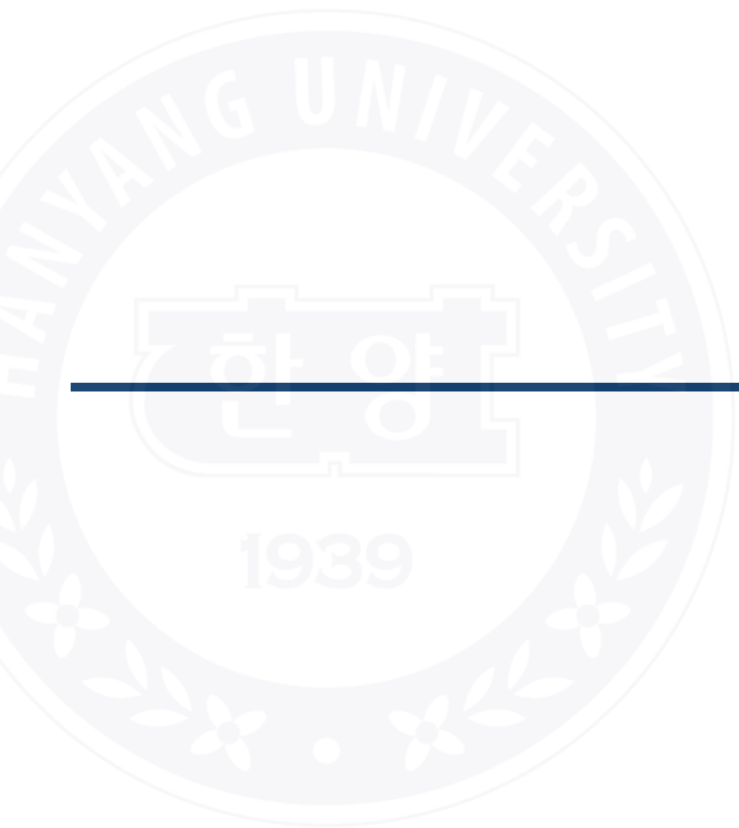


# Lab 2

2021.03.13

Python Programming



# Context

---

- 조건문
- 반복문
- Built-in 함수

# 조건문

구문	의미
if~ 구문	만약 조건을 만족하면 ~ 작업을 수행하라.
if~ else~ 구문	만약 조건을 만족하면 ~ 작업을 수행하고, 그렇지 않으면 ~ 작업을 수행하라.
if~ elif~ else~ 구문	만약 조건 1을 만족하면 작업 1을 수행하고, 조건 2를 만족하면 작업 2를 수행하고, 조건 3을 만족하면 작업 3을 수행하고, ..., 그렇지 않으면 작업 n을 수행하라.

# 조건문

---

연산자	종류
비교 연산자	>, <, ==, !=, <=, >=
논리 연산자	and, or, not

# 조건문

연산자	종류
비교 연산자	>, <, ==, !=, <=, >=
논리 연산자	and, or, not

비교 연산자	의미
a == b	a와 b는 같다
a != b	a와 b는 같지 않다
a > b	a는 b보다 크다
a >= b	a는 b보다 크거나 같다
a < b	a는 b보다 작다
a <= b	a는 b보다 작거나 같다

논리 연산자	의미
조건1 and 조건2	논리곱(and) 조건1과 조건2가 둘 다 참인 경우에만 참이 된다
조건1 or 조건2	논리합(or) 조건1 또는 조건2 중 하나만 참이어도 참이 된다
not 조건	논리부정(not) 조건이 참이면 거짓, 거짓이면 참으로 해서 논리 값을 반대로 변경한다

# 연습 문제 1 - 조건문

---

- 키와 몸무게를 입력 받아 다이어트의 필요성을 판정하는 프로그램을 작성하시오. 판단 기준은 키에서 100을 뺀 값에 0.9를 곱한 값보다 몸무게가 크면 다이어트가 필요하다고 판정한다.
- 입력 조건 : 키와 몸무게를 나타내는 정수 2개를 입력받는다.
- 출력 조건 : 키와 몸무게를 출력하고 다이어트가 필요한지에 따라 메시지를 출력한다.

# 연습 문제 1 - 조건문

---

- 예제 입력 1
  - 175 65
- 예제 출력 1
  - 키 : 175  
몸무게 : 65  
표준 또는 마른 체형입니다!
- 예제 입력 2
  - 175 80
- 예제 출력 2
  - 키 : 175  
몸무게 : 80  
건강을 위해 다이어트가 필요합니다!

# 연습 문제 2 - 조건문

- 구매 금액에 따른 할인율을 적용하여 지불 금액을 계산하는 프로그램을 작성하시오.

금액에 따른 할인율은

10,000~50,000원 미만 : 5%

50,000~300,000원 미만 : 7.5%

300,000원 이상 : 10%

10,000 미만 : 0%

라고 가정한다.

- 입력 조건 : 구매 금액에 해당하는 정수를 입력 받는다.
- 출력조건 : 구매금액, 할인율, 할인금액, 지불금액을 차례대로 출력한다.



# 연습 문제 2 - 조건문

---

- 예제 입력 1
  - 150000
- 예제 출력 1
  - 구매금액 : 150000  
할인율 : 7.5  
할인금액 : 11250  
지불금액 : 138750
- 예제 입력 2
  - 350000
- 예제 출력 2
  - 구매금액 : 350000  
할인율 : 10  
할인금액 : 35000  
지불금액 : 315000

# 연습 문제 3 - 조건문

---

- 물의 온도와 단위(섭씨 or 화씨)를 입력 받아 물의 상태를 판별하는 프로그램을 작성하시오. 섭씨는 1, 화씨는 2로 구분하고, 화씨로 입력되었을 때는 섭씨로 변환한 다음 물의 상태를 판별하여 처리하도록 한다.  
(섭씨 = (화씨 - 32) \* 5 / 9)
- 입력 조건 : 온도의 단위와 온도에 해당하는 정수 2개를 입력 받는다.
- 출력조건 : 물의 섭씨온도와, 상태를 출력한다.

# 연습 문제 3 - 조건문

---

- 예제 입력 1

- 단위를 입력해 주세요(1:섭씨, 2:화씨) : 1  
온도를 입력해주세요 : 30

- 예제 출력 1

- 물의 섭씨 온도 : 30.0도, 상태 : 액체

- 예제 입력 2

- 단위를 입력해 주세요(1:섭씨, 2:화씨) : 2  
온도를 입력해주세요 : 25

- 예제 출력 2

- 물의 섭씨 온도 : -3.9도, 상태 : 고체

# 반복문(for)

반복문을 사용하지  
않은 경우

```
print("안녕하세요!")  
print("안녕하세요!")  
print("안녕하세요!")  
print("안녕하세요!")  
print("안녕하세요!")
```

반복문을 사용한  
경우

```
for x in range(5) :  
    print("안녕하세요!")
```

# x가 0~4의 값을 가지고 5번 반복

# 반복문(for)

- range() 함수

- range(a, b, t) : a부터 b미만까지 t 간격으로 증가

```
for i in range(10) :                # 0 ~ 9
    print(i, end = " ")
print()                             # 줄 바꿈

for i in range(1, 11) :             # 1 ~ 10
    print(i, end = " ")
print()                             # 줄 바꿈

for i in range(1, 10, 2) :          # 1, 3, 5, 7, 9
    print(i, end = " ")
print()                             # 줄 바꿈

for i in range(20, 0, -2) :         # 20, 18, 16, ..., 2
    print(i, end = " ")
print()
```

# 연습 문제 1 - 반복문(for)

---

- 영어 문장을 입력 받아 세로로 한 글자씩 출력하는 프로그램을 작성하시오.
- 입력 조건 : 영어로 이루어진 문장(띄어쓰기 포함 가능)을 입력받는다.
- 출력조건 : 입력 받은 문장을 한 글자씩 줄바꿈하여 출력한다.

# 연습 문제 1 - 반복문(for)

---

- 예제 입력 1

- Python

- 예제 출력 1

- P  
y  
t  
h  
o  
n

- 예제 입력 2

- Hey, Hello

- 예제 출력 2

- H  
e  
y  
,  
  
H  
e  
l  
l  
o

# 연습 문제 2 - 반복문(for)

- 다음과 같이 for문을 이용하여 센티미터(1~100cm, 1씩 증가)를 밀리미터(mm), 미터(m), 인치(inch)로 환산하는 표를 만드는 프로그램을 작성하시오.  
(1cm = 0.3937inch)
- 출력조건 : 다음과 같이 출력한다.

cm	mm	m	inch
1	10	0.01	0.4
2	20	0.02	0.8
3	30	0.03	1.2
4	40	0.04	1.6
...			
99	990	0.99	39.0
100	1000	1.00	39.4



# 연습 문제 3 - 반복문(for)

---

- 첫째 줄에는 별 1개, 둘째 줄에는 별 2개, N번째 줄에는 별 N개를 찍는 프로그램을 작성하시오.
- 입력 조건 : 줄 수를 나타내는 정수를 입력 받는다.
- 출력조건 : 첫째 줄부터 N번째 줄까지 차례대로 별을 출력한다.

# 연습 문제 3 - 반복문(for)

- 예제 입력 1

- 5

- 예제 출력 1

- \*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

- 예제 입력 2

- 7

- 예제 출력 2

- \*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

# 연습 문제 4 - 반복문(for)

---

- 첫째 줄에는 별 N개, 둘째 줄에는 별 N-1개, N번째 줄에는 별 1개를 찍는 프로그램을 작성하시오.
- 입력 조건 : 줄 수를 나타내는 정수를 입력 받는다.
- 출력조건 : 첫째 줄부터 N번째 줄까지 차례대로 별을 출력한다.

# 연습 문제 4 - 반복문(for)

- 예제 입력 1

- 5

- 예제 출력 1

- \*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*\*  
\*

- 예제 입력 2

- 7

- 예제 출력 2

- \*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*

# 연습 문제 5 - 반복문(for)

---

- For문을 이용하여 키보드로 입력된 숫자에서 홀수의 개수를 세는 프로그램을 작성하시오.
- 입력 조건 : 정수 하나를 입력 받는다.
- 출력조건 : 입력 받은 숫자에 존재하는 홀수의 개수를 출력한다.

# 연습 문제 5 – 반복문(for)

---

- 예제 입력 1
  - 477569040
- 예제 출력 1
  - 4
- 예제 입력 2
  - 11100101
- 예제 출력 2
  - 5

# 연습 문제 6 - 반복문(for)

---

- 두 수 A와 B를 입력 받아, 구구단 A단부터 B단까지 출력하는 프로그램을 작성 하시오.
- 입력 조건 : 정수 A와 B를 입력 받는다. ( $A \leq B$ )
- 출력조건 : 구구단 A단부터 B단까지 차례대로 출력한다.

# 연습 문제 6 - 반복문(for)

- 예제 입력 1

- 2 2

- 예제 출력 1

- $2 \times 1 = 2$   
 $2 \times 2 = 4$   
 $2 \times 3 = 6$   
 $2 \times 4 = 8$   
 $2 \times 5 = 10$   
 $2 \times 6 = 12$   
 $2 \times 7 = 14$   
 $2 \times 8 = 16$   
 $2 \times 9 = 18$   
=====

- 예제 입력 2

- 4 5

- 예제 출력 2

- $4 \times 1 = 4$   
 $4 \times 2 = 8$   
 $4 \times 3 = 12$   
 $4 \times 4 = 16$   
 $4 \times 5 = 20$   
 $4 \times 6 = 24$   
 $4 \times 7 = 28$   
 $4 \times 8 = 32$   
 $4 \times 9 = 36$   
=====  
 $5 \times 1 = 5$   
 $5 \times 2 = 10$   
 $5 \times 3 = 15$   
 $5 \times 4 = 20$   
 $5 \times 5 = 25$   
 $5 \times 6 = 30$   
 $5 \times 7 = 35$   
 $5 \times 8 = 40$   
 $5 \times 9 = 45$   
=====



# 연습 문제 7 - 반복문(for)

---

- 첫째 줄에는 숫자  $N \sim 1$ , 둘째 줄에는 숫자  $N-1 \sim 1$ ,  $N$ 번째 줄에는 숫자 1를 출력하는 프로그램을 작성하시오.
- 입력 조건 : 줄 수를 나타내는 정수를 입력 받는다.
- 출력조건 : 첫째 줄부터  $N$ 번째 줄까지 차례대로 숫자를 출력한다.

# 연습 문제 7 - 반복문(for)

- 예제 입력 1

- 5

- 예제 출력 1

- 5 4 3 2 1  
4 3 2 1  
3 2 1  
2 1  
1

- 예제 입력 2

- 7

- 예제 출력 2

- 7 6 5 4 3 2 1  
6 5 4 3 2 1  
5 4 3 2 1  
4 3 2 1  
3 2 1  
2 1  
1

# 연습 문제 1 - 반복문(while)

- 다음과 같이 while문을 이용하여 1 ~ 100 정수 중 홀수의 누적 합계를 구하는 프로그램을 작성하시오.
- 출력조건 : 다음과 같이 출력한다.

1	4	9	16	25	36	49	64	81	100
121	144	169	196	225	256	289	324	361	400
441	484	529	576	625	676	729	784	841	900
961	1024	1089	1156	1225	1296	1369	1444	1521	1600
1681	1764	1849	1936	2025	2116	2209	2304	2401	2500

## 연습 문제 2 - 반복문(while)

---

- while문을 이용하여 키보드로 입력된 숫자에서 홀수의 개수를 세는 프로그램을 작성하시오.
- 입력 조건 : 정수 하나를 입력 받는다.
- 출력조건 : 입력 받은 숫자에 존재하는 홀수의 개수를 출력한다.

# 연습 문제 2 – 반복문(while)

---

- 예제 입력 1
  - 477569040
- 예제 출력 1
  - 4
- 예제 입력 2
  - 11100101
- 예제 출력 2
  - 5

# Built-In 함수

- Python 자료형

	설명	모습
숫자형	정수, 실수 등의 숫자를 다루는 자료형	0 or 1.25 or -123
문자열	문자열을 다루는 자료형	'alghost'
리스트	다른 자료형의 모음을 다루는 자료형	[1, 'alghost', 123]
튜플	리스트와 같지만 수정이 불가능한 자료형	(1, 'alghost', 123)
딕셔너리	키와 값으로 이루어진 자료형	{'name' : 'alghost'}
...	...	...

# Built-In 함수(Common)

---

- `sum(리스트, 튜플, 딕셔너리)`
  - Iterable한 자료형 안 원소의 합을 계산
- `max(정수, 리스트, 튜플, 딕셔너리)`
  - Iterable한 자료형 안 원소들 중 최댓값 계산
- `min(정수, 리스트, 튜플, 딕셔너리)`
  - Iterable한 자료형 안 원소들 중 최솟값 계산
- `round(정수)`
  - 정수를 반올림하여 계산(사사오입 원칙)
- `type(변수)`
  - 변수의 type을 return

# Built-In 함수(List)

함수명	설명	사용방법	result 값
append	요소를 뒤에 추가	<pre>result = [1,2,3] result.append(4)</pre>	[1,2,3,4]
sort	요소들을 정렬	<pre>result = ['a', 'c', 'b'] result.sort()</pre>	['a','b','c']
reverse	요소들을 뒤집음	<pre>result = [1,10,100] result.reverse()</pre>	[100,10,1]
index	입력값의 위치를 반환 (첫번째로 찾은 위치)	<pre>a = [10,11,11,100] result = a.index(11)</pre>	1



# Built-In 함수(List)

함수명	설명	사용방법	result 값
insert	특정 위치에 요소를 추가	<pre>result = [100,192,101] result.insert(1, 'a')</pre>	[100,'a',192,101]
remove	입력값을 삭제 (첫번째로 찾은 위치)	<pre>result = [10,11,100,11] result.remove(11)</pre>	[10,100,11]
pop	마지막 요소를 꺼내고 삭제	<pre>a = [10,101,102,103] result = a.pop()</pre>	103 (a: [10,101,102])
count	입력값의 갯수	<pre>a = [10,10,101,102,10,'a'] result = a.count(10)</pre>	3

# Built-In 함수(Dictionary)

함수명	설명	사용방법	result 값
keys	딕셔너리의 key들을 반환	<code>a = { 'a' : 123, 'b' : 456 }</code> <code>result = a.keys()</code>	<code>['a', 'b']</code>
values	딕셔너리의 value들을 반환	<code>a = { 'a' : 123, 'b' : 456 }</code> <code>result = a.values()</code>	<code>[123, 456]</code>
items	key,value 쌍을 반환	<code>a = { 'a' : 123, 'b' : 456 }</code> <code>result = a.items()</code>	<code>[('a', 123), ('b', 456)]</code>
get	key에 대한 값을 반환 (값이 없는 경우 기본값 지정 이 가능)	<code>a = { 'a' : 123, 'b' : 456 }</code> <code>result = a.get('c', 789)</code>	<code>789</code>

# 연습 문제 1 – Built-In 함수

---

- 임의의 수열을 입력 받아, 상위 K개의 수의 평균을 구하는 프로그램을 작성하시오.
- 입력 조건 : 첫번째 줄에 임의의 수열을 입력 받고, 두번째 줄에 정수 K를 입력 받는다.
- 출력조건 : 수열 중 상위 K개의 수의 평균을 출력한다.

# 연습 문제 1 – Built-In 함수

---

- 예제 입력 1

- 3 4 1 5 7 8 9 8  
3

- 예제 출력 1

- 8.333

- 예제 입력 2

- 5 2 1 5 7 8 12 3 4 18

- 예제 출력 2

- 11.25