

# 04 변수 & 집합!

- ◎ 학습목표
  - 변수를 이용해 변하는 대상을 만들 수 있다.
  - 집합을 활용해 큐브를 배치할 수 있다.

## 변하는 대상 만들기

생각 열기

먼 옛날 사람들은 동물이나 식물을 신성시하며, 이를 상징적으로 표현한 조형물을 곁에 두고는 했다.



알아봅시다!

중세시대에도 상징적 조각을 건물에 배치하고는 하였습니다. 노트르담 성당에는 가고일 석상이, 경복궁에는 해태상이 있습니다.



이들은 단순히 건물을 꾸미는 용도를 넘어서, 사람들에게 경계심과 신앙심을 가지게 합니다.

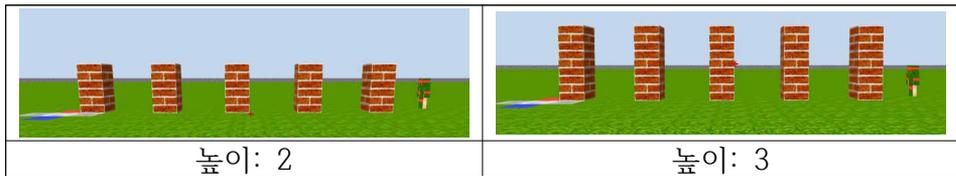
이를 토템이라고 한다. 토템을 만드는 방법의 하나는, 비슷한 모양을 크기별로 다르게 해서 배치하는 것이다.

지난 시간에 우리는 치환을 학습했다. 같은 모양을 크기만 다르게 해서 여러 개를 만들 수 있을까?

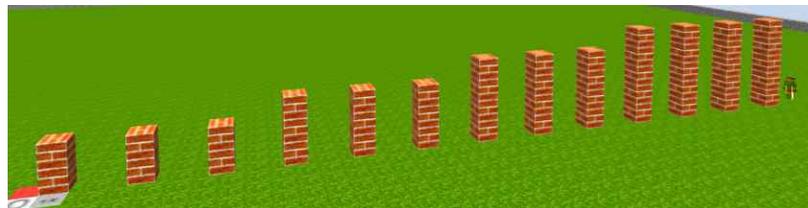
### 1. 변수! 변할 수 있는 수(크기)

성곽을 꾸밀 수 있는 조각상을 만들어 보자. 같은 모양을 크기만 다르게 해서 배치하려면 어떻게 해야 할까?

시작은 최대한 간단하게 해보자.



위 그림을 보면 왼쪽은 높이가 2인 기둥이 5개, 오른쪽은 높이가 3인 기둥이 5개 배열된 것을 알 수 있다.



심지어 이번 그림은 2, 3, 4, 5 크기의 기둥이 배열되어 있다. 모양은 똑같다. 다만, 그 크기만 다르다. 즉, 위로 올라간 **수가 변한다.**

이처럼 ‘**변할 수 있는 수**’를 우리는 **변수**라고 부른다. 우리는 이를 통해 크기가 변하는 대상을 만들어 볼 수 있다.

예를 들어 앞서 만든 구조물은 다음과 같은 코드를 가진다.

1	X = 's[u] 2T'	상자 만들고 위로 한 칸, 앞으로 두 칸 점프
2	doit(5X)	
3	Y = 's[2u] 2T'	상자 만들고 위로 두 칸, 앞으로 두 칸 점프
4	doit(5Y)	

1번과 3번의 치환을 보면 구조는 같으며, 올라간 숫자만 다르다는 것을 알 수 있다. 그렇다면 이를 다음과 같이 표현해 보면 어떨까?

1	X = 's[(n)u] 2T'	... 위로 n칸, ...
---	------------------	----------------

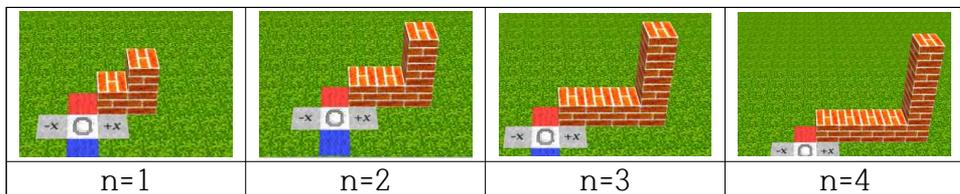
여기서 n을 우리는 변수라고 한다. 이를 표기할 때는 다른 코드와 다르게 괄호 안에 들어감을 주의하자.

이를 이용하면, 우리는 쉽게 대상을 변화시킬 수 있다.

1	X = 's[(n)u] 2T'	상자 만들고 위로 n 칸, 앞으로 두 칸 점프
2	n=1	n=1로 하고,
3	doit(5X)	X를 5번 실행
4	n=2	n=1로 하고,
5	doit(5X)	X를 5번 실행

다소 코드의 길이가 늘어났다고 보일 수 있으나, 입력한 코드가 훨씬 경제적으로 변했다는 것을 알 수 있다. 변하는 대상을 코딩하기 위해서는 우선 규칙성부터 인식하는 것이 중요하다.

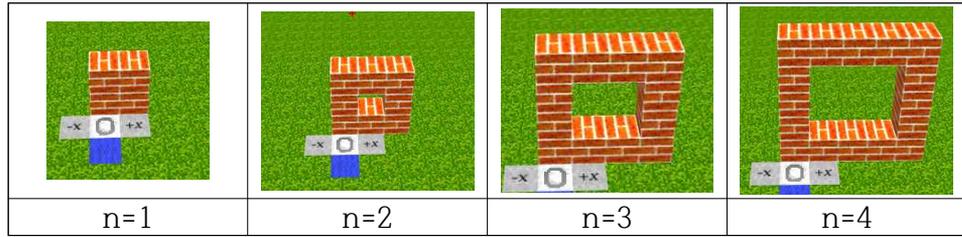
**활동 1** 다음 그림과 같이 n의 값에 따라 변하는 구조물을 코딩하자.



\*단, 저장할 때, n의 값은 5로 해보자.

## 활동②

다음 그림과 같이 n의 값에 따라 변하는 구조물을 코딩하자.



알아봅시다!

### ■ 코딩에서 사칙연산

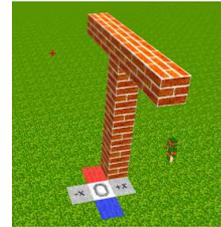
코딩에서는 수학기호가 자주 사용됩니다. 덧셈과 뺄셈은 +, -로 키보드에 있습니다. 곱셈은 \* 나눗셈은 / 를 사용합니다.

특히! 2 곱하기 n은 2\*n으로 써야 합니다. 2n으로 입력하면 실행되지 않습니다.

\*단, 저장할 때, n의 값은 7로 해보자.

변하는 대상을 배치하는 방법은 간단하다. 우리는 goto를 통해서 거북이의 위치를 바꾸고, T를 이용해서 거북이를 점프시킬 수 있다. 예를 들어, 다음과 같은 나만의 토템을 만들었다고 해보자.

```
1 X = 's[(2*n)u [(n)l] [(n)r] ]'
2 n=3
3 doit(X)
4
```



점프(T)를 이용하면, 각 구조물이 떨어지게 할 수 있다.

```
1 X = 's[(2*n)u [(n)l] [(n)r] ] 3T'
2 n=3
3 doit(10X)
```



## 활동③

나만의 토템(구조물)을 만들고, 10개를 배치해 보자.

**TIP!**

앞뒤로 가지를 뺀 경우, 토템끼리 겹칩니다. 이 경우, 3T보다 큰 값으로 점프를 해야 합니다.

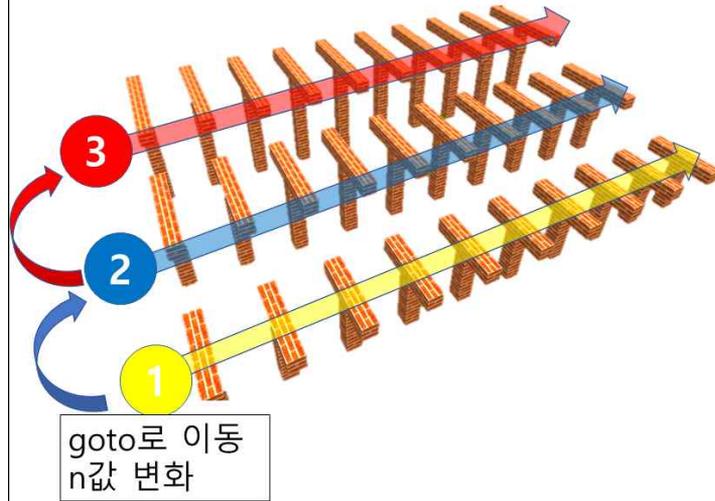
goto를 이용하면, 크기가 다른 토템을 배치할 수 있다.

**TIP!**

꼭 goto를 이용해  
위쪽 방향으로  
이동할 필요는 없다.

내가 원하는 위치로  
이동시키자.

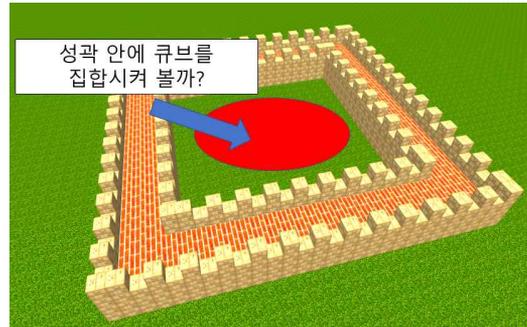
```
1 X = '...(생략)...'  
2 n=2  
3 doit(10X)  
4 goto(0,10,1)  
5 n=3  
6 doit(10X)  
7 goto(0,20,1)  
8 n=4  
9 doit(10X)
```



**활동④** 3개 이상의 크기를 가진 나만의 토템을 최대한 많이 만들어 보자.

## 조건에 맞는 큐브 집합!

생각 열기      지난 시간에 우리는 치환이라는 강력한 도구로, 성곽을 만들어 보았다. 치환을 통해 내가 생각하는 패턴을 쉽게 만들 수 있었다. 이번에는 또 다른 강력한 도구를 사용해 볼 것이다. 오른쪽 그림과 같이 성곽 가운데에 큐브를 집합시켜 보자. 정확하게 이야기해서 ‘정사각형 형태로 큐브를 집합시키자!’



컴퓨터가 이 말을 알아들을 수 있을까?

### 1. 집합이 뭐길래?

우리는 ‘**집합**’이라는 평소에도 사용한다. ‘거북 중학교 1학년 3반 학생들 집합!’이라고 선생님이 외치면, 선생님이 말한 조건에 맞는 사람들이 모인다.

거북중 1-3 **집합!**



수학에 말하는 집합은 ‘어떤 조건에 따라 결정되는 요소의 모임’이라고 할 수 있다. 그리고 집합에 들어있는 요소를 **원소**라고 한다.

키가 큰사람의 집합에 들어가도 될까?



170cm가 넘는 사람의 집합에 들어갈 수 있어!



이때, 주의해야 할 것은 **조건을 정확히 말해줘야 한다**는 것이다!

‘키 큰 사람의 집합은’ 수학적 집합이 아니다.

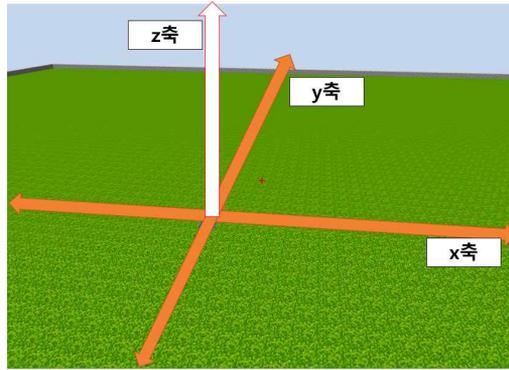
듣는 사람이 정확한 기준을 알

수 있게 해야 정확한 사람이 모임 것이다.

## 2. 큐브 집합시키기!

컴퓨터는 논리로 작동한다. 앞 장에서 살핀 것과 같이, ‘키가 큰 사람’이라는 용어는 알아듣지 못한다. 키가 150, 160, 170 이상인 사람과 같이 정확한 수치를 이야기 해주어야 한다.

터틀크래프트에서도 이와 같은 방식으로 큐브를 집합시킬 수 있다.



왼쪽은 첫 시간에 배웠던 3차원 공간이다. x축, y축은 각각 좌우, 앞뒤를 나타내는 좌표축이고, z축은 높이를 나타낸다.

우리는 거북이가 알아듣는 조건

을 x,y,z를 통해서 제시한다.

### 알아봅시다!

■ 왜 == 인가?

수학에서 사용되는 보통 두 개의 의미로 쓰인다.

a=1이라고 하자.(대입)  
a+1=2인가? (비교)

따라서 컴퓨터는 이를 구분한다.

= : 대입  
== : 비교

일단, 거북이에게 조건을 주겠다는 뜻으로 다음 코드를 입력한다.

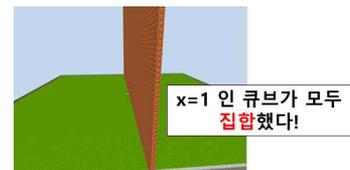
```
1 beginxyz
2
3
```

※ 지금부터 xyz로 조건을 주겠다!

조건은 다음과 같이 줄 수 있다.

```
1 beginxyz
2 집합{x==1}
3
```

→ beginxyz 클릭 →



조건을 줬는데, 너무 많은 큐브가 만들어진 듯하다. 다음 활동을 하고, 이유를 생각해보자.

### 활동⑤

다음을 코딩하고, 댓글에 집합의 원소 4개를 적어보자.

<‘4-5 내이름 집합’로 저장>

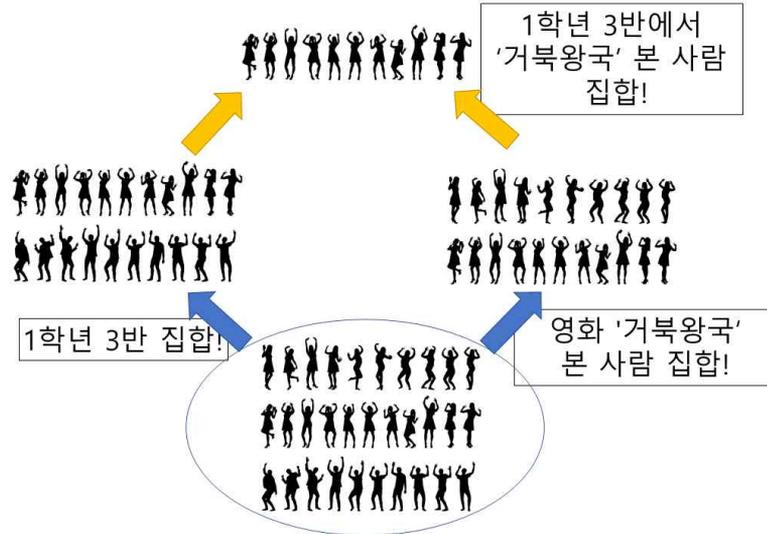
```
1 beginxyz
2 집합{z==1}
```

### TIP!

원소는 집합에 포함되는 요소를 뜻합니다. 예를 들어 (0,0,1)은 주어진 집합의 원소입니다.

### 3. 여러 조건 주기

앞서 살펴보았듯이 조건이 적으면, 집합에 들어갈 수 있는 원소는 많아진다. 반대로 이야기하면 여러 가지 조건을 줄수록, 그 안에 들어갈 수 있는 원소는 적어진다.



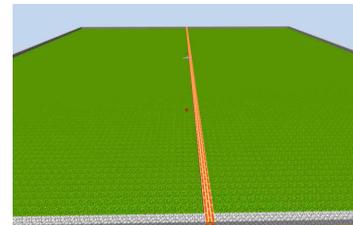
많은 조건이 주어질 때, 원소가 적어지는 이유는 그 조건을 모두 만족하기 어려워지기 때문이다.

즉, '1학년 3반 집합', '거북왕국 본 사람 집합' 보다는 '1학년 3반 **그리고** 거북왕국 본 사람 집합'에 포함되는 원소가 더 적어진다.

집합의 조건을 나열하기 위해서는 그리고를 코드로 표현해야 한다. 이는 다음과 같이 쓸 수 있다.

```
1 beginxyz
2 집합{x==1&&z==1}
3
```

→ beginxyz →



x==1인 조건을 만족하는 동시에 1층에 있는 큐브를 나열하면 위와 같은 모양이 나온다.

## 4. 집합으로 간단한 도형 만들기

집합에는 간단히 다음과 같은 정해진 조건이 들어갈 수 있다.

<pre>1 beginxyz 2 집합{정(a, b, c)}</pre>	<pre>1 beginxyz 2 집합{원(a, b, c)}</pre>
 <p data-bbox="475 609 707 721"><b>-정사각 기둥-</b> 중심: (a, b) 한변 : 2c</p>	 <p data-bbox="938 609 1169 721"><b>-원 기둥-</b> 중심: (a, b) 반지름 길이 : c</p>

**알아봅시다!**

■ 1층부터 5층

코딩을 할 때는 될 수 있으면 키보드를 적게 쓰고자 합니다. 즉 1층~5층이면

```
1<=z && z<=5
```

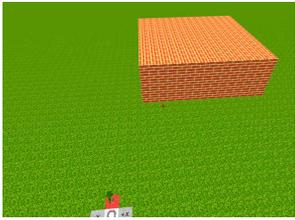
이라고 써도 된다.

그러나, 키보드를 적게 사용하려는 성향으로

```
0<z && z<6이라고 표현 한다.
```

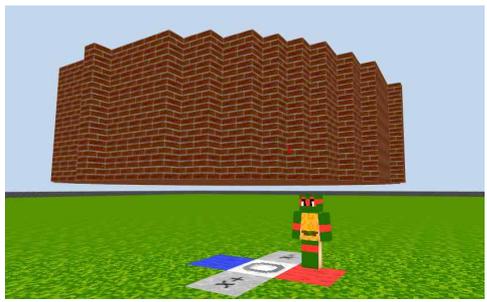
만약 중심이 (10, 20), 한 변의 길이가 7인 5층의 정사각기둥을 만들고자 하면 다음과 같은 코드를 실행하면 된다.

```
1 beginxyz
2 집합{정(10, 20, 7) && 0<z && z<6}
3
```



### 활동⑥

오른쪽 그림은 중심이 (-20, -15), 반지름이 15이고, 3층~10층에 만들어진 원기둥이다. 이를 코딩하자.



**TIP!**

1<z<10과 같은 표현은 인식 못합니다.

꼭 1<z && z<10과 같이 입력하세요.

## 창의 과제 ①

집합 명령을 이용해서, 오른쪽과 같이 피라미드를 만들어 보세요. 자신만의 아이디어를 발휘하여 크고 멋지게 만들어 볼까요?

**TIP!**

중심이 같고, 길이는 줄어들어야 합니다.

