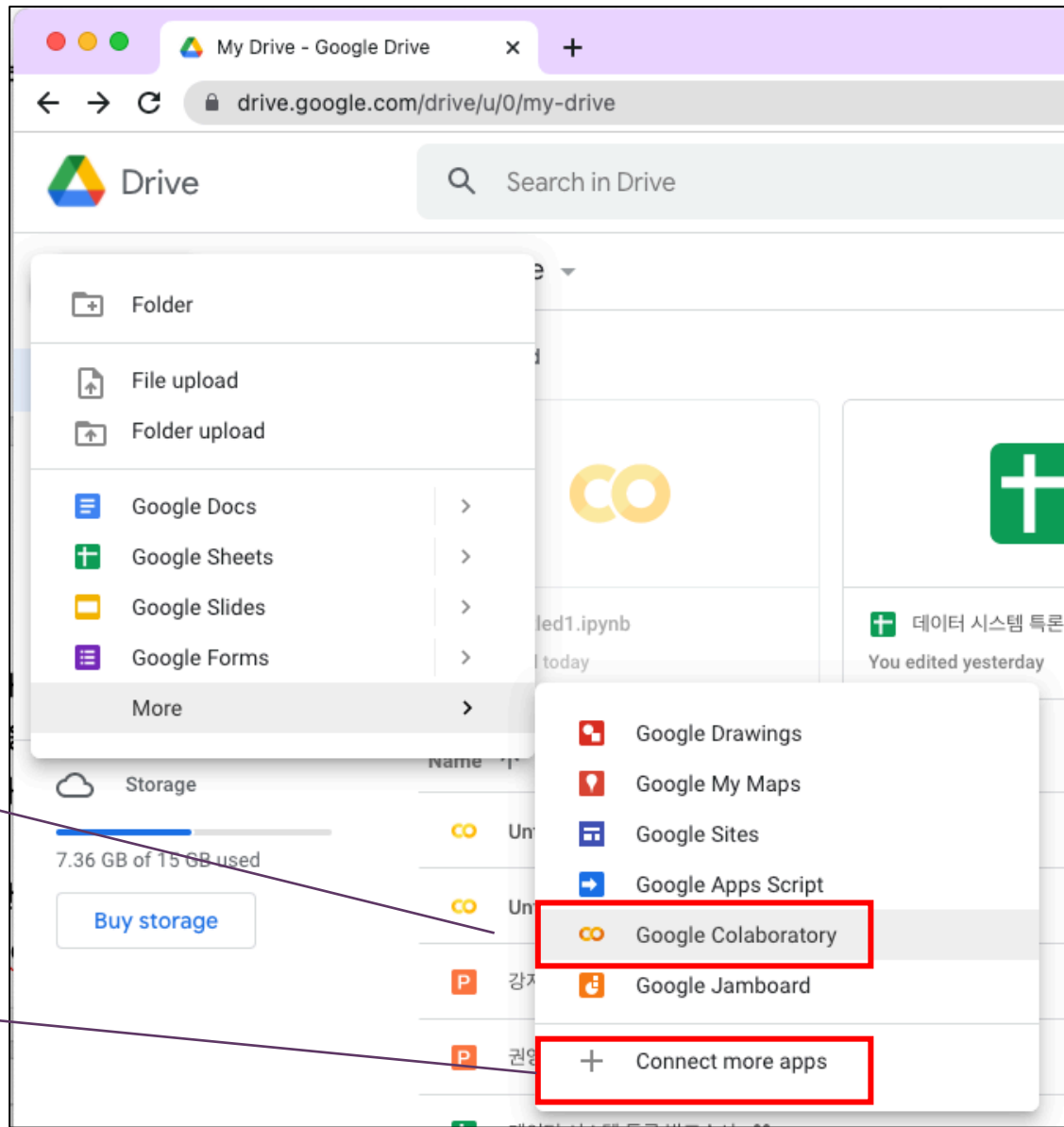
Eunji Lee

실습 환경

- Google colab

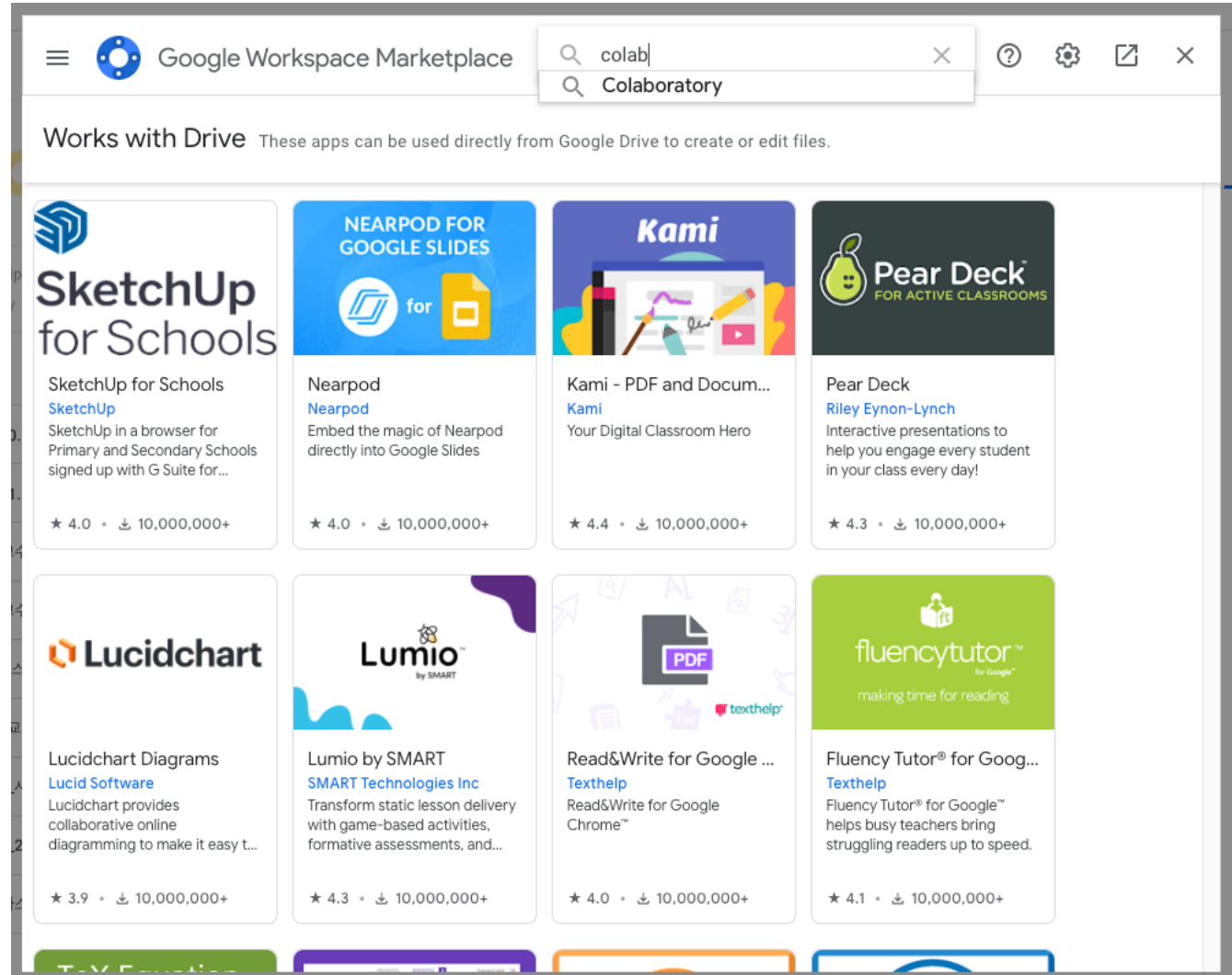
Google Colaboratory
선택

Google Colaboratory 가
보이지 않으면 이걸 선택
해서 extension 설치



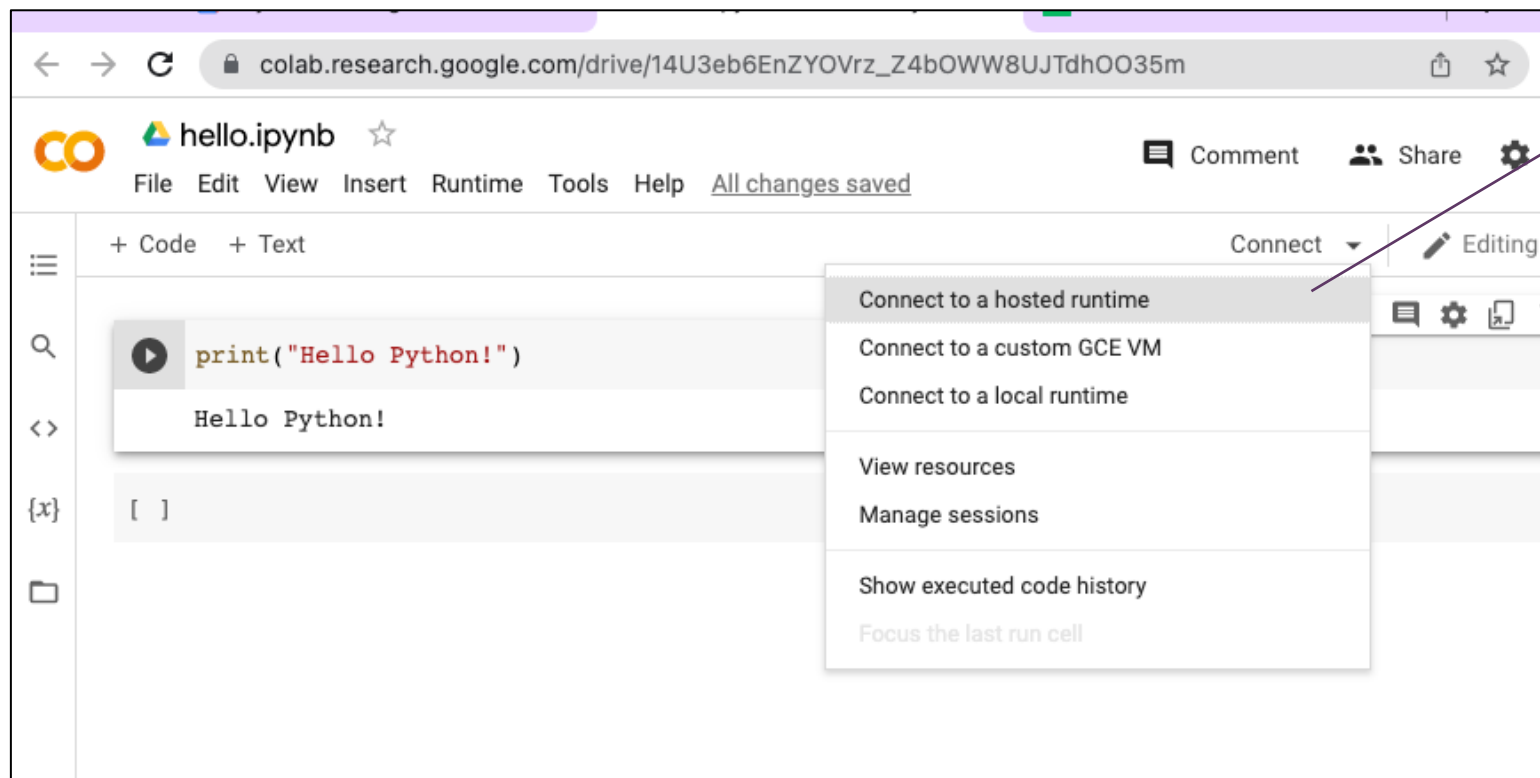
실습 환경

- colab 으로 검색 후 설치



실습 환경

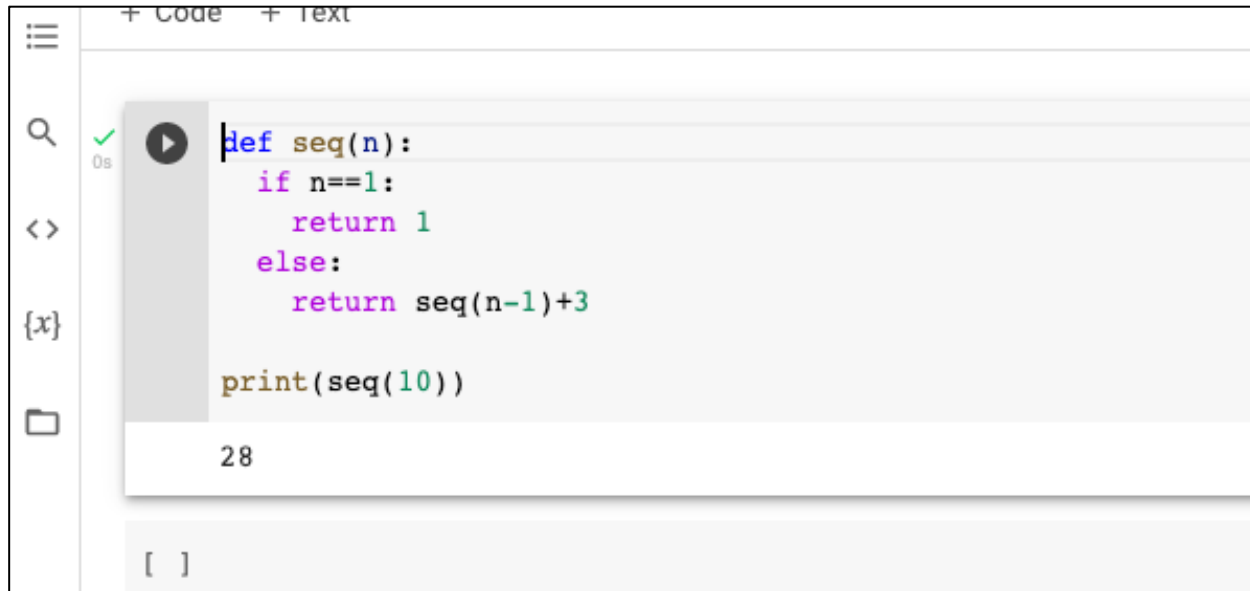
- Hello Python 출력해보기
- 코드 입력 후 shift + Enter



실행이 안되는 경우
connect 설정

연습 문제

- 공차가 3인 등차수열의 n 번째 원소를 구하는 프로그램을 재귀적으로 코딩해 보기



The screenshot shows a code editor with a sidebar on the left containing icons for a menu, search, code view, variable view, and file explorer. The main editor area has a tab labeled '+ Code + Text'. Inside the editor, a Python function `def seq(n):` is defined. It uses a base case `if n==1: return 1` and a recursive case `else: return seq(n-1)+3`. Below the function definition, `print(seq(10))` is called. A play button icon is visible to the left of the code. Below the code, the output `28` is displayed. At the bottom of the editor, there is a small input field containing `[]`.

```
def seq(n):  
    if n==1:  
        return 1  
    else:  
        return seq(n-1)+3  
  
print(seq(10))
```

28

[]

연습 문제

- 하노이의 타워 문제를 코딩해보기



```
# move n plates from src to dest  
def move(n, src, tmp, dest):  
    print("move %d from %c to %c" % (n, src, dest))
```

```
move(3, 'a', 'b', 'c' )
```

```
move 1 from a to c  
move 2 from a to b  
move 1 from c to b  
move 3 from a to c  
move 1 from b to a  
move 2 from b to c  
move 1 from a to c
```

이 코드를 수정하여 실행 시 아래의 결과가 나오도록 해보기

연습 문제

- n 을 입력으로 받아 2^n 을 계산해주는 함수 `pow2(n)`을 재귀적으로 구현하여라.

✓
0s



```
def pow2(n):
```



```
print(pow2(100))
```



```
1267650600228229401496703205376
```

연습 문제

- 리스트에 다음과 같이 숫자가 저장되어 있을 때, 최대값을 찾는 프로그램을 재귀적으로 만들어 보시오.

```
num = [2, 4, 1, 8, 9, 3]
```

```
def max(num, n):
```



```
print(max(num, len(num)-1))
```

✓
0초



```
num = [2, 4, 1, 8, 9, 3]
```

```
def max(num, n):
```

```
    if n==1:
```

```
        return num[0]
```

```
    return num[0] if num[0] > max(num[1:], n-1) else max(num[1:], n-1)
```

```
print(max(num, len(num)-1))
```

9