

공학적 문제해결 기법

13주차 2차시

그리기 코드 다시 사용하기





13주차 2차시 : 그리기 코드 다시 사용하기

I. 개요

색으로 채워진 도형그리기

II. 학습 개요

1) 학습 목표

펜 색상을 바꾸어 그려진 도형에 색을 채워보고, 함수를 한 번 호출해 다양한 색상과 모양으로 쉽게 도형을 그릴 수 있도록 그리기 코드를 재사용해 본다.

2) 학습 목차(세부 목차)

- 색칠하기
- 색으로 채워진 정사각형 그리기
- 색으로 채워진 별 그리기



13주차 1차시 : 기하학적 도형 그리기

III. 학습 1 - 색칠하기

color 함수

- ✓ 그림에 색상을 추가하는 기능들을 살펴볼까요?
- ✓ color 함수는 세 개의 매개변수를 사용
- ✓ 첫 번째는 빨간색, 두 번째는 녹색, 세 번째는 파란색의 양을 지정
- ✓ 100%의 빨간색 펜을 사용한다면, `color(1, 0, 0)`을 사용
- ✓ 빨간색, 녹색 및 파란색의 조합을 RGB(빛의 3원색이라고도 함)라고 함
- ✓ 이는 색상이 컴퓨터 모니터에 표시되는 방식
- ✓ 자주색: 파란색 + 빨간색
- ✓ 주황색: 노란색 + 빨간색
- ✓ 빨간색, 녹색, 파란색은 다른 색상과 혼합해 만들 수 없기 때문에 기본색이라 함



13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 1 - 색칠하기

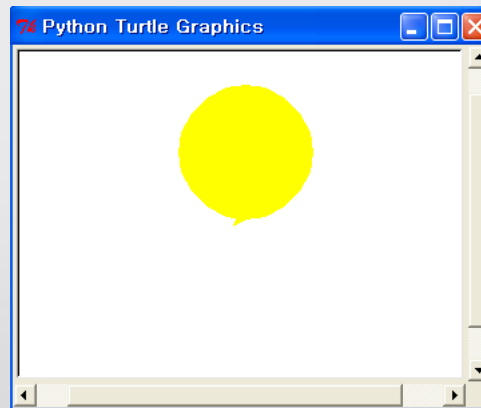
color 함수

- ✓ 거북이가 노란색 원을 그리려면?
- ✓ 파란색을 제외한 빨간색과 녹색을 둘 다 100% 사용해야 함

➔

```
>>> t.color(1,1,0)    빨간색과 녹색 100%이고, 파란색은 0%  
>>> t.begin_fill()  
>>> t.circle(50)  
>>> t.end_fill()
```

- ✓ `t.end_fill()`은 RGB 색상으로 원을 채우라고 하는 것





13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 1 - 색칠하기

색으로 채워진 원을 그리기 위한 함수

☒ 다른 색상으로 실습을 쉽게 할 수 있도록 색으로 채워진 원을 그리는 코드는?

➡

```
>>> def mycirde(red, green, blue):  
    t.color(red, green, blue)  
    t.begin_fill()  
    t.cirde(50)  
    t.end_fill()
```

☒ 위 코드로 빨간색 50%의 원을 그린다면?

➡

```
>>>> mycirde(0.5, 0, 0)
```



13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 1 - 색칠하기

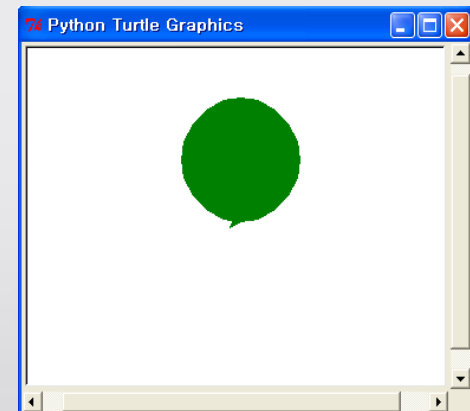
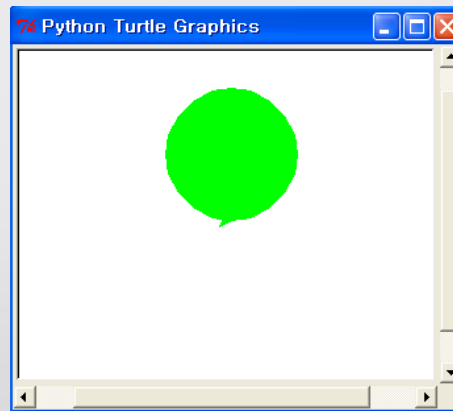
색으로 채워진 원을 그리기 위한 함수

- ☒ 녹색페인트만을 사용해 밝은 녹색 원을 그린다면?

➡ `>>> mycircle(0, 1, 0)`

- ☒ 녹색페인트를 절반(0.5)만 사용해 어두운 녹색 원을

➡ `>>>> mycircle(0, 0.5, 0)`





13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 1 - 색칠하기

색으로 채워진 원을 그리기 위한 함수

☒ 빨강과 녹색 그리고 파랑의 다양한 조합으로 여러가지 다양한 색상들을 만들 수 있다.

☒ 다음을 입력하고 색상을 확인해보라.

➡ `>>>>mycirde(0.9,0.75,0)`
`>>>>mycirde(1,0.7,0.75)`
`>>>>mycirde(1,0.5,0)`
`>>>>mycirde(0.9,0.5,0.15)`



13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 1 - 색칠하기

검정색과 흰색 만들기

✓ 빛이 없으면 색상도 없다.

✓ 모든 기본 색상이 0으로 지정된 원은 검정색

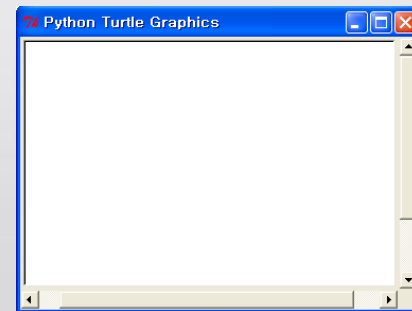
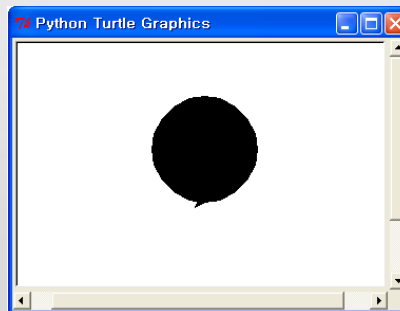
➡ `>>>> mycirde(0,0,0)`

✓ 세가지 기본 색상을 100%로 사용하는 경우에는 그 반대 색이 된다. → 흰색

✓ 흰색으로 그려진 원은 배경색이 같은 흰색이어서 그려도 보이지 않게 됨

✓ 검정색 원을 지우는 효과

➡ `>>>> mycirde(1,1,1)`





13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 2 - 색으로 채워진 정사각형 그리기

매개변수를 이용한 사각형의 크기 조절

- ✓ 앞서 익혔던 for 루프문으로 사각형을 그리는 코드

➡

```
>>> t.reset()
>>> for x in range(1,5):
    t.forward(50)
    t.left(90)
```

- ✓ 이 코드를 수정하여 시작 부분에서 매개변수를 이용해 사각형의 크기를 조절

➡

```
>>> def mysquare(size):
    for x in range(1,5):
        t.forward(size)
        t.left(90)
```

- ✓ 사이즈 50으로 함수를 호출 ->

```
>>> mysquare(50)
```

- ✓ 작은 정사각형이 만들어짐



13주차 2차시 : 그리기 코드 다시 사용하기

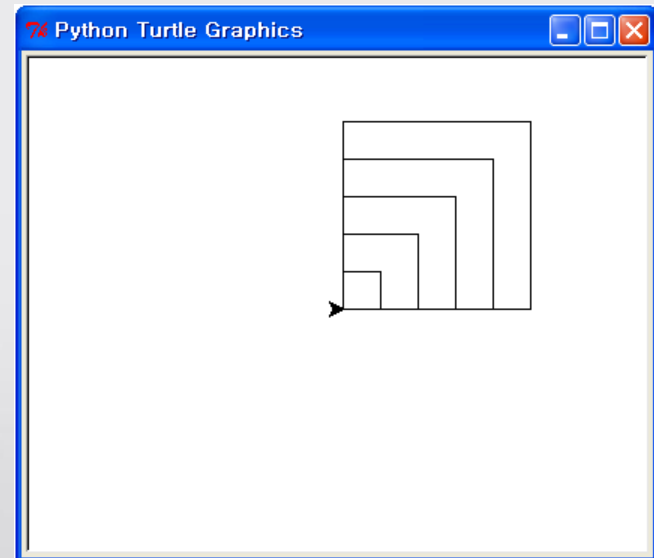
III. 학습 2 - 색으로 채워진 정사각형 그리기

매개변수를 이용한 사각형의 크기 조절

- ✓ 이제 다른 크기로 함수를 실행해 본다.
- ✓ 25, 50, 75, 100, 125의 크기로 정사각형을 만들



```
>>> t.reset()  
>>> mysquare(25)  
>>> mysquare(50)  
>>> mysquare(75)  
>>> mysquare(100)  
>>> mysquare(125)
```





13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 2 - 색으로 채워진 정사각형 그리기

색으로 채워진 정사각형

- ☒ 먼저, 캔버스를 리셋하고 색칠을 시작한 후 정사각형 함수를 다시 호출

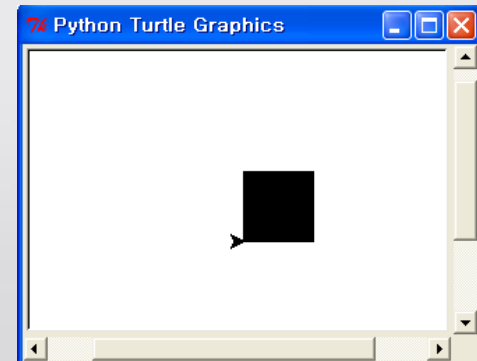


```
>>>t.reset()  
>>>t.begin_fill()  
>>>mysquare(50)
```

- ☒ 다음의 코드로 채우기를 끝내지 않으면 색이 칠해지지 않은 정사각형이 보임



```
>>>tend_fill()
```





13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 2 - 색으로 채워진 정사각형 그리기

색으로 채워진 정사각형

- ☒ 이 코드를 변경해서 색이 채워지거나 비어있는 정사각형 그리기
- ☒ 이를 위해 또 다른 매개변수와 조금 더 복잡한 코드가 필요



```
>>>t.begin_fill()  
>>>def mysquare(size, filled):  
    if filled == True:  
        t.begin_fill()  
    for x in range(0,4):  
        t.forward(size)  
        t.left(90)  
    if filled == True:  
        t.end_fill()
```



13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 2 - 색으로 채워진 정사각형 그리기

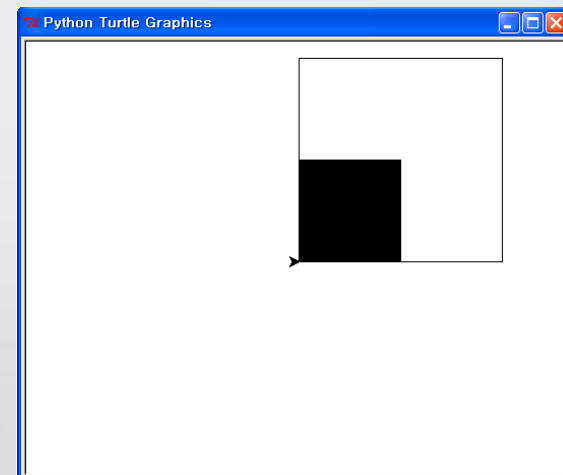
색으로 채워진 정사각형

☒ 다음 코드와 같이 색이 채워진 정사각형을 그릴 수 있다.

➡ `>>> mysquare(100, True)`

☒ 아니면 다음과 같이 색이 채워지지 않은 정사각형을 그릴 수도 있다.

➡ `>>> mysquare(200, False)`







13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 3 - 색으로 채워진 별 그리기

별에 색 추가하기

 앞서 그렸던 코드



```
>>> for x in range(1, 19):  
    t.forward(100)  
    if x % 2 == 0:  
        t.left(175)  
    else:  
        t.left(225)
```



13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 3 - 색으로 채워진 별 그리기

별에 색 추가하기

- ✓ mystar 함수를 만들고, if 문을 사용, 매개변수 size를 추가

➡

```
>>> def mystar(size, filled):  
    if filled == True:  
        t.begin_fill()  
    for x in range(1, 19):  
        t.forward(size)  
        if x % 2 == 0: t.left(175)  
        else: t.left(225)  
    if filled == True:  
        t.end_fill()
```

- ✓ filled가 True인지 확인하고, True이면 색 채우기를 시작
- ✓ 마지막 두 행에서 다시 확인하고, filled가 True이면 색 채우기를 끝냄

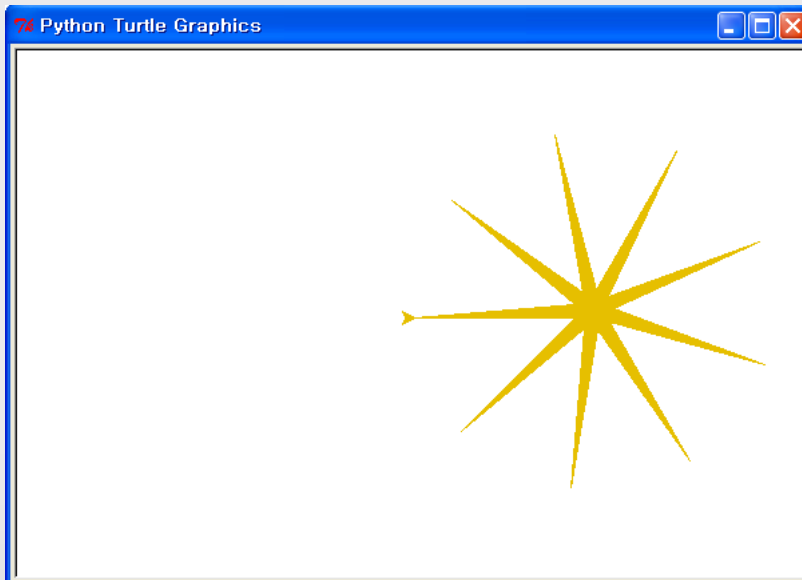


13주차 2차시 : 그리기 코드 다시 사용하기

III. 학습 3 - 색으로 채워진 별 그리기

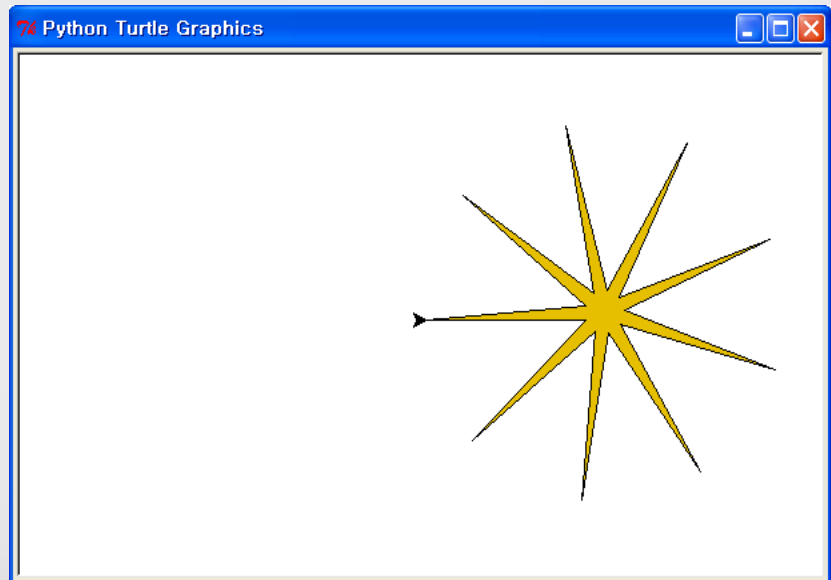
별에 색 추가하기

외곽선을 추가하기 위해 색상을 검정색으로 바꾸고 채우기 없는 별을 그리면, 다음과 같이 외곽선이 있는 금색 별이 됨



→

```
>>>t.color(0.9,0.75,0)
>>>mystar(120,True)
```



→

```
>>>t.color(0,0,0)
>>>mystar(120,False))
```



『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』

