제공 모듈을 이용한 코딩 과정에 대한 이해

[코딩 과정] 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화

> 알고리즘이 코드로 구현되면서 일반화, 구조화 되어가는 과정을 이해

> 외부 패키지 모듈을 임포트(import) 하여 사용하는 개념 이해

> 변수 사용하는 이유 이해

> 패턴을 찾아 코드를 일반화 하는 개념 이해

> 특정 코드를 함수로 독립시켜 재사용하는 개념 이해

[A] 제공되는 패키지 모듈 사용

> import 모듈명 # 손쉽게 그래픽 구현이 가능한 함수 모듈을 빌려서 사용

In [7]:
```
## [A-1] turtle 모듈 임포트
import turtle

################### 메인 시작 ############
turtle.shape('turtle')  #Turtle 모양

turtle.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [19]:
```
## [A-2] turtle 모듈 임포트: 모듈 함수 사용
import turtle as t  #별명을 r로 사용

################### 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.forward(100)  #앞으로 100 pixcel 이동

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [3]:
```
## [A-3] turtle 모듈 임포트: 모듈 함수 사용
import turtle as t  #Turtle 모양

################### 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.forward(100)  #앞으로 100 pixcel 이동
t.left(90)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

[B] 4-각형 그리기 작업 절차 구성

> '전진 100 > 좌로 회전 90도'를 4번 반복

In [1]:
```
## [B-1] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리rl: 작업 절차 구성
import turtle as t

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.forward(100)  #앞으로 100 pixcel 이동
t.left(90)  #좌로 90도 회전
t.forward(100)  #앞으로 100 pixcel 이동
t.left(90)  #좌로 90도 회전
t.forward(100)  #앞으로 100 pixcel 이동
t.left(90)  #좌로 90도 회전
t.forward(100)  #앞으로 100 pixcel 이동
t.left(90)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

[C] 변수 사용

> 변수(Variable)는 명령어가 실행되는 동안 사용할 값을 임시적으로 저장해 놓는 메모리 공간 이름

> 상수(Constant) 대신 변수를 사용함으로서 알고리즘의 패턴을 얻을 수 있고, 일반화가 가능하다.

In [27]:
```
## [C-1] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용
import turtle as t

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

step = 100
angle = 90

t.forward(step)  #앞으로 100 pixcel 이동
t.left(angle)  #좌로 90도 회전
t.forward(step)  #앞으로 100 pixcel 이동
t.left(angle)  #좌로 90도 회전
t.forward(step)  #앞으로 100 pixcel 이동
t.left(angle)  #좌로 90도 회전
t.forward(step)  #앞으로 100 pixcel 이동
t.left(angle)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

[D] 코드의 일반화

> 변수(Variable)를 사용함으로서 동일 알고리즘(패턴)으로 다양의 결과를 얻을 수 있다.

In [29]:
```
## [D-1] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성
import turtle as t
```

```
################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

step = 100
angle = 90

for _ in range(4):  #4회 반복
    t.forward(step)  #앞으로 100 pixcel 이동
    t.left(angle)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [33]:
```
## [D-2] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화
import turtle as t

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

n = 4
step = 100
angle = 360 // n  # //는 나눈 몫

for _ in range(n):  #n회 반복
    t.forward(step)  #앞으로 100 pixcel 이동
    t.left(angle)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [35]:
```
## [D-3] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화
## 키보드 입력 처리
import turtle as t

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

n = int(input(">몇 각형(3~n)? "))  #키보드 입력 처리
step = 100
angle = 360 // n  # //는 나눈 몫

for _ in range(n):  #n회 반복
    t.forward(step)  #앞으로 100 pixcel 이동
    t.left(angle)  #좌로 90도 회전

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

[E] 함수를 사용한 구조화

> 함수(**Function**)는 유사 기능을 하는 명령어들을 묶어놓은 모듈이다.

> 필요 시마다 함수를 호출하여 사용할 수 있다.

> 함수의 재사용성으로 프로그램 구조가 단순화 된다.

In [1]:
```
## [E-1] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수를 사용한 구조화
import turtle as t

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

n = int(input(">몇 각형(3~n)? "))
step = 100
render(n, step)  #함수 호출

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [3]:
```
## [E-2] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수의 재사용
import turtle as t

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

while True:  #무한 반복: 반복 조건 True
    n = int(input(">몇 각형(3~n)? "))
    if n < 3:  #반복 탈출 조건
        break  #반복 탈출
    step = 100
    render(n, step)  #함수 호출

t.exitonclick()  # 실행 창을 닫지 않도록
################## 메인 끝 ##############
```

In [1]:
```
## [E-3] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 반복문 재구성(1)
import turtle as t

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

n = int(input(">몇 각형(3~n)? "))
```

```
while True:  #무한 반복: 반복 조건 True
    if n < 3:  #반복 탈출 조건
        break  #반복 탈출
    step = 100
    render(n, step)  #함수 호출
    n = int(input(">몇 각형(3~n)? "))

t.exitonclick()  # 실행 창을 닫지 않도록
################# 메인 끝 ##############
```

In [5]:
```
## [E-4] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 반복문 재구성(2)
import turtle as t

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

n = int(input(">몇 각형(3~n)? "))
while n >= 3:  #무한 반복: 반복 조건 n >= 3
    step = 100
    render(n, step)  #함수 호출
    n = int(input(">몇 각형(3~n)? "))

t.exitonclick()  # 실행 창을 닫지 않도록
################# 메인 끝 ##############
```

[F] 프로그램 업그레이드

>> 마우스 클릭에 반응하는 함수 추가

> 프로그램에 새로운 기능을 추가한다는 것은 주로 필요한 함수를 추가 생성 후 호출로 사용한다.

> 즉, 프로그램의 메인(main) 부분은 작업의 절차에 따른 함수 호출 구문 위주로 구성되게 된다.

In [10]:
```
## [F-1] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수 사용: 마우스 클릭에 반응하는 함수 > left_click()
import turtle as t
import random

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

def left_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    t.goto(x, y) #(x, y) 좌표로 이동
```

```
################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.onscreenclick(left_click, 1)  #마우스 왼쪽 버튼(1번) 클릭 시 함수(left_click() 호출

t.done()  #계속 동작
################## 메인 끝 ##############
```

In [6]:
```
## [F-2] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수 사용: 마우스 클릭에 반응하는 함수 > left_click()
import turtle as t
import random

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

def left_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    t.goto(x, y) #(x, y) 좌표로 이동

def mid_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    pass  #아무 일 안함

def right_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    pass  #아무 일 안함

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.onscreenclick(left_click, 1)  #마우스 왼쪽 버튼(1번) 클릭 시 함수(left_click() 호출
t.onscreenclick(mid_click, 2)  #마우스 가운데 버튼(2번) 클릭 시 함수(mid_click() 호출
t.onscreenclick(right_click, 3)  #마우스 오른쪽 버튼(3번) 클릭 시 함수(right_click()

t.done()  #계속 동작
################## 메인 끝 ##############
```

In [14]:
```
## [F-3] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수 사용: 마우스 클릭에 반응하는 함수 > mid_click()
import turtle as t
import random

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

def left_click(x, y):  #마우스 클릭 위치 (x, y)로 이동
    t.goto(x, y) #(x, y) 좌표로 이동

def mid_click(x, y):  #마우스 클릭 위치 (x, y)로 펜 들고 이동
    t.penup()    #펜 들기
    t.goto(x, y) #(x, y) 좌표로 이동
    t.pendown()  #펜 내리기

def right_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    pass  #아무 일 안함

################## 메인 시작 ############
```

```
t.shape('turtle')  #Turtle 모양

t.onscreenclick(left_click, 1)  #마우스 왼쪽 버튼(1번) 클릭 시 함수(left_click() 호출
t.onscreenclick(mid_click, 2)  #마우스 가운데 버튼(2번) 클릭 시 함수(mid_click() 호출
t.onscreenclick(right_click, 3)  #마우스 오른쪽 버튼(3번) 클릭 시 함수(right_click()

t.done()  #계속 동작
################## 메인 끝 ##############
```

In [12]:
```
## [F-4] turtle 모듈 임포트: 모듈 함수 사용
## turtle이 4각형 그리기: 작업 절차 구성 > 변수 사용 > 패턴 구성 > 일반화 > 구조화
## 함수 사용: 마우스 클릭에 반응하는 함수 > right_click()
import turtle as t
import random

def render(n, step):
    angle = 360 / n
    for _ in range(n):  #n번 반복
        t.forward(step) #90pixcel 전진
        t.left(angle)  #좌로 90도 회전

def left_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    t.goto(x, y) #(x, y) 좌표로 이동

def mid_click(x, y):  #마우스 클릭 위치 (x, y)로 펜 들고 이동
    t.penup()    #펜 들기
    t.goto(x, y) #(x, y) 좌표로 이동
    t.pendown()  #펜 내리기

def right_click(x, y):  #마우스 클릭 위치 좌표값 x, y
    t.penup()    #펜 들기
    t.goto(x, y) #(x, y) 좌표로 이동
    t.pendown()  #펜 내리기
    n = random.randrange(3, 8)  #랜덤 수 발생
    step = random.randrange(10, 200)
    render(n, step)

################## 메인 시작 ############
t.shape('turtle')  #Turtle 모양

t.onscreenclick(left_click, 1)  #마우스 왼쪽 버튼(1번) 클릭 시 함수(left_click() 호출
t.onscreenclick(mid_click, 2)  #마우스 가운데 버튼(2번) 클릭 시 함수(mid_click() 호출
t.onscreenclick(right_click, 3)  #마우스 오른쪽 버튼(3번) 클릭 시 함수(right_click()

t.done()  #계속 동작
################## 메인 끝 ##############
```

In [ ]: