Lab4. Selection and Repetition

CSED101 LAB

들어가기 전 (1)

- 주사위 (1~6)를 던져서 나온 면의 숫자가 홀인지 짝인지 출력하는 프로그램을 작성하시오. (random 모듈 사용)

```
주사위  숫자: 5 (홀)
```

들어가기 전 (2)

- 입력한 숫자의 구구단을 출력하는 프로그램 작성하기
- 실행 예시 (밑줄은 사용자 입력을 말함)

```
출력하고 싶은 단을 입력하세요: 7
7 * 1 =  7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
```

조건문

bool 자료형

- bool 자료형
  - bool은 True와 False의 두 가지 값을 나타내는 자료

```
>>> result = 3 > 2
>>> print(result)
True
>>> type(result)
<class 'bool'>
```

- 조건문이 다음과 같을 때 거짓(False)으로 평가
  - False, None, 숫자 0 예) 0, 0.0
  - 비어 있는 순서열 예) '', (), []
  - 비어 있는 딕셔너리 : 예) {}

- 어떤 객체가 참인지 거짓인지 알고 싶을 때 bool() 함수 이용

```
>>> bool(0.0)
False
>>> bool([1,2,3])
True
```

관계 연산자와 논리 연산자

- 연산의 결과는 항상 참(True) 또는 거짓(False)
- 관계 연산자
  - <, >, ==, <=, >=, !=
    - '=='와 '='는 서로 다른 연산
- 논리 연산자
  - not, and, or

```
>>> a = 99
>>> (a > 100) and (a < 200)
False
>>> (a > 100) or (a < 200)
True
>>> not(a == 100)
True
```

if 문

- 조건문의 결과가 참(True)이면 A를 수행

```
if 조건문 :
    수행할 문장1
    수행할 문장2
      …
    수행할 문장n
```

A

- 블록(Block)
  - 여러 코드가 이루는 일정한 구역
  - 파이썬의 경우, 들여쓰기로 구역을 나눔
  - 들여쓰기는 스페이스(Space)나 탭(Tab) 둘 다 사용 가능
    - 스페이스 4 칸 사용 권장

if-else

- 조건의 결과가 참(True)이면 if문 바로 다음의 문장(if 블록)들을 수행하고, 거짓(False)이면 else문 다음의 문장(else 블록)들을 수행

```
if 조건문 :
    수행할 문장1
    수행할 문장2
        …
else :
    수행할 문장A
    수행할 문장B
        …
```

```
a = 200

if a < 100 :
    print("참")
    print("100보다 작다")
else :
    print("거짓")
    print("100보다 크거나 같다")

print("프로그램 끝")
```

if-elif-else

- 다중 조건 판단

```
if 조건A :
    수행할 문장 A
elif 조건B :
    수행할 문장 B
…

else :
    수행할 문장 C
```

```
num = int(input("정수를 입력하시오: "))

if num > 0 :
    print("양수입니다.")
elif num == 0 :
    print("0입니다.")
else :
    print("음수입니다.")
```

중첩 if

```
if 조건A :
    if 조건1 :
        수행할 문장 A1
    else :
        수행할 문장 A2
else :
    수행할 문장 B
```

if 문 응용

- 리스트와 함께 사용

if 항목 in 리스트 :

리스트 안에 항목이 있으면 True를 반환

```
>>> 1 in [1, 2, 3]
True
```

if 항목 not in 리스트 :

리스트 안에 항목이 없으면 True를 반환

```
>>> 1 not in [1, 2, 3]
False
```

실습 1

- 세 개의 정수를 입력 받아, 가장 큰 수를 출력하는 프로그램을 작성하시오.

  단, max() 함수, 정렬함수 사용할 수 없음

```
세 개의 수를 입력하시오: 30 22 9
가장 큰 수는 30입니다.
```

```
세 개의 수를 입력하시오: 3 9 9
가장 큰 수는 9입니다.
```

실습 2

- 학생수준평가 시험에서 영어 점수와 수학 점수가 합해서 110점 이상이면 합격이다. 단, 각 점수가 40점 미만이면 불합격이다. 영어(eng), 수학(math)점수를 입력 받아 합격여부를 출력하는 프로그램을 작성하시오.

```
영어 점수 입력: 80
수학 점수 입력: 20
불합격: 총합 점수 부족
```

```
영어 점수 입력: 90
수학 점수 입력: 30
불합격: 수학 점수 부족
```

```
영어 점수 입력: 70
수학 점수 입력: 80
합격
```

```
영어 점수 입력: 35
수학 점수 입력: 30
불합격: 총합 점수 부족
```

```
영어 점수 입력: 35
수학 점수 입력: 95
불합격: 영어 점수 부족
```

Problem 1

- 프로그래밍 과목의 중간고사와 기말고사 점수를 입력 받아 평균과 학점을 구하는 프로그램을 작성하시오.

```
- 평균이 90이상: A
- 80이상~90미만: B
- 70이상~80미만: C
- 60이상~70미만: D
- 60미만: F
```

```
중간고사 점수 입력: 77
기말고사 점수 입력: 66
평균: 71.5
학점: C
```

- 요구사항
  - 평균은 소수 첫째 자리까지 나타낼 것
  - 아래의 함수를 정의해서 사용할 것
    - calc_average(??) : 2개의 점수를 전달 받아 평균(float)을 계산하여 반환
    - calc_grade(??) : 평균을 전달 받아 학점(str)을 반환
  - 제출파일명: 학번_Lab4_1.py

```
# 함수 정의
def calc_average(??):
    pass

def calc_grade(??):
    pass

# 시작 코드
```

반복문

while 문

- 조건이 참(True)인 동안 A를 반복 수행

```
while 조건식 :
    반복할 문장들 A
```

- 예제) 1부터 5까지 반복해서 숫자를 출력하는 프로그램

```
number = 1
while number <= 5 :
    print(number)
    number = number + 1
```

실행결과)

```
1
2
3
4
5
```

- 실행결과를 한 줄에 이어서 출력하려면?

```
print(number, end=' ')
```

for 문

- 순서열을 순회하다가 순서열의 끝에 도달하면 반복을 종료 함

```
for 반복변수 in 순서열 :
    반복할 문장
```

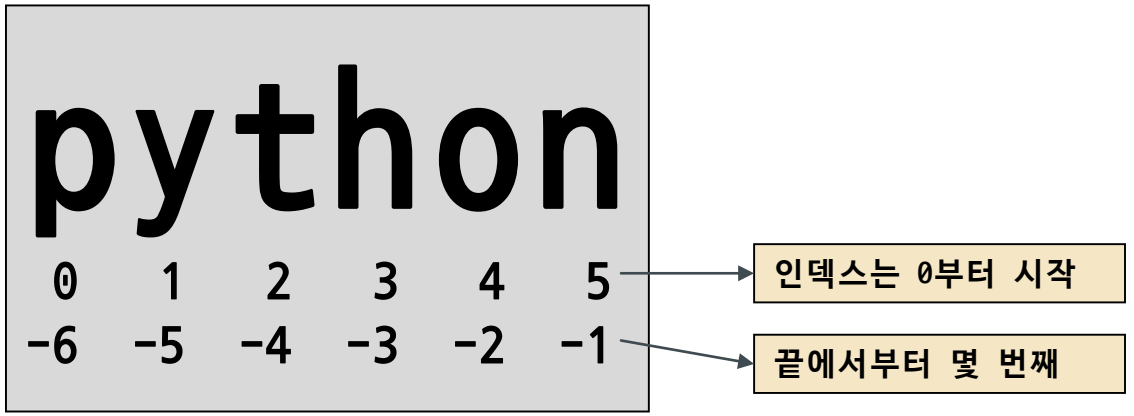
- 실습

```
for x in [1, 2, 3] :
    print(x)
```

```
for x in "python" :
    print(x, end=' ')
```

```
for x in range(3) :
    print(x, end=' ')
```

※ 문자열 인덱싱
: 인덱스를 통해 해당 값에 접근




```
>>> word = "python"
>>> word[0]
'p'
>>> word[-1]
'n'
```

range() 함수

- 형식

```
range(start, stop, step)
```

start부터 step만큼씩 (증가/감소)하여 stop전까지의 정수들을 생성

```
>>> range(1, 7, 2)
```

- start와 step은 생략 될 수 있음
  - 생략된 경우 start의 값은 0, step은 1이라고 간주

```
>>> range(5)
```

→ `range(stop)`

```
>>> range(5, 10)
```

→ `range(start, stop)`

실습 3

- 아래와 같이 출력되는 프로그램을 작성하시오. 단, for 문을 이용할 것!

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, Happy new year!!
```

- 입력 받은 양의 정수를 뒤집어 출력하는 프로그램을 작성하시오.

```
양의 정수 입력: 13452
25431
```

- 입력 받은 양의 정수의 각 자릿수를 더한 값을 출력하는 프로그램을 작성하시오.

```
양의 정수 입력: 13452
15
```

중첩 반복문

- 반복문 내부의 반복문

```
while 조건1 :
    반복할 문장1
    반복할 문장2
    while 조건2 :
        반복할 문장A
        반복할 문장B
    반복할 문장3
```

```
for x in [1, 2, 3]:
    for y in [4, 5]:
        print(x, y)
    print()
```

실습 4

- 2단부터 9단까지 구구단 출력

```
== 2단 ==
2 * 1 =  2
2 * 2 =  4
2 * 3 =  6
2 * 4 =  8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
== 3단 ==
3 * 1 =  3
3 * 2 =  6

… 중략 …

9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
```

- ★출력 프로그램

정수 입력: 3259
★★★
★★
★★★★★
★★★★★★★★★★

정수 입력: 54321
★★★★★
★★★★
★★★
★★
★

break, continue

- break
  - 반복문 순회 도중 break 문을 만나면 내부 블록 벗어남

```
for i in range(1, 11) :
    if i == 5:
        break
    print(i)
```

- continue
  - 반복문 순회 도중 continue 문을 만나면 그 아래의 문장은 해당 반복에 한해서 건너뜀

```
for i in range(1, 11) :
    if i % 2 == 0 :
        continue
    print(i)
```

Problem 2

- 구구단을 2단부터 시작하여 9단까지 아래와 같이 화면에 출력하는 프로그램을 작성하시오.

  (힌트) 중첩 반복문 사용

```
2*1= 2  3*1= 3  4*1= 4  5*1= 5  6*1= 6  7*1= 7  8*1= 8  9*1= 9
2*2= 4  3*2= 6  4*2= 8  5*2=10  6*2=12  7*2=14  8*2=16  9*2=18
2*3= 6  3*3= 9  4*3=12  5*3=15  6*3=18  7*3=21  8*3=24  9*3=27
2*4= 8  3*4=12  4*4=16  5*4=20  6*4=24  7*4=28  8*4=32  9*4=36
2*5=10  3*5=15  4*5=20  5*5=25  6*5=30  7*5=35  8*5=40  9*5=45
2*6=12  3*6=18  4*6=24  5*6=30  6*6=36  7*6=42  8*6=48  9*6=54
2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49  8*7=56  9*7=63
2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64  9*8=72
2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81
```

- 제출 파일명: 학번_Lab4_2.py