



▶ 학습내용

01. 반복문의 필요성과 특성을 이해합니다.
02. for문을 사용하여 정해진 횟수만큼 반복하는 방법을 학습합니다.
03. while문을 사용하여 조건으로 반복하는 방법을 학습합니다.
04. 반복문의 흐름을 제어하는 방법에 대해 이해합니다.

▶ LAB

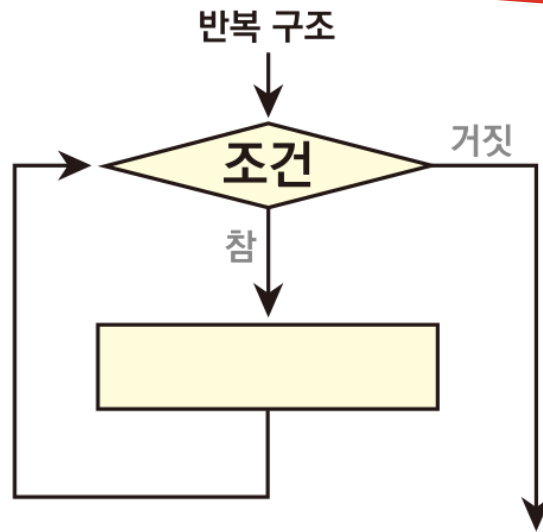
01. 코드를 줄여보아요
02. 도돌이표
03. n각형 그리기
04. 랜덤 워크 시뮬레이션
05. 범인 찾기 게임
06. 몬드리안 터틀
07. 모든 약수 구하기
08. 최대공약수 구하기
09. 별 그리는 터틀
10. 숫자 맞추기 게임

01. 왜 반복이 중요할까요?



- 루프(loop)
- 동일한 작업을 여러 번 수행할 때 무엇이 더 간결할까?

똑같은 문장 여러 번 쓰기 vs 반복 구조



01. 왜 반복이 중요할까요?



반복 구조를 이용하면 프로그램이 더 간결해집니다.

같은 작업을 여러 번 나열하여 작성

```
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")
```

반복 구조를 이용하여 작성한 코드

```
for i in range(5):  
    print("방문을 환영합니다!")
```

i j k l ...

02. 횟수 제어 반복-for



- 횟수 제어 반복 : 반복의 횟수를 미리 아는 경우에 사용, for 루프

많은 언어에서 횟수 제어 반복에 for를 키워드로 사용합니다.

for 문

for 변수 in range (종료 값) :
문장

0에서 (종료 값-1)까지의 숫자를 반환합니다.

반복되는 문장으로 들여쓰기 하여야 합니다.

02. 횟수 제어 반복-for



예를 들어서 "방문을 환영합니다" 문장을 5번 반복하여 출력해보겠습니다.

코드

```
for i in range(5):  
    print("방문을 환영합니다.")
```

끝에 콜론(:)이 있음
들여쓰기하세요.

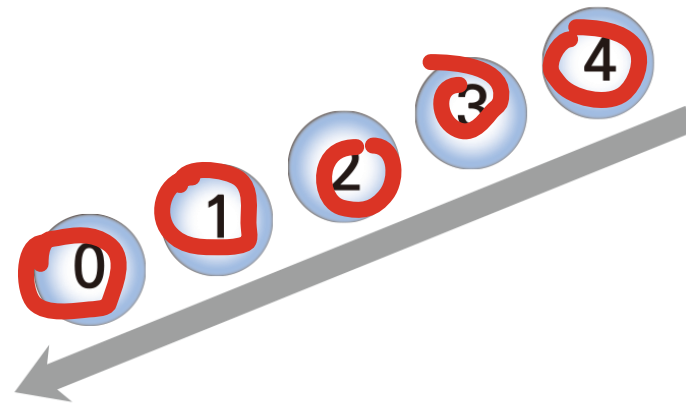
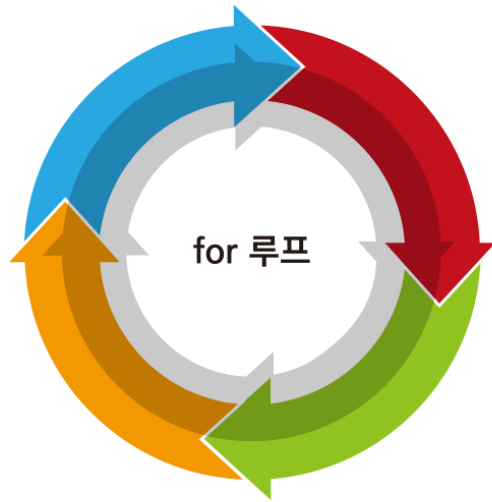
실행 결과

방문을 환영합니다.
방문을 환영합니다.
방문을 환영합니다.
방문을 환영합니다.
방문을 환영합니다.

03. range() 함수



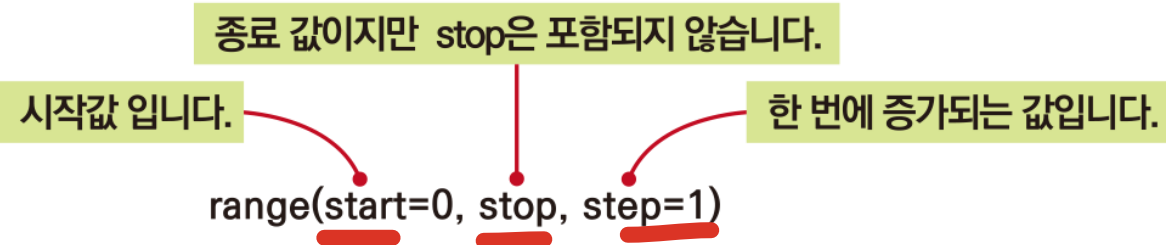
- range() 함수: 숫자들을 생산하는 공장
- range(5)



03. range()함수



range() 함수



03. range() 함수



만약 1부터 시작하여서 5까지 반복하고 싶다면 어떻게 하면 될까요?

코드

```
for i in range(1, 6, 1):  
    print(i, end=" ")
```

실행 결과

1 2 3 4 5



03. range() 함수



만약 10부터 시작하여서 1까지 반복하고 싶다면 어떻게 하면 될까요?

코드

```
for i in range(10, 0, -1):  
    print(i, end=" ")
```

실행 결과

10 9 8 7 6 5 4 3 2 1



04. 횟수 제어 반복을 좀 더 이해시켜 줄 예제

11



1부터 100까지의 합을 구하는 프로그램

1에서 100까지의 자연수의 합을 구하여 출력하는 프로그램을 작성해 봅시다.

코드

```
sum = 0  
for i in range(1, 101):  
    sum += i  
print("1부터 100까지의 합은", sum, "입니다.")
```

실행 결과

1부터 100까지의 합은 5050 입니다.

sum = sum + i

04. 횡수 제어 반복을 좀 더 이해시켜 줄 예제

12



팩토리얼 계산 프로그램

팩토리얼 $n!$ 은 1부터 n 까지의 정수를 모두 곱한 것을 의미합니다. 즉, 다음과 같습니다.

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

n 값을 입력하면 $n!$ 를 계산하는 프로그램을 작성해 보세요.

코드

```
n = int(input("정수를 입력하시오: "))  
fact = 1  
  
for a in range(1, n + 1):  
    fact = fact * a  
  
print(n, "!은", fact, "이다.")
```

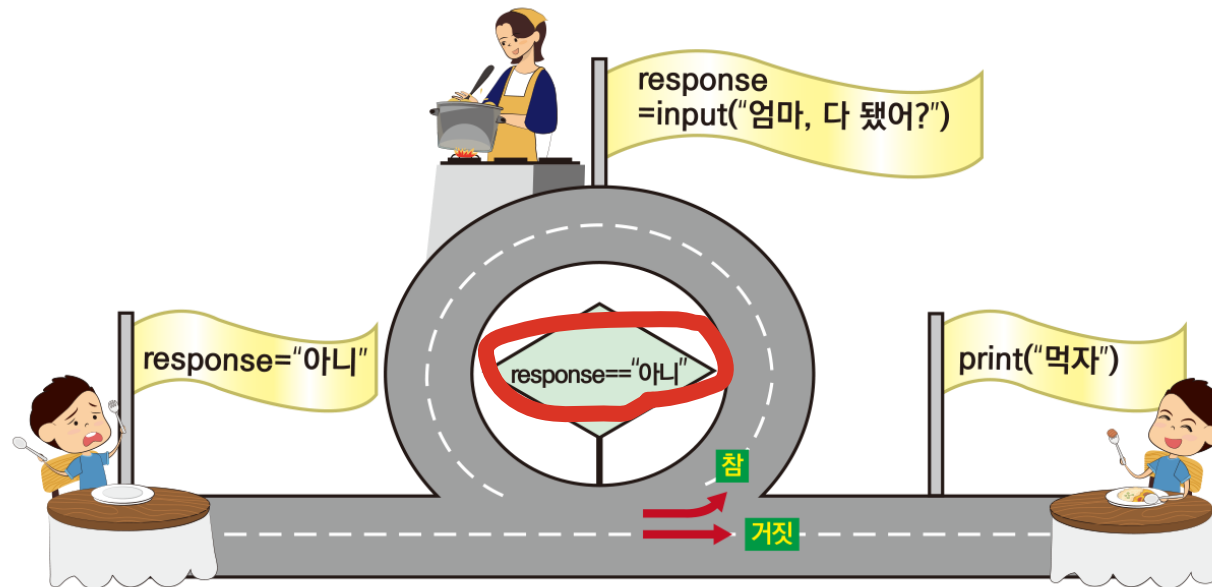
실행 결과

정수를 입력하시오: 10
10 !은 3628800 이다.

05. 조건 제어 반복-while



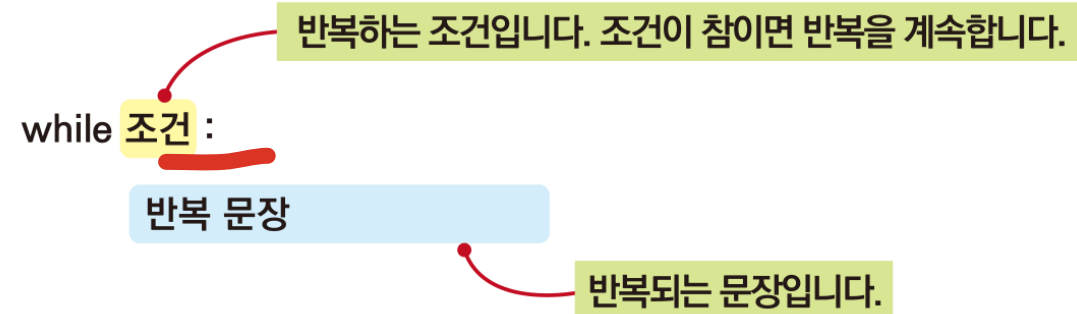
- 어떤 조건이 만족되는 동안 반복
- 반복의 횟수는 모르지만, 반복의 조건은 알고 있는 경우에 주로 사용하는 반복



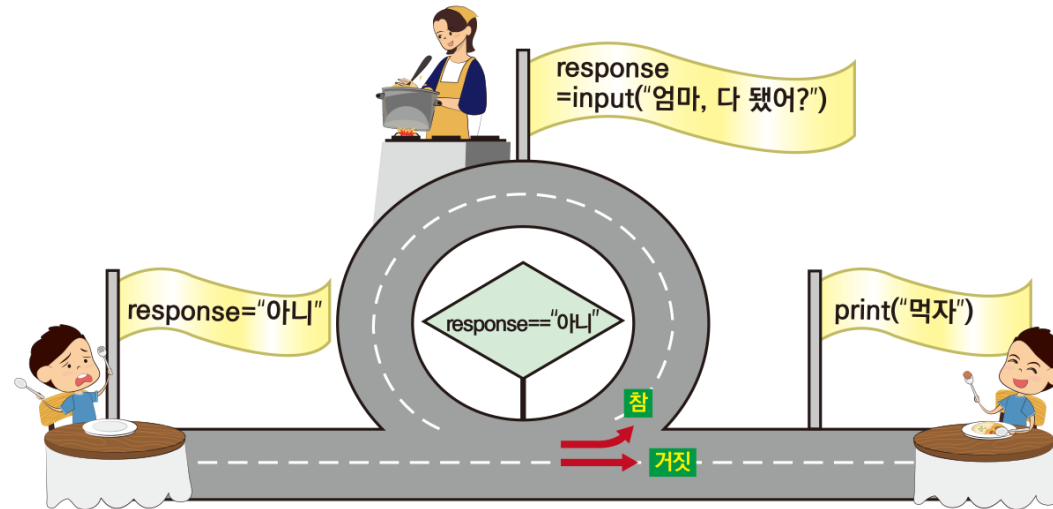
05. 조건 제어 반복-while



while 루프



05. 조건 제어 반복-while



위의 상황을 프로그램으로 작성해 보면 다음과 같습니다.

코드

```
response = "아니"  
while response == "아니":  
    response = input("엄마, 다됐어? ");  
    print("먹자")
```

실행 결과

```
엄마, 다됐어? 아니  
엄마, 다됐어? 아니  
엄마, 다됐어? 그래  
먹자
```

06. 조건 제어 반복을 좀 더 이해시켜 줄 예제

16



1부터 100까지의 합을 구하는 프로그램

1에서 100까지의 자연수의 합을 구하여 출력하는 프로그램을 작성해 봅시다.

코드

```
count = 1
sum = 0
while count <= 100 :
    sum = sum + count
    count = count + 1
print("1부터 100까지의 합은",sum,"입니다.")
```

실행 결과

1부터 100까지의 합은 5050 입니다.

06. 조건 제어 반복을 좀 더 이해시켜 줄 예제

17



로그인 프로그램

코드

```
password = ""  
while password != "pythonisfun":  
    password = input("암호를 입력하시오: ")  
print("로그인 성공")
```

실행 결과

암호를 입력하시오: idontknow
암호를 입력하시오: 12345678
암호를 입력하시오: pythonisfun
로그인 성공



07. 중첩 반복문



다음과 같이 '*'을 이용하여 사각형을 출력하는 프로그램을 작성해 봅시다.

한 줄에 '*'이 10개씩 총 5줄을 출력하여 사각형을 만들겠습니다.

코드

```
for i in range(5):  
    for j in range(10):  
        print("*", end=" ")  
    print("")
```

실행 결과

```
*****  
*****  
*****  
*****  
*****
```

Print

행 5

(5, 10)
10-10
5

07. 중첩 반복문



다음과 같이 '*'을 이용하여 직각삼각형을 출력하는 프로그램을 작성해 봅시다.

코드

```
for i in range(1, 6):  
    for j in range(1, i+1):  
        print("*", end=" ")  
    print("")
```

실행 결과

```
*  
* *  
* * *  
* * * *  
* * * * *
```

08. 무한 반복



- 반복이 무한히 발생하는 것 = 무한 루프(infinite loop)
- 특정 조건에서 그 무한 반복을 멈추는 것을 생각해줘야 함.



무한 반복을 이용하여 신호등 프로그램을 작성해 보겠습니다.

코드

```
sign = True

while sign:
    light = input('신호등 색상을 입력하시오: ')
    if light == 'blue':
        sign = False

print('전진!!')
```

실행 결과

신호등 색상을 입력하시오: red
신호등 색상을 입력하시오: red
신호등 색상을 입력하시오: blue
전진!!

09. break와 continue



- break : 강제로 반복 중지

코드

```
while True:
    light = input('신호등 색상을 입력하시오: ')
    if light == 'blue':
        break
    print('전진!!')
```

실행 결과

신호등 색상을 입력하시오: red
신호등 색상을 입력하시오: red
신호등 색상을 입력하시오: blue
전진!!

09. break와 continue



- continue : 해당 차례를 건너뛰고 새롭게 다음 차례의 반복을 수행

코드

```
for n in range(10):  
    if n % 2 == 0:  
        continue ✓  
    print(n)
```

실행 결과

1
3
5
7
9

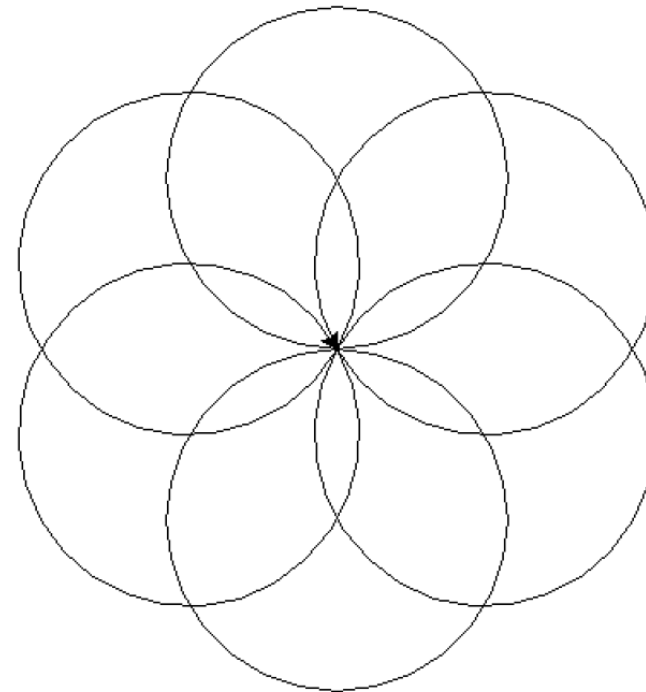
Lab. 코드를 줄여보아요.



코드

```
import turtle
t = turtle.Turtle()
t.circle(100) # 반지름이 100인 원을 그림
t.left(60)   # 60도만큼 터틀을 왼쪽으로 회전
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
```

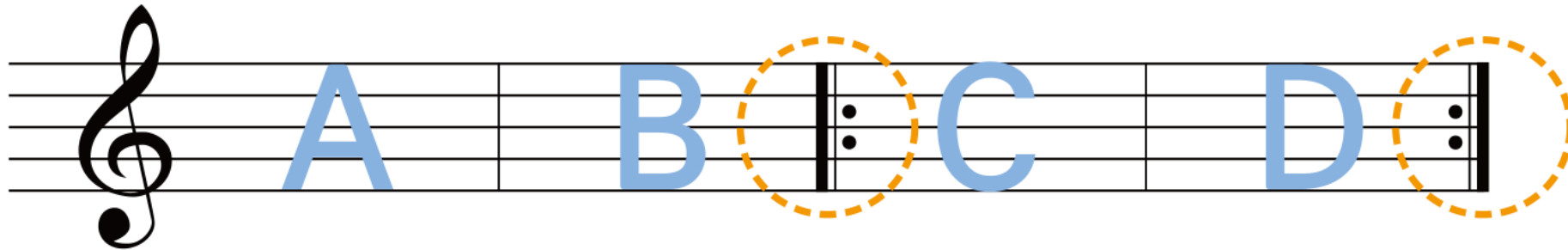
실행 결과



Lab. 도돌이표



- 악보에서 반복을 표현할 때 사용하는 기호가 도돌이표입니다.
- 다음 악보를 연주하는 순서를 출력하는 프로그램을 반복문을 이용하여 작성해 봅시다.

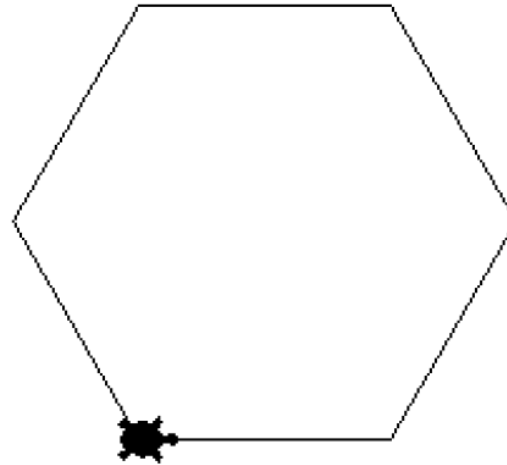
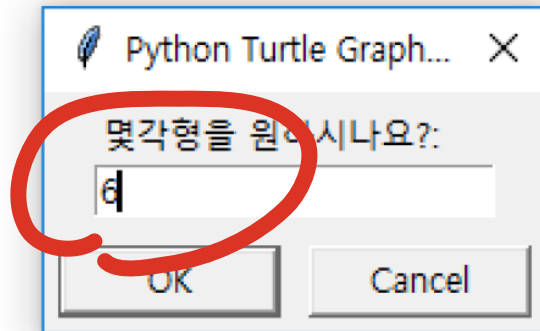


연주 순서: A - B - C - D - C - D

Lab. n각형 그리기



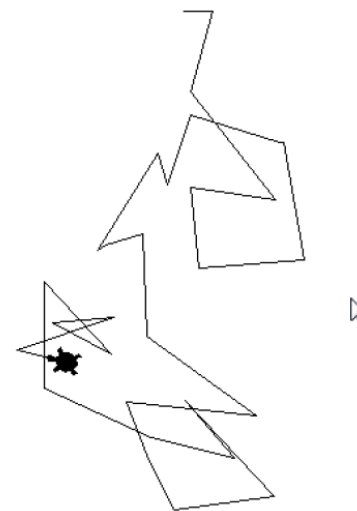
- 사용자로부터 정수 n 을 입력받아서 한 변의 길이가 100인 정 n 각형을 그리는 프로그램을 작성해 보세요.



Lab. 랜덤 워크 시뮬레이션



- 랜덤 워크(random walk) : 수학, 컴퓨터 과학, 물리학 분야에서 임의의 방향으로 향하는 연속적인 걸음을 나타내는 수학적 개념
- 랜덤 워크의 성질을 시뮬레이션하는 프로그램을 작성해 봅시다.



생각 1 : 프로그램의 순서를 생각해 봅시다.

터틀 그래픽을 사용할 준비 작업

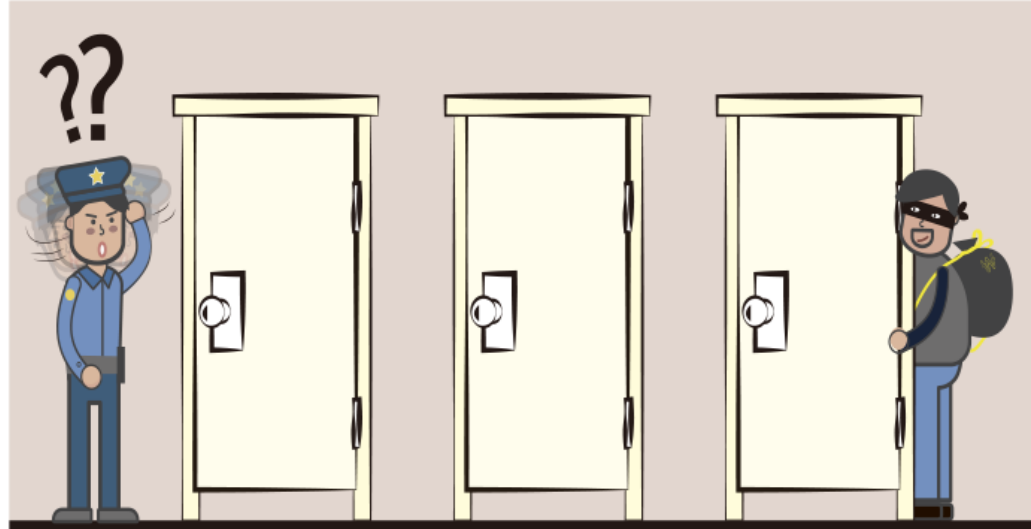


반복할 부분 구조화(터틀이 이동할 위치를 랜덤 값 받기, 터틀 이동)

Lab. 범인 찾기 게임



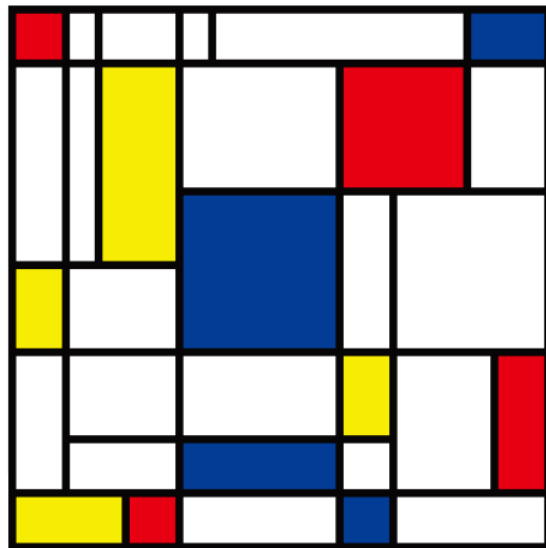
- 3개의 방 중 한 곳에 숨어 버린 범인을 찾아라!
- 범인이 숨은 방을 맞추면 100점 추가 후 게임 종료.
- 하지만 틀리면 범인은 다른 방에 숨고 10점 감점 후 다시 맞추기.



Lab. 몬드리안 터틀



- 몬드리안 : 현대 추상 미술의 거장, 구성주의 회화의 거장
순수한 점, 선, 면 색채를 이용하여 그림을 그린 것으로 유명
- 터틀 그래픽을 이용하여 선과 면을 사용하여 추상화를 그려보세요.



Lab. 몬드리안 터틀



생각 1: 프로그램의 순서를 생각해봅시다.

터틀 그래픽을 사용할 준비 작업



반복문을 이용하여 사각형의 길이, 위치, 색상 정보를 무작위로 생성



생성된 정보를 이용하여 추상화 그리기 반복



생각 2 : 터틀 그래픽과 랜덤 모듈을 사용하기 위해 import를 한 번에 해 줄 수 있습니다.

```
import turtle, random
```

Lab. 몬드리안 터틀



생각 3 : 사각형은 t.forward()와 t.right(90)를 반복하여 그립니다.



생각 4 : 사각형의 위치값은 (-300, 300) 사이의 무작위 정수로 생성합니다. 사각형 한 변의 길이는 10 ~ 300 사이의 무작위 정수로 생성합니다. 그리고 penup()과 pendwon()을 적절히 이용하여 터틀이 깔끔하게 작품활동을 할 수 있도록 도와줍니다.

```
x = random.randint(-300,300)
y = random.randint(-300,300)
length = random.randint(10,300)
```

Lab. 몬드리안 터틀



생각 5 : 터틀 그래픽에서 RGB 값을 `random()`으로 무작위로 생성하여 색을 칠합니다. 색을 칠할 때는 `color()`, `begin_fill()`, `end_fill()`을 이용합니다.

```
r = random.random( )  
g = random.random( )  
b = random.random( )
```

```
t.color(r, g, b)  
t.begin_fill()  
# 사각형 그리기  
t.end_fill()
```

Lab. 모든 약수 구하기



- 약수: 어떤 수를 나누어 떨어지게 하는 수를 그 수의 약수라고 함. 나머지를 '0'으로 하는 수
- 사용자로부터 어떤 자연수를 입력 받아 약수를 모두 출력하는 프로그램을 작성해 보세요.

실행 결과

자연수 입력: 6

1 2 3 6

Lab. 최대공약수 구하기



- 유클리드 호제법

- ① x 와 y 의 최대공약수를 (x, y) 라고 나타내기로 함

- ② a 와 b 를 자연수라고 하고 a 를 b 로 나눈 나머지를 r 이라고 하면 $\rightarrow (a, b) = (b, r)$

- 예 : 722와 190의 최대공약수

$$722 \div 190 = 3 \quad \text{나머지 } 152$$

$$190 \div 152 = 1 \quad \text{나머지 } 38$$

$$152 \div 38 = 4 \quad \text{나머지 } 0$$

따라서 722와 190의 최대공약수는 나머지가 0이 될 때의 나눈 수인 38입니다.

Lab. 최대공약수 구하기



math

실행 결과 1

정수1 입력: 190

정수2 입력: 722

두 수의 최대공약수: 38

실행 결과 2

정수1 입력: 3

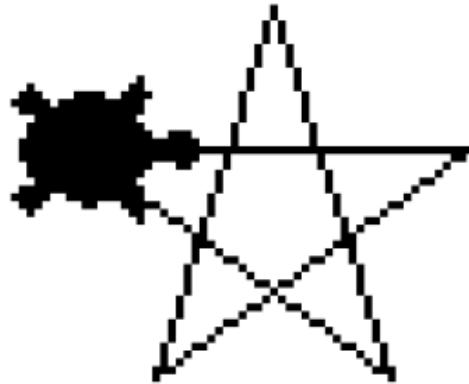
정수2 입력: 5

두 수는 서로소이다

Lab. 별 그리는 터틀



- 거북이가 별을 그리는 프로그램을 작성해 보겠습니다.



Lab. 숫자 맞추기 게임



- 1에서 100사이의 숫자를 무작위로 컴퓨터가 추출하면 사용자가 그 숫자를 맞추는 게임을 작성해 보겠습니다.

실행 결과

1부터 100 사이의 숫자를 맞추시오

숫자를 입력하시오: 50

낮음!

숫자를 입력하시오: 86

낮음!

⋮

숫자를 입력하시오: 87

축하합니다. 시도횟수= 3