

# [Opensource Basic Project]

# 오픈소스 기초 프로젝트

## - 3주차 : Control Statement

2024. 3. 20(수).

우창우

Dr.woo@chungbuk.ac.kr

# □ 주차별 강의계획

※ 4.10(총선), 5.1(근로자의날), 5.15(스승의날)

주차(변경 전)	주차(변경 후)	강의계획	과제	번호	제출 마감일
1주(03.06)	1주(03.06)	Python Started	VScode 설치화면, "Hello World" 출력화면 및 작성코드	1	03.06~03.09(토)
2주(03.13)	2주(03.13)	Variables and simple data types	다이아몬드 모양, 진수 변환 프로그램 및 동전 변환 프로그램의 출력화면/작성코드	2	03.13~03.16(토)
3주(03.20)	3주(03.20)	Control Statement		3	03.20~03.23(토)
4주(03.27)	4주(03.27)	Lists, Dictionaries, String		4	03.27~03.30(토)
5주(04.03)	5주(04.03)	Functions and module		5	04.03~04.06(토)
6주(04.10)	6주(04.17)	Classes		6	04.17~04.20(토)
7주(04.17)	7주(04.24)	Window programming		7	04.24~04.27(토)
8주(04.24)	8주(05.01)	Midterm		-	-
9주(05.01)	9주(05.08)	Files and Exceptions		8	05.08~05.11(토)
10주(05.08)	10주(05.22)	Database		9	05.22~05.25(토)
11주(05.15)	11주(05.29)	Data visualization		10	05.29~06.01(토)
12주(05.22)	12주(06.05)	Team project implementation I		-	-
13주(05.29)	13주(06.12)	Team project implementation II		-	-
14주(06.05)	14주(06.17)	Team project implementation III		-	-
15주(06.12)	15주(06.18)	Project Presentation		-	-

## Chapter 5. 조건문

**Selection 01.** 이장에서 만들 프로그램

**Selection 02.** 기본 if 문

**Selection 03.** 중첩 if 문

**Selection 04.** if 문 응용

# Selection 01. 이장에서 만들 프로그램 (실습&과제#1)

## 1. 종합계산기

- 입력한 수식대로 계산하는 계산기와, 두 숫자 사이의 수를 더하는 계산기

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [icon] [icon] ... ^ x

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
1. 입력한 수식 계산 2. 두 수 사이의 합계 : 1
*** 수식을 입력하세요 : 3*4/2-5
3*4/2-5 결과는 1.0입니다.
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
1. 입력한 수식 계산 2. 두 수 사이의 합계 : 2
*** 첫 번째 숫자를 입력하세요 : 1
*** 두 번째 숫자를 입력하세요 : 10
1+...+10는 55입니다.
PS C:\Users\rryan> [icon]
```

# Selection 02. 기본 if 문

## 1. if 문

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  a = 99
2  if a < 100:
3      print("100보다 작습니다.")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python
100보다 작습니다.
PS C:\Users\rryan>
```

- ✓ '만약 ~라면' 라는 의미처럼 조건을 만들 때 사용
- ✓ 조건의 결과는 참(True) 또는 거짓(False) 두 가지

```
graph TD
    A[a에 99 대입] --> B{a가 100보다 작은가?}
    B -- 참 --> C["“100보다 작군요.” 출력"]
    B -- 거짓 --> D[프로그램 종료]
    C --> D
```

if 문 실행 과정

# Selection 02. 기본 if 문

## 1. if 문

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1 a = 200
2 if a < 100:
3     print("100보다 작습니다.")
4 print("거짓이므로 이 문장은 안보이겠죠?")
5 print("프로그램 끝")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
거짓이므로 이 문장은 안보이겠죠?
프로그램 끝
PS C:\Users\rryan>

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1 a = 200
2 if a < 100:
3     print("100보다 작습니다.")
4     print("거짓이므로 이 문장은 안보이겠죠?")
5 print("프로그램 끝")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
프로그램 끝
PS C:\Users\rryan>

✓ 조건이 참이고 실행할 문장이 2개일 때 들여쓰기를 명확하게 해야 의도한 결과 출력이 가능

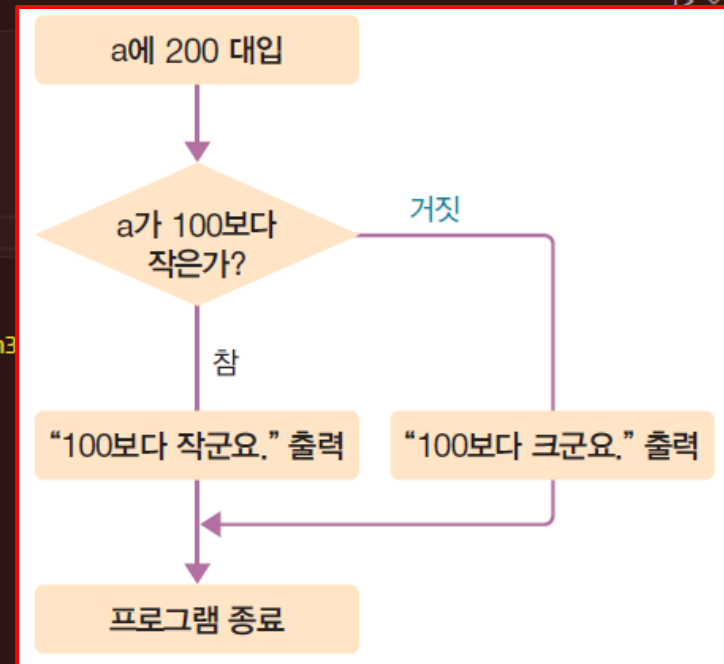
# Selection 02. 기본 if 문

## 2. if~else 문

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  a = 200
2  if a < 100:
3      print("100보다 작군요.")
4  else:
5      print("100보다 크군요.")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python3
100보다 크군요.
PS C:\Users\rryan>
```

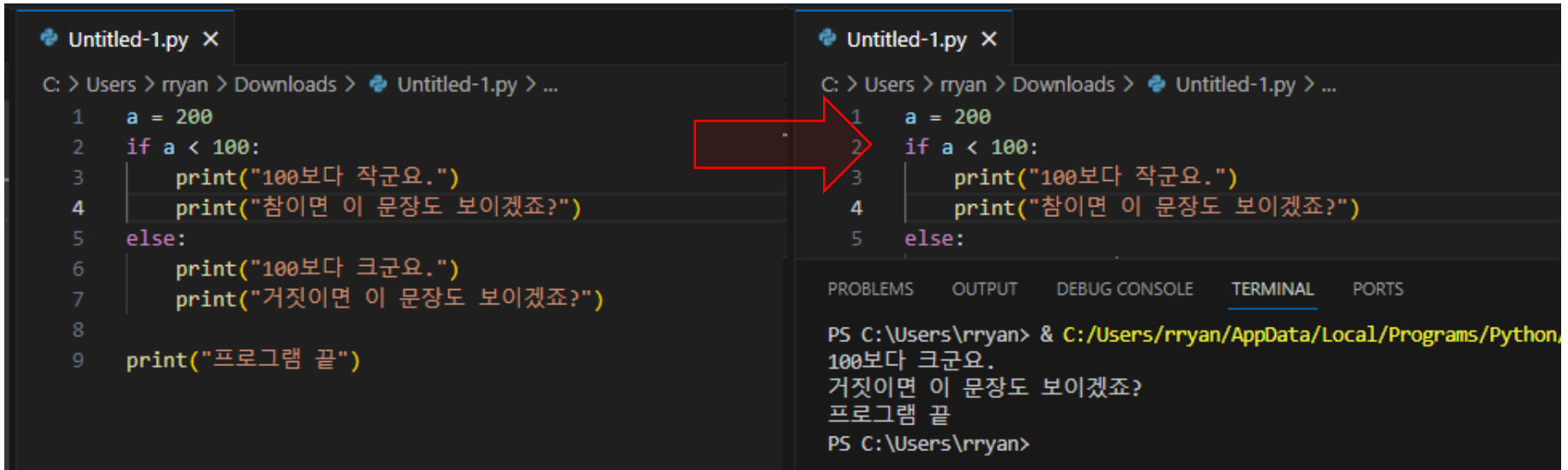
✓ 조건이 참일 때와 거짓일 때 실행할 문장이 다름



**if~else 문 실행 과정**

## Selection 02. 기본 if 문

### 2. if~else 문



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 a = 200
2 if a < 100:
3     print("100보다 작군요.")
4     print("참이면 이 문장도 보이겠죠?")
5 else:
6     print("100보다 크군요.")
7     print("거짓이면 이 문장도 보이겠죠?")
8
9 print("프로그램 끝")
```

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 a = 200
2 if a < 100:
3     print("100보다 작군요.")
4     print("참이면 이 문장도 보이겠죠?")
5 else:
6     print("100보다 크군요.")
7     print("거짓이면 이 문장도 보이겠죠?")
8
9 print("프로그램 끝")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

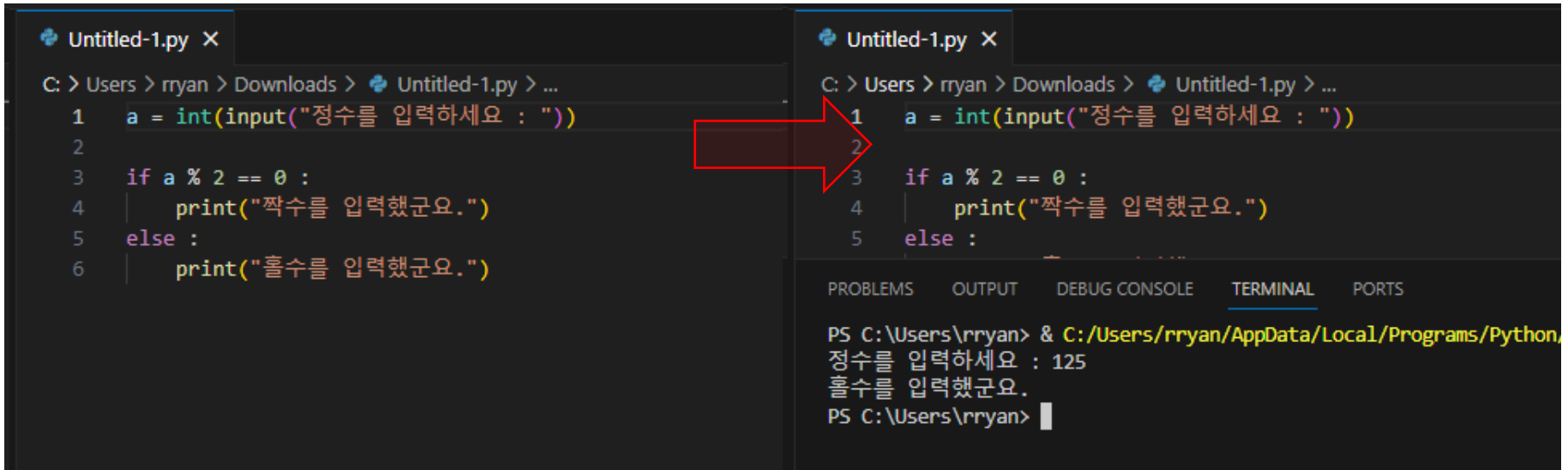
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-6/python.exe C:/Users/rryan/Downloads/Untitled-1.py
100보다 크군요.
거짓이면 이 문장도 보이겠죠?
프로그램 끝
PS C:\Users\rryan>
```

✓ 조건이 참일 때와 거짓일 때 실행할 문장이 다름



## Selection 02. 기본 if 문

### 2. if~else 문



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 a = int(input("정수를 입력하세요 : "))
2
3 if a % 2 == 0 :
4     print("짝수를 입력했군요.")
5 else :
6     print("홀수를 입력했군요.")
```

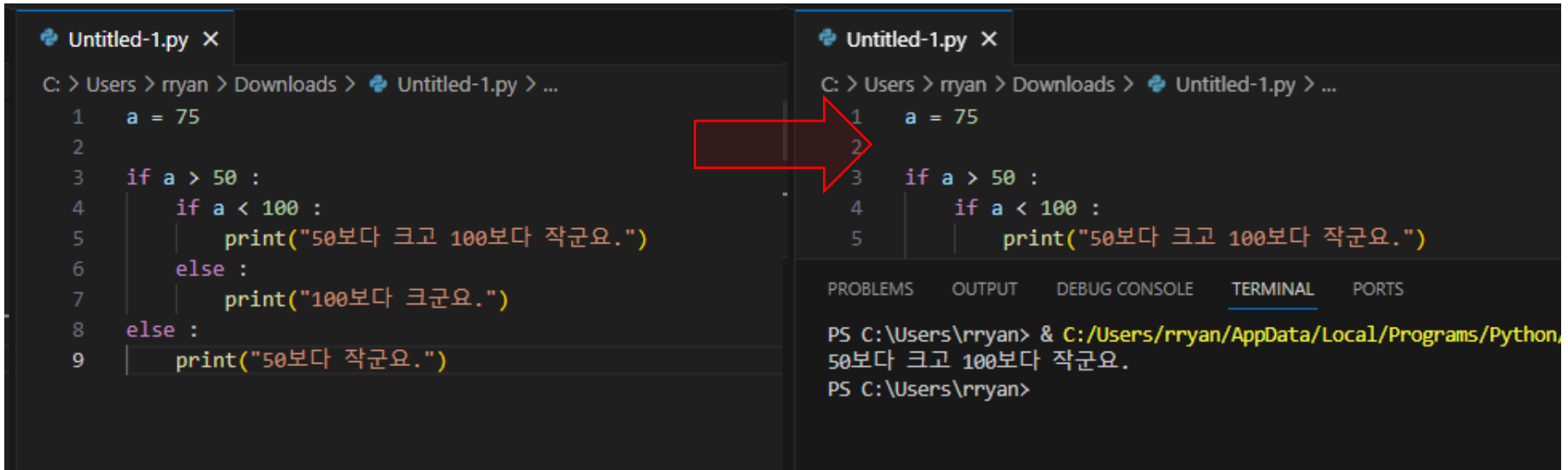
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python311/Python.exe C:/Users/rryan/Downloads/Untitled-1.py
정수를 입력하세요 : 125
홀수를 입력했군요.
PS C:\Users\rryan>
```

✓ 조건이 참일 때와 거짓일 때 실행할 문장이 다름

## Selection 03. 중첩 if 문

### 1. if~else~if~else 문



```
1 a = 75
2
3 if a > 50 :
4     if a < 100 :
5         print("50보다 크고 100보다 작군요.")
6     else :
7         print("100보다 크군요.")
8 else :
9     print("50보다 작군요.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

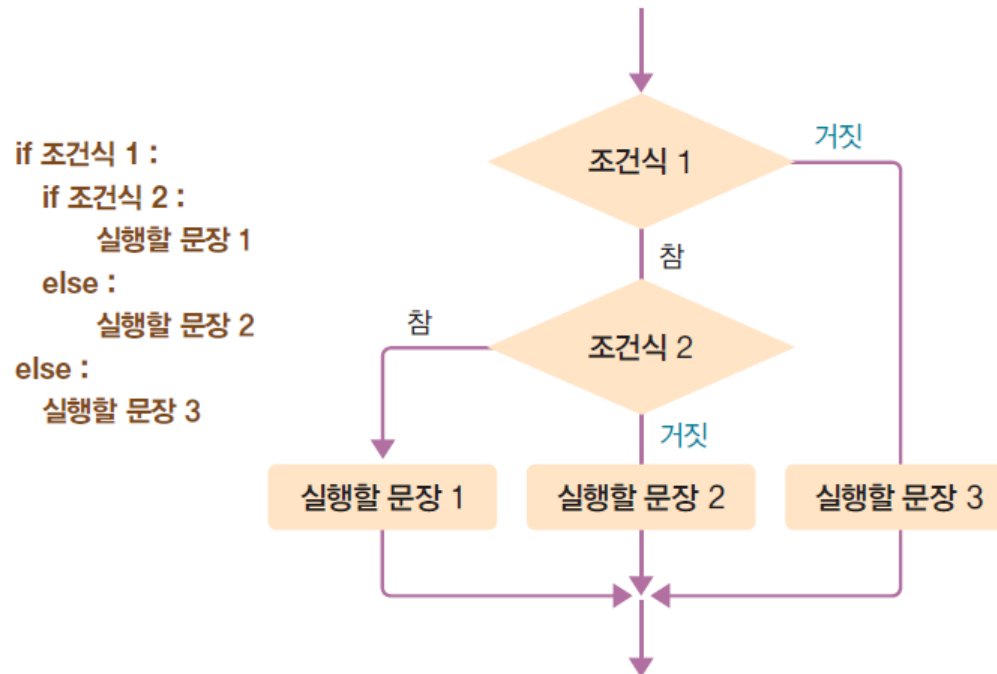
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python38-32/python.exe C:/Users/rryan/AppData/Local/Programs/Python/Python38-32/python.exe C:\Users\rryan\Downloads\Untitled-1.py
50보다 크고 100보다 작군요.
PS C:\Users\rryan>
```

✓ if 문을 한 번 실행한 후 그 결과에서 if 문을 다시 실행하는 것

## Selection 03. 중첩 if 문

### 1. if~else~if~else 문

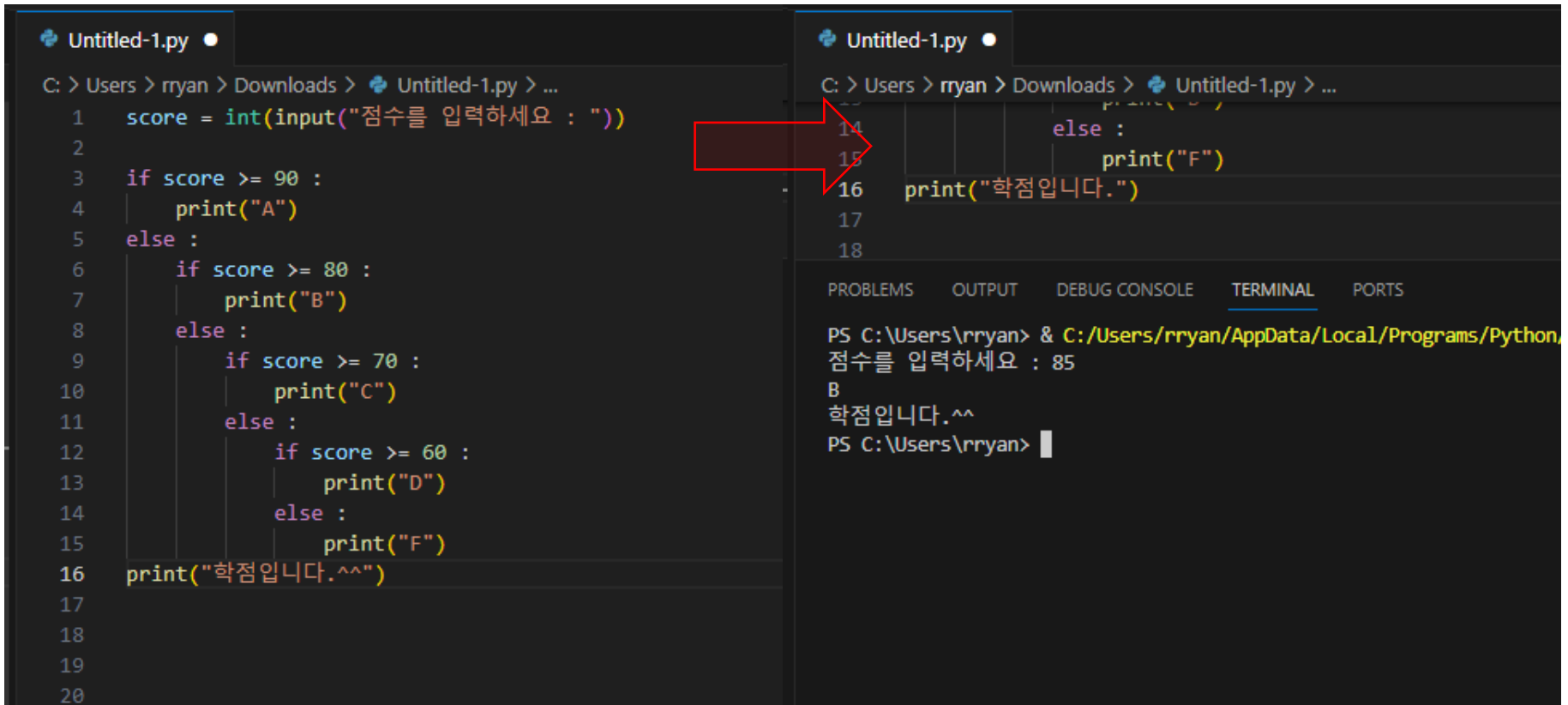
- if 문을 한 번 실행한 후 그 결과에서 if 문을 다시 실행하는 것



중첩 if 문의 형식과 순서도

## Selection 03. 중첩 if 문

### 1. if~else~if~else 문



The image shows a Python IDE with two panels. The left panel displays a Python script named 'Untitled-1.py' with a nested if-else statement. The right panel shows the execution output in the 'TERMINAL' tab, with a red arrow pointing from the code to the output.

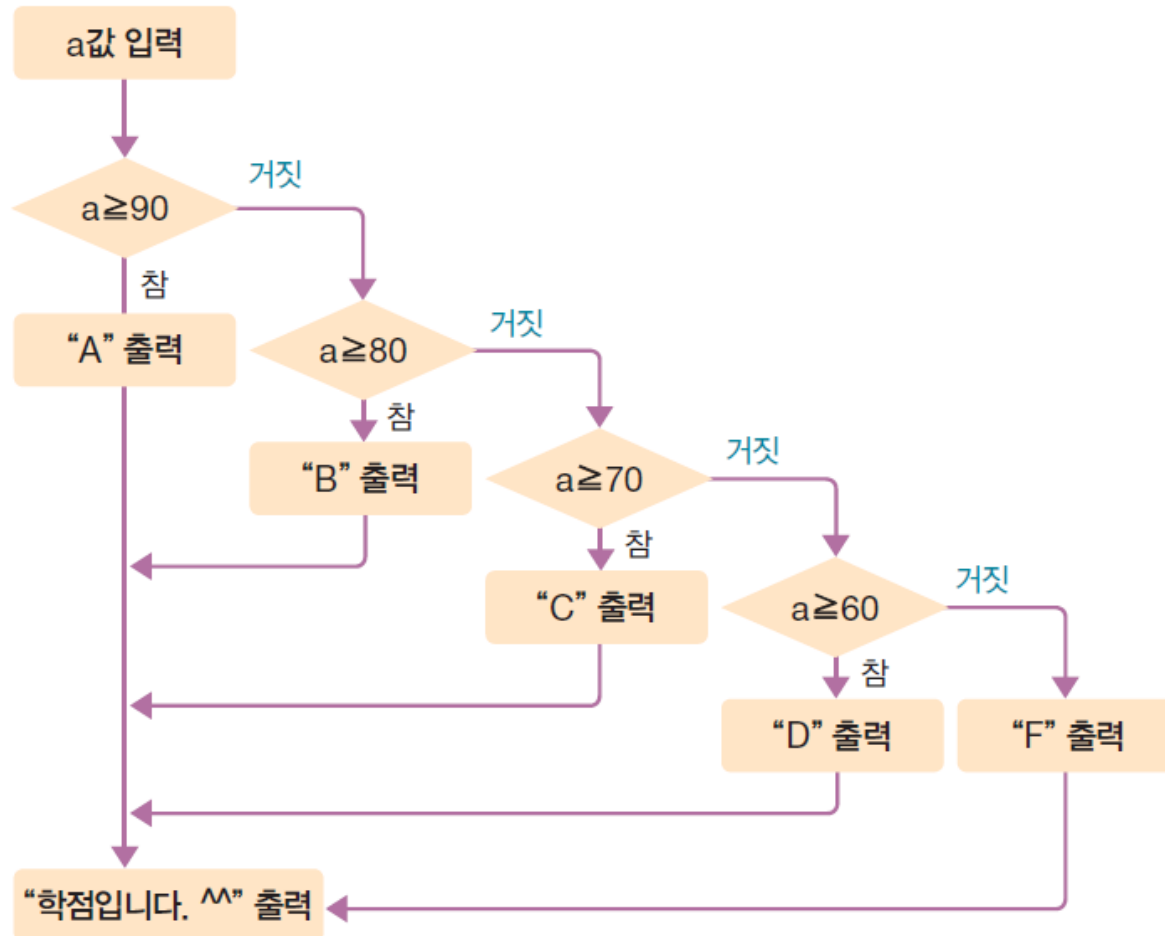
```
1 score = int(input("점수를 입력하세요 : "))
2
3 if score >= 90 :
4     print("A")
5 else :
6     if score >= 80 :
7         print("B")
8     else :
9         if score >= 70 :
10            print("C")
11        else :
12            if score >= 60 :
13                print("D")
14            else :
15                print("F")
16 print("학점입니다.^^")
17
18
19
20
```

Execution Output (Terminal):

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/python.exe C:/Users/rryan/Downloads/Untitled-1.py
점수를 입력하세요 : 85
B
학점입니다.^^
PS C:\Users\rryan>
```

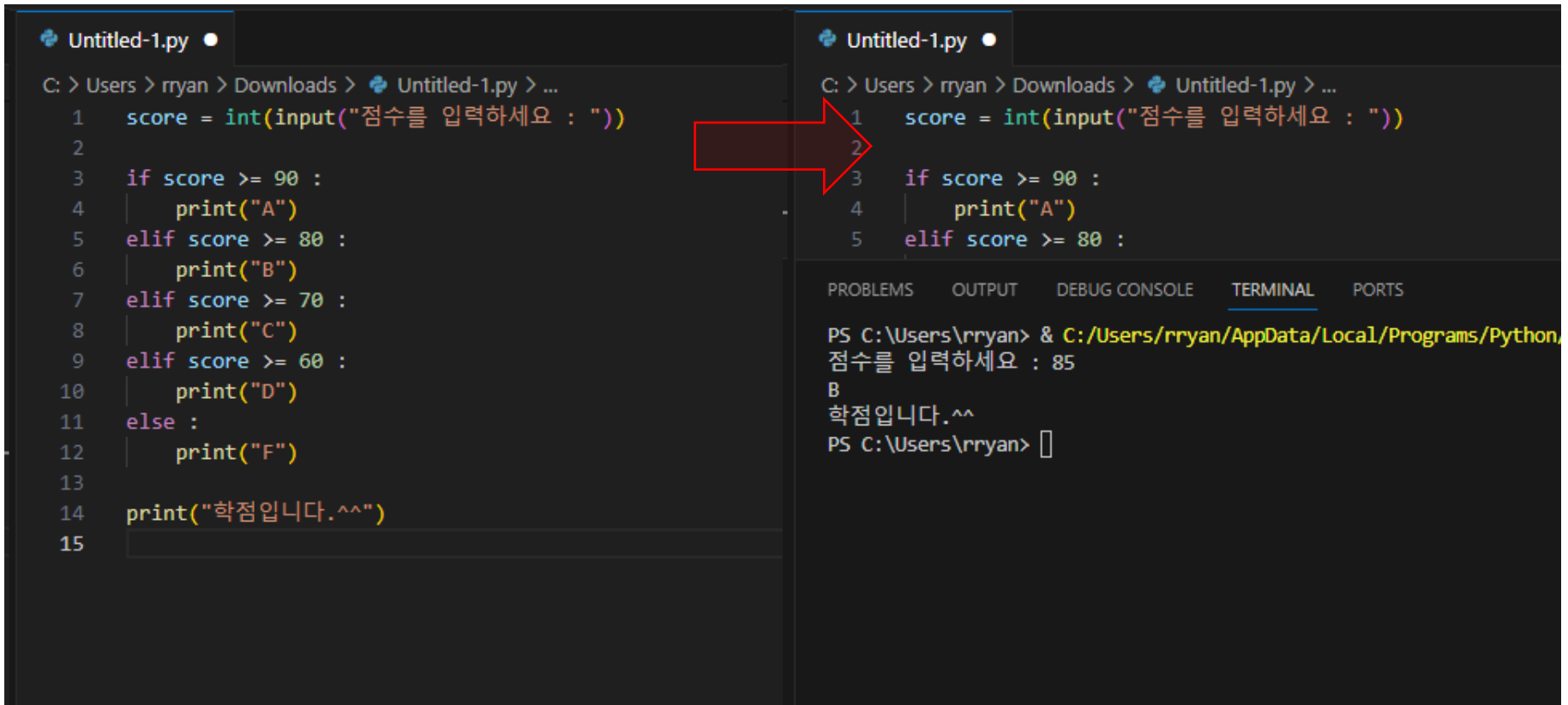
## Selection 03. 중첩 if 문

### 1. if~else~if~else 문



## Selection 03. 중첩 if 문

### 2. if~elif~else 문



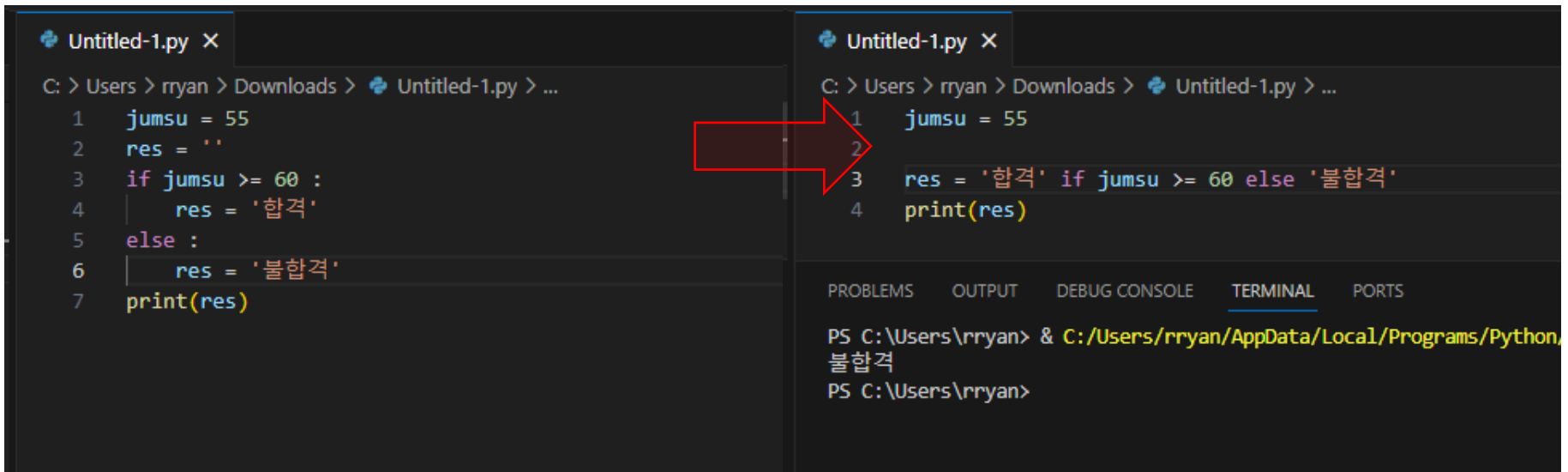
```
Untitled-1.py
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  score = int(input("점수를 입력하세요 : "))
2
3  if score >= 90 :
4      print("A")
5  elif score >= 80 :
6      print("B")
7  elif score >= 70 :
8      print("C")
9  elif score >= 60 :
10     print("D")
11 else :
12     print("F")
13
14 print("학점입니다.^^")
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/
점수를 입력하세요 : 85
B
학점입니다.^^
PS C:\Users\rryan> 
```

## Selection 03. 중첩 if 문

### 3. 삼항 연산자를 사용한 if 문



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 jumsu = 55
2 res = ''
3 if jumsu >= 60 :
4     res = '합격'
5 else :
6     res = '불합격'
7 print(res)
```

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 jumsu = 55
2
3 res = '합격' if jumsu >= 60 else '불합격'
4 print(res)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

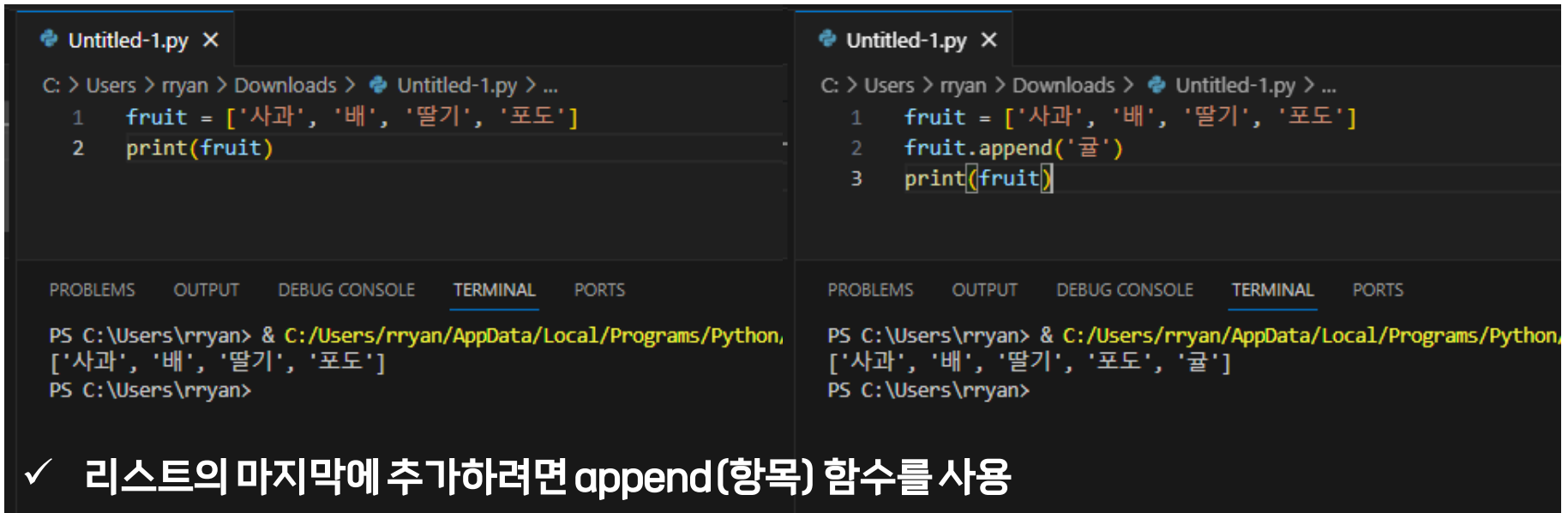
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-6/python.exe C:/Users/rryan/Downloads/Untitled-1.py
불합격
PS C:\Users\rryan>
```

- ✓ 3~6행의 if~else 문을 삼항 연산자를 사용해 한 줄로 줄일수도 있음
- ✓ 삼항연산자: ① 참인 경우 값, ② if 조건, ③ else 거짓인 경우 값

## Selection 04. if 문 응용

### 1. 리스트와 함께 사용

- 리스트(list) : 데이터 여러 개를 한 곳에 담아놓는 것
- 방법 : 대괄호 []로 묶고 그 안에 필요한 것들을 한꺼번에 넣음
- 예 : fruit 변수에 값 4개를 리스트 하나로 묶어서 대입



The image shows two side-by-side screenshots of a Python IDE. The left screenshot shows a file named 'Untitled-1.py' with the following code:

```
1 fruit = ['사과', '배', '딸기', '포도']
2 print(fruit)
```

The right screenshot shows the same file with an additional line of code:

```
1 fruit = ['사과', '배', '딸기', '포도']
2 fruit.append('귤')
3 print(fruit)
```

Below the code editor, the terminal output is shown for both. The left terminal shows the output of the first code block:

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/python.exe C:\Users\rryan\Downloads\Untitled-1.py
['사과', '배', '딸기', '포도']
PS C:\Users\rryan>
```

The right terminal shows the output of the second code block:

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/python.exe C:\Users\rryan\Downloads\Untitled-1.py
['사과', '배', '딸기', '포도', '귤']
PS C:\Users\rryan>
```

At the bottom of the image, there is a checkmark icon followed by the text:

✓ 리스트의 마지막에 추가하려면 append(항목) 함수를 사용



# Selection 04. if 문 응용

## 1. 리스트와 함께 사용

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 fruit = ['사과', '배', '딸기', '포도']
2 fruit.append('귤')
3
4 if '딸기' in fruit :
5     print("딸기가 있네요.^^")

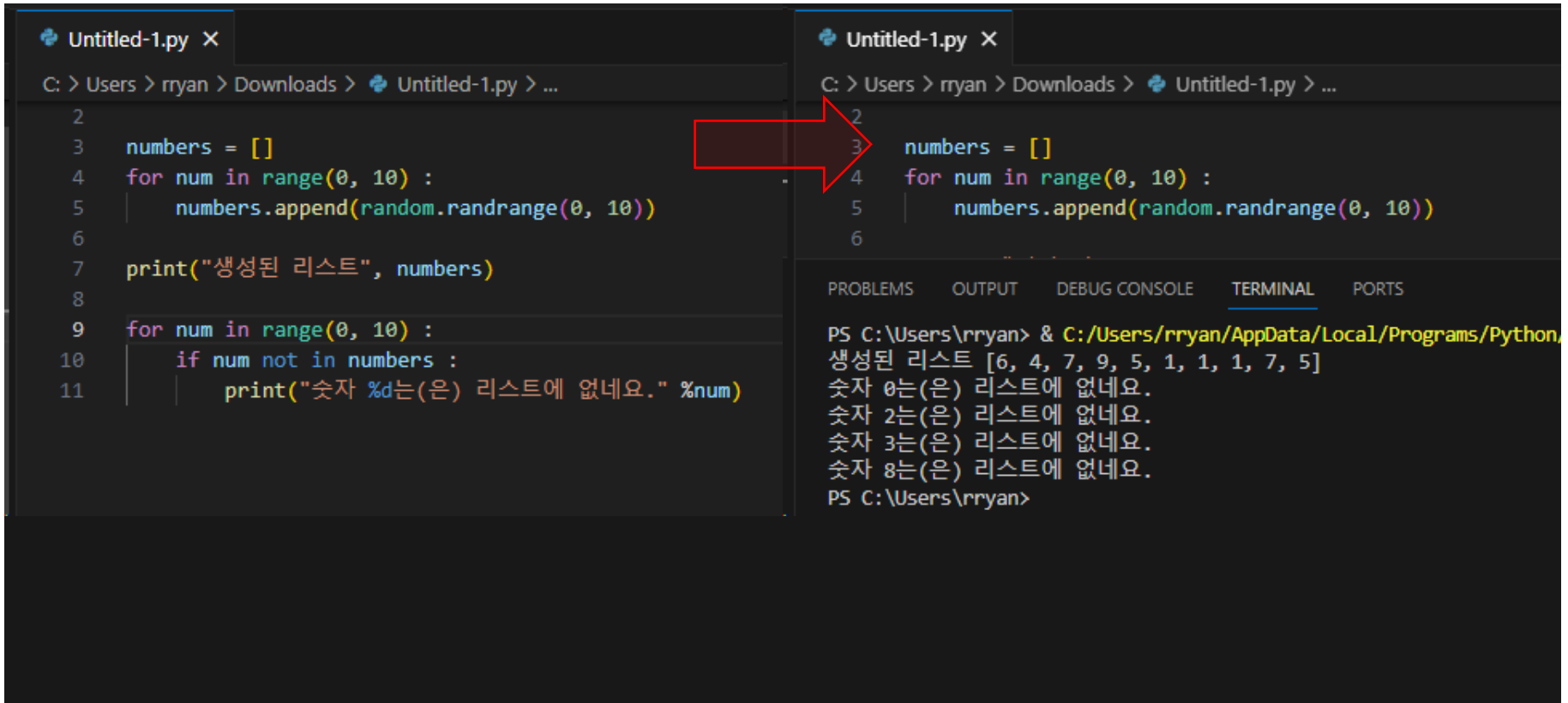
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + v [ ] [ ] ... ^ X

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
딸기가 있네요.^^
PS C:\Users\rryan>
```

✓ if 항목 in 리스트의 경우 리스트에 해당 항목이 있다면 True를 반환하고,  
if 항목 not in 리스트의 경우 리스트에 해당 항목이 없다면 True를 반환

# Selection 04. if 문 응용

## 1. 리스트와 함께 사용



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...

2
3 numbers = []
4 for num in range(0, 10) :
5     numbers.append(random.randrange(0, 10))
6
7 print("생성된 리스트", numbers)
8
9 for num in range(0, 10) :
10     if num not in numbers :
11         print("숫자 %d는(은) 리스트에 없네요." %num)
```

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...

2
3 numbers = []
4 for num in range(0, 10) :
5     numbers.append(random.randrange(0, 10))
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-32/python.exe C:\Users\rryan\Downloads\Untitled-1.py
생성된 리스트 [6, 4, 7, 9, 5, 1, 1, 1, 7, 5]
숫자 0는(은) 리스트에 없네요.
숫자 2는(은) 리스트에 없네요.
숫자 3는(은) 리스트에 없네요.
숫자 8는(은) 리스트에 없네요.
PS C:\Users\rryan>
```

## Chapter 5. 요약

- ✓ if 문의 형식은 다음과 같고, 조건식의 결과는 참 또는 거짓 두 가지임

if 조건식 :  
실행할 문장

- ✓ if 문에서 문장을 여러 개 실행하려면 들여쓰기를 정확히 해야 함
- ✓ 참일 때 실행하는 문장과 거짓일 때 실행하는 문장을 다르게 할 때는 if~else 문을 사용하거나 중첩 if 문 (if~else~if~else)을 사용하여야 함
- ✓ 중첩 if 문을 간략히 표현하려면 if~elif~else 문을 사용하여야 함
- ✓ 리스트 (list)는 여러 데이터를 한 곳에 담아 놓은 것으로 대괄호 []로 묶어야 함
- ✓ 'if 항목 in 리스트 :'는 리스트 안에 항목이 있으면 True를 반환함
- ✓ random.randrange(시작값, 끝값) 함수는 시작값 부터 끝값 -1까지 숫자 중에서 임의의 숫자 하나를 반환하고, eval(수식) 함수는 해당 수식을 계산함
- ✓ 'for num in range(0, 10) :'은 num에 0부터 9까지 숫자를 넣어서 반복함

## Chapter 6. 반복문

**Selection 01.** 이장에서 만들 프로그램

**Selection 02.** 기본 for 문

**Selection 03.** 중첩 for 문

**Selection 04.** while 문

**Selection 05.** break 문과 continue 문

# Selection 01. 이장에서 만들 프로그램 (실습&과제#2)

## 1. 구구단 출력

- For 문을 사용하여 구구단을 출력

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [] [X] ... ^ X
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
# 2단 ## 3단 ## 4단 ## 5단 ## 6단 ## 7단 ## 8단 ## 9단 #
2 X 1 = 2 3 X 1 = 3 4 X 1 = 4 5 X 1 = 5 6 X 1 = 6 7 X 1 = 7 8 X 1 = 8 9 X 1 = 9
2 X 2 = 4 3 X 2 = 6 4 X 2 = 8 5 X 2 = 10 6 X 2 = 12 7 X 2 = 14 8 X 2 = 16 9 X 2 = 18
2 X 3 = 6 3 X 3 = 9 4 X 3 = 12 5 X 3 = 15 6 X 3 = 18 7 X 3 = 21 8 X 3 = 24 9 X 3 = 27
2 X 4 = 8 3 X 4 = 12 4 X 4 = 16 5 X 4 = 20 6 X 4 = 24 7 X 4 = 28 8 X 4 = 32 9 X 4 = 36
2 X 5 = 10 3 X 5 = 15 4 X 5 = 20 5 X 5 = 25 6 X 5 = 30 7 X 5 = 35 8 X 5 = 40 9 X 5 = 45
2 X 6 = 12 3 X 6 = 18 4 X 6 = 24 5 X 6 = 30 6 X 6 = 36 7 X 6 = 42 8 X 6 = 48 9 X 6 = 54
2 X 7 = 14 3 X 7 = 21 4 X 7 = 28 5 X 7 = 35 6 X 7 = 42 7 X 7 = 49 8 X 7 = 56 9 X 7 = 63
2 X 8 = 16 3 X 8 = 24 4 X 8 = 32 5 X 8 = 40 6 X 8 = 48 7 X 8 = 56 8 X 8 = 64 9 X 8 = 72
2 X 9 = 18 3 X 9 = 27 4 X 9 = 36 5 X 9 = 45 6 X 9 = 54 7 X 9 = 63 8 X 9 = 72 9 X 9 = 81
PS C:\Users\rryan>
```

## Selection 02. 기본 for 문

### 1. 반복문의 개념과 필요성

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py
1 print("안녕하세요? for 문을 공부 중입니다. ^^")
2 print("안녕하세요? for 문을 공부 중입니다. ^^")
3 print("안녕하세요? for 문을 공부 중입니다. ^^")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
안녕하세요? for 문을 공부 중입니다. ^^
안녕하세요? for 문을 공부 중입니다. ^^
안녕하세요? for 문을 공부 중입니다. ^^
PS C:\Users\rryan>
```

**반복문을 사용하지 않는 경우**

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 for i in range(0, 3, 1) :
2     print("안녕하세요? for 문을 공부 중입니다. ^^")
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
안녕하세요? for 문을 공부 중입니다. ^^
안녕하세요? for 문을 공부 중입니다. ^^
안녕하세요? for 문을 공부 중입니다. ^^
PS C:\Users\rryan>
```

**반복문을 사용한 경우**

✓ 반복문은 문장을 반복해서 만드는 것으로, 반복문이 없다면 코드가 너무 길어짐

# Selection 02. 기본 for 문

## 2. For 문의 개념

- 기본 형식

```
for 변수 in range(시작값, 끝값+1, 증가값) :  
    이 부분을 반복
```

```
Untitled-1.py X  
C: > Users > rryan > Downloads > Untitled-1.py > ...  
1 for i in range(0, 3, 1) :  
2 |     print("안녕하세요? for 문을 공부 중입니다. ^^")  
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/Python.exe Untitled-1.py  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
PS C:\Users\rryan>
```

```
Untitled-1.py X  
C: > Users > rryan > Downloads > Untitled-1.py > ...  
1 for i in [0, 1, 2] :  
2 |     print("안녕하세요? for 문을 공부 중입니다. ^^")  
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/Python.exe Untitled-1.py  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
PS C:\Users\rryan>
```

✓ range() 함수를 사용하는 경우와, []를 사용하는 경우는 모두 값이 같게 출력

## Selection 02. 기본 for 문

### 2. For 문의 개념

The image displays two side-by-side screenshots of a Python IDE, likely VS Code, showing the execution of a for loop. Both screenshots show a file named 'Untitled-1.py' and a terminal window.

**Left Screenshot:** The code defines a for loop with a loop variable `i`.

```
1 for i in range(0, 3, 1) :  
2     print("%d : 안녕하세요? for 문을 공부 중입니다. ^^" % i)  
3
```

The terminal output shows the loop executing three times, with the value of `i` (0, 1, 2) printed before each message.

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Pythc  
0 : 안녕하세요? for 문을 공부 중입니다. ^^  
1 : 안녕하세요? for 문을 공부 중입니다. ^^  
2 : 안녕하세요? for 문을 공부 중입니다. ^^  
PS C:\Users\rryan>
```

**Right Screenshot:** The code defines a for loop using an underscore `_` as a placeholder for the loop variable.

```
1 for _ in range(0, 3, 1) :  
2     print("안녕하세요? for 문을 공부 중입니다. ^^" )  
3
```

The terminal output shows the loop executing three times, but no values are printed before the messages, as the loop variable is not used.

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Pyt  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
안녕하세요? for 문을 공부 중입니다. ^^  
PS C:\Users\rryan>
```

✓ 각 행의 맨 앞에 | 값을 출력할 수 있고, 이를 생략하기 위해서는(=변수 없이 사용하기 위해서는) `_`(언더바)도 사용이 가능



# Selection 02. 기본 for 문

## 2. For 문의 개념



The screenshot shows a Python IDE with a file named 'Untitled-1.py'. The code in the editor is a for loop that iterates from 2 down to 0, printing a message for each iteration. The output window shows the execution results, confirming that the loop runs three times with values 2, 1, and 0.

```
Untitled-1.py X
```

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
```

```
1  for i in range(2, -1, -1) :
2      print("%d : 안녕하세요? for 문을 공부 중입니다. ^^" % i)
3
```

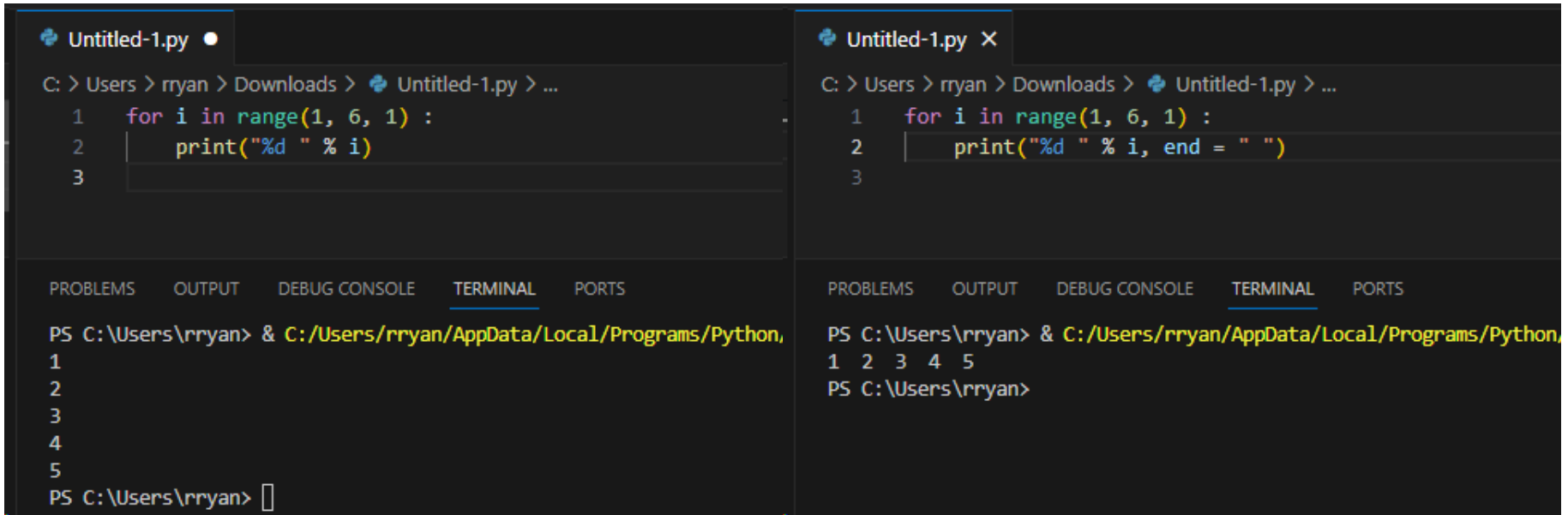
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + v [ ] [ ] ... ^ X

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
2 : 안녕하세요? for 문을 공부 중입니다. ^^
1 : 안녕하세요? for 문을 공부 중입니다. ^^
0 : 안녕하세요? for 문을 공부 중입니다. ^^
PS C:\Users\rryan> [ ]
```

✓ range() 함수의 시작 값을 2, i 값을 1씩 줄여 0이 될 때까지 3번 실행

## Selection 02. 기본 for 문

### 2. For 문의 개념



The image shows two side-by-side screenshots of a Python IDE, likely VS Code, with a file named 'Untitled-1.py'. Both screenshots show the same Python code:

```
1 for i in range(1, 6, 1):  
2     print("%d " % i)  
3
```

The left screenshot shows the output in the terminal as five separate lines, each containing a number from 1 to 5 followed by a space:

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python...  
1  
2  
3  
4  
5  
PS C:\Users\rryan> 
```

The right screenshot shows the output in the terminal as a single line containing the numbers 1 through 5 separated by spaces:

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python...  
1 2 3 4 5  
PS C:\Users\rryan> 
```

The difference between the two is the addition of the parameter `end = " "` in the `print` statement in the right screenshot, which changes the default newline character to a space.

- ✓ 1~5의 숫자들을 차례로 출력하기 위해 `range(1, 6, 1)`을 통해 값을 출력하고, `print()` 함수이 마지막에 `end = " "`을 추가함으로써 마지막을 공백으로 처리해 한 줄로 출력

## Selection 02. 기본 for 문

### 3. For 문을 활용한 합계 구하기

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1 hap = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
2 print("1에서 10까지의 합계 : %d" % hap)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python
1에서 10까지의 합계 : 55
PS C:\Users\rryan>
```

▶ ◂ ▢ ...

1부터 10까지 변할 i 변수 준비

for i 변수가 1을 시작으로 10까지 1씩 증가  
hap값에 i값을 더해 줌

hap값 출력

For 문 작성 내용

✓ For 문을 배우기 전의 방식으로 1~10까지의 합계를 구하는 코드  
10까지가 아닌 1,000,000까지의 합계라면...?

## Selection 02. 기본 for 문

### 3. For 문을 활용한 합계 구하기

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  i = 0
2
3  for i in range(1, 11, 1) :
4      hap = hap + i
5
6  print("1에서 10까지의 합계 : %d" % hap)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python
Traceback (most recent call last):
  File "c:\Users\rryan\Downloads\Untitled-1.py", line 4, in <module>
    hap = hap + i
    ^^^
NameError: name 'hap' is not defined. Did you mean: 'map'?
PS C:\Users\rryan>
```

1부터 10까지 변할 i 변수 준비

for i 변수가 1을 시작으로 10까지 1씩 증가  
hap값에 i값을 더해 줌

hap값 출력

**For문 작성 내용**

✓ hap = hap + i 를 계산 해야하는데 hap 변수가 선언되지 않아서 오류 발생

## Selection 02. 기본 for 문

### 3. For 문을 활용한 합계 구하기

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, hap = 0, 0
2
3 for i in range(1, 11, 1) :
4     hap = hap + i
5
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

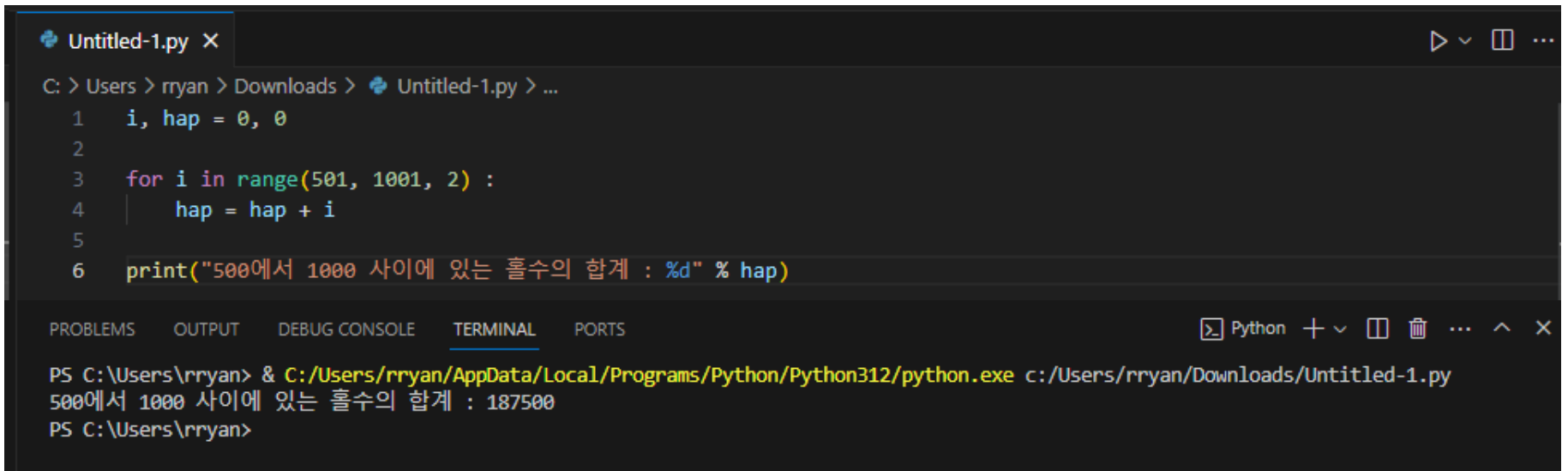
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe
1에서 10까지의 합계 : 55
PS C:\Users\rryan>
```

✓ hap 변수를 선언함으로써 오류를 수정

Iteration	i	Operation	hap
Initial			0
제1회	1	+	1
제2회	2	+	3
제3회	3	+	6
...	...	...	...
제9회	9	+	45
제10회	10	+	55

## Selection 02. 기본 for 문

### 3. For 문을 활용한 합계 구하기



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  i, hap = 0, 0
2
3  for i in range(501, 1001, 2) :
4      hap = hap + i
5
6  print("500에서 1000 사이에 있는 홀수의 합계 : %d" % hap)
```

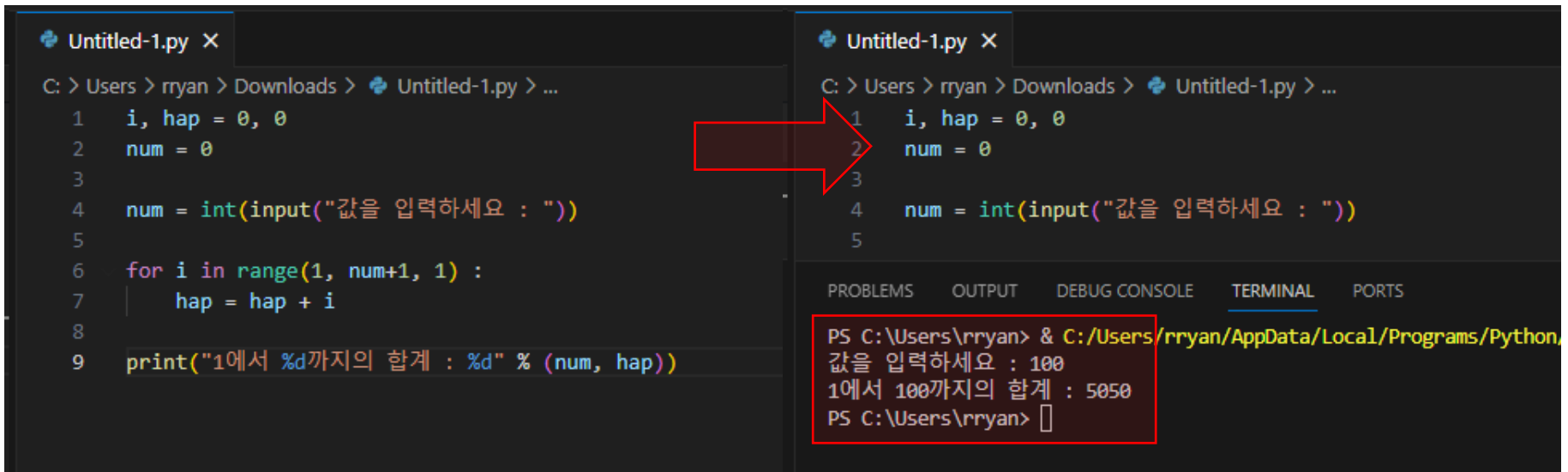
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - □ □ ... ^ X

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
500에서 1000 사이에 있는 홀수의 합계 : 187500
PS C:\Users\rryan>
```

- ✓ 500과 1000사이에 있는 홀수의 합계를 구하는 프로그램
- ✓ For 문의 범위를 지정하는 range() 함수의 시작값, 끝값+1, 증가값만 적절히 변경하면 다양한 형태의 합계를 구할 수 있음

## Selection 02. 기본 for 문

### 4. 키보드로 입력한 값까지 합계 구하기



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, hap = 0, 0
2 num = 0
3
4 num = int(input("값을 입력하세요 : "))
5
6 for i in range(1, num+1, 1) :
7     hap = hap + i
8
9 print("1에서 %d까지의 합계 : %d" % (num, hap))
```

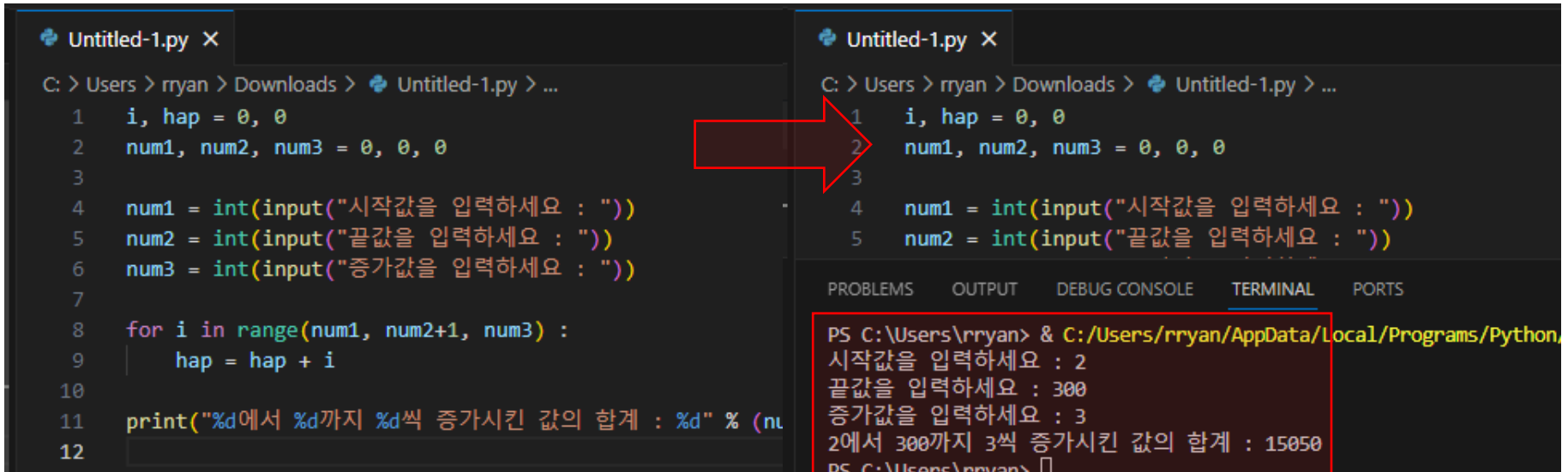
```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, hap = 0, 0
2 num = 0
3
4 num = int(input("값을 입력하세요 : "))
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python38-32/python.exe C:\Users\rryan\Downloads\Untitled-1.py
값을 입력하세요 : 100
1에서 100까지의 합계 : 5050
PS C:\Users\rryan>
```

✓ input() 함수로 1부터 사용자가 입력한 수까지 합계를 구하는 프로그램

## Selection 02. 기본 for 문

### 4. 키보드로 입력한 값까지 합계 구하기



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, hap = 0, 0
2 num1, num2, num3 = 0, 0, 0
3
4 num1 = int(input("시작값을 입력하세요 : "))
5 num2 = int(input("끝값을 입력하세요 : "))
6 num3 = int(input("증가값을 입력하세요 : "))
7
8 for i in range(num1, num2+1, num3):
9     hap = hap + i
10
11 print("%d에서 %d까지 %d씩 증가시킨 값의 합계 : %d" % (num1, num2, num3, hap))
12
```

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, hap = 0, 0
2 num1, num2, num3 = 0, 0, 0
3
4 num1 = int(input("시작값을 입력하세요 : "))
5 num2 = int(input("끝값을 입력하세요 : "))
6 num3 = int(input("증가값을 입력하세요 : "))
7
8 for i in range(num1, num2+1, num3):
9     hap = hap + i
10
11 print("%d에서 %d까지 %d씩 증가시킨 값의 합계 : %d" % (num1, num2, num3, hap))
12
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

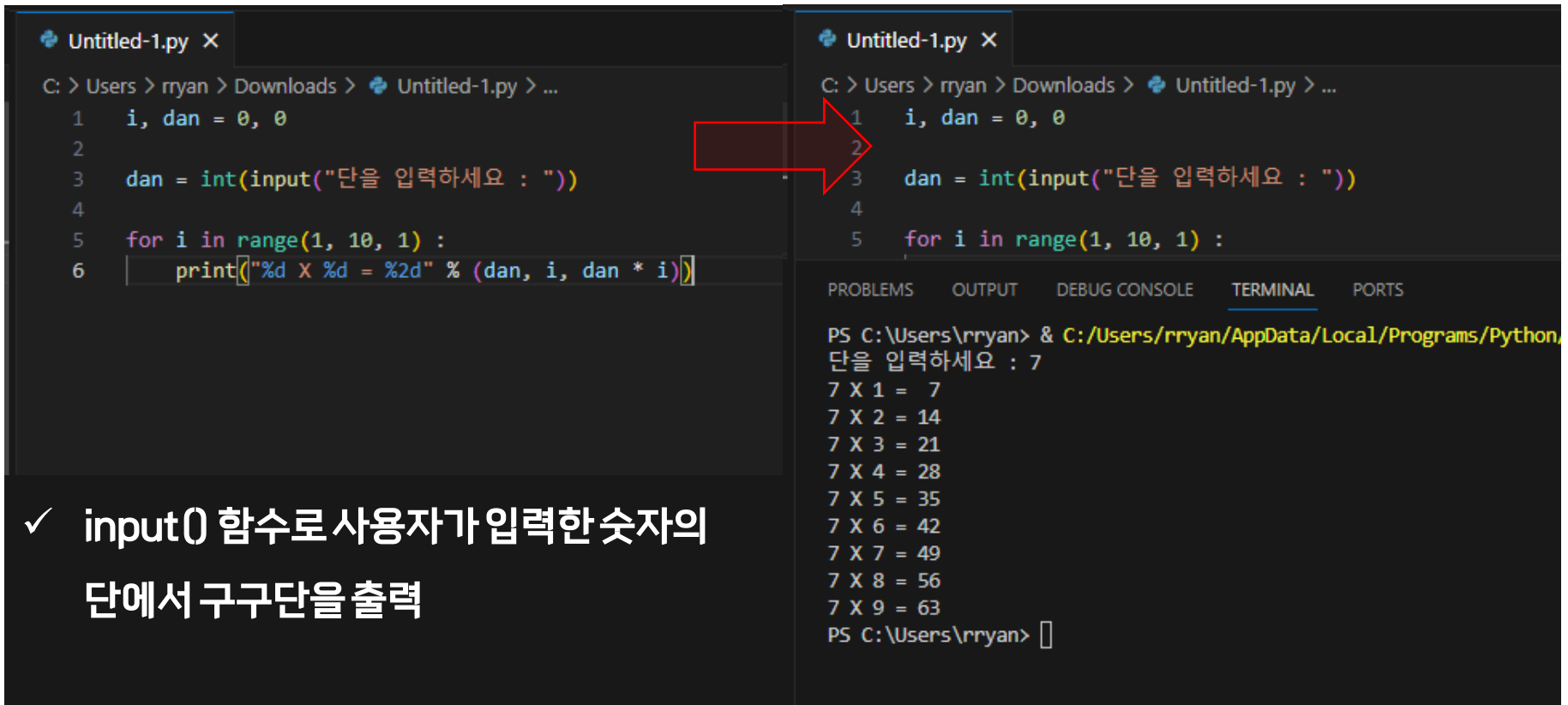
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python38-32/python.exe C:/Users/rryan/Downloads/Untitled-1.py
시작값을 입력하세요 : 2
끝값을 입력하세요 : 300
증가값을 입력하세요 : 3
2에서 300까지 3씩 증가시킨 값의 합계 : 15050
PS C:\Users\rryan>
```

- ✓ input() 함수로 사용자에게 시작 값, 끝 값, 증가 값까지 입력 받은 수의 합계를 구하는 프로그램



## Selection 02. 기본 for 문

### 4. 키보드로 입력한 값까지 합계 구하기



```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, dan = 0, 0
2
3 dan = int(input("단을 입력하세요 : "))
4
5 for i in range(1, 10, 1) :
6     print("%d X %d = %2d" % (dan, i, dan * i))
```

✓ input() 함수로 사용자가 입력한 숫자의  
단에서 구구단을 출력

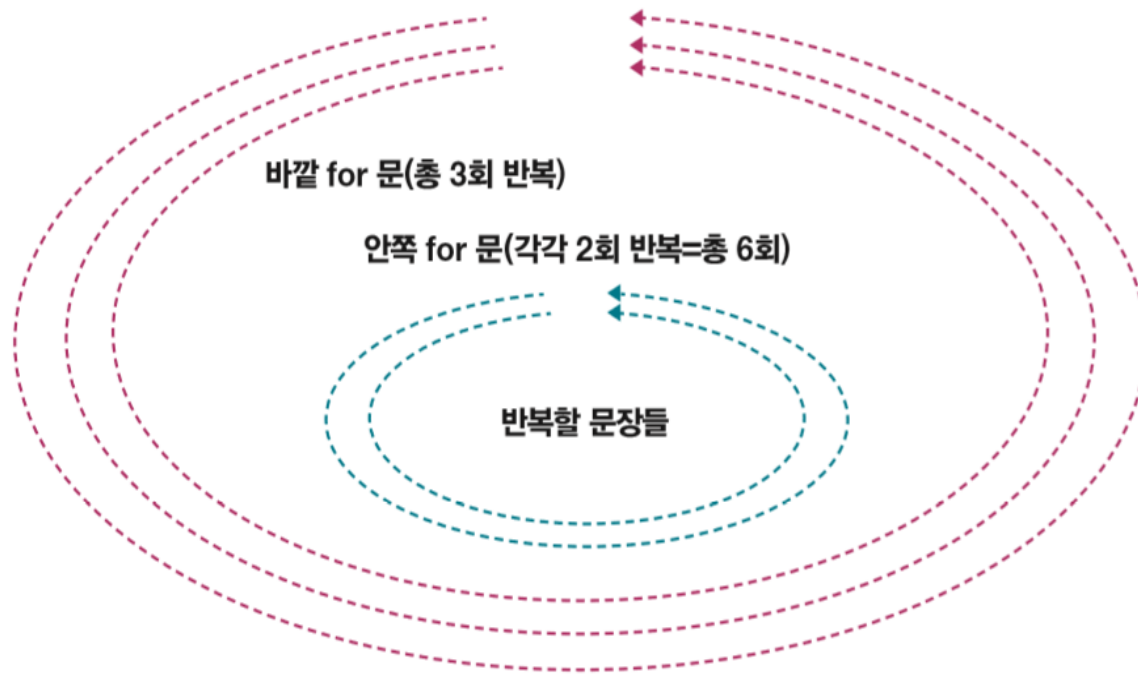
```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 i, dan = 0, 0
2
3 dan = int(input("단을 입력하세요 : "))
4
5 for i in range(1, 10, 1) :
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\rryan\Downloads\Untitled-1.py
단을 입력하세요 : 7
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63
PS C:\Users\rryan> 
```

## Selection 03. 중첩 for 문

### 1. 중첩 for 문의 개념



중첩 for 문의 작동 개념

## Selection 03. 중첩 for 문

### 1. 중첩 for 문의 개념

```
Untitled-1.py
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 for i in range(0, 3, 1) :
2     for k in range(0, 2, 1) :
3         print("파이썬은 꿀잼입니다.^ (i값 : %d, k값 : %d)"
4             )
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Pyt
파이썬은 꿀잼입니다.^ (i값 : 0, k값 : 0)
파이썬은 꿀잼입니다.^ (i값 : 0, k값 : 1)
파이썬은 꿀잼입니다.^ (i값 : 1, k값 : 0)
파이썬은 꿀잼입니다.^ (i값 : 1, k값 : 1)
파이썬은 꿀잼입니다.^ (i값 : 2, k값 : 0)
파이썬은 꿀잼입니다.^ (i값 : 2, k값 : 1)
```

❶ 외부 for 문 1회 : i에 0을 대입  
내부 for 문 1회 : k에 0을 대입 후 print() 함수 수행  
내부 for 문 2회 : k에 1을 대입 후 print() 함수 수행

❷ 외부 for 문 2회 : i에 1을 대입  
내부 for 문 1회 : k에 0을 대입 후 print() 함수 수행  
내부 for 문 2회 : k에 1을 대입 후 print() 함수 수행

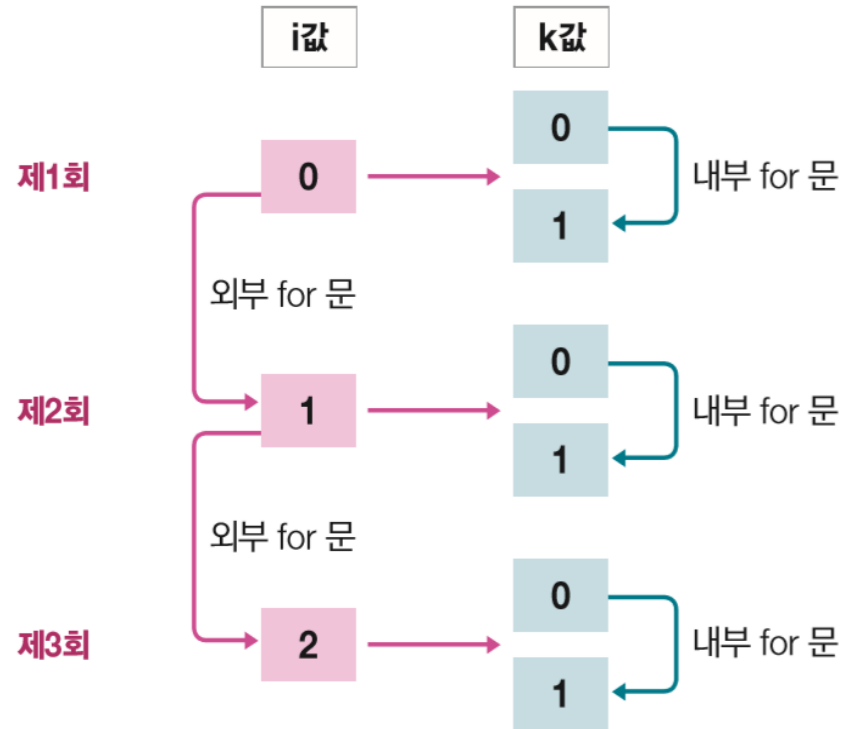
❸ 외부 for 문 3회 : i에 2를 대입  
내부 for 문 1회 : k에 0을 대입 후 print() 함수 수행  
내부 for 문 2회 : k에 1을 대입 후 print() 함수 수행

✓ 외부 변수인 i는 계속 0, 1, 2로 변경된 후 끝나지만,  
내부 변수인 k는 0과 1을 계속 반복

**처리 순서**

## Selection 03. 중첩 for 문

### 1. 중첩 for 문의 개념



중첩 for 문에서 i와 k값 변화

## Selection 03. 중첩 for 문

### 2. 중첩 for 문의 활용

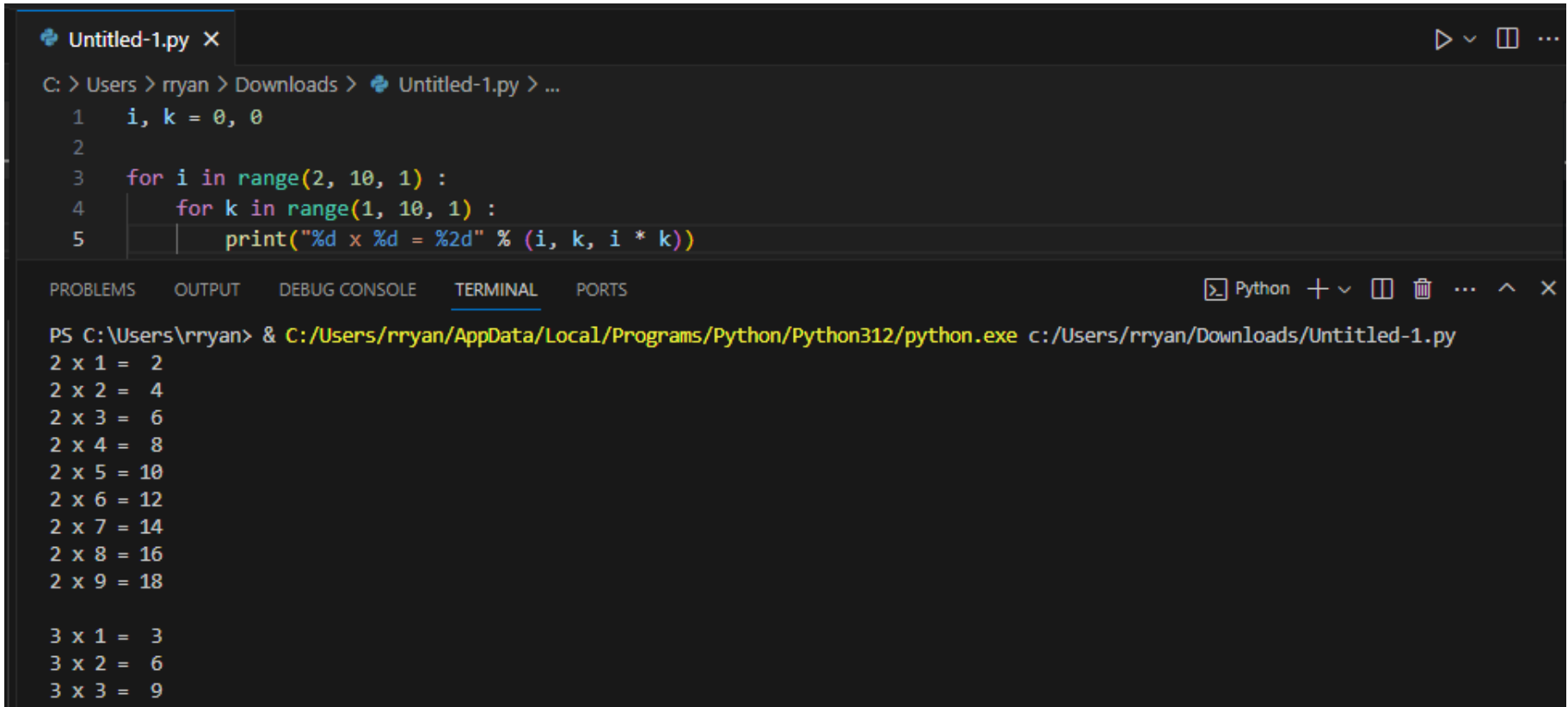
- 중첩 for 문을 활용해 2단부터 9단까지 구구단 출력



구구단에서 i와 k 변수 추출

## Selection 03. 중첩 for 문

### 2. 중첩 for 문의 활용



The screenshot shows a Python IDE with a file named 'Untitled-1.py'. The code defines two nested loops: an outer loop for 'i' ranging from 2 to 10 (exclusive) and an inner loop for 'k' ranging from 1 to 10 (exclusive). The inner loop prints the multiplication of 'i' and 'k'. The output in the terminal shows the results for i=2 (k=1 to 9) and i=3 (k=1 to 3).

```
1 i, k = 0, 0
2
3 for i in range(2, 10, 1) :
4     for k in range(1, 10, 1) :
5         print("%d x %d = %2d" % (i, k, i * k))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [Icons] ^ X

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
```

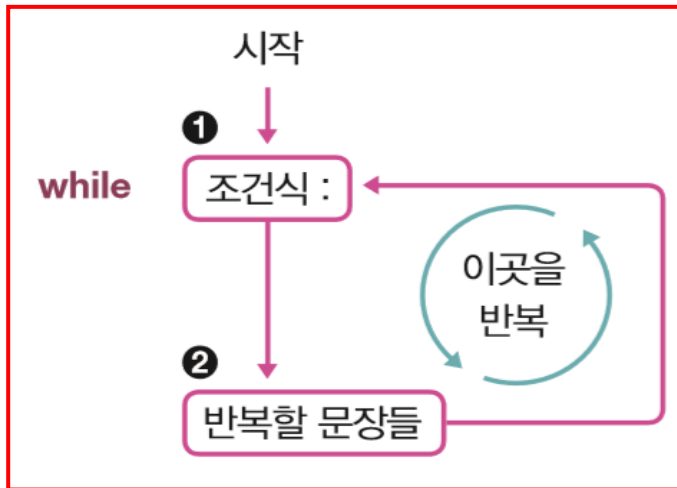
## Selection 04. while 문

### 1. for 문과 while 문 비교

- For 문의 형식

```
for 변수 in range(시작값, 끝값+1, 증가값) :
```

- For 문은 반복할 횟수를 range() 함수에서 결정한 후 그 횟수만큼 반복하지만, while 문은 반복 횟수를 결정하기보다 조건이 참일 때 반복하는 방식



- For 문과 비슷하게 사용할 수 있는 while 문의 형식

```
변수 = 시작값  
while 변수 < 끝값 :  
    이 부분을 반복  
    변수 = 변수 + 증가값
```

# Selection 04. while 문

## 1. for 문과 while 문 비교

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1 for i in range(0, 3, 1) :
2     print("%d : 안녕하세요? for 문을 공부 중입니다." % i)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
0 : 안녕하세요? for 문을 공부 중입니다.
1 : 안녕하세요? for 문을 공부 중입니다.
2 : 안녕하세요? for 문을 공부 중입니다.
PS C:\Users\rryan>

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1 i = 0
2 while i < 3 :
3     print("%d : 안녕하세요? while 문을 공부 중입니다.")
4     i = i + 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
%d : 안녕하세요? while 문을 공부 중입니다.
%d : 안녕하세요? while 문을 공부 중입니다.
%d : 안녕하세요? while 문을 공부 중입니다.
PS C:\Users\rryan>



# Selection 04. while 문

## 1. for 문과 while 문 비교

Untitled-1.py

C: > Users > rryan > Downloads > Untitled-1.py > ...  
1 hap = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10  
2 print("1에서 10까지의 합계 : %d" % hap)  
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python, 1에서 10까지의 합계 : 55  
PS C:\Users\rryan>

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...  
1 i, hap = 0, 0  
2  
3 i = 1  
4 while i < 11 :  
5 hap = hap + i  
6 i = i + 1  
7  
8 print("1에서 10까지의 합계 : %d" % hap)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python, 1에서 10까지의 합계 : 55  
PS C:\Users\rryan>

## Selection 04. while 문

### 2. 무한루프를 하는 while 문

- 무한루프 적용: ' while 조건식: '에 들어가는 조건식을 True로 지정

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py
1 while True :
2     print("ㅋ", end = " ")
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

✓ 중단하기 위해서는 Ctrl + C 를 입력

```
graph TD
    Start[시작] --> Cond[1 조건식:]
    Cond --> Statements[2 반복할 문장들]
    Statements -- "이곳을 반복" --> Cond
```

While 문을 이용한 무한루프

## Selection 04. while 문

## 2. 무한루프를 하는 while 문

The image shows two side-by-side screenshots of a Python IDE. The left window, titled 'Untitled-1.py', displays the following code:

```
1 hap = 0
2 a, b = 0, 0
3
4 while True :
5     a = int(input("더할 첫 번째 수를 입력하세요 : "))
6     b = int(input("더할 첫 번째 수를 입력하세요 : "))
7     hap = a + b
8     print("%d + %d = %d" % (a, b, hap))
9
10
```

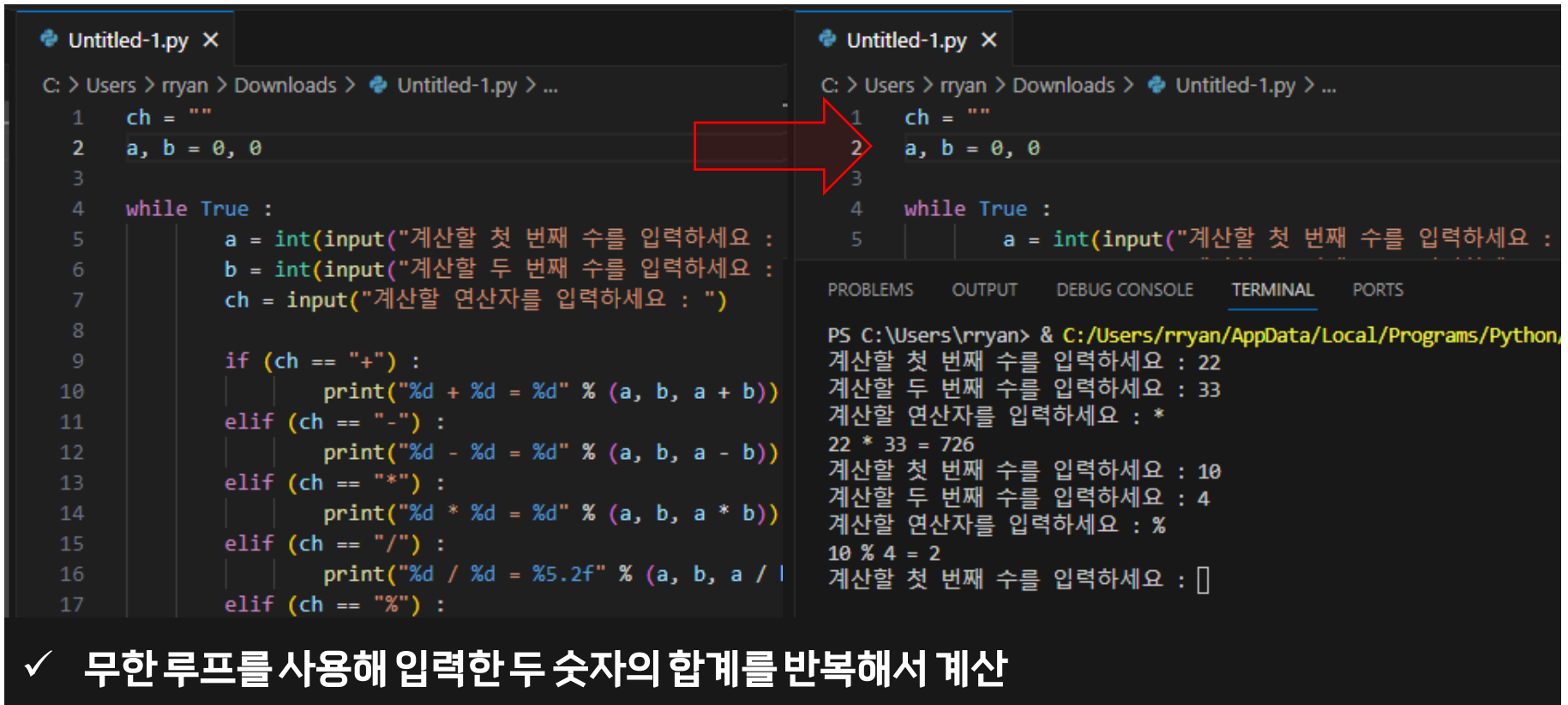
The right window, also titled 'Untitled-1.py', shows the same code but with a red arrow pointing to the `while True :` line. Below the code editor is a terminal window with the following output:

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/
더할 첫 번째 수를 입력하세요 : 55
더할 첫 번째 수를 입력하세요 : 22
55 + 22 = 77
더할 첫 번째 수를 입력하세요 : 77
더할 첫 번째 수를 입력하세요 : 128
77 + 128 = 205
```

✓ 무한루프를 사용해 입력한 두 숫자의 합계를 반복해서 계산

## Selection 04. while 문

### 2. 무한루프를 하는 while 문



The image shows a code editor with two panes. The left pane displays the source code of a Python program named 'Untitled-1.py'. The code uses a `while True` loop to repeatedly prompt the user for two numbers and an operator, then performs the corresponding arithmetic operation. The right pane shows the terminal output of the program, where the user has entered '22', '33', and '\*', resulting in the calculation '22 \* 33 = 726'. A red arrow points from the code in the left pane to the terminal output in the right pane.

```
1 ch = ""
2 a, b = 0, 0
3
4 while True :
5     a = int(input("계산할 첫 번째 수를 입력하세요 : "))
6     b = int(input("계산할 두 번째 수를 입력하세요 : "))
7     ch = input("계산할 연산자를 입력하세요 : ")
8
9     if (ch == "+") :
10         print("%d + %d = %d" % (a, b, a + b))
11     elif (ch == "-") :
12         print("%d - %d = %d" % (a, b, a - b))
13     elif (ch == "*") :
14         print("%d * %d = %d" % (a, b, a * b))
15     elif (ch == "/" ) :
16         print("%d / %d = %5.2f" % (a, b, a / b))
17     elif (ch == "%") :
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

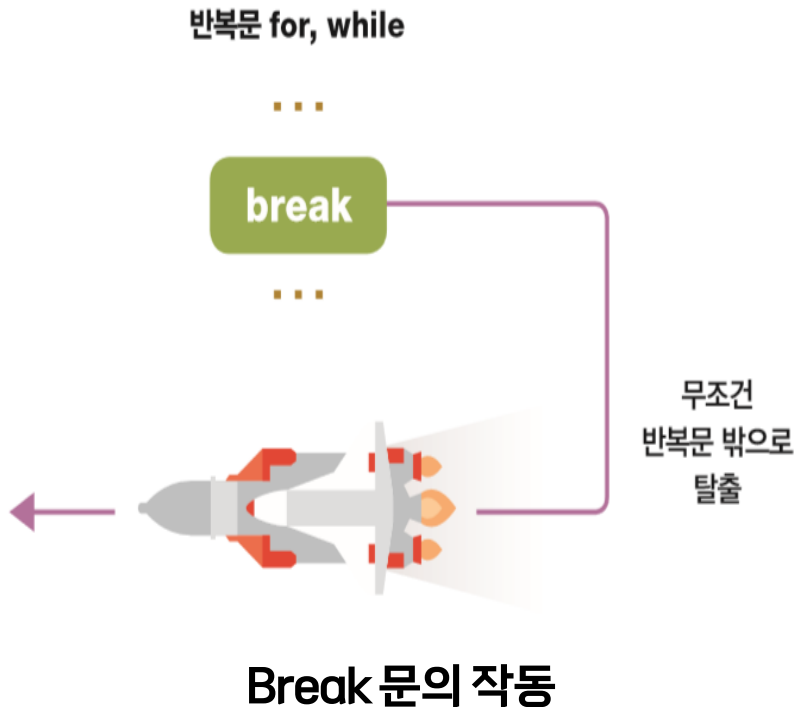
```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/Python.exe
계산할 첫 번째 수를 입력하세요 : 22
계산할 두 번째 수를 입력하세요 : 33
계산할 연산자를 입력하세요 : *
22 * 33 = 726
계산할 첫 번째 수를 입력하세요 : 10
계산할 두 번째 수를 입력하세요 : 4
계산할 연산자를 입력하세요 : %
10 % 4 = 2
계산할 첫 번째 수를 입력하세요 : [
```

✓ 무한 루프를 사용해 입력한 두 숫자의 합계를 반복해서 계산

## Selection 05. break 문과 continue 문

### 1. 반복문을 탈출시키는 break 문

- 계속되는 반복을 논리적으로 빠져나가는 방법



```
for i in range(1, 100) :  
    print("for 문을 %d번 실행했습니다." % i)  
    break
```

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

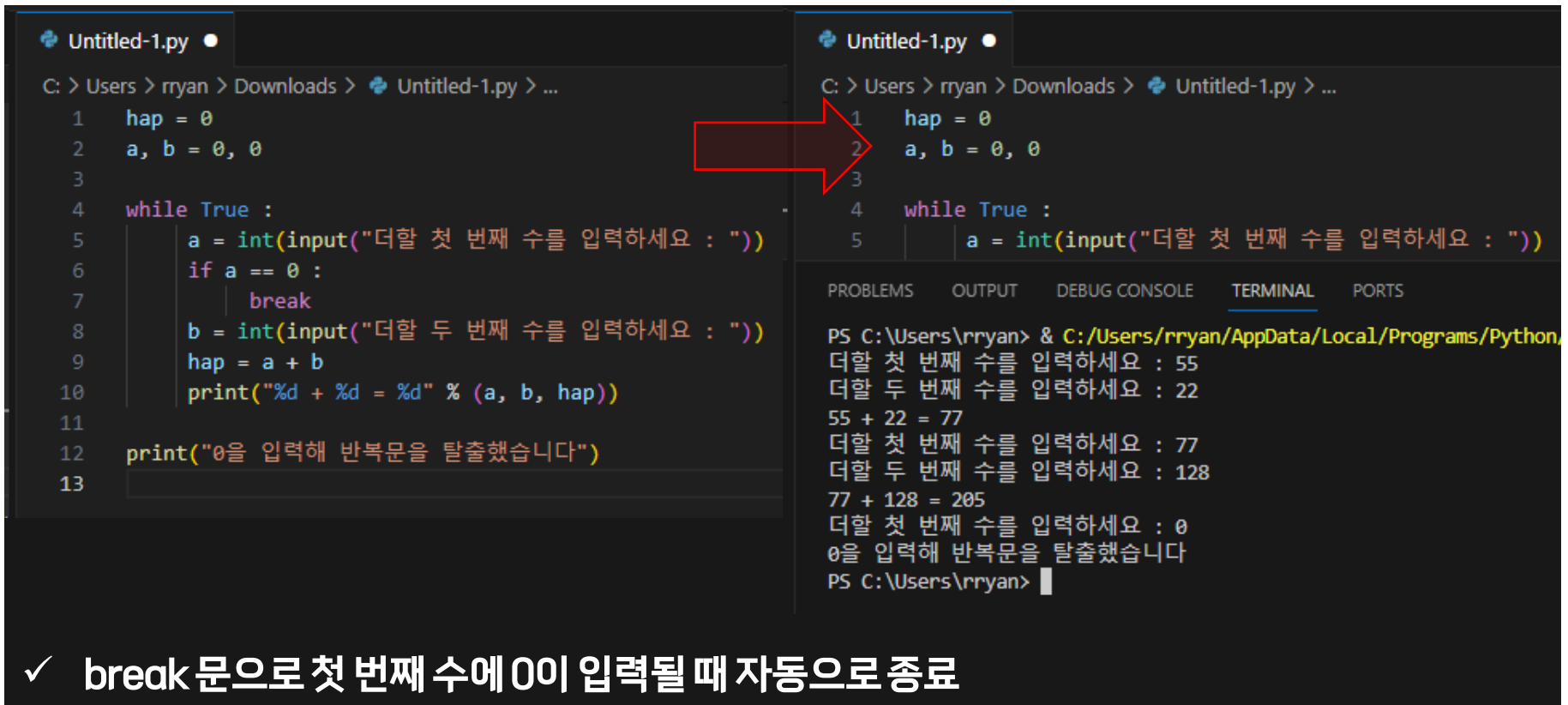
```
1  for i in range(1, 100) :  
2      print("for 문을 %d번 실행했습니다." % i)  
3      break  
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python38-32/Python.exe C:\Users\rryan\Downloads\Untitled-1.py  
for 문을 1번 실행했습니다.  
PS C:\Users\rryan>
```

## Selection 05. break 문과 continue 문

### 1. 반복문을 탈출시키는 break 문



```
Untitled-1.py
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  hap = 0
2  a, b = 0, 0
3
4  while True :
5      a = int(input("더할 첫 번째 수를 입력하세요 : "))
6      if a == 0 :
7          break
8      b = int(input("더할 두 번째 수를 입력하세요 : "))
9      hap = a + b
10     print("%d + %d = %d" % (a, b, hap))
11
12     print("0을 입력해 반복문을 탈출했습니다")
13
```

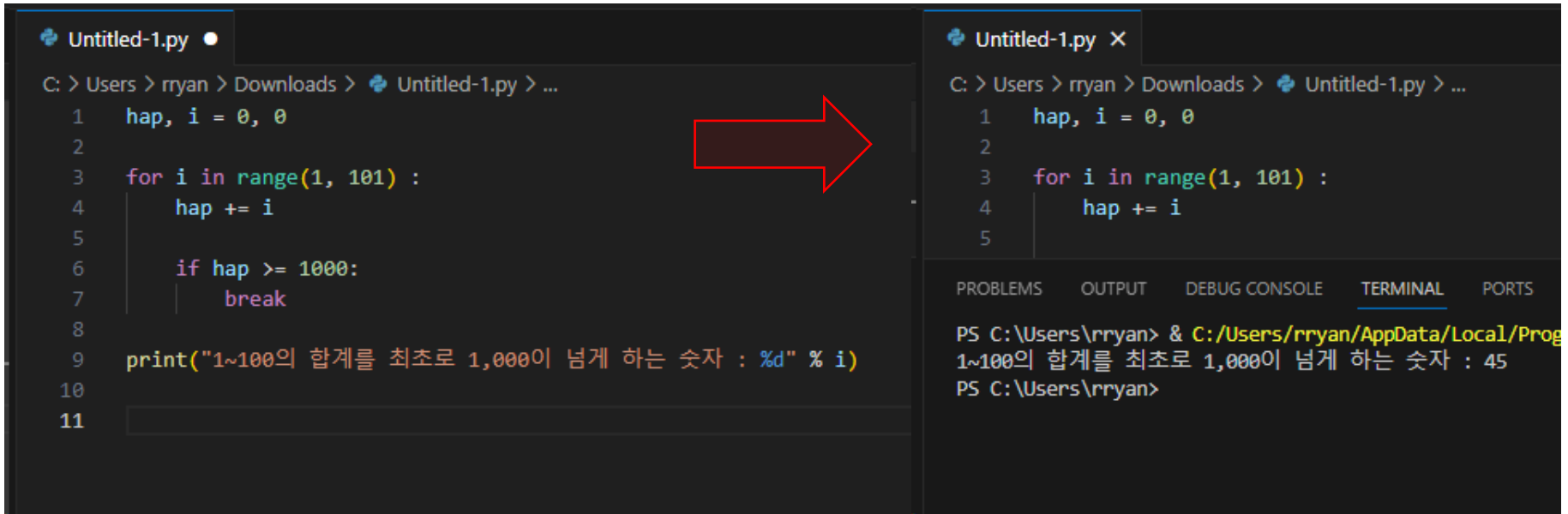
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\rryan\Downloads\Untitled-1.py
더할 첫 번째 수를 입력하세요 : 55
더할 두 번째 수를 입력하세요 : 22
55 + 22 = 77
더할 첫 번째 수를 입력하세요 : 77
더할 두 번째 수를 입력하세요 : 128
77 + 128 = 205
더할 첫 번째 수를 입력하세요 : 0
0을 입력해 반복문을 탈출했습니다
PS C:\Users\rryan>
```

✓ break 문으로 첫 번째 수에 0이 입력될 때 자동으로 종료

## Selection 05. break 문과 continue 문

### 1. 반복문을 탈출시키는 break 문



```
Untitled-1.py
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 hap, i = 0, 0
2
3 for i in range(1, 101) :
4     hap += i
5
6     if hap >= 1000:
7         break
8
9 print("1~100의 합계를 최초로 1,000이 넘게 하는 숫자 : %d" % i)
10
11
```

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 hap, i = 0, 0
2
3 for i in range(1, 101) :
4     hap += i
5
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-6/Python.exe C:\Users\rryan\Downloads\Untitled-1.py
1~100의 합계를 최초로 1,000이 넘게 하는 숫자 : 45
PS C:\Users\rryan>
```

✓ 누적 합계가 1,000 이상이 되는 시작 지점을 계산하기 위한 코드

## Selection 05. break 문과 continue 문

### 2. 반복문으로 다시 돌아가게 하는 continue 문

- continue 문을 만나면 블록의 남은 부분을 무조건 건너뛰고 반복문의 처음을 실행



continue 문의 작동

```
Untitled-1.py X
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  hap, i = 0,0
2
3  for i in range(1, 101):
4      if i % 3 == 0:
5          continue
6
7      hap += i
8
9  print("1~100의 합계(3의 배수 제외) : %d" % hap)
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python39-6/python.exe C:/Users/rryan/Downloads/Untitled-1.py
1~100의 합계(3의 배수 제외) : 3367
PS C:\Users\rryan>
```

✓ 1~100의 합계를 구하되, 3의 배수를 제외하고 더하기



## Chapter 6. 요약

- ✓ for 문의 형식은 다음과 같음

```
for 변수 in range(시작값, 끝값+1, 증가값) :  
    이 부분을 반복
```

- ✓ range() 함수는 지정된 범위의 값을 반환하고, range(0, 3, 1)의 경우 0에서 시작해 2까지 1씩 증가하는 값을 반환
- ✓ while 문의 형식은 다음과 같음

```
변수 = 시작값  
while 변수 < 끝값 :  
    이 부분을 반복  
    변수 = 변수 + 증가값
```

- ✓ while 문에서는 무한 루프를 사용할 수 있고, 'while 조건식:'에 들어가는 조건을 True로 지정
- ✓ 반복문 안에서 break 문을 만나면 반복문을 탈출하고, continue 문을 만나면 블록의 남은 부분을 건너뛰어 반복문의 처음으로 되돌아감

111001100110

# 감사합니다.

우창우

Dr.woo@chungbuk.ac.kr

## 팀프로젝트 현황: (제출처) Dr.woo@chungbuk.ac.kr, (기한) 3.23(토) 까지

조 번호	프로젝트 주제	조장	성명	성명	성명
1조			고태경	조형준	김다민
2조			김민혁	전영우	김정민
3조		홍성진	김태영	-	정세연
4조			우태현	이규민	전수혁
5조			박상인	서범교	배정민
6조			김건우	박조현	오다영
7조		박주현	권정욱	-	정현준
8조			김규현	김준후	조윤정
9조			신종환	신승우	한강민
10조			육광민	송설희	김준홍
11조		한준영	고태영	-	이관학
12조			전준석	김민경	윤시훈
13조		김준호	-	황지연	이용희
14조			이한결	신혜원	배수환
15조		박성범	-	이태정	김민석

# 실습&과제 : (제출처) Dr.woo@chungbuk.ac.kr, (기한) 3.23(토) 까지

## 1. 종합계산기

## 2. 구구단 출력

Untitled-1.py X

C: > Users > rryan > Downloads > Untitled-1.py > ...

```
1  ## 변수 선언 부분 ##
2  select, answer, numStr, num1, num2 = 0, 0, "", 0, 0
3
4  ## 메인 코드 부분 ##
5  select = int(input("1. 입력한 수식 계산  2. 두 수 사이의"))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
1. 입력한 수식 계산  2. 두 수 사이의 합계 : 1
*** 수식을 입력하세요 : 3*4/2-5
3*4/2-5 결과는  1.0입니다.
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
1. 입력한 수식 계산  2. 두 수 사이의 합계 : 2
*** 첫 번째 숫자를 입력하세요 : 1
*** 두 번째 숫자를 입력하세요 : 10
1+...+10는 55입니다.
PS C:\Users\rryan> |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python,
# 2단 ## 3단 ## 4단 ## 5단 ## 6단 ## 7단 ## 8
2 X 1 = 2  3 X 1 = 3  4 X 1 = 4  5 X 1 = 5  6 X 1
2 X 2 = 4  3 X 2 = 6  4 X 2 = 8  5 X 2 = 10  6 X 2
2 X 3 = 6  3 X 3 = 9  4 X 3 = 12  5 X 3 = 15  6 X 3
2 X 4 = 8  3 X 4 = 12  4 X 4 = 16  5 X 4 = 20  6 X 4
2 X 5 = 10  3 X 5 = 15  4 X 5 = 20  5 X 5 = 25  6 X 5
2 X 6 = 12  3 X 6 = 18  4 X 6 = 24  5 X 6 = 30  6 X 6
2 X 7 = 14  3 X 7 = 21  4 X 7 = 28  5 X 7 = 35  6 X 7
2 X 8 = 16  3 X 8 = 24  4 X 8 = 32  5 X 8 = 40  6 X 8
2 X 9 = 18  3 X 9 = 27  4 X 9 = 36  5 X 9 = 45  6 X 9
PS C:\Users\rryan>
```