

5주차

python에서 딥러닝까지

Python 기초반

주제 | 조건문 ~ 반복문

Contents

1. 조건문

1. 조건문이란?
2. if/ if-else/ if-elif-else 조건문
3. 조건문 mission

2. 리스트 자료형

- 1) 리스트 개념과 활용하는 이유
- 2) 리스트 제어하기
- 3) 리스트 mission

3. 반복문

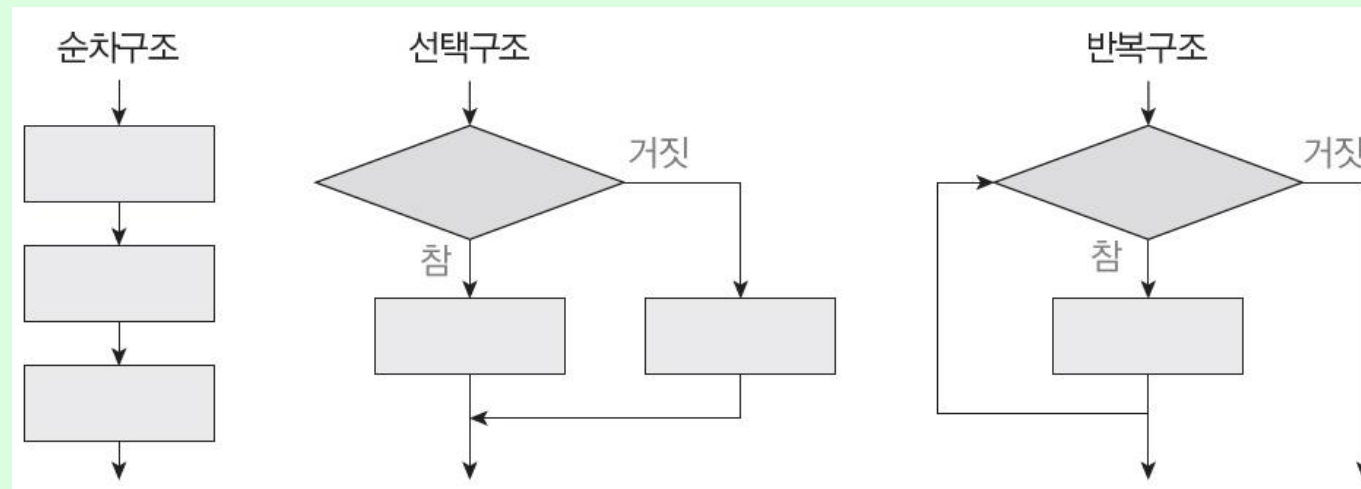
- 1) for 반복문
- 2) while 반복문
- 3) 반복문 추가 문제

Next

조건문 (교재 77p 참고)

프로그램의 3가지 제어구조

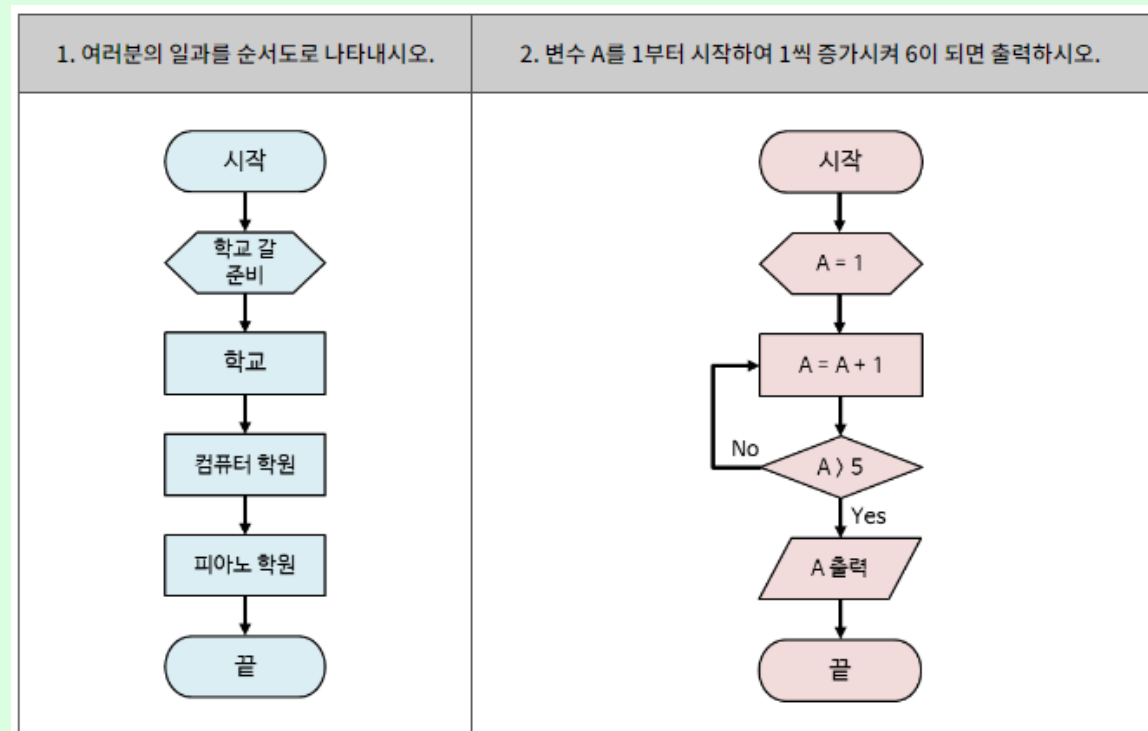
- 순차구조(sequence): 명령어들이 순차적으로 실행되는 구조
- 선택구조(selection): 둘 혹은 다수의 명령어 흐름을 선택하여 실행하는 구조
- 반복구조(iteration): 동일한 명령어들이 반복되며 실행되는 구조
- 위와 같은 제어 구조들은 외우는 것이 아닌 “익숙해 져야 한다”



순서도(flowchart)

- 어떠한 일을 처리하는 과정을 순서대로 간단한 기호와 도형으로 도식화한 것

기호	명칭	설명
	단말	순서도의 시작과 끝을 나타냄.
	흐름선	순서도 기호 간의 연결 및 작업의 흐름을 표시함.
	준비	작업 단계 시작 전 해야 할 작업을 명시함.
	처리	처리해야 할 작업을 명시함.
	입출력	데이터의 입출력 시 사용함.
	의사 결정	비교 및 판단에 의한 논리적 분기를 나타냄.
	표시	화면으로 결과를 출력함.



의사코드(pseudo code)

- 컴퓨터 프로그램이나 알고리즘이 수행해야할 내용을 우리가 사용하는 언어로 간략히 서술해 놓은 것
- 주석으로 중간중간 순서대로 표시

```
import turtle

# 변수들 선언
width = 500
height = 500
lineLenght = 100
diffAngle = 90

# 스크린과 거북이 선언
win = turtle.Screen()
win.setup(width, height)
t = turtle.Turtle('turtle')
```

```
# 그림그리기
t.forward(lineLenght)
t.right(diffAngle)
t.forward(lineLenght)
t.right(diffAngle)
t.forward(lineLenght)
t.right(diffAngle)
t.forward(lineLenght)
t.right(diffAngle)

turtle.mainloop()
```

조건문

- 조건문이란?

: 조건에 따라 명령이 달라지는 제어문

- 제어문 및 함수의 주요 특징

: 제어문에 포함된 문장은 “띄어쓰기”로 구분한다.

```
if a == 4:  
    # 조건이 참이므로 실행됩니다.  
    print('한국 사람이 싫어하는 숫자 입니다.')  
    print('그래서 4층을 "F"층으로 표기하기도 합니다.')  
    print()
```

if 조건문

- if 조건문(Conditional)
 - if: '만약 ~라면(if)', 첫번째 조건
 - elif: '그렇지 않고(else) 만약 ~라면(if)'
 - else: '그렇지 않다면(else)'
 - 조건이 True(1)가 되는지를 판단
- 주의사항
 - 조건 뒤 ":" (콜론) 반드시 기입
 - 조건 충족 시, 실행할 명령들은 들여쓰기로 구분

```
if 조건A:  
    조건A가 참이면 수행할 코드  
...  
if 조건A2:  
    조건A와 조건A2가 모 참이면 수행할 코드  
...  
elif 조건B:  
    조건A가 거짓이고, 조건B가 참이면 수행할 코드  
...  
elif 조건C:  
    이전 조건들(조건A, 조건B)가 거짓이고, 조건C가 참이면 수행할 코드  
...  
else:  
    이전 조건들이 모두 거짓이면 수행할 코드  
...
```


조건문 연습문제

- 교재 80p “덧셈 문제를 맞히는 프로그램 ”
- if-else로 판단하는 프로그램을 작성해 봅시다.
- 조건
 - input을 활용하여 “12+23=”을 출력하고 사용자로부터 답을 입력 받는다.
 - x가 12+23의 결과와 일치하는 경우: “천재!”를 출력
 - x가 12+23과 일치하지 않는 경우: “바보?”를 출력

조건문 Mission1

- Mission

: 구독자 수를 입력 받고 수익창출이 되는지 여부를 판단하는 프로그램을 작성해 보자

구독자 수를 입력하세요 >> 995
수익창출을 할 수 없습니다.

구독자 수를 입력하세요 >> 2365
수익창출을 할 수 있는 계정입니다.

조건문 Mission2

- Mission

: 현재 가진 금액을 통해 먹을 수 있는 음식을 출력하는 프로그램을 작성해 보자

- 20000원 이상: 오늘 저녁은 치킨이닭!
- 10000원 이상: 이제는 고오급 음식인 떡볶이를 먹으러 가자!
- 2000원 이상: 그래도 굶지는 않는다! 편의점 삼각김밥!
- 2000원 미만: 돈이 없다고 굶어야 하는 것은 아니다... 친구에게 “한입만”을 외쳐보자

현재 가진 금액을 입력하세요 >> 30000

오늘 저녁은 치킨이닭!!

조건문 Mission3

- Mission

: 국어, 영어, 수학 점수를 입력 받고 합계와 평균을 출력한 뒤

- 평균이 60점이 넘을 경우: 보충 대상자가 아닙니다
- 평균이 60점을 넘지 않을 경우: 보충 대상자입니다.

를 출력하는 프로그램을 작성해 보자

```
국어>>> 95
```

```
수학>>> 100
```

```
영어>>> 85
```

```
합계: 280, 평균: 93.33
```

```
보충 대상자가 아닙니다. 즐거운 방학 보내세요 :D
```

Next

리스트 자료형

리스트 자료형

- 리스트 자료형이란?

: 데이터가 묶여 일렬로 들어있는 것과 같은 군집형 자료형 중 하나

- 리스트를 사용하는 이유

- 여러 개의 데이터를 한 번에 저장하기 위해서
- (여러 개를 한 번에 관리하므로써) 데이터를 효율적으로 관리하고 제어하기 위해서
- 예시) 아파트 호수를 방송



리스트 자료형

- 리스트 선언
 - 리스트이름 = [데이터1, 데이터2, ...]
 - 빈 리스트도 생성 가능: 리스트 이름 = []
- 리스트 제어1: 데이터 추가하기
 - 리스트이름.append()
 - ※ append: '추가'라는 의미

리스트 자료형

- 리스트 제어2: 인덱싱

- 순서 번호를 가지고 원하는 데이터만 가져오기. 이 때, 인덱스 번호는 0부터 시작
- 리스트이름[인덱스 번호]

- 리스트 제어3: 데이터 수정하기

- 리스트이름[인덱스 번호] = 대신 넣을 데이터

리스트 자료형

- 리스트 제어4: 데이터 삭제하기
 - `del 리스트이름[삭제할 인덱스 번호]`
- 리스트 제어5: 리스트 슬라이싱
 - `리스트이름[시작인덱스:끝인덱스+1]`
 - `리스트이름[:끝인덱스+1]`
 - `리스트이름[시작인덱스:]`
 - `리스트이름[:]`

리스트 자료형

- 리스트 제어6: 리스트 길이 구하기
 - `len(리스트이름)`
- 리스트 제어7: 리스트 정렬하기
 - 오름차순 정렬: `리스트이름.sort()`
 - 내림차순 정렬: `리스트이름.sort(reverse=True)`

리스트 Mission1

- Mission
 - : RGB 색상(red, green, blue)을 리스트에 저장하고 turtle 모듈을 활용하여 색상이 서로 다른 직선을 그려보자
- 굵기: 30 // 선길이: 200

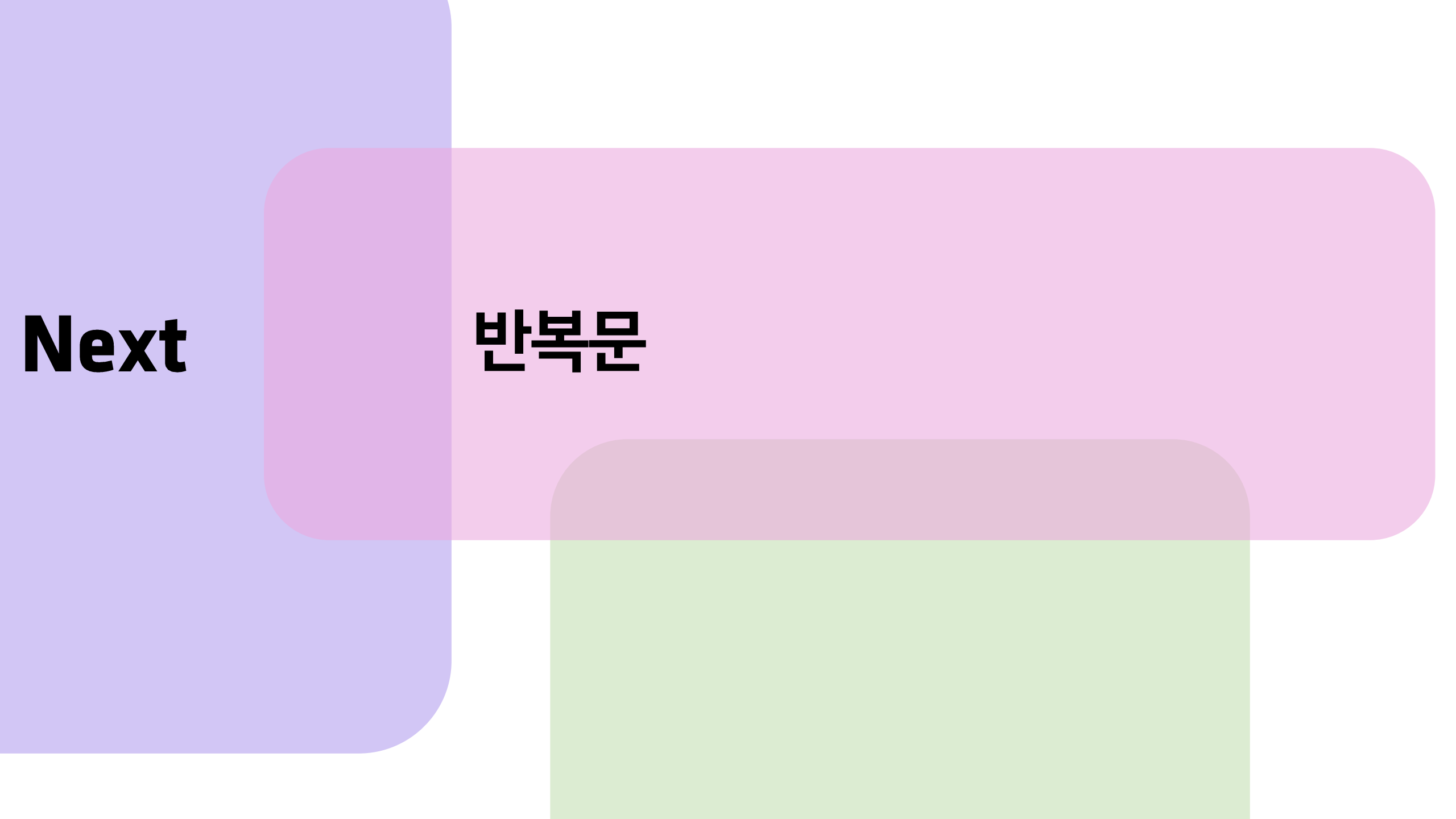
```
color = ['red', 'green', 'blue']
```

```
t.penup()  
t.setpos(0, 30)  
t.pencolor( )  
t.pendown()  
t.fd(200)
```



리스트 Mission2

- 리스트 Mission2라고 적힌 문제를 풀어보자



Next

반복문

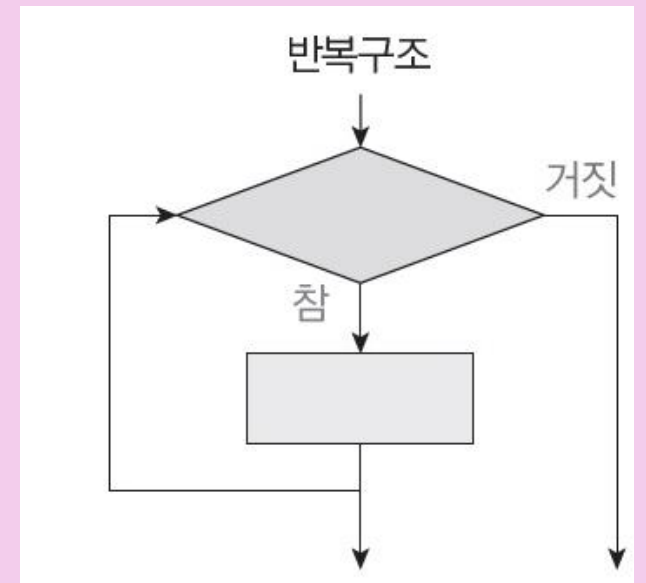
반복문(loop)의 개념

- 정의

: 동일한 기능이나 문장을 계속 해서 반복하도록 하는 명령어
즉, 반복적인 업무를 컴퓨터에게 시키도록 하는 것.

- 종류

- for 문: '횟수'에 의한 반복
- while 문: '조건'에 의한 반복



for 반복문(교재 42p)

- 기능: '정해진 횟수' 만큼 반복을 수행하는 반복문
- 문법: 아래 반복할 명령들은 들여쓰기(공백 4번)으로 구분
 - for <변수> in <리스트(or 튜플 or dict)>
 - for <변수> in <range()함수>
- 예시
 - for l in [1,2,3,4,5,6]:
반복할 명령들
 - for j in range(10):
반복할 명령들

```
for 변수 in 리스트(또는 튜플, 문자열):  
    수행할 문장1  
    수행할 문장2  
    ...
```

```
>>> test_list = ['one', 'two', 'three']  
>>> for i in test_list:  
...     print(i)  
...  
one  
two  
three
```

range() 함수 (교재 49p)

- 기능

: 숫자를 순서대로 발생시키는 함수

- 활용: range(start, stop, step)

- start: 시작하는 숫자, 입력 X시 기본 0으로 입력 됨. 음수 가능
- stop: 끝나길 원하는 숫자+1. 음수 가능
- step: 발생시킬 숫자들의 차이. 음수 가능

range() 함수: 활용법

- range(stop)
: 0부터 stop전까지 1만큼의 간격으로 정수를 생성
- range(start, stop)
: start에서 stop전까지 1만큼의 간격으로 정수를 생성
- range(start, stop, step)
: start에서 stop까지 step만큼의 간격으로 정수를 생성

range() 함수: range 함수로 결과 확인

- 아래의 함수들을 실행시켜보고 그 결과를 실행 코드에 주석으로 달아보자.
- **range(10)**
- **range(3,11)**
- **range(3, 15, 2)**
- **range(10, 1, -1)**

for문 연습문제

- 연습문제 1 : `range()`를 활용한 “횟수” 반복을 사용하여,
원하는 문자열을 10번 반복 출력해보자
- 연습문제 2: `list`를 활용하여 `for` 반복문을 실행시켜보자.
- 연습문제 3: 문자열 “안녕하세요”를 활용하여, `for` 반복문을 실행시켜 보자.

while 반복문

- 기능: '주어진 조건'을 만족하면 반복을 하는 반복문

- 문법

while 조건:
반복할 명령들

- 예시

- **while num<100:**
 print("아직 100에 도달하지 못했습니다")

- **while 1:**
 print("무한 loop 출력, 프로그램을 종료하려면 실행 중지 버튼을 누르세요")

while 조건:

조건이 참이면 반복 수행할 코드

...

```
i = 0;
while i < 5 :
    print("환영합니다.")
    i = i + 1
print("반복이 종료되었습니다.")
```

while문 연습문제1: 기본적인 활용

- while문을 활용하여 왼쪽 사진과 같은 출력이 나오도록 프로그램을 작성해보자

1꼬마

2꼬마

3꼬마

인디언!

break와 continue

- break

- 기능: 반복문 실행 중 반복문에서 강제로 빠져나오기 위한 키워드
- 활용: 특정 조건일 때, 반복문에서 빠져나오도록 하고 싶은 경우 if 문과 같이 활용

- continue

- 기능
 - : while문을 빠져나가지 않고 while문의 맨 처음(조건문)으로 다시 돌아가게 하는 키워드
- 활용: 특정 조건일 때, 현재 반복되는 실행을 멈추고 다음 반복 실행을 하고 싶은 경우

while문 연습문제2: 무한루프와 break

- 무한 루프와 break를 활용하여 게임 시작메뉴를 만들어 보자

```
1을 입력하여 게임을 종료하세요>> 0  
1을 입력하여 게임을 종료하세요>> 5  
1을 입력하여 게임을 종료하세요>> 7  
1을 입력하여 게임을 종료하세요>> 1  
끝
```

while문 연습문제3: continue

- 연습문제

: continue문을 활용하여 1~10까지 숫자 중 홀수만 출력하도록 만들기

- 조건: while문과 if문 continue를 활용할 것

1 3 5 7 9

반복문 Mission1-1

- Mission: 원하는 단을 입력 받고 그 숫자의 구구단 출력하기
- 조건: for문을 활용할 것.

```
구구단 단 수를 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
```

반복문 Mission1-2

- Mission: 2~9단을 모두 출력하는 프로그램을 작성해보자
- 조건: 이중 for문을 활용할 것.

[구구단 전체 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
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
```



[9단]

```
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
```

반복문 Mission2-1

- Mission: 원하는 단을 입력 받고 그 숫자의 구구단 출력하기
(반복문 Mission1-1과 같은 출력)
- 조건: 단, while문을 활용할 것.

```
구구단 단 수를 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
```

반복문 Mission2-2

- Mission: 2~9단을 모두 출력하는 프로그램을 작성해보자
- 조건: while문을 2번 활용할 것.

[구구단 전체 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
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
```



[9단]

```
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
```

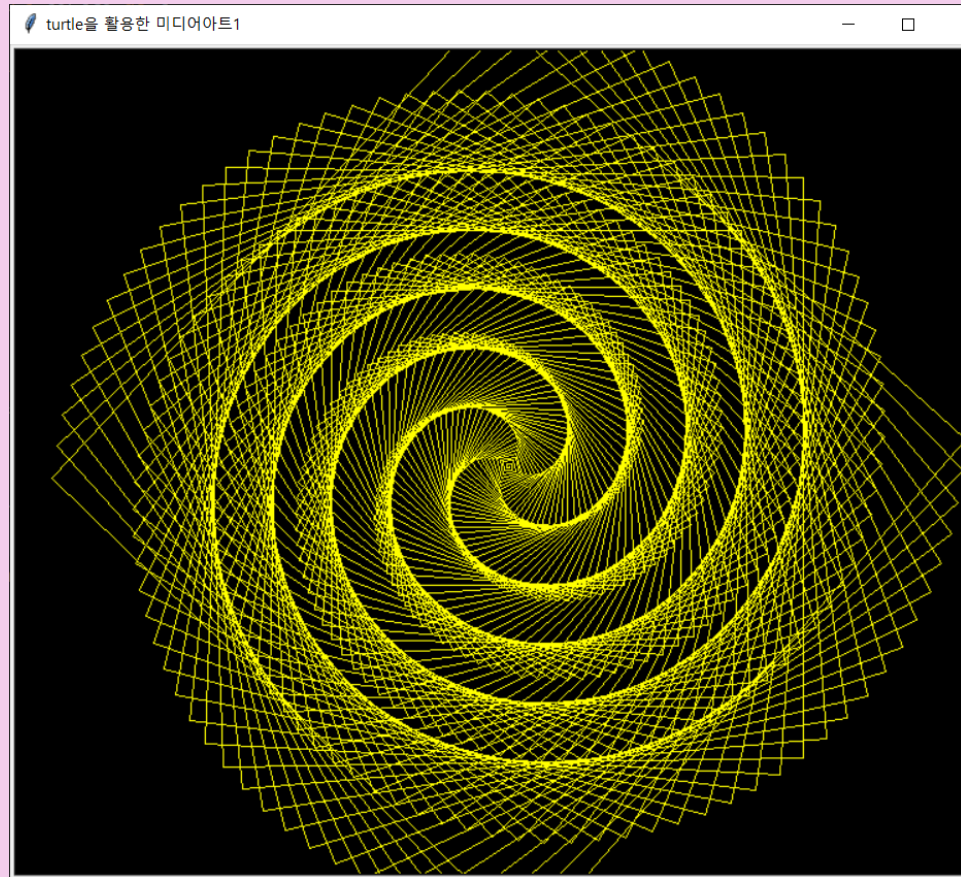
반복문 Mission3

- Mission 영어 타자 연습을 할 수 있는 프로그램을 작성해보자
- 규칙
 1. 연습할 영단어가 담긴 list를 만든다.
 2. 리스트에서 순서대로 단어를 가져와 화면에 출력한다.
 3. 프로그램 사용자는 단어를 그대로 입력한다.(input)
 4. 입력이 끝나면 전체 문제, 맞은 문제, 틀린 문제의 수가 출력되도록 한다.

```
Let's Learning English
study
study
pizza
pizza
overwatch
overwatcha
I love chicken
I love chicken
전체 문제 개수: 4
맞힌 문제 개수: 3
틀린 문제 개수: 1
```

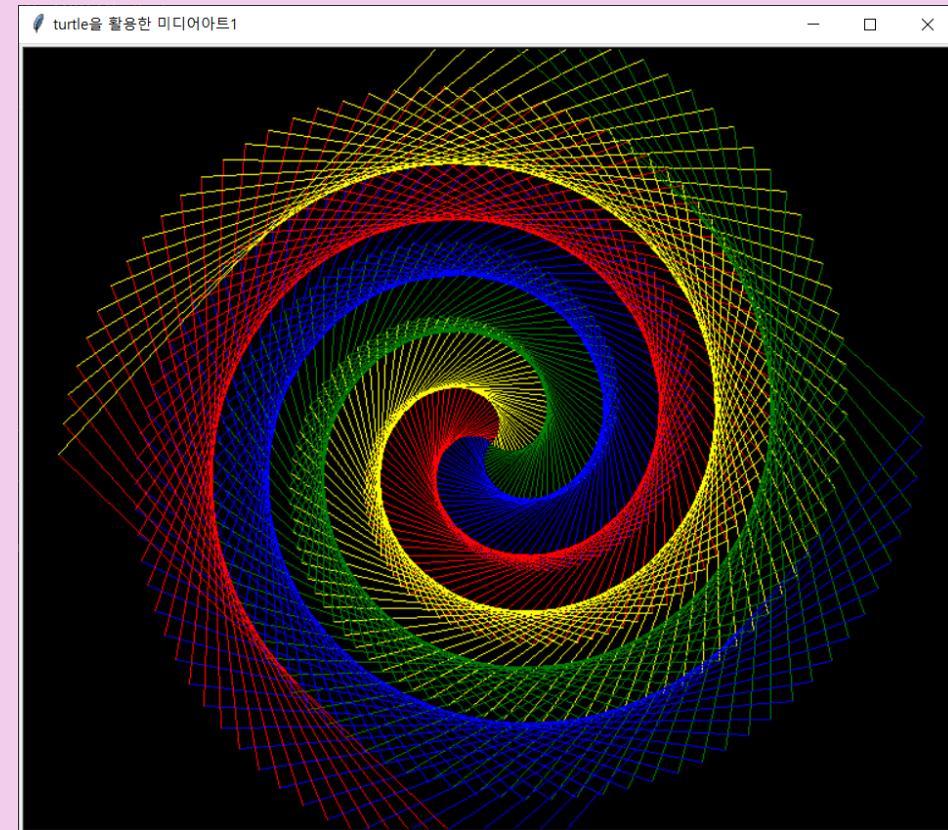
반복문 Mission4-1: 미디어아트1

- turtle 명령어에 대한 내용은
교재 55p, 111p 참고
- for 반복문을 활용해서 다음과 같은 문양을
출력해보자.
 - range를 활용하여 300번 반복할 것.
 - 꺾는 각도를 “89도”로 설정해 줄 것.
 - 사용 명령: t.forward(이동거리), t.right(각도)



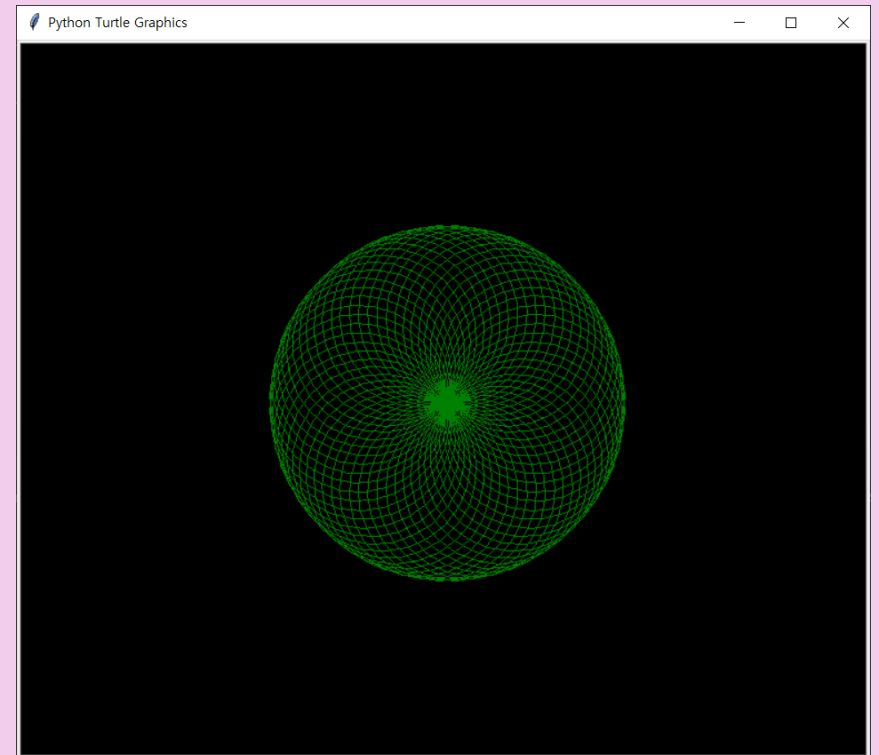
반복문 Mission4-2: 미디어아트1

- for 반복문을 활용해서 다음과 같은 문양을 출력해보자.
 - list를 활용하여 'red', 'yellow', 'green', 'blue'를 각각 설정해 줄 것.
 - range를 활용하여 300번 반복할 것.
 - 꺾는 각도를 "89도"로 설정해 줄 것.



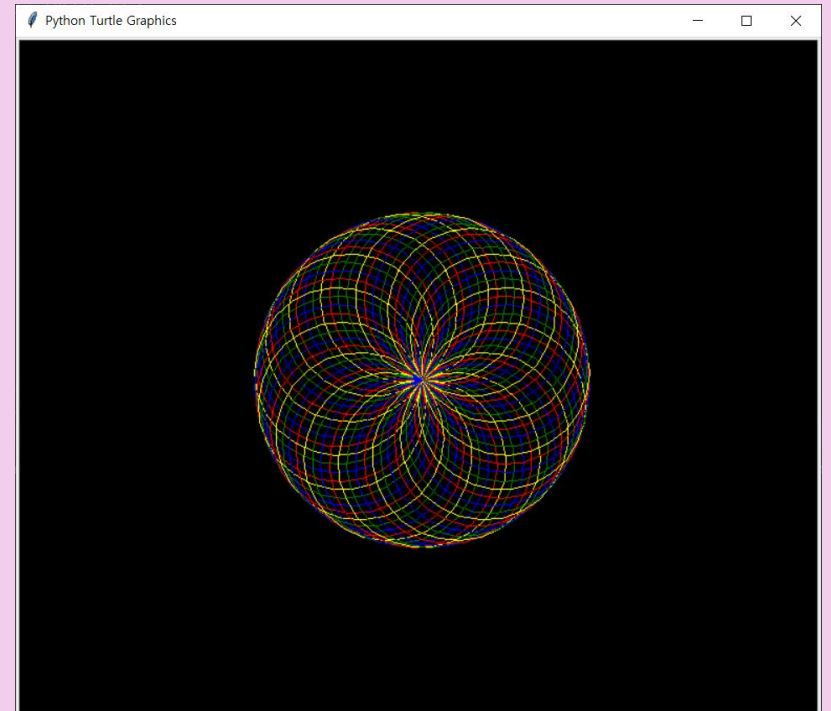
반복문 Mission5: 미디어아트 2

- turtle 명령어를 활용하여 원을 반복해서 그리는 프로그램을 작성해보자.
- 활용 명령어
 - `t.circle(반지름 길이)`, 여기서는 80~100 입력
 - `t.left(각도)`
- `n`은 50 이상으로 설정할 것.



반복문 Mission5-2: 미디어아트 2

- turtle 명령어를 활용하여 원을 반복해서 그리는 프로그램을 작성해보자.
- color = ['red', 'yellow', 'green', 'blue']로 색상 제어
- 활용 명령어
 - t.circle(반지름 길이), 여기서는 80~100 입력
 - t.left(각도)
- n은 50 이상으로 설정할 것.



Next

반복문 추가문제

이중 for문 (중첩 loop)

- for문 안에 for문을 반복하는 것. (while문도 중첩으로 사용 가능)
- 활용하는 경우: 차원이 있는 구조를 다루는 경우

```
array = [[101, 102, 103, 104, 105],  
         [201, 202, 203, 204, 205],  
         [301, 302, 303, 304, 305]]  
  
for i in range(3):  
    for j in range(5):  
        print(array[i][j], end='  ')  
    print()
```

이중 for문 연습문제

- Mission: 이중 for문을 활용하여 별 모양으로 직각삼각형 만들기
 - 조건1: 이중 for 문을 활용할 것
 - 조건2: 4줄 정도로 끝낼 것

```
C:\Anaconda3\envs\python_deep\python.exe "D:/Google 드라이브/Github_asdfrv20/Python_deep/3rd week mission.py"  
*  
**  
***  
****  
*****
```

랜덤 함수

- 난수(random number): 무작위 숫자

- 필요 모듈: import random

```
import random
```

- random 명령어

```
random_num = random.randint(1, 100)
```

- random.random(): 0이상 1미만의 숫자 중에서 무작위 숫자를 돌려주는 함수
- random.randint(시작숫자, 끝나는 숫자의 다음숫자)
: 정해진 범위에서 하나의 정수를 랜덤으로 반환하는 함수
 - 주의: randrange()에 들어가는 숫자는 정수이다.

Mission1: Up-Down Game 만들기

- Mission

: 랜덤으로 정해진 숫자(1~100)사이 숫자를 맞추는 Up-Down 게임

- computer가 1~100 사이의 랜덤정수를 하나 발생시킨다.
- 유저가 임의의 숫자를 입력한다. 이때 입력한 숫자가
 - 랜덤 정수와 같으면 Win!!
 - 랜덤 정수보다 작으면 Up
 - 랜덤 정수보다 크면 Down

```
import random
```

```
random_num = random.randint(1, 100)
```

※ while을 활용하여 무한루프를 만들 것.

Mission2: ASCII 코드를 활용한 슬롯머신

- Mission: ASCII 코드를 활용한 슬롯머신 만들기
- 조건
 1. 3개의 random 정수를 발생시킨 후, 이를 ASCII 코드로 변환하여 슬롯머신을 출력하는 프로그램을 작성한다
 2. 시작화면: "1.게임시작 // 2.나가기 >> " 를 출력 후 입력을 받는다
 1. 1. 선택 시, <SLOT MACHINE> 기능 실행
 2. 2. 나가기 실행 시 프로그램 종료
 3. 이외의 숫자 입력 시, "잘못 입력하셨습니다."를 출력하고 다시 시작화면에서 입력을 받기

Mission2: ASCII 코드를 활용한 슬롯머신

3. <SLOT MACHIN> 기능: “코인을 넣어주세요!(1.코인넣기//2.그만하기)>>”

출력 후 입력받기

- 1 입력 시, 3개의 랜덤한 정수를 발생시키고 이를 ASCII코드로 전환, 화면에 출력 후. 결과까지 출력 & <SLOT MACHIN> 기능 계속 반복
 - 셋 다 일치할 경우, 'JACKPOT!!' 출력)
 - 이외의 경우, '아쉽습니다. 다음 기회에 ' 출력
- 2 입력 시, 메인 화면으로 이동
- 이외의 입력에 대해, “잘못 입력하셨습니다.” 출력하기

Mission2: ASCII 코드를 활용한 슬롯머신

4. ASCII 코드로 자료형 변환하기

- `random.randint()`를 활용하여 “33~39”(총 7개) 사이의 정수를 임의로 3개 할당한다. (| “ # \$ % & ‘ 로 총 7개)
- 생성된 변수를 `print(“%c” %chr(자료형 변환할 변수))`로 출력해 준다.
- 이를 슬롯머신과 같은 모양의 틀에 넣어 출력해 줄 것!

Mission2: ASCII 코드를 활용한 슬롯머신

• 출력 화면

```
[Slot Machine GAME]
1.게임시작 // 2.나가기>>1
```

```
1.게임시작 // 2.나가기>>2
게임을 종료합니다:D
```

```
1.게임시작 // 2.나가기>>5
잘못 입력하셨습니다.
```

```
<SLOT MACHINE>
코인을 넣어주세요!(1.코인넣기//2.그만하기)>>1
-----
| # | ! | $ |
-----
아쉽습니다. 다음 기회에!
```

```
<SLOT MACHINE Start!>
코인을 넣어주세요!(1.코인넣기//2.그만하기)>>5
잘못 입력하셨습니다.
```

```
<SLOT MACHINE Start!>
코인을 넣어주세요!(1.코인넣기//2.그만하기)>>
```

```
<SLOT MACHINE Start!>
코인을 넣어주세요!(1.코인넣기//2.그만하기)>>2
슬롯머신 게임을 중단합니다.
```

```
[Slot Machine GAME]
1. 게임시작 // 2. 나가기>>|
```

Mission3: turtle 모듈로 무지개 그리기

- Mission: turtle 모듈을 활용하여 무지개 그리기
 - rainbow_size=500 // pen_size=30
 - rainbow_color = ['red', 'orange', 'yellow', 'green', 'blue', 'navy', 'purple']
 - for문 Hint!
: 위쪽 보기>펜 들기 > 무지개 시작점으로 이동 > 색 지정 > 펜 내리기 >그리기



Mission3: turtle 모듈로 무지개 그리기

```
for i in range(10):  
    t.setheading(0)  
    t.penup()  
    t.setpos(0, 0)  
    t.pencolor('red')  
    t.pendown()  
    t.circle(100, 180)
```

- setheading(각도): 거북이가 바라보는 방향을 각도로 설정
- penup(): 거북이 객체가 움직일 때 선이 남지 않도록 하기 펜을 드는 명령
- setpos(x좌표, y좌표): 거북이 객체를 화면의 (x좌표, y좌표)로 이동시키는 명령
- pencolor(색상명): 거북이 객체가 움직이면서 그리는 선의 색상을 설정하는 명령어(색상명은 문자열)
- pendown(): 거북이 객체의 움직임에 따라 선이 남도록 펜을 내리는 명령어
- circle(반지름길이, 이동각도): 현재 위치에서 시작하여 입력된 반지름 길이를 가지는 원을 이동각도만큼 움직이도록 하는 명령어