

02 집합

- ◎ 학습목표 • 조건을 추가하여 원하는 해집합을 구할 수 있다.
- 집합을 활용해 다양한 구조물을 만들 수 있다.

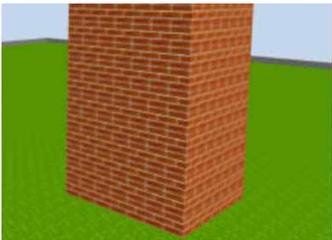
정/직/구/원 code + 조건추가

생각 열기

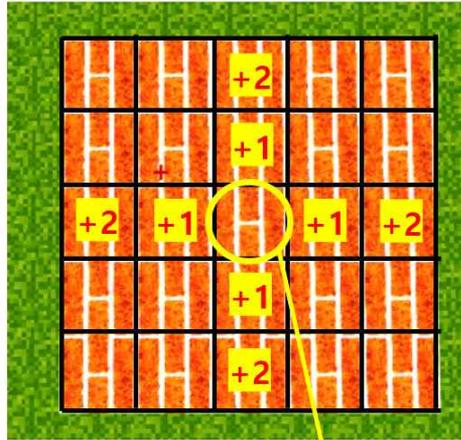
지난 시간에 우리는 집합의 개념에 대해 학습해 보았고 조건들을 추가할수록 원소들이 어떻게 변하는지 살펴보았다.(해집합) 또한 정사각기둥과 원기둥을 만드는 코드에 조건들을 추가하며 자유자재로 위치와 모양을 변형해보았다. 이와 더불어 직사각기둥과 구를 만드는 코드를 더 소개하고 &&(and)로 조건을 추가하여 다양한 구조물들을 만들어 보자.

1. 집합으로 간단한 도형 만들기 : 정/직/구/원 !

지난 시간에 학습했던 정사각기둥과 원기둥 명령어를 복습해보자.

<pre>1 beginxyz 2 집합{정(a, b, c)}</pre>	<pre>1 beginxyz 2 집합{원(a, b, c)}</pre>
	
<p>- 정사각 기둥 중심이 (a, b)이고 중심으로부터 상하로 c만큼, 좌우로 c만큼씩 커짐</p>	<p>-원 기둥- 중심: (a, b) 반지름 길이 : c</p>

예를 들어, 집합{원(10, 10, 2)&&z==1}은 아래의 그림과 같음을 알 수 있다.



중심: (10, 10)

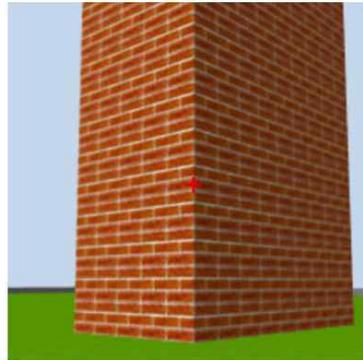
이제 직사각기둥과 구를 만들어주는 명령을 살펴보자. 특히 명령어를 입력할 때 고려해야하는 사항이 4가지로 늘어났음을 주의하고 각각의 기능을 정확하게 숙지해야한다.

알아봅시다!

① 직(a,b,c,d)의 이해를 돕기 위해 정(10, 10, 3)은 직(10, 10, 3, 3)과 같음을 기억하자. (직사각형은 가로와 길이와 세로의 길이가 달라도 됨)

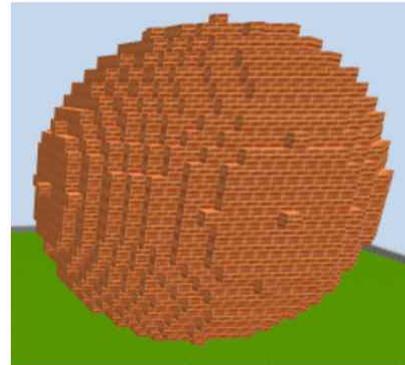
② 터틀크래프트의 좌표계는 정수를 사용하기 때문에 원기둥의 경우처럼 구의 모양이 예쁘지 않을 때에는 크기를 키우면 좀 더 매끄러워진다.

```
1 beginxyz
2 집합{직(a, b, c, d)}
```



- 직사각 기둥
중심이 (a, b)이고
가로가 중심으로부터 c만큼,
세로가 중심으로부터 d만큼씩 커짐

```
1 beginxyz
2 집합{구(a, b, c, d)}
```



- 구
중심이 (a, b, c)이고 반지름이 d

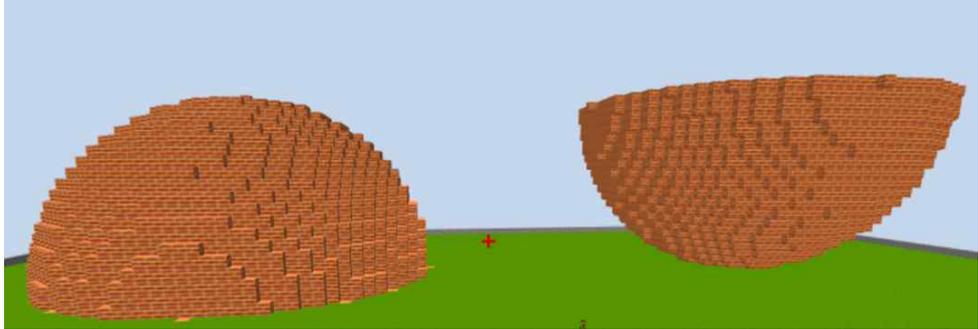
직(a, b, c, d)에서 a와 b는 x, y 좌표를 뜻하고 c와 d는 가로, 세로의 길이와 관련이 있는 한편 구에서는 a, b, c가 x, y, z(높이/층) 좌표를 뜻하고 d만 길이(반지름)와 관련이 있음을 유의하자.

활동① 다음은 반구(구의 절반)의 모습이다. 다음 코드의 ??? 부분을 고치거나 조건들을 추가하여 반구 2개를 만들어보자.
(위치와 반지름은 바꾸지 마세요.)

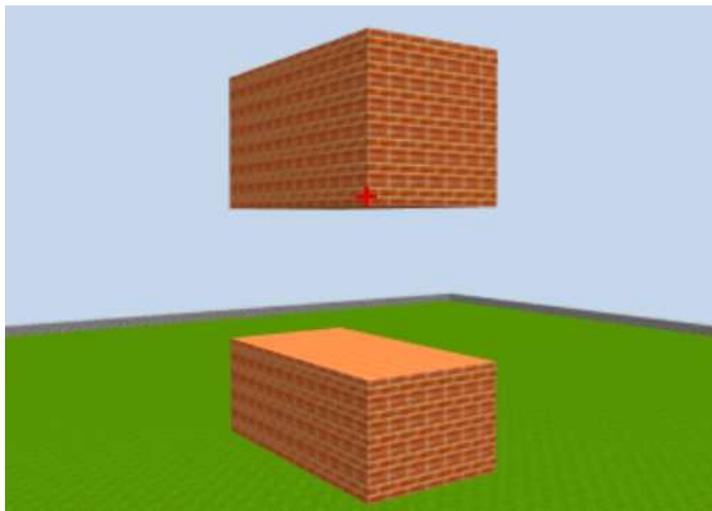
```

1 beginxyz
2 집합{구(30, 30, ???, 20)}
3 집합{구(-30, 30, ???, 20)}

```



활동② 아래의 그림과 같이 **집합{직(a, b, c, d)}** 명령어를 사용하여 두 개의 직육면체를 만들어보자. 만약 조건을 한 번에 다 적용하려고 했다면, 안 되는 이유를 댓글로 설명해보자.(집합의 개념)



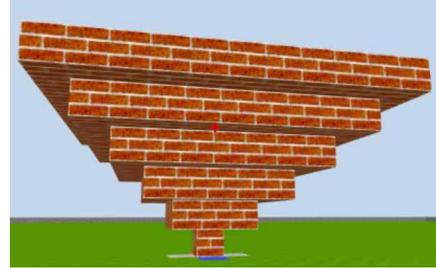
2. 정/직/구/원의 확장

우리는 앞서 정, 직, 구, 원의 명령어에 조건을 추가하여 다양한 도형들을 만들어 낼 수 있음을 살펴보았다. 그런데 꼭 a, b, c, d에는 숫자만 넣어야 할까? 좀 더 깊은 탐구를 위해 아래의 예시를 살펴보자. 아래의 코드를 실행하면 다음과 같이 피라미드를 뒤집어 놓은 모양이 나온다.

```

1 beginxyz
2 집합{정(0, 0, 5)&& z==6}
3 집합{정(0, 0, 4)&& z==5}
4 집합{정(0, 0, 3)&& z==4}
5 집합{정(0, 0, 2)&& z==3}
6 집합{정(0, 0, 1)&& z==2}
7 집합{정(0, 0, 0)&& z==1}

```



알아봅시다!

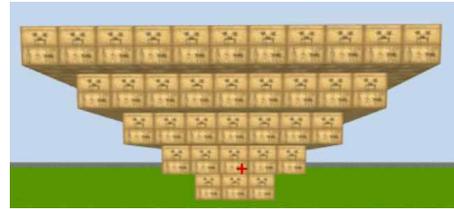
오른쪽 그림을 보면
아이템이 변한 것을
알 수 있다.
그 이유는 코드의
두 번째 줄 마지막에
빨간 부분 때문이다.
: 뒤에 숫자는
아이템의 번호로써
앞으로 편하게
아이템을 바꿀 때
해당 명령어를
사용해보자.

코드를 자세히 살펴보면 무언가 규칙성이 보이는 것 같다. 다음의 코드를 살펴보자.

```

1 beginxyz
2 집합 정(0, 0, z) && z<6 ; 16 }
3

```

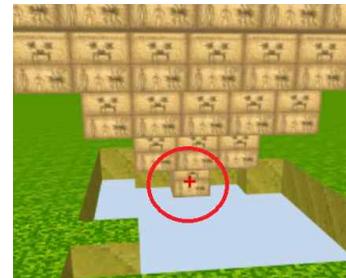


알아봅시다!

alt키를 누른 채
마우스 왼쪽을
누르면 구멍을 뚫을
수 있다.

집합 {정(0, 0, z) && z<6} 명령이 실행될 때에는 $z=-63$ 부터 $z=63$ 까지 한 층씩 올라가면서 각 z 층에 정(0, 0, z) 조건에 맞는 정사각형을 만든다. $z<6$ 조건을 각각 살펴보자. $z=5$ 는 5층(위의 그림의 맨 윗층)을 뜻하고 정(0, 0, z)에 $z=5$ 를 대입하면 정(0, 0, 5)이므로 5층에서는 중심으로부터 길이가 5씩 늘어난 정사각형이 만들어진다. 같은 방법으로 4층에는 정(0, 0, 4), 3층에는 정(0, 0, 3), ... , 1층에는 정(0, 0, 1)의 형태로 나타나게 되는 것이다. 따라서 1층에서는 중심으로부터 1씩 늘어난 길이 3의 정사각형 모양이 형성되었다.

한편, $z=0$ 즉 0층인 경우에는 정(0, 3, 0)조건에 의해 큐브 하나가 생기지만 0층이기에 땅을 파보아야 알 수 있다. 또한 z 가 음수인 경우에는 정(0, 0, z)에서 길이가 음수일 수 없으므로 더 이상 큐브가 생기지 않는다.



< 피라미드 >

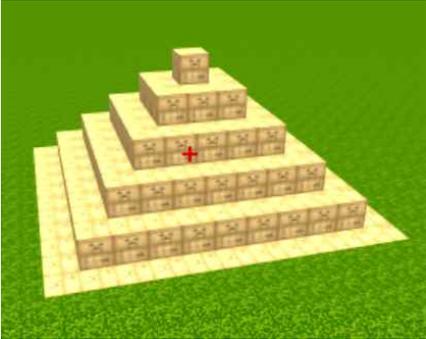


그렇다면 뒤집히지 않은 피라미드는 어떻게 만들 수 있을까? 앞서 살펴 보았듯이 $z=-63$ 부터 $z=63$ 까지 한 층 한 층 다 적용되는 것을 활용해 보면 단서를 찾을 수 있다. 다음의 코드를 살펴보자.

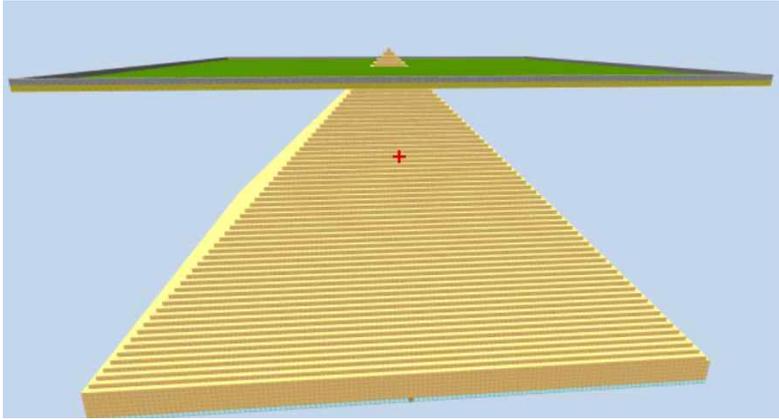
```

1 beginxyz
2 집합{ 정(0, 0, 5-z) ; 16 }
3

```



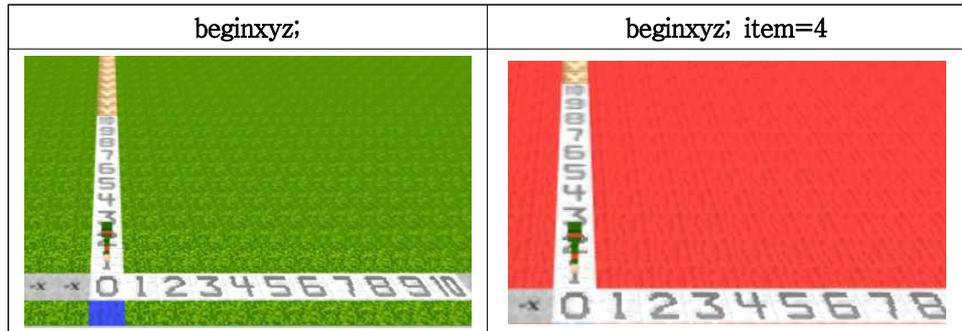
$z>6$ 인 경우에는 정(0, 0, 5-z)에 z 를 대입하면 길이가 음수가 나오므로 불가능하기에 $z\leq 5$ 인 경우를 살펴보자. $z=5$ 이면 정(0, 0, 0)의 5층의 모양, 즉 큐브 1개가 생긴다. 마찬가지로 $z=4$ 이면 정(0, 0, 1)이 되어 중심으로부터 1칸씩 늘어난 가로세로 3칸의 정사각형이 4층에 생기고 같은 방법으로 $z=1$ 까지 이해할 수 있다. $z=0$ 인 경우에는 정(0, 0, 5)에서 $1+(5\times 2)=11$, 즉 가로세로 11칸의 정사각형이 0층에 만들어져 위와 같은 결과가 나오게 된 것이다. 피라미드가 뒤집힌 경우와 다르게 이번 경우에는 z 가 음수여도 정(0, 0, 5-z)에서 길이가 음수가 나오지 않으므로 가능함을 예측할 수 있다. 정말로 그러할까?



$z=-63$ 부터 $z=63$ 까지 모든 z 에 대하여 실행한 것이기에 당연한 결과이고, 지상세계에서만 피라미드를 만들기 위해서는 조건을 추가하면 된다. 또한 집합{원(a, b, c)} 명령어를 비슷한 방법으로 사용하면 원뿔도 만들 수 있다.

활동3을 하기에 앞서 이해를 돕기 위해 한 가지를 더 소개한다.

beginxyz의 옆에 ; 를 붙여 beginxyz;를 입력하면 다음과 같이 좌표를 알아볼 수 있다. 또한 그 옆에 beginxyz; item=4와 같이 아이템을 바꾸는 명령어를 입력하면 땅의 아이템이 변함을 알 수 있다.



활동③ 아래의 그림과 위치와 층수까지 완벽히 같도록 5층짜리 피라미드를 지상에만 만들어 보자. 집합{정(a, b, c)}의 a, b, c에 대한 정확한 이해를 보기 위한 것이니 그림을 자세히 보고 그림과 똑같이 만들어보자. (TIP을 참조하세요.)

단, 집합{정(?, ?, ?)} 명령어를 한 번만 사용하되 조건은 추가하여 지하에는 없도록 해야한다.

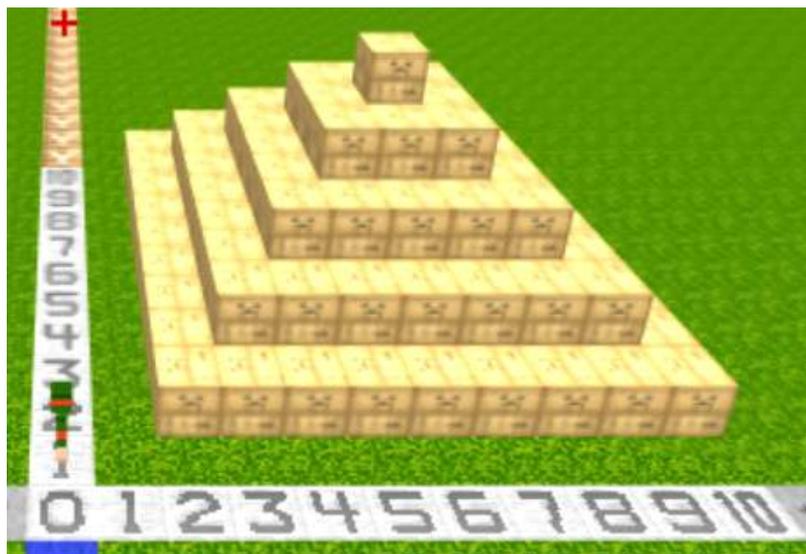
TIP!

활동3의 주의사항

① 중심좌표(위치)

② 길이와 층수
(가로가 9이면
중심으로부터 좌우로
몇씩 늘어났을까?)

③ item번호는 16
(4page '알아봅시다'
참조)



활동④

다음의 코드와 그림은 집합{정(a, b, c)}에서 c에 $15-0.25*z$ 와 $15-2*z$ 를 적용했을때의 모습이다. 이처럼 z앞에 계수를 곱해 자신이 원하는 탑모양의 뾰족한 피라미드와 고분 모양의 완만한 피라미드를 2개 자유롭게 만들어보자.

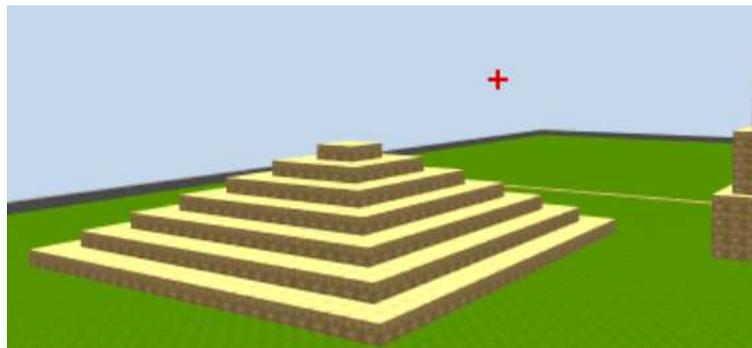
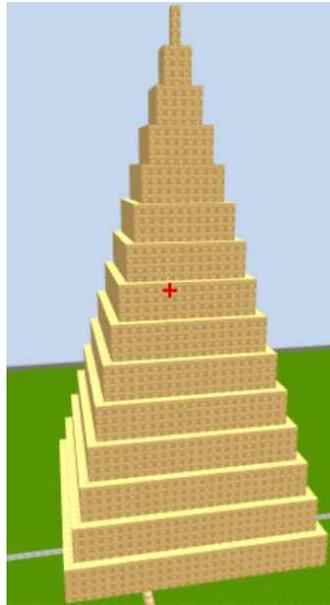
TIP!

■ 코딩에서 사칙연산

코딩에서는 수학기호가 자주 사용됩니다. 덧셈과 뺄셈은 +, -로 키보드에 있습니다. 곱셈은 * 나눗셈은 / 를 사용합니다.

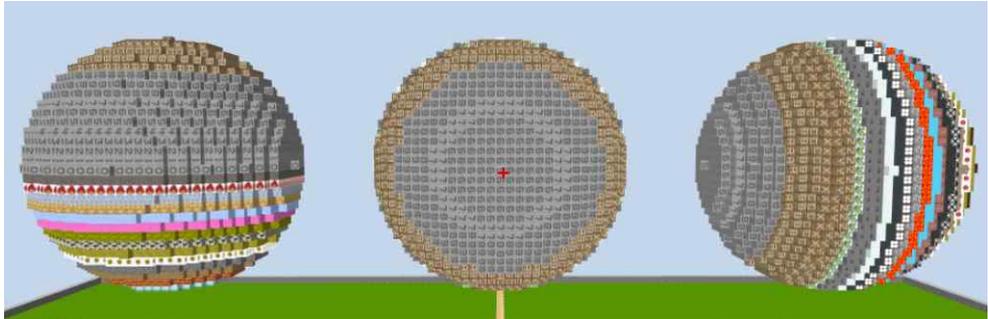
특히! 2 곱하기 n은 $2*n$ 으로 써야 합니다. $2n$ 으로 입력하면 실행되지 않습니다.

- ① 위치와 모양, item은 자유
- ② 집합 명령어는 2번 사용(2개의 피라미드)



활동 5 다음은 집합{구(a, b, c, d) ; item번호}에서 빨간 부분에 자리에 각각 x, y, z를 대입한 결과들이다. 다음 코드를 실행시켜 여러 방향에서 살펴보고 게시글에는 실행시킨 코드를 입력하고 댓글에 그 이유를 각각 입력해보자.

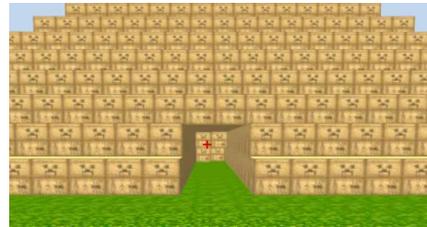
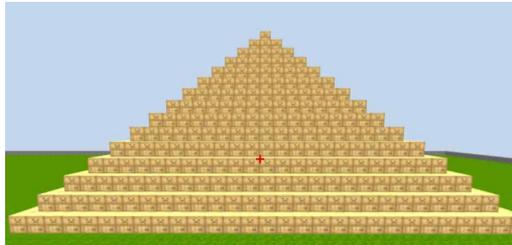
```
1 beginxyz
2 집합 구(40, 40, 20, 15) ; x }
3 집합 구(0, 40, 20, 15) ; y }
4 집합 구(-40, 40, 20, 15) ; z }
```



3. 도형의 내부 꾸미기

내가 만든 피라미드의 안은 어떠할까? 다음의 코드와 구멍을 뚫어본 모습을 살펴보자. 코드를 실행시켜서 직접 탐구해보기 바란다.

```
1 beginxyz
2 집합{ 정(10, 10, 15-z) && z>0 ; 16 }
```

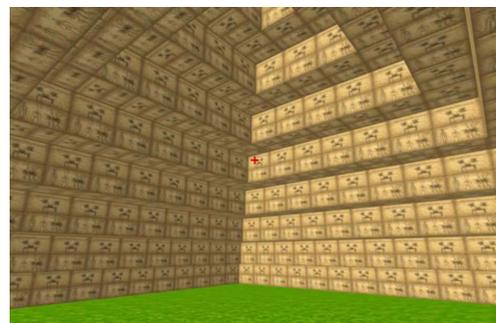
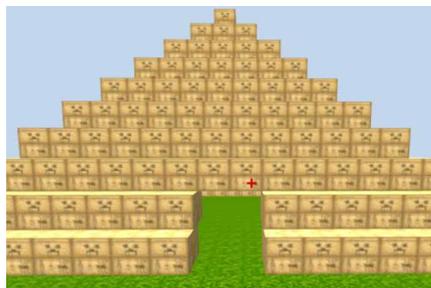


안이 짝 차있다. 그렇다면 피라미드 안을 비워보려면 어떻게 해야할까? 일일이 안에 들어가서 다 구멍을 뚫으면 되는 것일까?

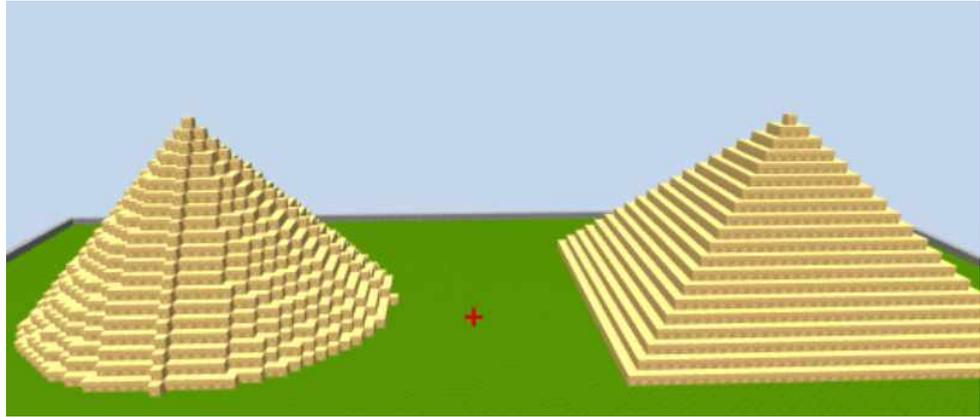
터틀크래프트에서 item=0은 투명한 블록으로서 0번 아이템을 사용하면 집합의 조건을 만족시키는 원소들이 투명한 것으로 출력된다. 다음의 코드를 실행시키기 전에 어떻게 그려질 것인지 상상해보고 다음 사항을 하나 유념해두자.

```
1 beginxyz
2 집합{ 정(10, 10, 8-z) && z>0 ; 0 }
3 집합{ 정(10, 10, 10-z) && z>0 ; 16 }
```

위를 보면 알 수 있듯이 해당코드는 먼저 좀 더 작은 투명 피라미드 (item=0)을 만들고 그다음 곁에 더 큰 피라미드를 만들게 된다. 2번 줄과 3번 줄의 순서를 바꾸면 다시 피라미드는 속이 짝 차는 것을 알 수 있다. 즉, 먼저 특정좌표에 채워진 아이템은 밑에 다시 명령어를 입력하여 변형시켜도 변하지 않음을 뜻한다. 따라서 이번 활동을 할 때는 순서에 반드시 유념하여 코딩을 해야 한다.



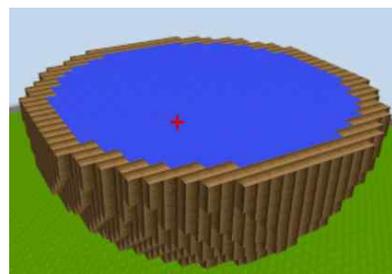
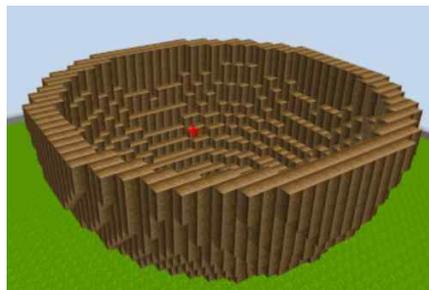
활동⑥ 피라미드를 크게(15층이상), 그리고 안에 공간이 있도록 만들어보자.
 그리고 멀리 떨어뜨려 원뿔안에도 공간이 있도록 만들어보자.
 (입구도 만들어보자 :)



마지막으로 피라미드의 안을 비우는 데 있어 반드시 ①투명아이템을 먼저 사용하고 ②다른 아이템으로 그보다 더 큰 피라미드를 만드는 순서를 지켜야함을 다시 한번 기억하자. 특히 집합{구(a, b, c, d)}명령을 이용하면 속이 빈 구를 만들 수 있고 조건을 더 추가하면 그릇과 같은 반구를 만들 수 있다. 한편 그릇에 물을 채우려면 첫 줄의 아이템을 5 번으로 바꾸면 된다.

```

1 beginxyz
2 집합{ 구(30, 30, 15, 13) && z>0 && z<15 ; 0 }
3 집합{ 구(30, 30, 15, 15) && z>0 && z<15 ; 2}
    
```

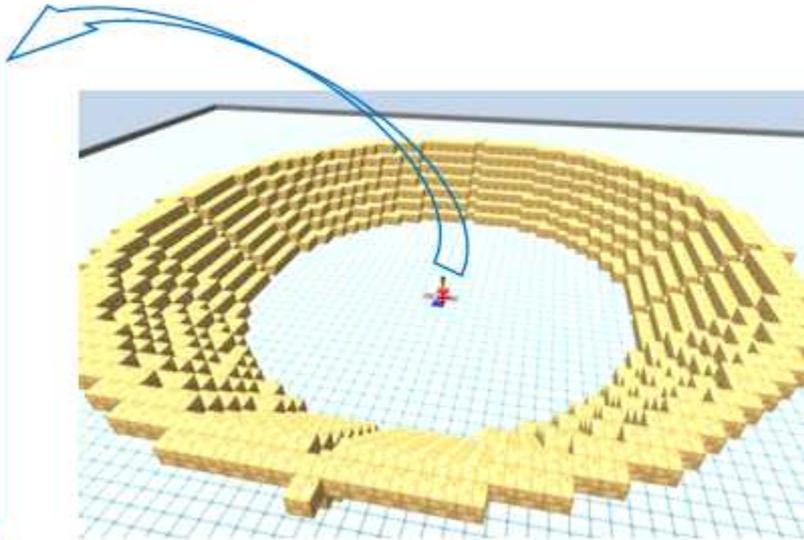


위의 코드를 반드시 실행해보고 다양한 상상력을 동원하여 더 많은 도형을 만들 수 있도록 지금부터 미리 연습하자.

item 번호는 마지막 페이지의 부록을 참고하세요.

창의 과제 ① 아래의 그림은 원형 경기장 속의 피겨왕 김터틀이다. 다양한 방법이 있겠지만 어렵다면 다음의 Hint 코드를 변형 및 이용하여 위치를 조정하고 빈 공간을 만들어 김터틀에게 원형경기장을 선물할 코딩을 해보자.
<5-분반 내이름 피겨왕 김터틀'로 저장>

Hint : 집합 { 원(0,0,z+???) && ??? && ??? ; 16 }



창의 과제 ② 화산 활동을 시작하려는 활화산의 모습을 상상해보자. 지금까지 배운 명령어들을 활용하여 안에 뜨거운 용암이 있도록 활화산을 만들어보자.
<5-분반 내이름 활화산'으로 저장>

