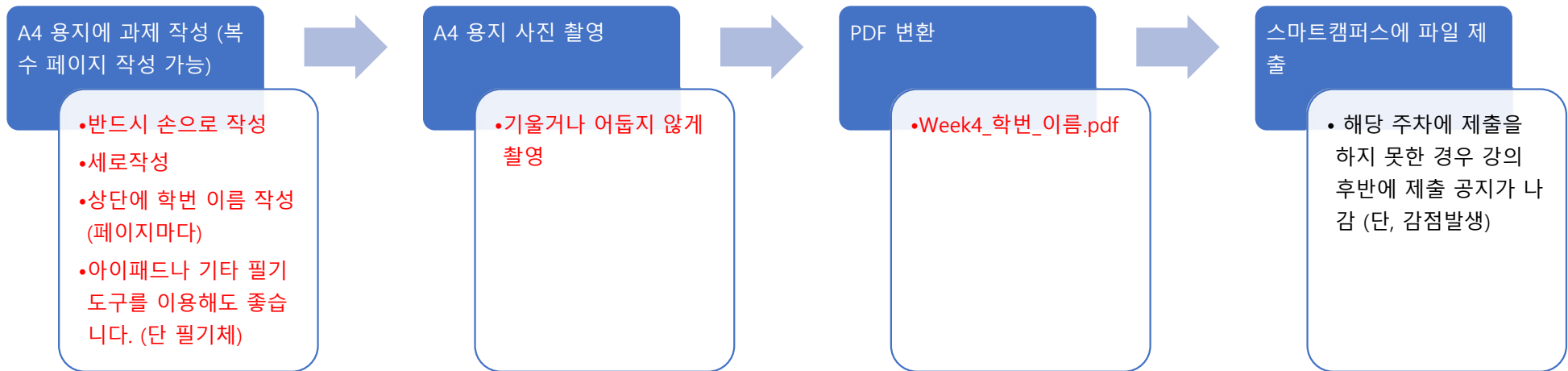


# 과제 제출 방법 안내

소프트웨어융합대학 김병정

# 작업 순서

- PDF 이외의 파일은 업로드 하지 마세요

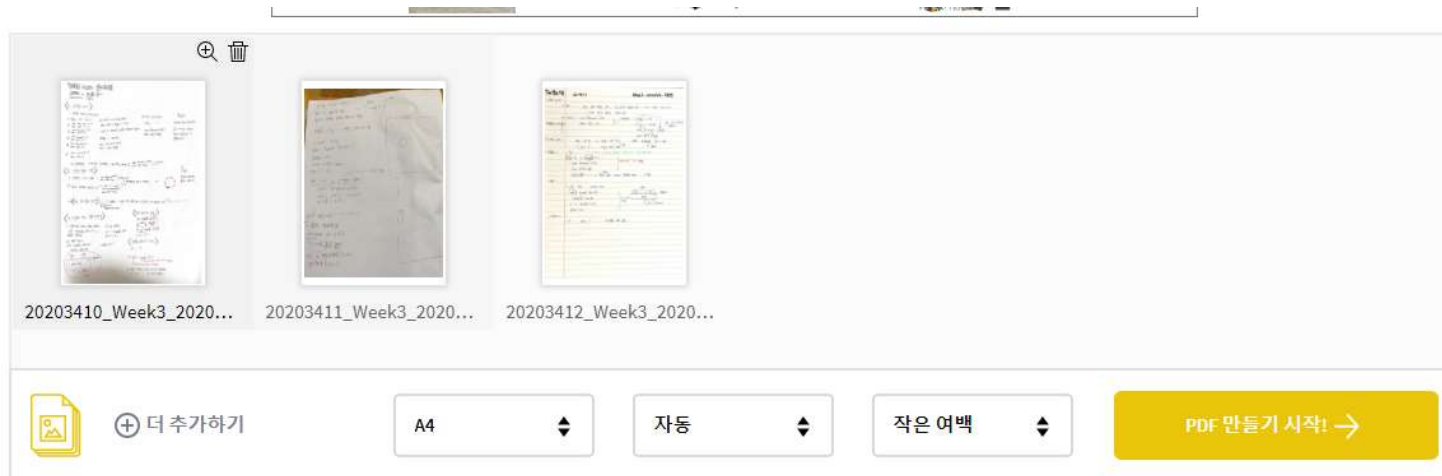


# 복수 이미지 하나 pdf 만들기

- <https://smallpdf.com/kr/jpg-to-pdf>



# 이미지 추가하고 PDF 생성



# 올바른 예

- 자유롭게 작성하면 됩니다.

3월 20/15/06

	A	B	Ex-OR	Ex-NOR	NAND	NOR
NAND	0	0	0	1	1	1
	0	1	1	0	1	0
	1	0	1	0	1	0
	1	1	0	1	0	1

AND와 NOT을 이용해 만들 수 있다.

OR과 NOT을 이용해 만들 수 있다.

AND (0):  $(\bar{A} + \bar{B}) \cdot (A + B)$

OR (1):  $\bar{A} \cdot \bar{B} + A \cdot B$

NAND:  $\bar{A} \cdot \bar{B}$

NOR:  $\bar{A} + \bar{B}$

AND와 NOT을 이용해 만들 수 있다.

OR과 NOT을 이용해 만들 수 있다.

AND (0):  $(\bar{A} + \bar{B}) \cdot (A + B)$

OR (1):  $\bar{A} \cdot \bar{B} + A \cdot B$

NAND:  $\bar{A} \cdot \bar{B}$

NOR:  $\bar{A} + \bar{B}$

## 컴퓨터설계에 필요한 논리표현방법 (2) 전산공학과

기본 논리 표현을 이용하여 만들 수 있다.

1. 1을 가지고 만든다. (OR 방법)

2. 0을 가지고 만든다. (AND 방법)

복제기

$\bar{A} \cdot \bar{B} = \bar{A+B}$

$\bar{A} \cdot B = \overline{A+B}$

AND와 NOT을 이용해 만들 수 있다.

OR과 NOT을 이용해 만들 수 있다.

NAND:  $\bar{A} \cdot \bar{B}$

NOR:  $\bar{A} + \bar{B}$

AND와 NOT을 이용해 만들 수 있다.

OR과 NOT을 이용해 만들 수 있다.

AND (0):  $(\bar{A} + \bar{B}) \cdot (A + B)$

OR (1):  $\bar{A} \cdot \bar{B} + A \cdot B$

NAND:  $\bar{A} \cdot \bar{B}$

NOR:  $\bar{A} + \bar{B}$

# 올바른 예

- 최대한 그림을 많이 사용하세요

## 포문과 함수 이용법

화학과 2022

### 1. 반복문 (여러 번의 카피맨 페이스트 → 간단하게)

import turtle

aaa = turtle.Turtle()

Cnt = 숫자 → Cnt라는 변수 정의.

Ang = 360 / Cnt → Ang라는 변수 정의.

Step = 50 → Step라는 변수 정의. (그림이 화면에 들어오도록 조정.)

for (kk) in range (Cnt):

→ 반복문을 쓸 것이라는 표현, 구간 설정시 사용

Cnt에서 하나씩 뺀 것 ex) Cnt=30

0~29를 담은 배열

aaa.forward (Step)

aaa.left (Ang)

Print (kk) → 오른쪽 하단에 0~29까지 출력됨

/ 프로그램 작성중 숫자나 결과물을 확인하고 싶을 때,

텍스트 확인. → 디버깅

### 2. 함수

→ 입력값을 여러 개 가능 \* del

def myadd (aa, bb):

→ 함수라는 표현 → 함수이름 (함수의 입력값)

return aa+bb

→ 값으로 출력

cc = myadd (3, 4)

Print (cc)

출력

7

#### \* 변수

- 변수를 사용하여 효율적으로 프로그램을 작성할 수 있다.
- 변수는 상황에 따라 설정

#### \* 반복문을 활용하여

다각형 그리기

Cnt=5 → 5각형

Cnt=7 → 7각형

Cnt=30 → 거의 원에 가깝음.

#### \* 단축키

# (컴퓨터 수행능력 제어)

→ 주석

→ Ctrl + /

# 올바른 예

- 과제는 새로운 페이지에 작성하거나, 과제라고 표시해 주세요

파이썬 과학 프로그래밍 기초 - 02 (603108) , 3주차 스마트IoT

\* 디버깅 : 변수들의 변화를 확인하고 수정할 수 있게 함  
 \* reversed() : 거꾸로 출력 함수

1) 반복문을 이용해 정사각형 그리고 한변을 그릴 때마다 4, 3, 2, 1 출력 import turtle  
a = turtle.Turtle()

① 4에서 i 빼기

```
for i in range(4):
    a.fd(150)
    a.left(90)
    print(4-i)
```

② reversed() 함수 이용

```
for i in reversed(range(4)):
    a.fd(150)
    a.left(90)
    print(i+1)
```

③ range 함수 범위 조작

```
for i in reversed(range(1,5)):
    a.fd(150)
    a.left(90)
    print(i)
```

1-1) 정사각형 그리고 2, 4, 6, 8 출력

```
for i in range(1,5):
    a.fd(150)
    a.left(90)
    a.print(i*2)
```

\* 선 색깔 바꾸기

```
a.pencolor("색")
```

\* 현재 펜 색 출력

```
a.pencolor()
```

\* 함수의 위치에 따른 결과 확인

\* or 연산 : 둘 중 하나만 참이어도 참

\* % 연산 (나머지 연산) : a % b의 나머지 값을 계산

2) 정사각형을 그릴 때, 세로선만 빨간색으로 짓기

① 직접 조건 내부에 지정

```
if (i==1 or i==3):
    a.pencolor("red")
else:
    a.pencolor("black")
```

② if의 드모르간 이용

```
if (not (i==1 and i!=3)):
    a.pencolor("red")
```

\* 드모르간 법칙

$$A \cup B = \overline{\overline{A} \cap \overline{B}}$$

③ 짝수를 관계로 확인

```
if (i % 2 == 1):
    a.pencolor("red")
else:
    a.pencolor("red")
```

\* 0 → false, 1 → true

\* # 주석처리

④ 짝수인 경우에 검은색으로 설정


```
if (i % 2 == 0):
    a.pencolor("black")
```

④-1

```
if (not (i % 2 != 0)):
    a.pencolor("black")
```

\* dot() : 그 자리에 점 찍기 함수  
 ↳ 차를 넣어 걸크 조정 가능

**16P 과제**




```
import turtle
a = turtle.Turtle()
for i in range(6):
    if (i % 3 == 0):
        a.pencolor("red")
    elif (i % 3 == 1):
        a.pencolor("green")
    else:
        a.pencolor("blue")
a.fd(100)
a.left(60)
```

**3) 두 객체가 반대 방향으로 동시에 전진**  
 (시분할 : 시간을 짧게 쪼개서 동시에 일어나게 함)

```
import turtle
a1 = turtle.Turtle()
a2 = turtle.Turtle()
a1.shape("turtle")
a2.shape("circle")
a1.dot(5)
a1.left(180)
for i in range(25):
    a1.fd(5)
    a2.fd(5)
```

**21P 과제**



```
import turtle
a1 = turtle.Turtle()
a1.pencolor("red")
a2 = turtle.Turtle()
a2.pencolor("blue")
for i in range(5):
    for s in range(100):
        a1.fd(1)
        a2.fd(1)
    a1.left(360/5)
    a2.right(360/5)
```

# 올바르지 못한 예

- 상단에 이름이 없습니다.

2주차  
컴퓨터 논리 표현방법  
설명

정답 = 0, 1 (거짓, 참) 오로지 이 둘에짐

AND = 교집합   OR = 합집합   NOT = 여집합 (거짓 논리)  
 논리식 =  $A \cdot B$     $A + B$     $\sim A (\bar{A})$   
 논리 =  $A \Rightarrow B \Rightarrow D \cdot F$     $A \Rightarrow F$     $A \Rightarrow F$

A	B	AND	OR	NOT	EX-OR	EX-NOR	NAND	NOR
0	0	0	0	1	0	1	1	1
0	1	0	1	1	1	0	1	0
1	0	0	1	0	1	0	1	0
1	1	1	1	0	0	1	0	0

진리표

$\Rightarrow D \Rightarrow D \Rightarrow D \Rightarrow D \Rightarrow D \Rightarrow D \Rightarrow D \Rightarrow D$

$A \cdot B \quad A + B \quad \sim A \quad A \oplus B \quad A \odot B \quad A \cdot \bar{B} \quad \bar{A} \cdot B$

기본오류   조합물 \* 공집합: 0   전체집합: 1

컴퓨터 논리 표현방법  
AND, OR, NOT (기본 논리)  
NAND, NOR (조합 논리)  
EX-OR, EX-NOR (조합 논리)

AND   OR   NOT

A	B	AND	OR	NOT	EX-OR	EX-NOR
0	0	0	0	1	0	1
0	1	0	1	1	1	0
1	0	0	1	0	1	0
1	1	1	1	0	0	1

논리식:  $A \cdot B$     $A + B$     $\sim A$     $A \oplus B$     $A \odot B$

논리회로:  $A \cdot B$     $A + B$     $\sim A$     $A \oplus B$     $A \odot B$

기본오류   조합물 \* 공집합: 0   전체집합: 1



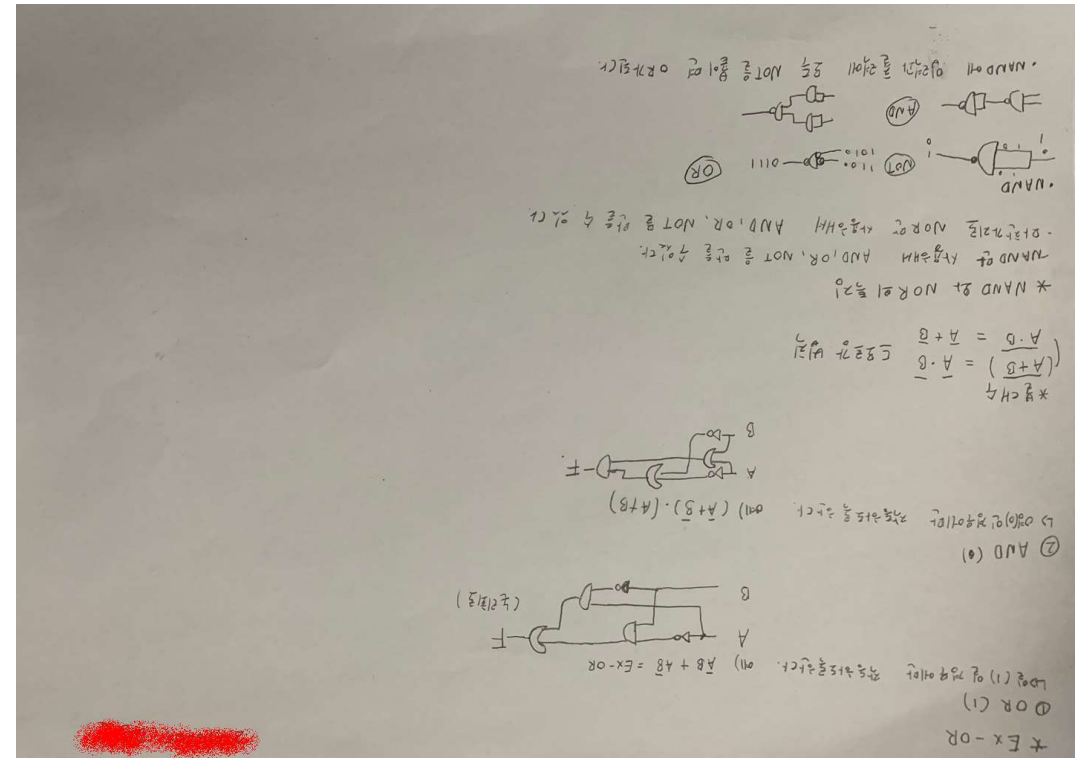
# 올바르지 못한 예

- 서술형으로 작성할 필요 없습니다.

3주차 수업을 2주차 수업때 배운 내용을 응용하여 도전해보는 것을 간단하게 나타낼수 있는 방법이다. 배웠습  
니다. 시작점을 만들거나 도착점 등 도착점을 만들때 turn이 90도를 돌면된다. 단지 몇번에 나누어 가는데 따라  
몇각형인지 정해진다. 저번 시간에 제가 도착점을 만들땐 `aaa.forward(150), a.rotate(90)`를 5번 반복했습니  
다. 여기서 72라는 숫자 대신 `angle`을 쓰고 `aaa.forward(150), angle=360/5`를 쓴다. 이것은 360도를 5번 나눠서 돌리  
는 것이다. 이 방법 훨씬 선택의 길이다. 150 대신 `step`라 하고 `aaa.forward(150), step=50`이라 적게 되면 `aaa.forward(step)`이  
전혀 5번씩 반복해진다. 여기서 `angle, step` 같은 것들은 변수라고 할수다. 만약 할작업을 만들수 있으면 `angle=360/7`  
이라 쓰면 됩니다. 그런데 기존에 `aaa.forward(step), a.rotate(angle)` 5번만 있기 때문에 이 상태로 진행하면  
선분 두개만 반복해진다. 그래서 `aaa.forward(step), a.rotate(angle)`를 3번씩 적어야 된다. 몇각형은 만드는  
대과 `aaa.forward(step), a.rotate(angle)`의 개수는 늘리거나 줄이거나 적기 때문에 불편하다. 그렇기 때문에 반복을  
사용하여 좀더 편하게 할수 있도록 만들것입니다. 반복문은 반복되는 대상을 `aaa.forward(step), a.rotate(angle)`  
을 어떤 수 만큼 반복을 하면됩니다. 프로그램은 위에서 아래로 실행이 되기 때문에 위에서 아래로 실행이 되  
가 `aaa.forward(step), a.rotate(angle)` 구간을 반복하면 됩니다. 반복되는 구간을 설정하기 위해 `for`이라는  
것을 사용합니다. `for kk in range(7):`를 하게 되면 구간이 7번 반복 하게 된다는 뜻입니다. 여기서  
`for kk in range(7)`과 `angle=360/7`에서 7개씩 `angle=360/cnt`에 `cnt=7`이라 쓰면  
중첩이 여차 반도의 `angle=360/cnt`에 `cnt`라는 변수가 됩니다. 만약 밑에다 사계 되면 프로그램이  
위에서 아래로 실행 되기 때문에 `angle=360/cnt`에서 `cnt` 값은 모르게 되어 오류가 발생할수다. 소괄에  
다라 계속 값이 변하라는 것 같다. 생각할 때 변수를 사용하는 것 훨씬 효율적 입니다. 만약 원본 만들고  
쓰면 `cnt=360`이라 쓰면 `angle=360/360`이기 때문에 turn이 1도 만큼 계속 돌면서 진행되어 다시 제자리로 오게 되는  
것이 됩니다. 여기서 `angle`은 함수인데 함수는 입력받은 값으로 돌아오는 형태입니다. 함수에 대해 추가적인  
설명 하려면 9번에 배운 프로그램도 보시면 됩니다. 키보드 `ctrl + /`를 같이 누르면 일련한것이 동작 원리가 각각  
#이 나타나게 됩니다. 이것을 주석이라 하고 #라는 문자를 사용해 설명하는 것을 하게 됩니다. 이렇게 주석을  
작성하고 설명하는 함수에 대해 만듭니다. 이항변 `myadd`라고 입력을 `aa, bb`라 하여 앞에 `def`라 적어  
`myadd(aa, bb)`가 함수라는 것을 표현해 줍니다. 그 다음 `return aa+bb`라 적습니다. 다른 `aa, bb`라 값들 대신  
= 값을 바꿀수도 출력해라는 의미입니다. 이제 `myadd`라는 함수를 불러와 `aa, bb` 대신 3, 4를 주고 `myadd(3, 4)`의 이름은  
크로 정해 줍니다. 그 다음 `print()`를 입력하고 실행을 시키면 프로그램이 3과 4를 더한 결과값이 나타나게 됩니다.  
아래 다시 본으로 돌아와서 turn에 따라 함수는 `(turn + 1)`를 늘려 확실하게 시킵니다. `range`라는 함수는  
`range(360)`라는 수가 들어 가면 0부터 359까지 자연수 360개를 같은 배구를 만드는 역할을 합니다. 360개의 자연수는  
444씩 씩 `aaa.forward(step), a.rotate(angle)`에 사용하라는 것입니다. 이렇게 액티 차례대로 한계까지  
반복하는 일을 합니다. 그래서 `aaa.rotate(angle)` 바로 밑에 문장에 `print(kk)`를 입력하고 실행하면  
오른쪽에 0부터 359까지 `kk`가 등장하게 됩니다. 대략을 보는 이유는 그림을 그려면서 중간중간에 숫자나  
결과물을 확인하고 쓸수있게 사용하고 이를 디버깅이 시나 합니다.

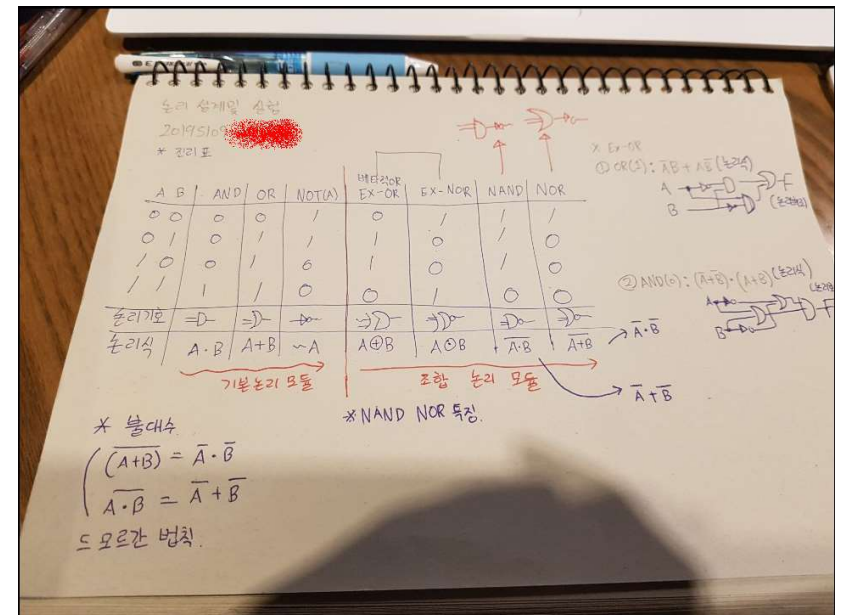
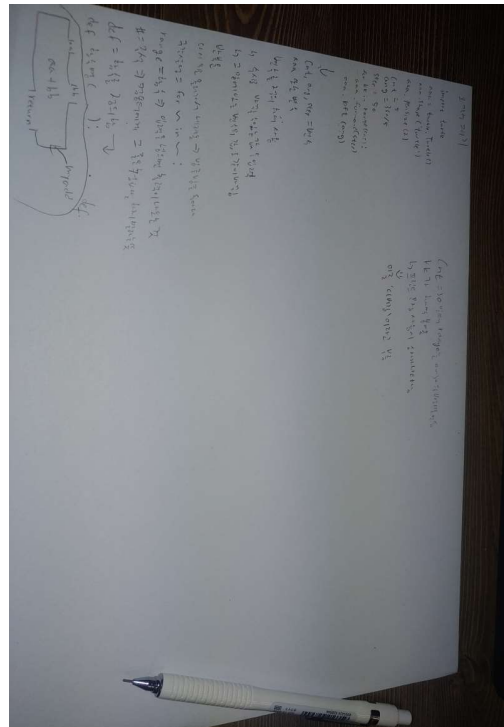
# 올바르지 못한 예

- 세로쓰기가 기본입니다.
- 이미지가 뒤집혀 있습니다.



# 올바르지 못한 예

- 촬영 각도를 평행하게 유지해 주세요



# 올바르지 못한 예

- 한장이미지에 두페이지를 촬영하지 마세요
- 각각 촬영해서 PDF 에서 합치면 됩니다.

1. **대방기** 프로그밍에서 변수의 크기는 프로그램 실행 시점에 결정된다.  
 2. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 3. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 4. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 5. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 6. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 7. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 8. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 9. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.  
 10. **대방기** 변수의 크기는 프로그램 실행 시점에 결정된다.

DATE. NO.

다비경 - 프로그램 내에서 변수들의 변화를 관망해보고, 수정할수있는 과정

```

import turtle
aaa = turtle.Turtle()
for kk in range(4):
    aaa.forward(150)
    aaa.left(90)
    print(kk)
        
```

→ 여기서 **출력된 (count) 0, 1, 2, 3**의 순서를 3, 2, 1, 0로 만드는 방법 (순서 역으로 만들기)

1번의 방법  

```

print(kk)를 print(4-kk)로 바꾸기.
        
```

이 모든 과정 = **다비경**

2번의 방법  

```

for kk in range(4): 를
for kk in reversed(range(4)):로 바꾸기.
        
```

→ **그 밖에도 (count) 0, 2, 4, 6, 8**으로 만드는 방법

3번  

```

print(kk)를 print(2*kk)로 바꾸기.
        
```

→ **이러한 모든 다비경**

\* **중요**의 위치 (순서) 유지하기

예) 순서가  
 - 앞도 나비여  
 - 뒤로도 회전하여  
 - 생장하기  
 → **원래의 순도**에 **예외**를 **바꿔주면**  
 생장하기"를 **직접** **정답**에 **바꿔주기**

DATE. NO.

만약 예이 이렇게 세로로만 색깔을 바꿔서 하고 싶다면?

```

import turtle
aaa = turtle.Turtle()
for kk in range(4):
    aaa.forward(150)
    aaa.left(90)
    print(kk)
        
```

→ 여기서 세로로만 색깔을 바꿔서 하고 싶다면? **KK의 값은 1, 3이다.**  
 만약 1, 3이 다른때만 색깔을 변경해주면 된다

→ **뒤로에 맞게 수정**

```

for kk in range(4):
    if kk == 1 or kk == 3:
        aaa.pencolor('red')
        aaa.forward(150)
        aaa.left(90)
    else:
        aaa.pencolor('black')
        aaa.forward(150)
        aaa.left(90)
        
```

다반한리 줄어는법 (공백을 빼주세요)

```

for kk in range(4):
    # kk == 1 or kk == 3:
    aaa.pencolor('red')
    aaa.forward(150)
    aaa.left(90)
    else:
        aaa.pencolor('black')
        aaa.forward(150)
        aaa.left(90)
        
```

→ **이렇게도 해볼수**

DATE. NO.

꼭대기 2가지 줄어이기

aaa.det(5) → aaa.4번째 줄어이기

```

import turtle
aaa1 = turtle.Turtle()
aaa2 = turtle.Turtle()
aaa1.forward(150)
aaa2.forward(150)
        
```

→ **aaa1이 먼저 돌리는데, aaa2가 나중에 돌림**

동시에 돌리려면 =>  
 (변환한, 돌리려 하는 순서로 바꿔기)

```

import turtle
aaa1 = turtle.Turtle()
aaa2 = turtle.Turtle()
aaa2.left(90)
aaa1.forward(150)
for kk in range(150):
    aaa1.forward(1)
    aaa2.forward(1)
        
```

### 숙제1.

```

import turtle
aaa = turtle.Turtle()

for kk in range(6):
    - if kk == 0 or kk == 3:
        - aaa.pencolor('red')
    - else:
        - if kk == 1 or kk == 4:
            - aaa.pencolor('green')
        - else:
            - aaa.pencolor('blue')

aaa.forward(100)
aaa.left(60)
        
```

### 숙제2.

```

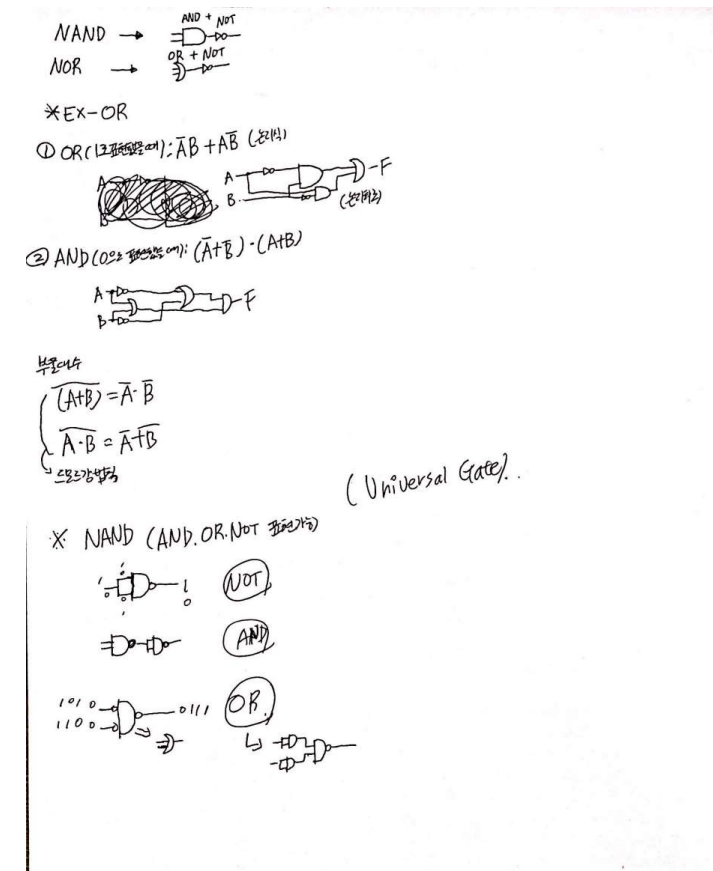
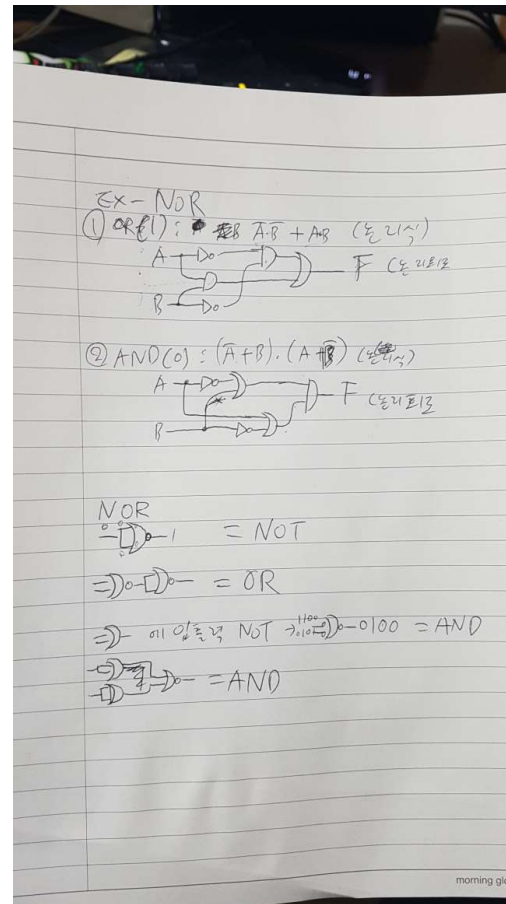
import turtle
aaa1 = turtle.Turtle()
aaa2 = turtle.Turtle()

aaa1.pencolor('red')
aaa2.pencolor('blue')

for kk in range(1, 501):
    aaa1.forward(1)
    aaa2.forward(1)
    if kk % 100 == 0:
        aaa1.left(92)
        aaa2.right(92)
        
```

# 올바르지 못한 예

- 노트 내용이 부실하네요.



## 올바르지 못한 예 (파일명 작성)

- week3주\_20141234\_홍길동.pdf
  - → week3\_20141234\_홍길동.pdf (올바른 예)
- 20201234\_122957.pdf
  - 파일명 앞에 week 없음
  - 파일명에 이름 없음