

LAB 02

2020-01 소프트웨어의 이해 01분반 / 조교 이경민

LAB 02

- 수식, 문자열 다뤄보기
- 리스트 생성 및 추가/제거
- 예제 실습

append / del

- 마일, 킬로미터 변환
- 은행 계좌 프로그램
- 시간-분-초 계산

→ 정수 나머지 연산

- Math 모듈
- 삼각형 면적 계산

- 실습 과제

변수 (저장공간) 이름 설정 규칙

- **변수** : 저장공간
- 변수에 값을 저장, 즉 데이터를 넣기 위해서 '='를 사용
- 변수 이름 설정 규칙
 - 한글 사용 (X)
 - **문자와 숫자**로 이루어질 수 있으며, 반드시 **문자로 시작!**
 - 알파벳, _문자 (O)
 - %, \$, &, # 등 특수문자 (X)
 - 띄어쓰기 (X)

변수 (저장 공간) 이름 설정 규칙

- 변수 이름 설정 Tip!
 - 변수에 저장된 데이터의 의미를 잘 나타내는 이름
 - ex) 이름 → name 전체 합 → sum 점수 → score
결과 → result 첫 번째 숫자 → num1
 - ex) t1 (△) 의미를 알기 어렵다
 - 가능한 명사의 형태
 - ex) 전체 합 → sum (O) plus(△)

~_~
or appleAndGrape
 ~~과~~
 과합자

수식, 문자열 다뤄보기

- 수식

- 정수와 실수 연산으로 나뉘는 것을 기억!

- 덧셈 + 뺄셈 - 곱셈 *

- 나눗셈 // (정수) / (실수)

- ex) $13 // 4 = 3$

$13 / 4 = 13.0 / 4.0 = 13.0 / 4 = 13 / 4.0 = 3.25$

- 나머지 %

- ex) $13 \% 4 = 1$

$13.0 \% 4.0 = 1.0$

$13.1 \% 4.1 = 0.8000000$

- 거듭제곱 **

- ex) $2 ** 5 = 32$

$2.5 ** 5 = 97.65625$

```
>>> 12 + 11
23
>>> 12 - 11
1
>>> 12 * 12
144
>>> 12 // 5
2
>>> 12 / 5
2.4
>>> 12.0 / 5.0
2.4
>>> 12 % 5
2
>>> 12.0 % 5.0
2.0
>>> 2 ** 10
1024
>>> |
```

수식, 문자열 다뤄보기

- 문자열 (string)

- 작은 따옴표 (') 또는 큰 따옴표 (") 로 표시
- 한글 문자열도 가능
- 특수문자 표현 : **백슬래시(w)** 사용

- Example

- Today's date 출력해보기
 - 1. "Today's date" : 따옴표 종류가 다르면 서로 구분됨
 - 2. 'Todayw's date' : 백슬래시 이용
 - cf) 'Today's date' (오류) : 'Today'를 하나의 문자열 + 미완성 문자열로 인식

수식, 문자열 다뤄 보기

- 문자열 연산
 - + : 두 문자열을 합쳐서 새로운 문자열 하나로 만들 수 있음
 - * : 문자열을 반복하여 새로운 문자열을 생성

```
>>> 'Merry Christmas' + 'and Happy New Year!'
'Merry Christmasand Happy New Year!'
>>> print('Merry Christmas' + 'and Happy New Year!')
Merry Christmasand Happy New Year!
>>> print('Merry Christmas and Happy New Year!')
Merry Christmas and Happy New Year!
>>> print('Merry Christmas','and Happy New Year!')
Merry Christmas and Happy New Year!
>>> "Merry Christmas" * 2
'Merry ChristmasMerry Christmas'
```

수식, 문자열 다뤄보기

- 구구단 프로그램 (multiplication.py)
 - 수식 연산을 이용하여 구구단을 작성하기

구구단 프로그램

→ 변수나 연산자 아 input 사용

```
print("7 x 1 =", 7*1)
print("7 x 2 =", 7*2)
print("7 x 3 =", 7*3)
print("7 x 4 =", 7*4)
print("7 x 5 =", 7*5)
print("7 x 6 =", 7*6)
print("7 x 7 =", 7*7)
print("7 x 8 =", 7*8)
print("7 x 9 =", 7*9)
```

```
===== RESTART: C:/
=
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
>>>
```

dan * 1 = dan

수식, 문자열 다뤄보기

- 1~10까지의 합 구하기

```
>>> print(1+2+3+4+5+6+7+8+9+10)
55
>>> sum = 1+2+3+4+5+6+7+8+9+10
>>> print(sum)
55
>>> |
```

리스트 (List)

- 값들의 리스트, 목록
- 연산
 - 1. 접합 ('+') : 두 리스트를 붙여서 새로운 리스트를 생성
 - 2. 곱하기 ('*') : 대상 리스트를 반복하여 새로운 리스트를 만듦

```
>>> America = ['New York','Chicago','LA']
>>> Korea = ['Seoul', 'Busan']
>>> print(America + Korea)
['New York', 'Chicago', 'LA', 'Seoul', 'Busan']
>>> print(Korea * 2)
['Seoul', 'Busan', 'Seoul', 'Busan']
>>> |
```

리스트 (List)

인덱스

- 리스트 내에 있는 특정 값을 지정하여 사용 (0부터 시작)
- 콜론 (:)
 - 인덱스의 범위를 지정하여 여러 개의 값들을 나타냄
 - 끝의 값은 포함 되지 않음 *이상이전의거*

```
>>> Korea = ['Seoul', 'Busan', 'Ulsan', 'Daegu']
>>> print(Korea[0], ",", Korea[3])
Seoul , Daegu
>>> print(Korea[2:3])
['Ulsan']
>>> print(Korea[2:4])
['Ulsan', 'Daegu']
>>> print(Korea[1:])
['Busan', 'Ulsan', 'Daegu']
>>> |
```

리스트 (List)

- 새로운 원소 추가
 - append 함수 이용
- 특정 원소 삭제
 - del 함수 이용

```
>>> Korea = ['Seoul', 'Busan', 'Ulsan', 'Daegu']
>>> Korea.append('Incheon')
>>> print(Korea)
['Seoul', 'Busan', 'Ulsan', 'Daegu', 'Incheon']
>>> del Korea[0]
>>> print(Korea)
['Busan', 'Ulsan', 'Daegu', 'Incheon']
>>>
```

예제 1) 마일, 킬로미터 변환

- 마일을 킬로미터로 변환하는 프로그램
 - `float()` : 입력 받은 값을 실수형으로 변환

pound

```
mile.py - C:/SW_1911123/mile.py (3.7.2)
File Edit Format Run Options Window Help
mile = float(input("마일 입력: "))
kilo = 1.6 * mile
print(mile, "마일은", kilo, "킬로미터입니다.")
```

```
===== RESTART:
마일 입력: 15
15.0 마일은 24.0 킬로미터입니다.
>>> |
```

예제 2) 은행 계좌 프로그램

- 은행 계좌 입금 프로그램
 - 1. 계좌 소유자 이름을 입력 받아 은행 계좌를 개설
 - 2. 입금액을 입력 받음
 - 3. 현재 잔액에 입금액을 추가
 - 4. 현재 잔액을 출력

523.

bank.py - C:/SW_1911123/bank.py (3.7.2)

File Edit Format Run Options Window Help

```
balance = 0
name = input("계좌 소유자 이름: ")
amount = int(input("입금액 입력: "))
balance = balance + amount
print(name, "님의 현재 잔액: ", balance)
```

===== RESTART

계좌 소유자 이름: 이경민

입금액 입력: 54000

이경민 님의 현재 잔액: 54000

>>> |

예제 2) 은행 계좌 프로그램

- 은행 계좌 출금 프로그램

bank2.py - C:/SW_1911123/bank2.py (3.7.2)

File Edit Format Run Options Window Help

```
balance = 0
name = input("계좌 소유자 이름: ")
amount = int(input("입금액 입력: "))
balance = balance + amount
print(name, "님의 현재 잔액: ", balance)
amount = int(input("출금할 금액 입력: "))
balance = balance - amount
print(name, "님의 현재 잔액: ", balance)
```

수정하기...

===== RESTART:

계좌 소유자 이름: 이경민

입금액 입력: 55000

이경민 님의 현재 잔액: 55000

출금할 금액 입력: 3000

이경민 님의 현재 잔액: 52000

>>>

예제 3) 시간-분-초 계산

- 초를 입력 받아 시간, 분, 초로 계산하는 프로그램

```
sec = int(input("시간을 초 단위로 입력하세요 : "))
minute = sec // 60
second = sec % 60
hour = minute // 60
minute = minute % 60
print(sec, "초는", hour, "시간", minute, "분", second, "초입니다")
```

```
시간을 초 단위로 입력하세요 : 6543
6543 초는 1 시간 49 분 3 초입니다
```

1시간 49분 3초

math 모듈

- **import math** : math 모듈 불러오기
- [주요 수학 내장 함수 및 모듈 제공 함수]

abs(x)	x의 절대값 반환
max(x1, x2)	가장 큰 값 반환
min(x1, x2)	가장 작은 값 반환
pow(x, y)	x의 y승 반환
round(x)	x의 반올림 값 반환
math.floor(x)	x의 내림 값 반환
math.ceil(x)	x의 올림 값 반환

```
>>> import math
>>> abs(-50)
50
>>> max(30,20)
30
>>> min(15,60)
15
>>> pow(2,10)
1024
>>> round(36.7)
37
>>> math.floor(28.9)
28
>>> math.ceil(67.1)
68
>>> |
```

예제 4) 삼각형 면적 계산

- $S = (a+b+c)/2$
- 삼각형 면적 = $\sqrt{s * (s - a) * (s - b) * (s - c)}$
 - 1. 세 변의 길이 입력 받기
 - 2. s 값 계산
 - 3. 면적 계산 및 출력

```
===== RESTART:
===
삼각형 세 변의 길이를 입력하세요
a : 3
b : 4
c : 5
삼각형 면적: 6.0
>>> |
```

```
triangle.py - C:/SW_1911123/triangle.py (3.7.2)
File Edit Format Run Options Window Help

import math

print("삼각형 세 변의 길이를 입력하세요")
a = int(input("a : "))
b = int(input("b : "))
c = int(input("c : "))
s = (a+b+c) / 2
area = math.sqrt(s * (s-a) * (s-b) * (s-c))

print("삼각형 면적:", area)
```

삼각형의 세 변을 입력
후 면적을 구함

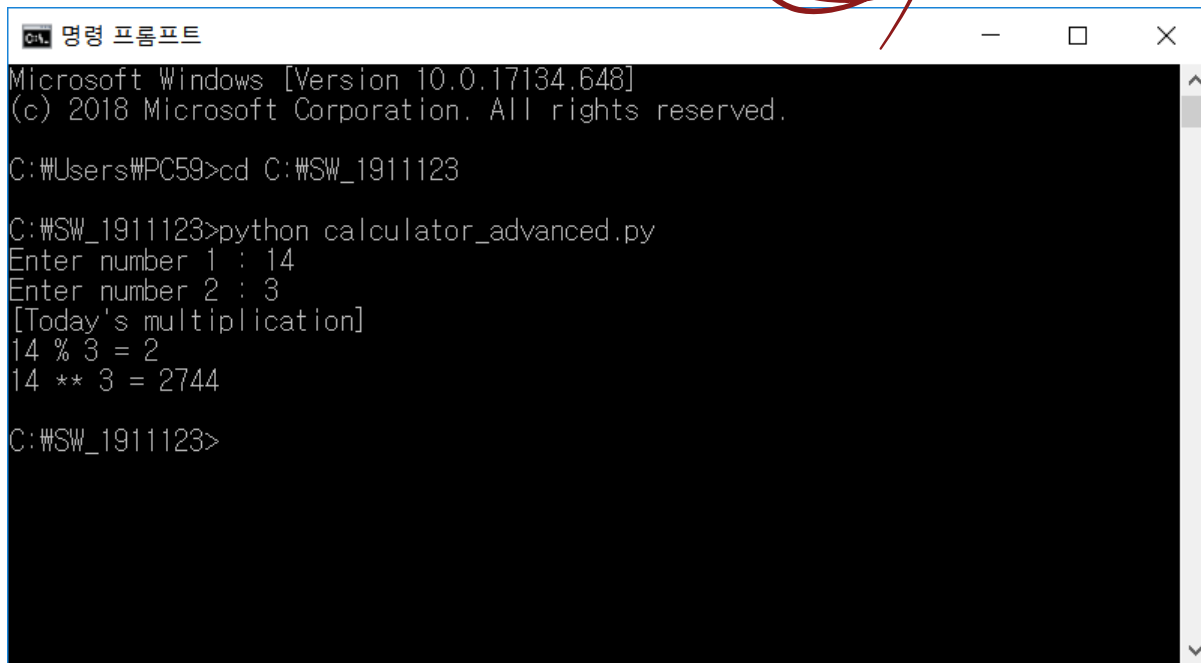
LAB 02 실습 과제 1 (calculator_basic.py)

- 숫자 2개를 입력 받아 덧셈, 뺄셈, 곱셈, 나눗셈(정수, 실수)를 수행하여 계산 결과를 출력하는 프로그램을 작성하여 실행

```
===== RESTART: C:/SW_1911123/calculator_basic.py =====  
Enter number 1 : 12  
Enter number 2 : 5  
[Today's multiplication]  
12 + 5 = 17  
12 - 5 = 7  
12 * 5 = 60  
12 // 5 = 2  
12 / 5 = 2.4  
>>> |
```

LAB 02 실습 과제 2 (calculator_advanced.py)

- 숫자 2개를 입력 받아 나머지 계산과 거듭제곱 계산을 수행하여
계산 결과를 출력하는 프로그램을 작성하여 cmd창에서 실행하여 제출



```
명령 프롬프트
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\PC59>cd C:\SW_1911123

C:\SW_1911123>python calculator_advanced.py
Enter number 1 : 14
Enter number 2 : 3
[Today's multiplication]
14 % 3 = 2
14 ** 3 = 2744

C:\SW_1911123>
```

LAB 02 실습 과제 3 (calculate_rate.py)

- S통신사 표준 요금제는 다음과 같이 월 이용 요금을 계산한다.
이 요금제의 월 이용 요금을 계산하는 프로그램을 작성하시오.
- 기본 요금: 12,100원
- 음성 통화: 1.98원/초
- 데이터 사용: 55원/MB
- 가입자 이름, 음성 통화 시간(초), 데이터 사용량(MB)를 입력 받아,
월 이용 요금을 계산하여 출력

```
===== RESTART: C:/SW_19
=
이름을 입력하세요 : 이경민
음성 통화 시간(초)을 입력하세요: 3600
데이터 사용량(MB)을 입력하세요: 1024
이경민 님의 이용 요금입니다: 75548.0
>>> |
```


LAB 02 실습 과제 4 (circle.py)

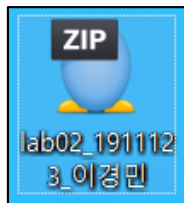
- 사용자로부터 반지름을 입력받아 원의 둘레와 면적을 구하는 프로그램을 작성하시오.
- 원의 둘레 공식 : $2\pi r$, 원의 면적 공식 : πr^2
- 원주율 값 : `math.pi`

15.7

```
===== RESTART: C:/SW_
===
원의 반지름을 입력하세요: 2.5
원의 면적: 19.634954084936208
원의 둘레: 15.707963267948966
>>> |
```

소스파일(.py)과 과제보고서(.docx) 합친 압축파일 제출

이름	수정한 날짜	유형
 calculate_rate	2020-03-23 오후 5:37	Python File
 calculator_advanced	2020-03-23 오후 5:37	Python File
 calculator_basic	2020-03-23 오후 5:37	Python File
 circle	2020-03-23 오후 5:37	Python File
 Lab02_학번_이름	2020-03-23 오후 5:38	Microsoft Word 문서



과제 채점 기준·기한

- 과제 제출 기한

- 실습 다음주 화요일 오후 11시까지
- 4월 14일 화요일 오후 11시까지 제출

- 제출 장소

- 스노우보드 해당 주차 과제 제출 페이지에 업로드

- 추가 제출

- 제출기한 이후 24시간 이내 업로드 : 만점에서 20% 감점
- 그 이후는 받지 않음

- 표절 X

과제 제출 형식 & 질문 메일

- **제출물** : 소스파일(.py)과 과제보고서(.docx) 합친 **압축파일 제출**
 - **소스파일 이름** : 매 실습 과제 마다 ppt에 제시 예정
 - ex) test.py
 - **과제 보고서 양식** : 스노우보드에서 다운로드
- **(소스파일 + 과제보고서) 압축 파일 이름 : Lab02_본인학번_이름**
 - ex) Lab02_1911123_이경민.zip
- **이메일** : newkml22@gmail.com
- **질문 시 주의사항**
 - **과목, 분반, 이름, 학번** 알려주세요.
 - 몇 번 과제에서, 어떤 부분이 막혔는지, 어떤 과정인지 **설명**과 함께 보내주세요.
 - 답장까지 시간이 걸릴 수도 있으니 제출 과제 질문은 미리 해주세요!