3D프린팅으로 배우는
코딩교육

(주)헬로앱스

1. SPL for 3D 프린팅
1.1 SPL for 3D 프린팅이란? ····· 3
1.2 SPL for 3D 프린팅 설치 ····· 3
1.3 SPL for 3D 프린팅 실행 ····· 4

2. SPL for 3D 프린팅 편집기 알아보기
2.1 상단 메뉴 ····· 5
2.2 로직 명령어 ····· 6
2.3 3차원 도형 명령어 ····· 7
2.4 코딩창 ····· 8
2.4 결과창 ····· 9

3. SPL for 3D 프린팅 기본 도형 익히기
3.1 기본 도형 명령어 ····· 10
3.2 기본 도형 명령어 활용 ····· 15

4. SPL for 3D 프린팅 연산 명령어 익히기
4.1 연산 명령어 알아보기 ····· 17
4.2 연산 명령어 활용 ····· 21

5. SPL for 3D 프린팅 로직 명령어 For 반복문 익히기
5.1 For 반복문 알아보기 ····· 29
5.2 For 반복문 활용 ····· 33

6. SPL For 3D 프린팅 로직 명령어 2중 For 반복문 익히기
6.1 2중 For 반복문 알아보기 ····· 39
6.2 2중 For 반복문 활용 ····· 42

7. SPL For 3D 프린팅 로직 명령어 If 조건문 익히기
7.1 If 조건문 알아보기 ····· 49
7.2 If 조건문 활용 ····· 55

8. SPL For 3D 프린팅 로직 명령어 수식 익히기
8.1 수식 알아보기 ····· 60

9. SPL For 3D 프린팅 로직 명령어 문자 출력 및 문자 라인 출력
9.1 문자 출력 알아보기 ····· 69
9.2 문자라인 출력 알아보기 ····· 70
9.3 문자 출력으로 수식 입력하기 ····· 71
9.4 문자 출력 & 문자라인 출력 응용하기 ····· 72

10. SPL For 3D 프린팅 로직 명령어 주석과 공백
10.1 주석과 공백 ····· 76

11. SPL For 3D 프린팅 로직 명령어 while문
11.1 while문 사용하기 ····· 78

12. SPL For 3D 프린팅 도형 불러오기 & 도형 변환 명령어
12.1 도형 불러오기 사용하기 ····· 79
12.2 도형 변환 명령어 사용하기 ····· 81
12.3 도형 불러오기 & 도형 변환 명령어 활용하기 ····· 94

13. SPL For 3D 프린팅 로직 명령어 함수
13.1 함수 정의 ····· 108
13.2 함수의 활용 ····· 110
13.3 함수를 이용하여 모델링 제작 ····· 119

14. 다른 창작물 만들어 보기
14.1 굴뚝 모양 꽃이 만들기 ····· 127
14.2 첨성대 만들기 ····· 131
14.3 체스 세트 만들기 ····· 138

1. SPL For 3D 프린팅

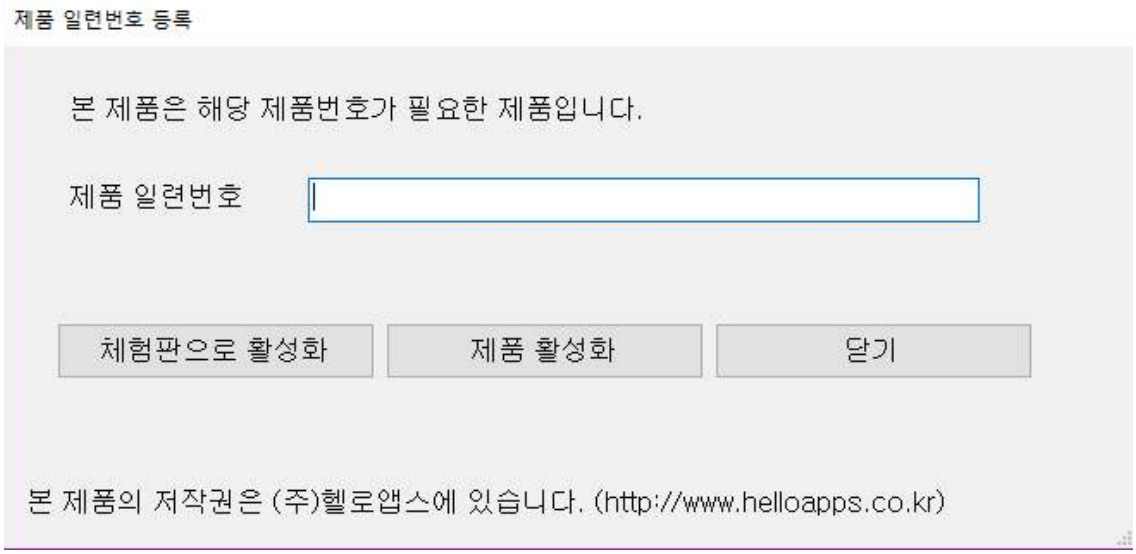
1.1 SPL For 3D 프린팅이란?

SPL For 3D 프링팅은 기존의 모델링 프로그램과 다른 방식을 채택하고 있습니다.
기본 도형과 사칙연산으로 모델링을 하는 방식을 취하고 있습니다.
사칙연산이 가능한 수준의 학생이면 사용이 가능합니다.
본 프로그램은 C언어를 기반으로 하고 있기에 모델링교육과 SW 코딩 교육을 동시에 진행이 가능합니다.
C언어의 로직들을 사용함으로써 코딩 학습에 도움을 줄 수 있습니다.

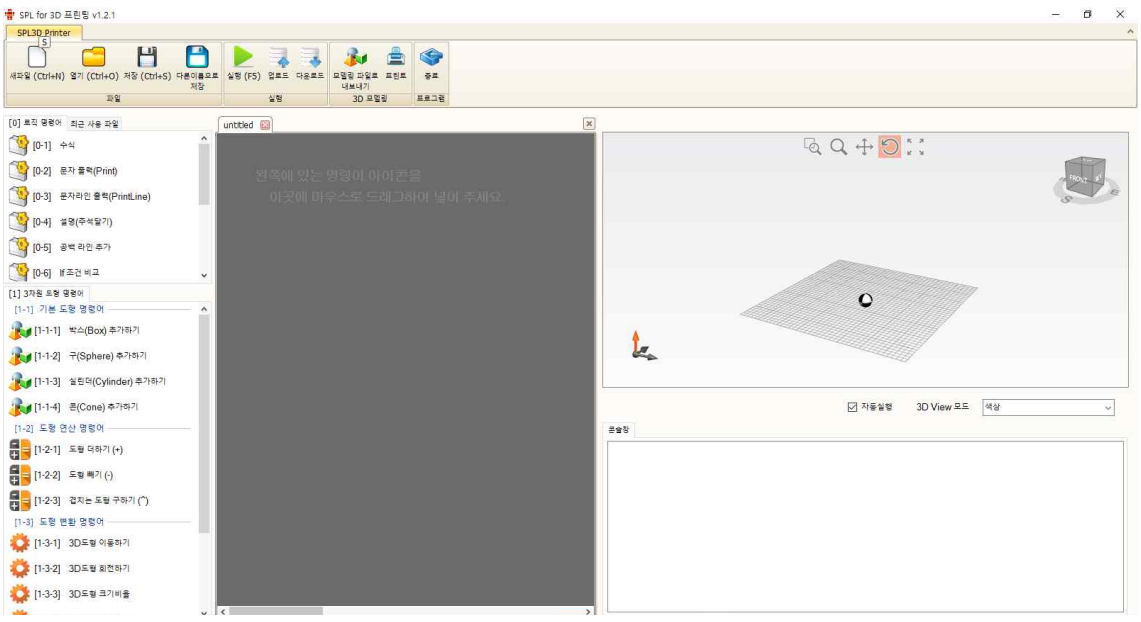
1.2 SPL For 3D 프린팅 설치

헬로앱스 SPL For 3D 프린팅 저작툴은 http://www.helloapps.co.kr/ 접속 후 **3D 프린팅** 페이지에서 다운로드 받을 수 있습니다.

1.3 SPL For 3D 프린팅 실행



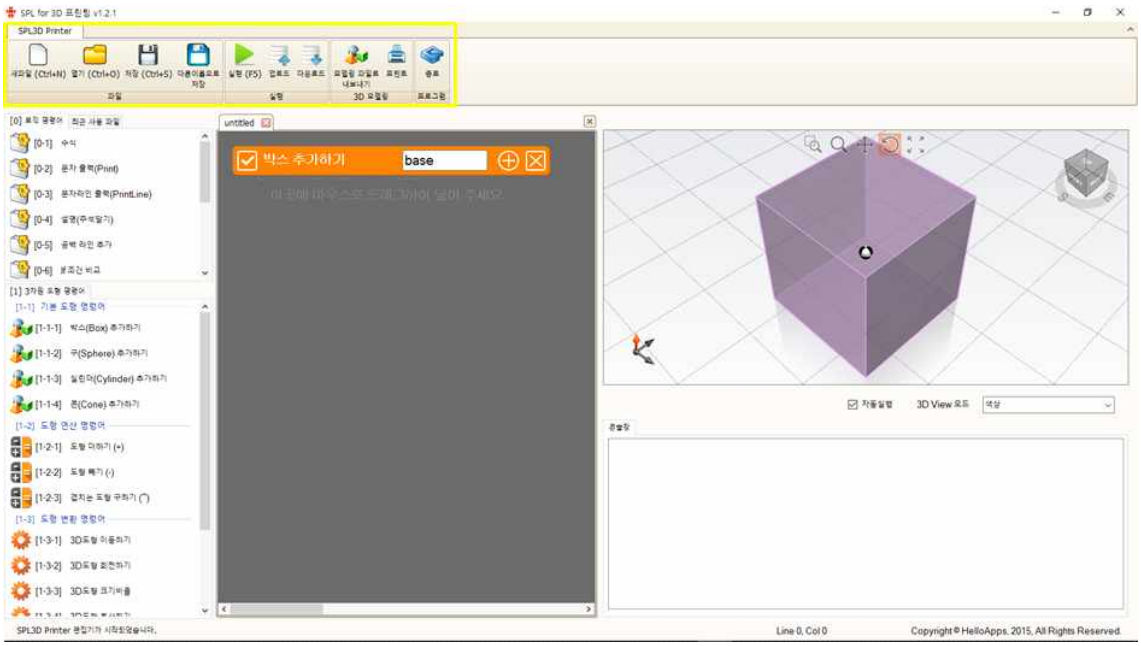
설치가 완료되고 실행하면 다음과 같은 창이 뜹니다.
체험판 활성화시 15번의 실행이 가능합니다.
본 교제의 앞면의 시리얼 넘버를 입력하여 제품 활성화를 해줍니다.
본 제품키는 제품키당 한 대의 PC이며 1년의 라이센스 기간을 가지고 있습니다.



제품키를 입력하고 나면 위 그림과 같은 화면으로 전환됩니다.

2. SPL For 3D 프린팅 편집기 알아보기

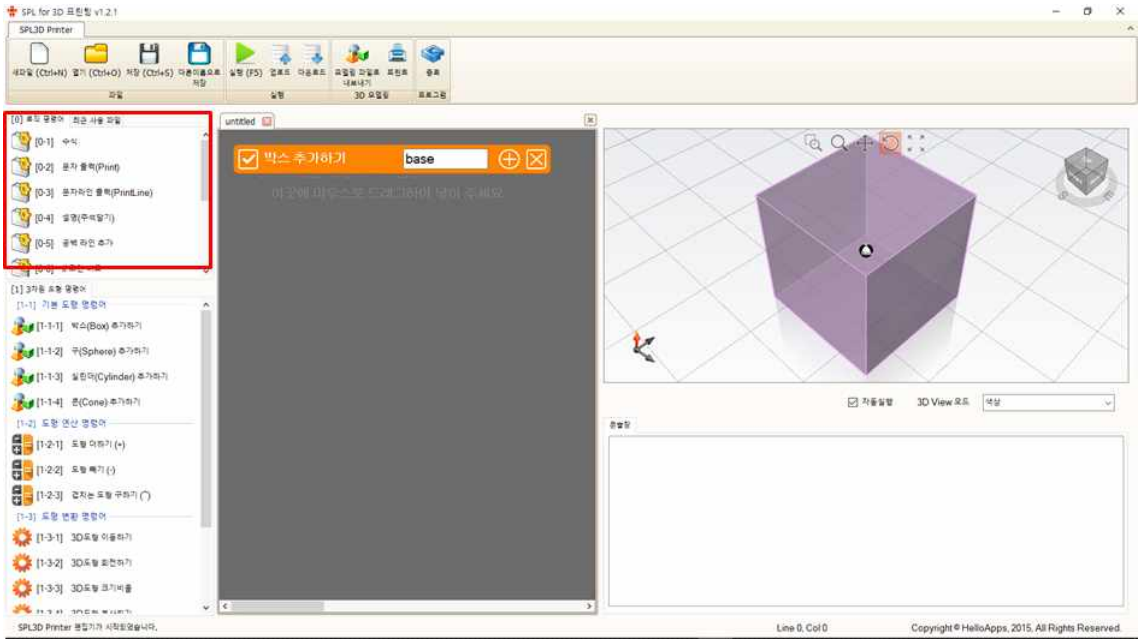
2.1 상단 메뉴



노란색 박스 표시된 상단메뉴는 다음과 같은 기능을 합니다.

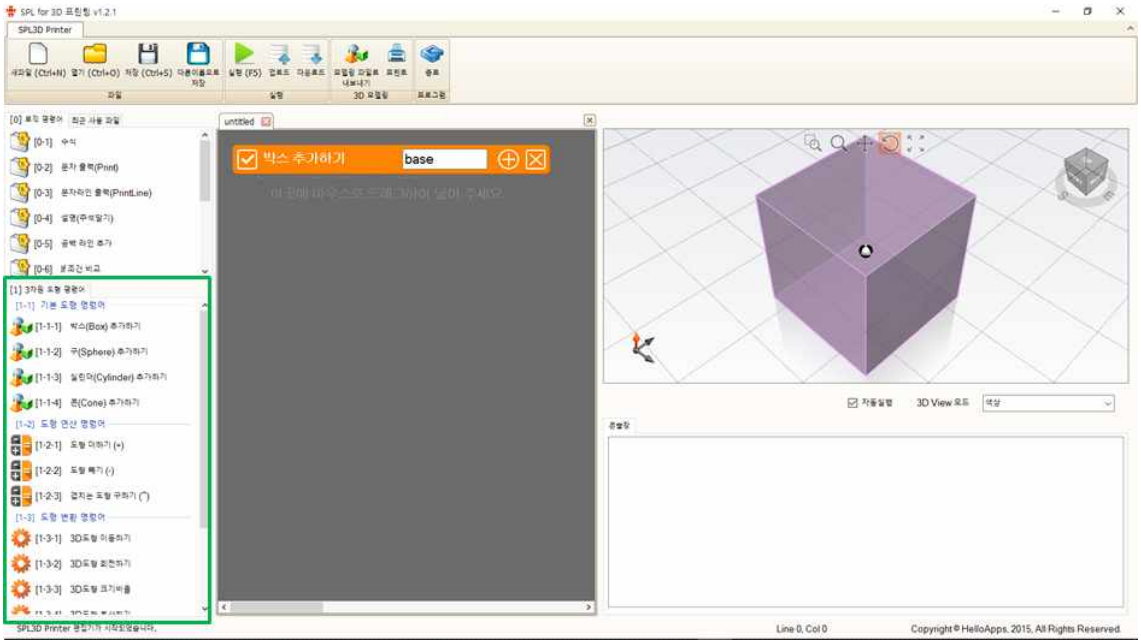
- 새파일 (Ctrl+N) : 공백의 새 파일을 생성합니다.
- 열기 (Ctrl+O) : 이전에 저장했던 파일을 엽니다.
- 저장 (Ctrl+S) : 현재 작업하고 있는 파일을 저장합니다.
- 다른 이름으로 저장 : 이미 저장한 현재의 파일을 다른 이름으로 저장합니다.
- 실행(F5) : 작업하거나 수정한 파일을 실행하여 오른쪽 미리보기창에 적용시킵니다.
- 업로드 : 클라우드 서버에 내가 작업한 파일을 업로드 시킵니다.
- 다운로드 : 클라우드 서버에 저장되어 있는 파일들을 다운로드 받습니다.
- 모델링 파일로 내보내기 : 현재까지 작업한 내용을 내보내기를 합니다.
  내보내기시 *.STL, *.emf, *.OBJ, *.iGS로 내보낼 수 있습니다.
- 프린트 : 현재 지원하지 않는 기능입니다.
- 종료 : SPL For 3D 프린팅 편집기를 종료합니다.

2.2 로직 명령어



빨간색 박스 표시된 왼쪽 상단 메뉴는 로직 명령어로 C언어의 명령어들입니다.
위 명령어를 사용함으로써 SPL For 3D 프린팅 편집기는 코딩을 기반으로 하는 모델링파일 편집기라 할 수 있습니다.
명령어들의 대한 설명은 로직 명령어 익히기 페이지에서 설명하도록 하겠습니다.

2.3 3차원 도형 명령어

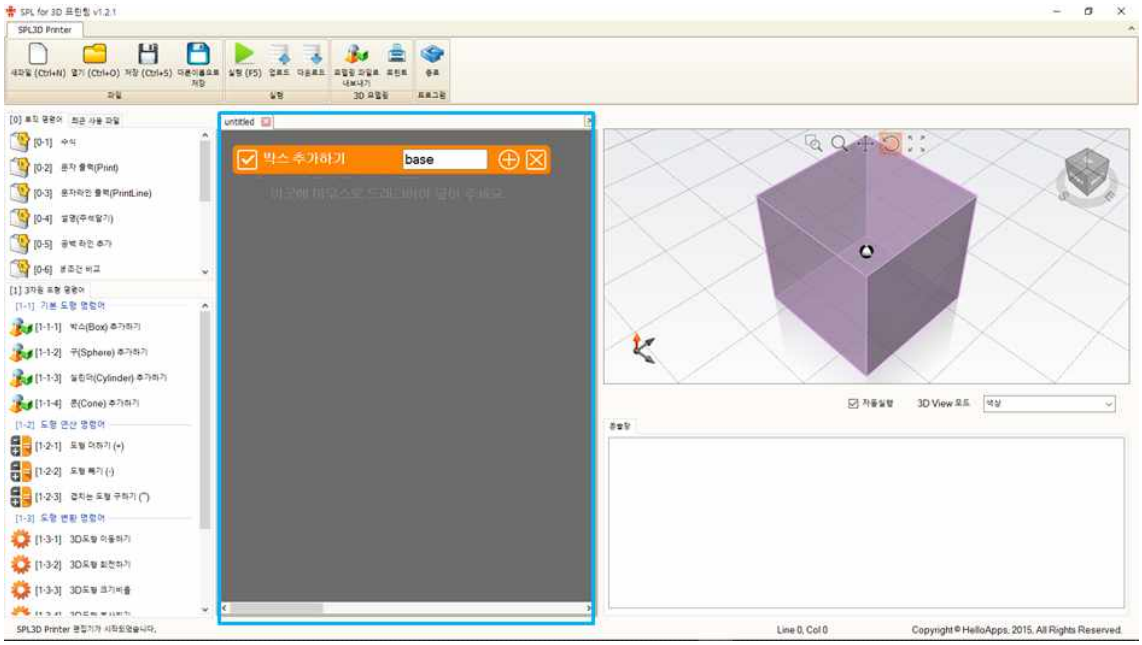


녹색 박스 표시된 왼쪽 하단의 3차원 도형 명령어는 기본 도형과 그 도형들에게 명령어를 내릴 수 있는 메뉴입니다.

[1-1] 기본도형명령어는 박스, 구, 실린더, 콘이 주어지며 이 도형들을 [1-2]연산 명령어(더하기, 빼기, 겹치는 부분 구하기)를 이용하여 다른 도형을 만들 수 있습니다.

별도로 [1-3] 도형 변환 명령어를 이용하다 기타 작업을 할 수 있습니다.

2.4 코딩창

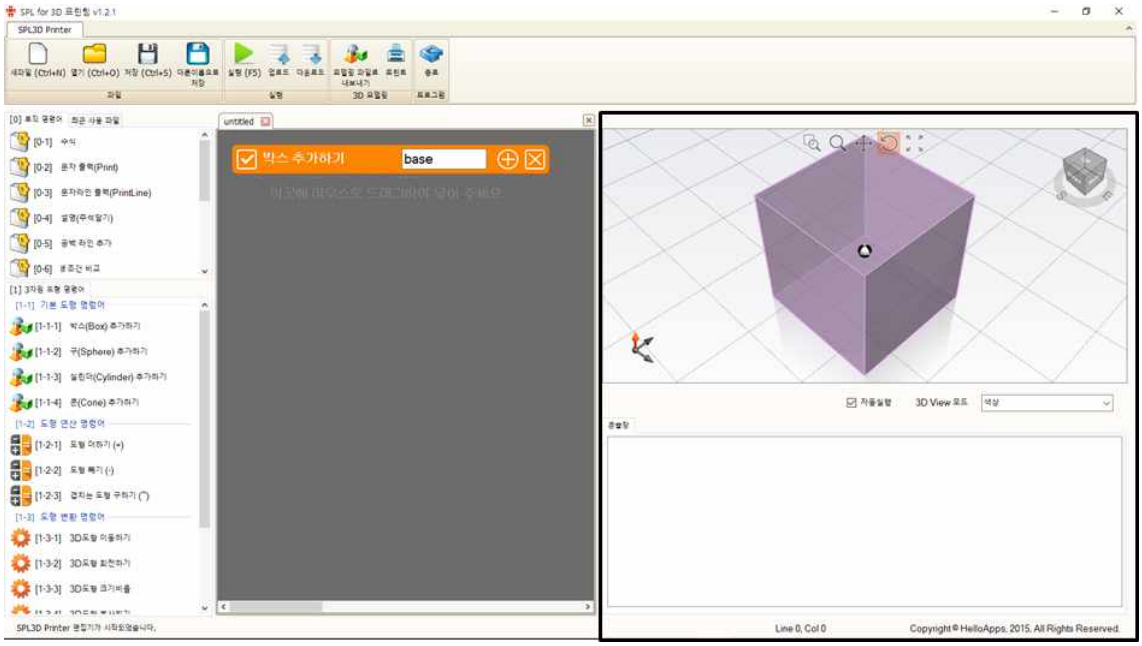


하늘색 박스 표시된 부분은 코딩창으로 도형 생성과 명령어를 설정하고 수행하기 위한 창입니다.

로직명령어와 3차원 도형명령어를 해당창에 드래그 앤 드롭하거나 더블클릭하면 명령어가 생성됩니다.

창 안의 명령어들을 드래그 앤 드롭으로 서로 위치를 바꿀 수 있습니다.

2.4 결과창

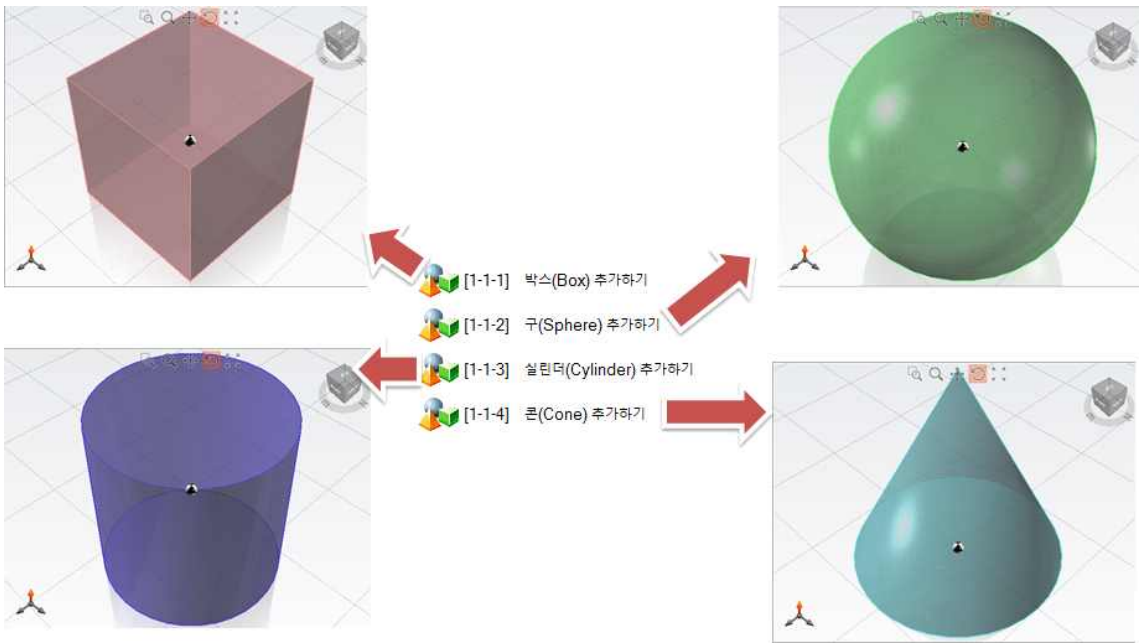


검은색 박스의 표시된 부분들은 결과 창으로 상단은 지금까지 작성한 명령어를 작용시켜 3D 도형을 미리보기해줍니다.
미리보기 창의 가운데 있는 흰색검은색 작은 구는 (0,0,0)의 위치를 표현한 곳입니다.
창의 왼쪽 하단은 X, Y, Z축을 의미합니다.
오른쪽 상단의 주사위모양을 이용하여 윗면 아랫면 정면 대각선 등 미리보기의 카메라 위치를 조정이 가능합니다.
바로 아래의 3D View 모드의 메뉴바를 선택하여 와이어와 외각선 모드로도 보기가 가능합니다.
하단의 콘솔창은 로직 명령어에서 [0-2] 문자출력과 [0-3] 문자라인 출력을 입력하였을 시 콘솔창에 해당 명령어가 나타납니다.
그 외에 명령어가 오류가 발생시 오류 메시지 또한 출력됩니다.

3. SPL For 3D 프린팅 기본 도형 익히기

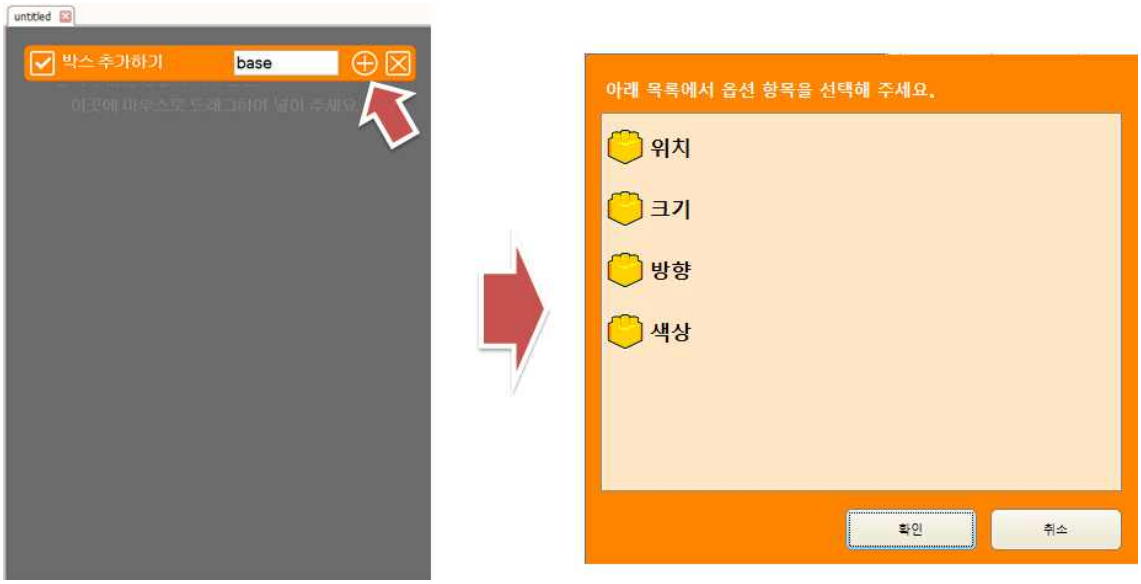
3.1 기본 도형 명령어

위 그림과 같이 왼쪽 하단의 메뉴를 보시면 **[1-1] 기본 도형 명령어**가 있습니다.
원하시는 도형을 더블클릭하거나 드래그 앤 드롭을 하여 중앙 창에 추가 하실 수 있습니다.

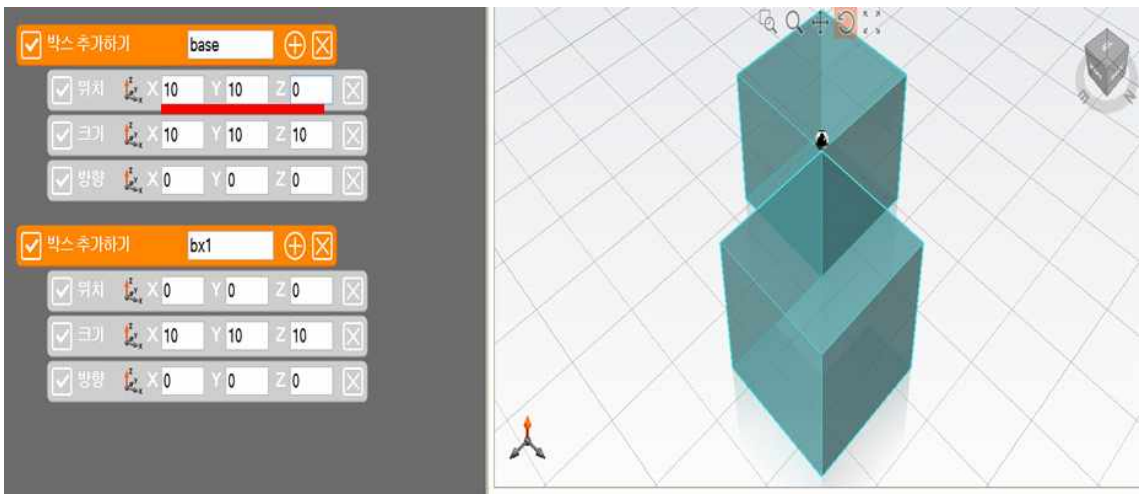


4가지의 기본도형들은 위 그림과 같은 모양을 지니게 됩니다.

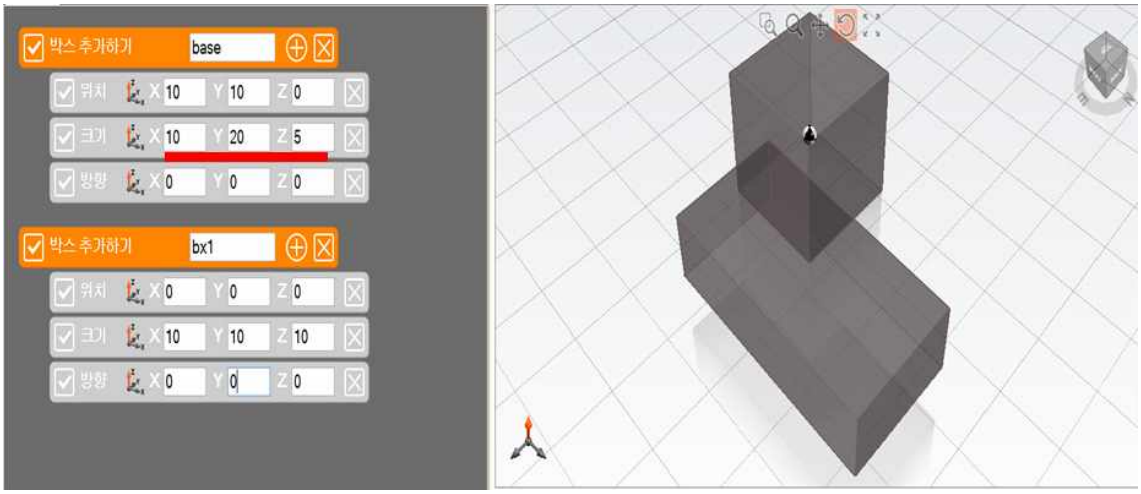
1) 박스



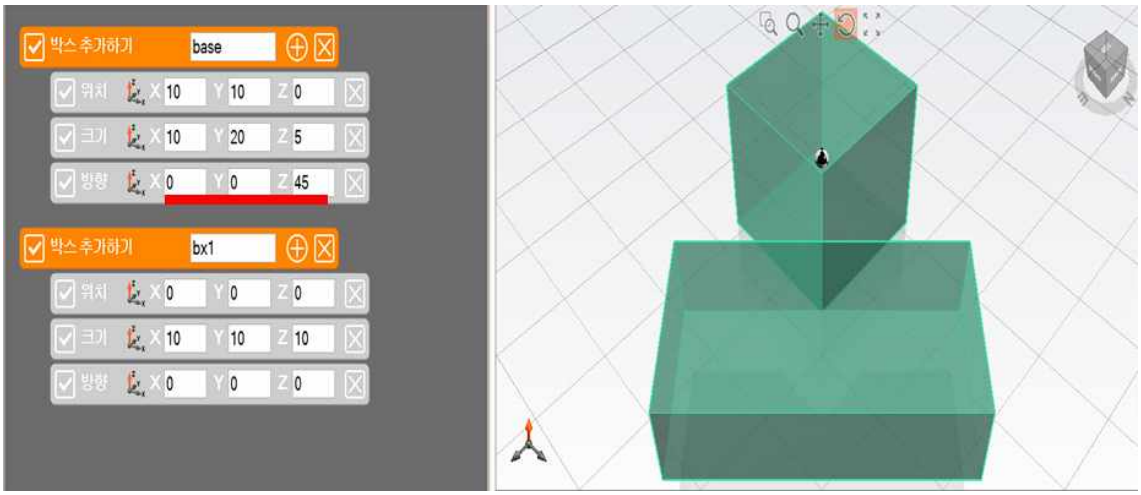
박스를 하나 추가합니다.
각 도형마다 옵션이 있으며 ⊕를 클릭하면 메뉴가 하나 뜨고 각 메뉴를 선택 할 수 있습니다.



위 그림의 도형과 같이 박스(base)의 위치를 X : 10, Y : 10으로 설정 하면 박스(bx1)과 다른 곳에 이동한 모습을 볼 수 있습니다.
위치의 수치는 (mm)입니다.



이번엔 박스(base)의 크기를 Y를 20, Z를 5로 설정하였습니다.
박스(base)sms Y축으로 20 늘어났고 Z축 5만큼 줄어들었습니다.
크기가 x : 10, y : 10, z : 10인 박스(bx1)과는 차이가 생김을 확인 할 수 있습니다.
크기의 수치는 (mm)입니다.

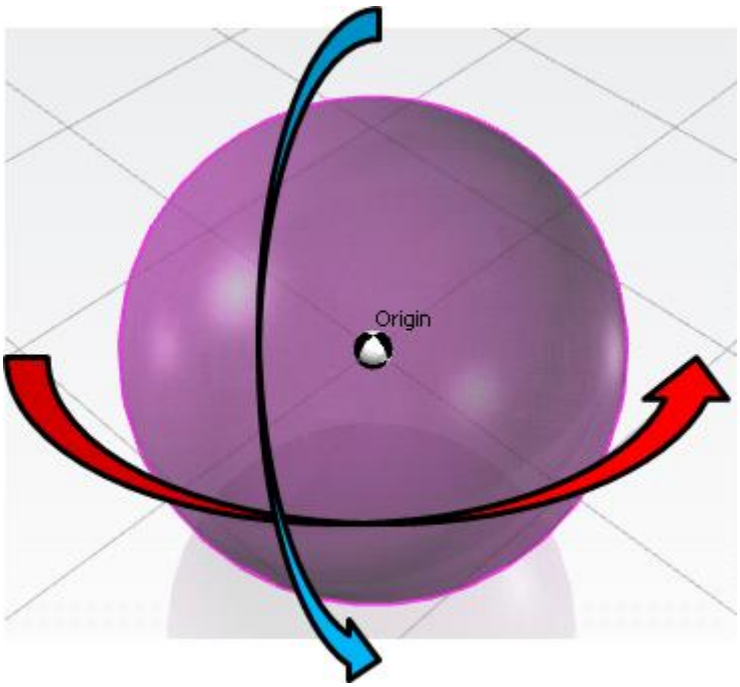


이번엔 박스(base)의 방향의 수치 Z를 45만큼 적용한 결과입니다.

방향은 위 그림의 X축 Y축 Z축을 기준으로 회전을 합니다.
방향의 값은 도(°)입니다.

2) 구

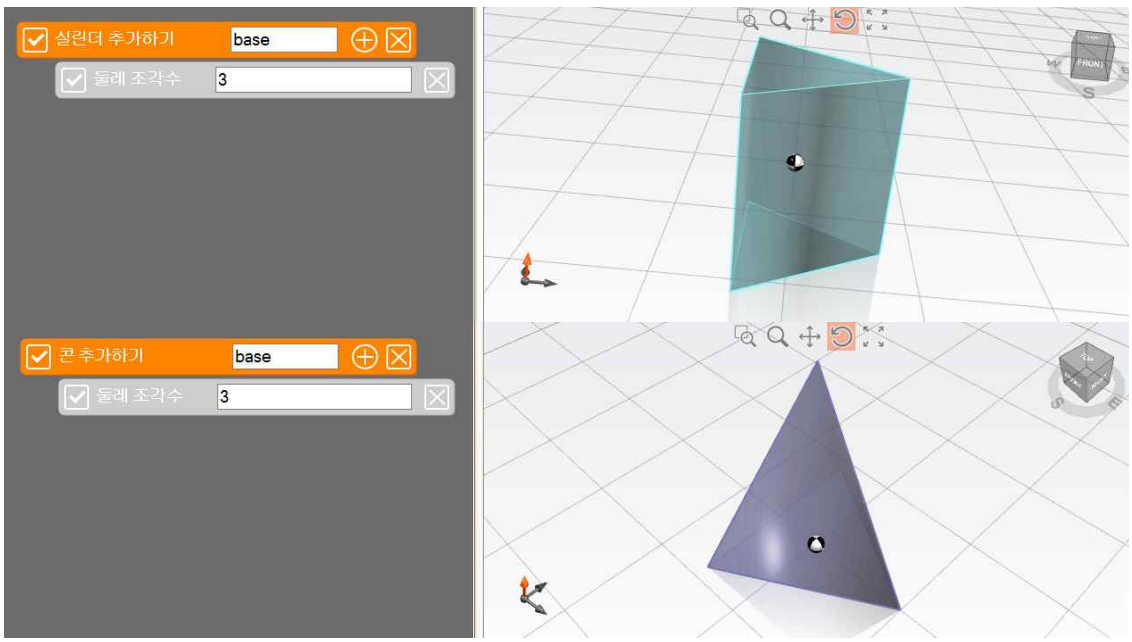
구 역시 박스와 마찬가지로 옵션을 가지며 클릭하면 위 그림과 같은 메뉴가 나옵니다.
위치, 크기, 방향은 박스에서 설명했던 원리과 같습니다.
반면 박스와 달리 둘레 조각수와 세로 조각수를 가지고 있습니다.



가로 방향이 둘레 조각수를 나타내며 세로 방향이 세로 조각수를 나타냅니다.

3) 실린더 & 콘

구와 달리 실린더와 콘은 둘레 조각수 하나만 가지게 됩니다.

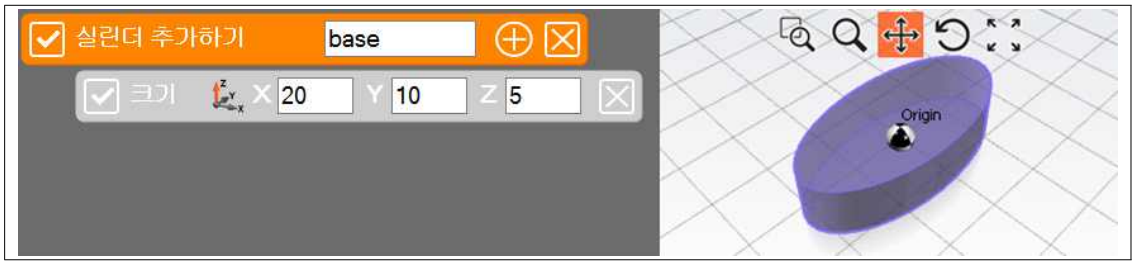


위 그림과 같이 둘레 조각수를 3으로 설정하면 실린더는 삼각통이 되고 콘은 삼각뿔이 됩니다.
4로 설정하면 사각통, 사각뿔 6으로 설정하면 육각통, 육각뿔이 됩니다.
둘레 조각수는 최소값 3을 가지며 최대값 36을 가집니다.

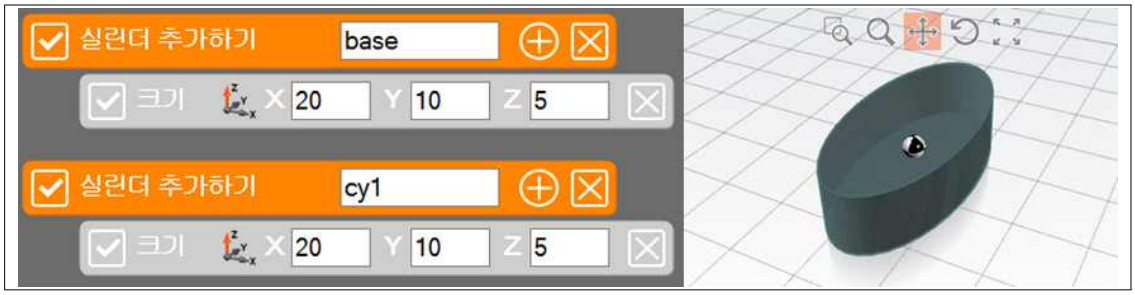
3.2 기본 도형 명령어 활용

1) 하트 만들기

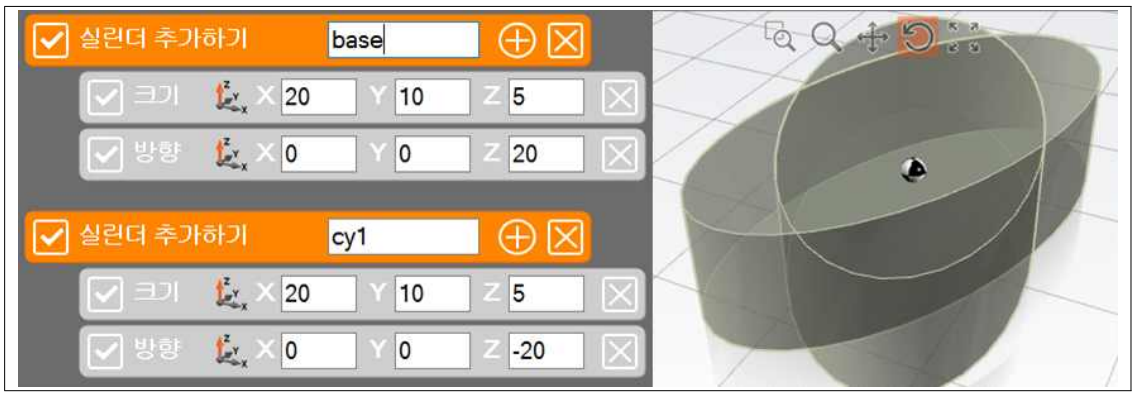
도형의 크기, 위치, 방향으로 하트모양을 만드는 방법입니다.



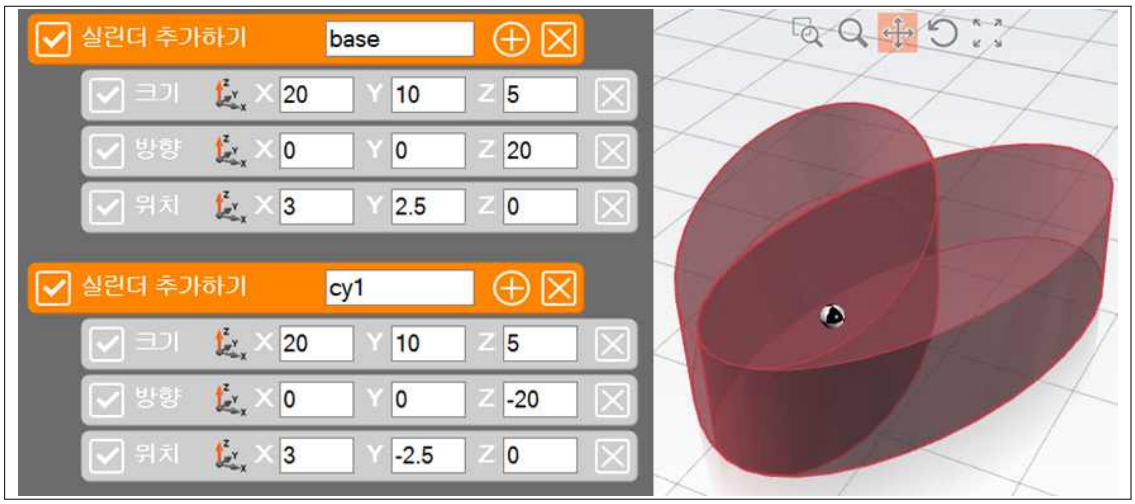
우선 실린더 하나를 생성하여 한축은 길고 한축은 짧고 얇은 실린더를 하나 만들어 줍니다.



같은 크기의 실린더를 생성해줍니다.
두 도형은 같은 위치와 같은 방향이므로 겹쳐진 상태입니다.
방향을 바꾸어 봅니다.



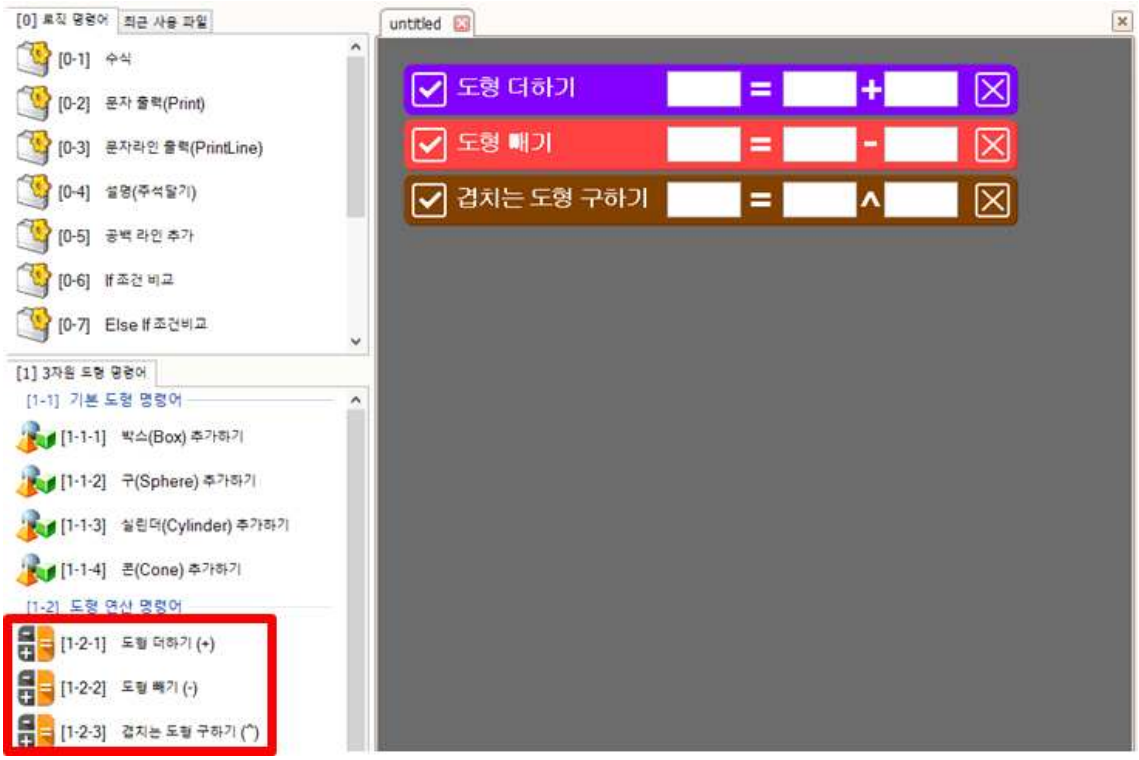
이제 두 도형을 이동하여 최종적으로 하트 모양을 만듭니다.



실린더 두 개를 이용하여 기존에 없던 하트 모양을 만들었습니다.
위와 같은 방법으로 다양한 도형을 만들 수 있습니다.

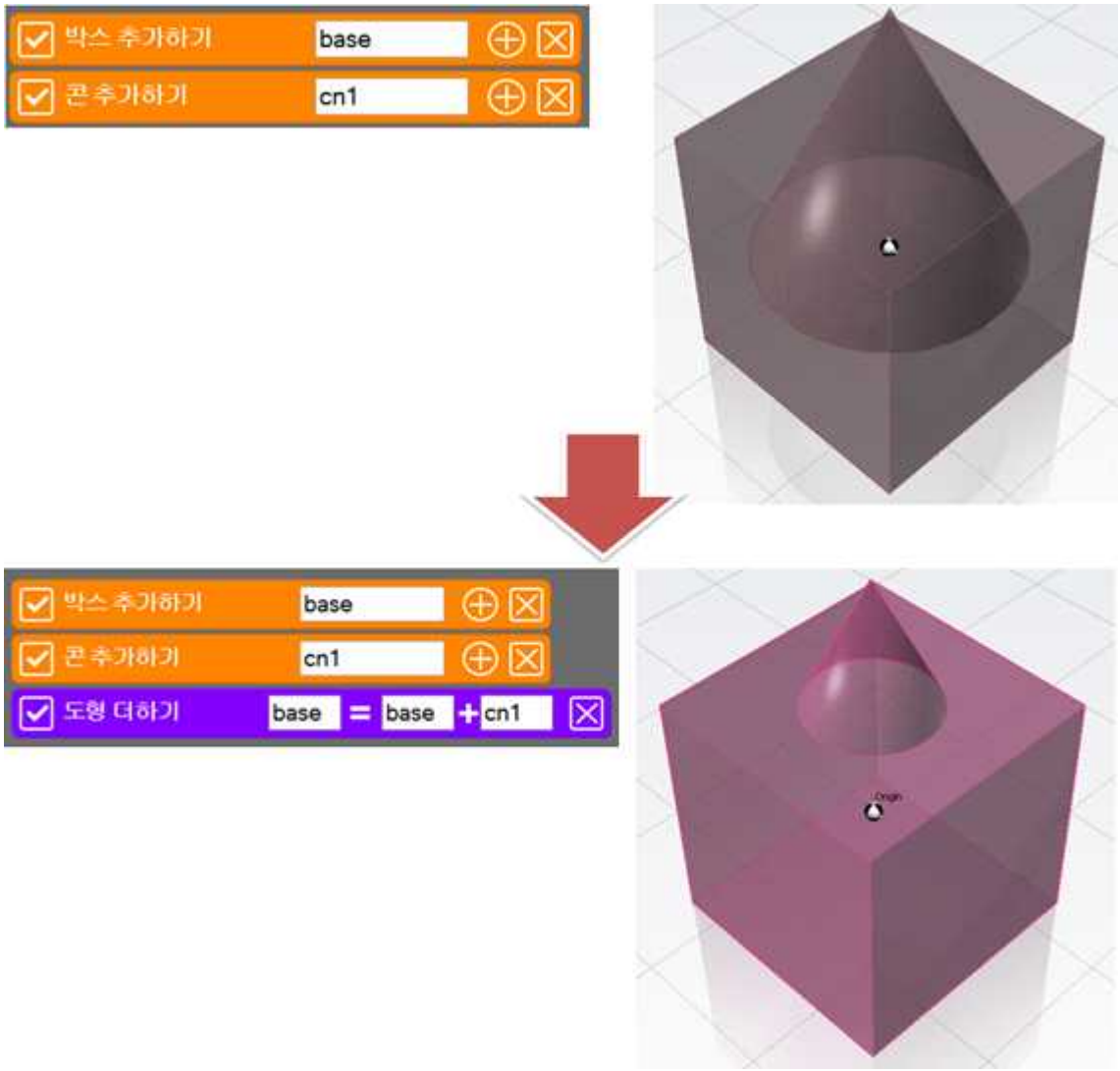
4. SPL For 3D 프린팅 연산 명령어 익히기

4.1 연산 명령어 알아보기



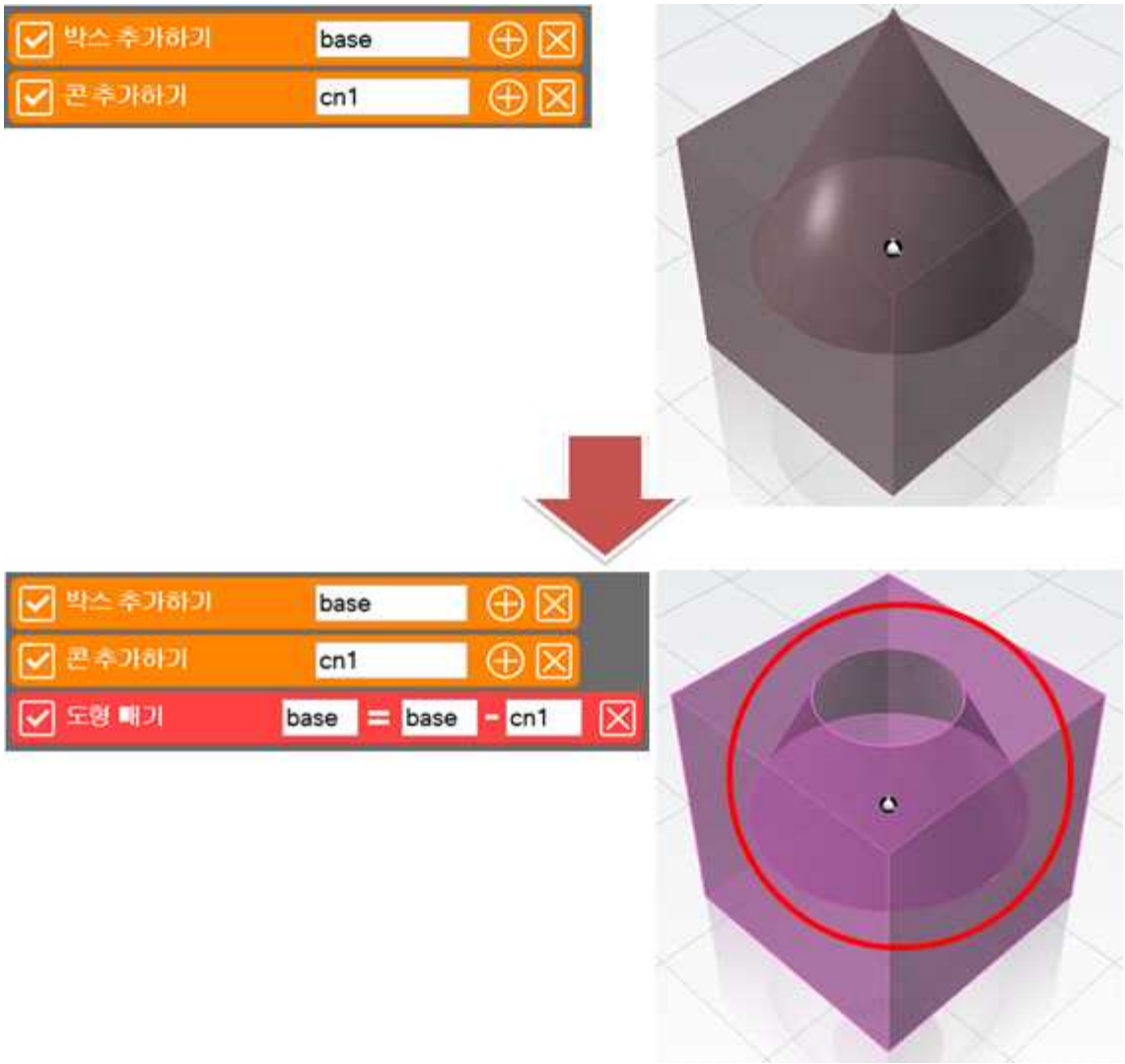
[1-2] 도형 연산 명령어는 두 개의 도형을 이용하여 도형을 더하거나 빼거나 겹치는 도형을 구하는 명령어입니다.

1) 도형 더하기



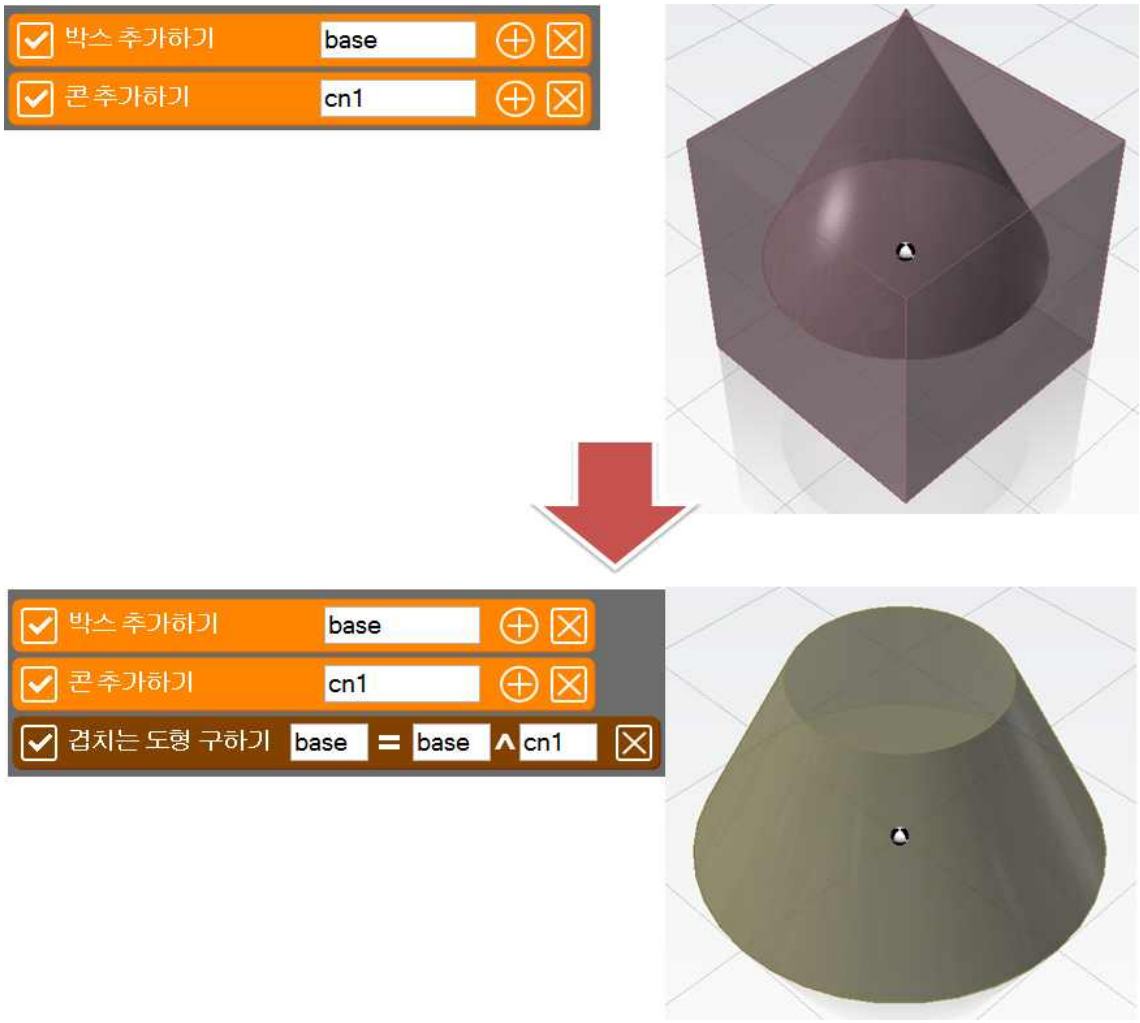
임의의 도형 두 개를 생성하여 **도형 더하기**를 명령어를 수행하면 박스(base)와 콘(cn1)의 겹쳐진 부분이 없어지고 하나의 새로운 도형 base가 생성되게 됩니다.
도형 더하기 “base = base + cn1"에서 ”base + cn1“는 더할 두 도형의 대상의 이름을 입력하고 두 도형의 더한 값 ”base =“를 만들게 됩니다.
여기서 ”base =“의 base는 이름을 바꿀 수 있습니다.

2) 도형 빼기



이번엔 박스(base)와 콘(cn1)을 **도형 빼기** 명령어를 사용하였습니다.
이전의 도형 더하기는 겹쳐진 부분이 없어지고 두 도형이 하나의 도형이 되었다면 도형 빼기의 경우 “base-cn1"의 앞 base에서 겹쳐진 cn1의 부분을 빼서 겹쳐진 부분이 없어진 base라는 새로운 도형을 생성하였습니다.
박스(base)에서 콘(cn1)의 겹쳐진 부분이 없어짐으로써 박스(base)안에 공간이 생겼습니다.
더하기와 마찬가지로 결과값 base의 이름은 바꿀 수 있습니다.

2) 겹치는 도형 구하기



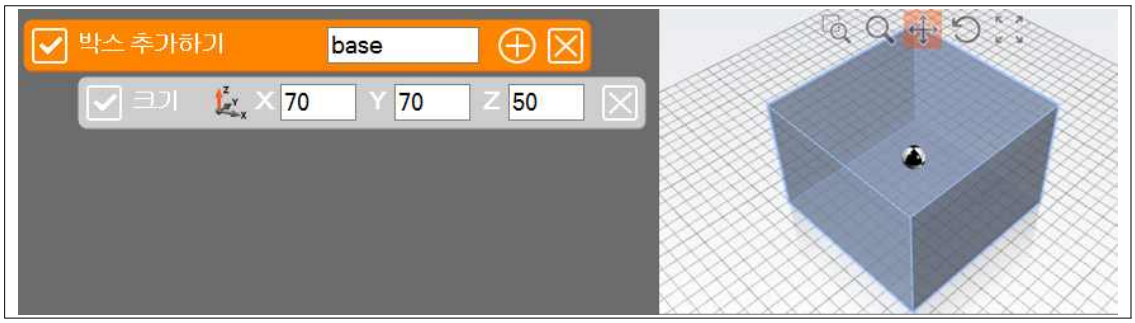
겹치는 도형 구하기는 이제까지 했던 도형 더하기와 도형 빼기에서 겹쳐진 부분만 가지고 오는 명령어입니다.

도형 더하기는 두 도형의 겹쳐진 부분과 나머지를 합치는 명령어였고 도형 빼기는 겹쳐진 부분을 빼는 명령어였다면 겹치는 도형 구하기는 그 겹쳐진 부분의 값을 얻고자 할 때 쓰이는 명령어입니다.

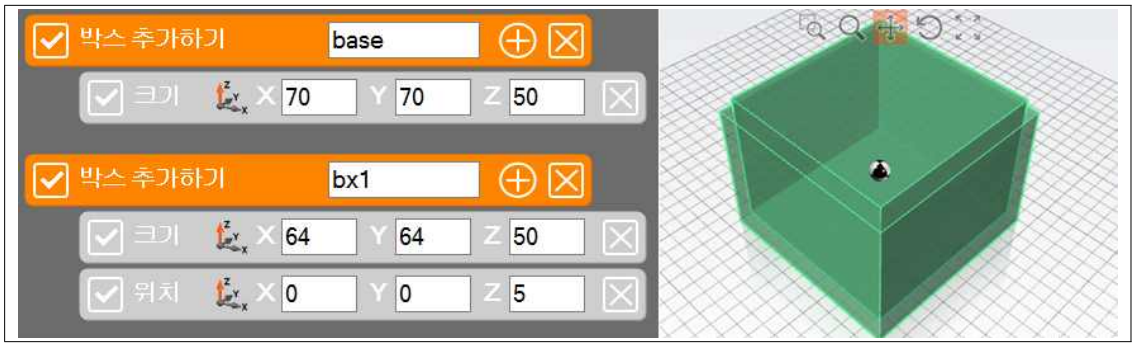
4.2 연산 명령어 활용

1) 연필통 만들기

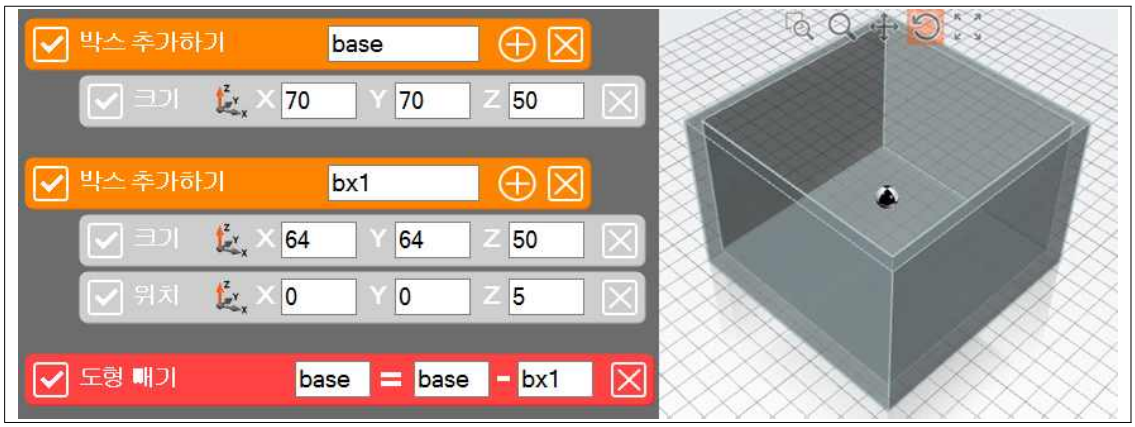
도형 빼기를 이용하여 간단한 연필통을 만들 수 있습니다.



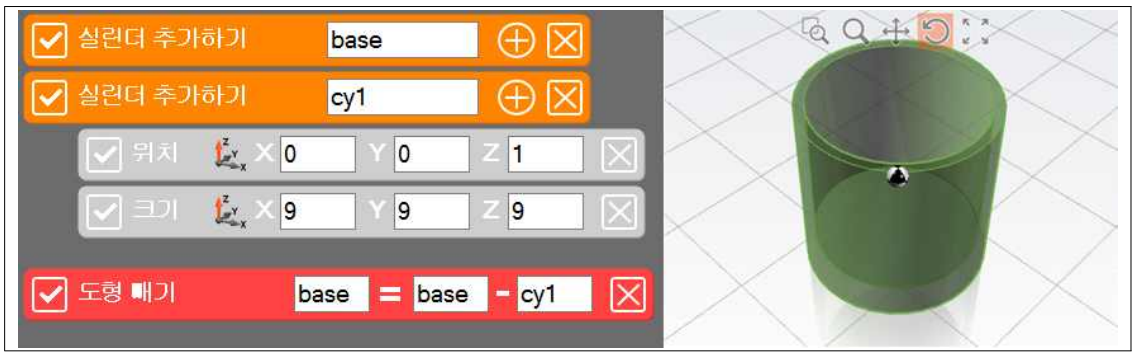
임의의 크기의 박스를 생성해줍니다.



첫 번째 박스(base)보다 작은 박스(bx1)를 만들어줍니다.
작은 박스(bx1)의 위치를 살짝 올려서 큰 박스의 바닥은 남기고 천장은 뚫 수 있게 합니다.



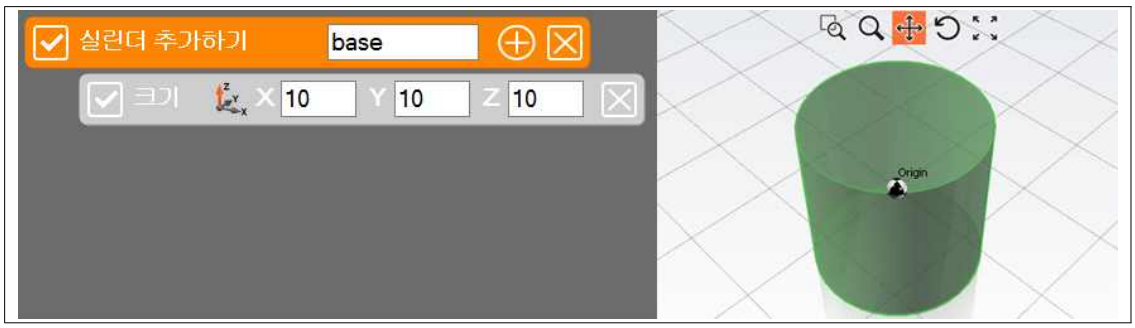
마지막으로 빼기를 해주면 사각 연필통이 완성됩니다.
박스 두 개와 명령어 하나로 간단한 연필통을 만들 수 있습니다.



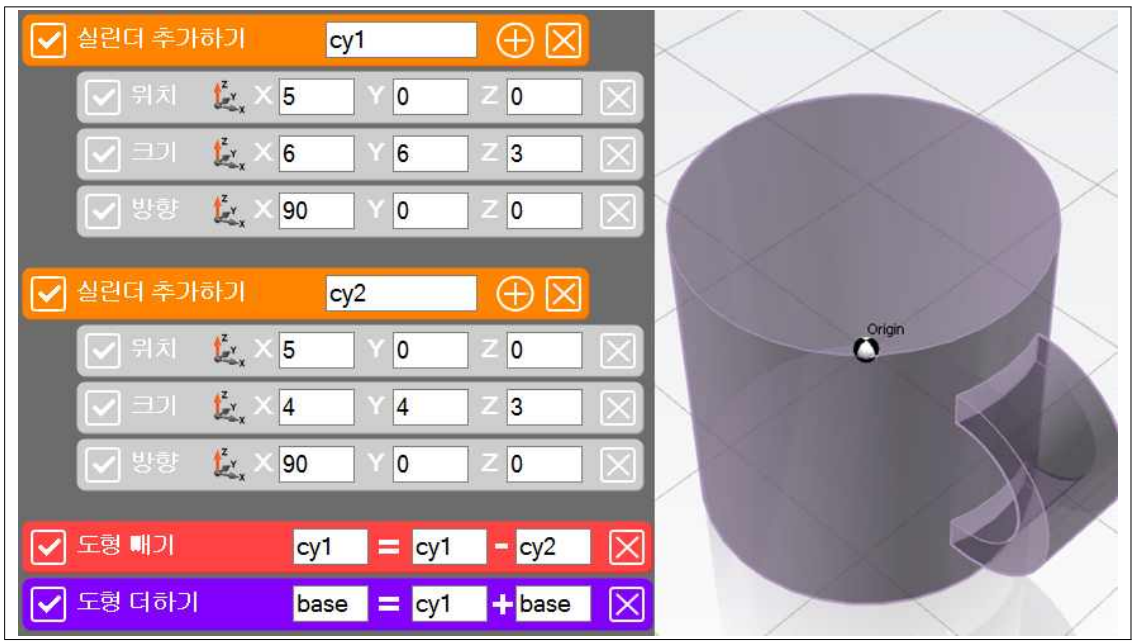
같은 방법으로 도형만 바꾸어 원통으로 된 연필통을 만들 수 있습니다.

2) 머그컵 만들기

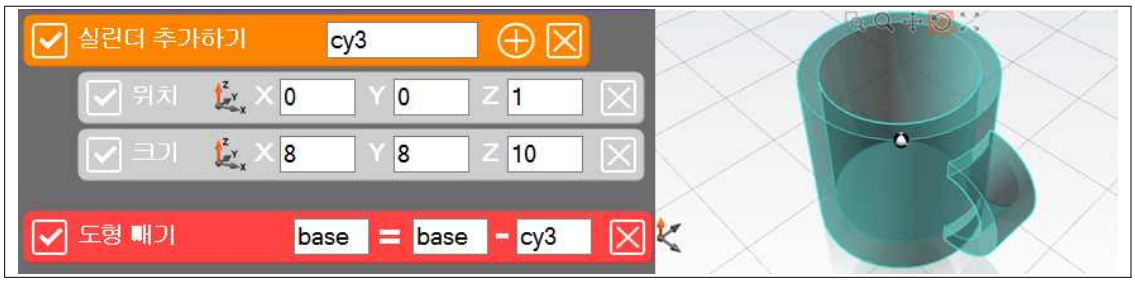
여러 가능을 이용하여 머그컵을 만들 수 있습니다.



컵 모양이 될 실린더를 만들어 줍니다.
크기는 임의로 지정해줍니다.



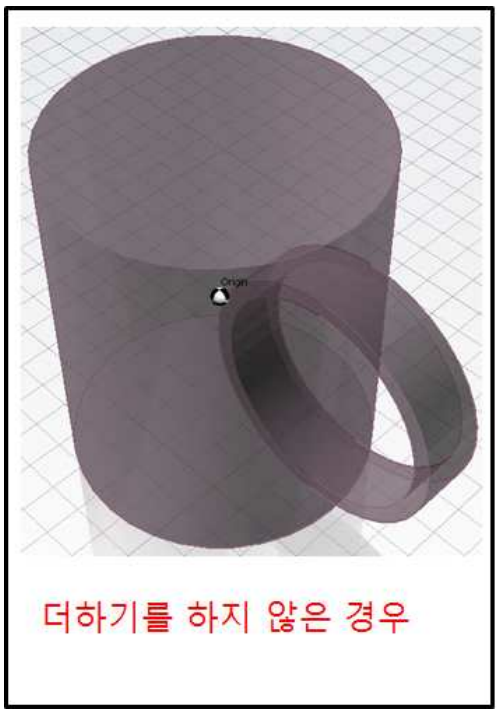
두 개의 실린더(cy1, cy2)를 이용하여 손잡이를 만들어 줍니다.
두 개의 실린더를 눕혀서 큰 실린더(base)의 외각으로 이동 시킨 뒤 두 실린더를 빼줍니다.
그리고 만든 손잡이(cy1)와 큰 실린더(base)를 **도형 더하기(중요)** 해줍니다.



마지막으로 컵의 공간을 만들어줄 실린더(cy3)를 만들어 빼주면 머그컵이 완성됩니다.

※ 중요

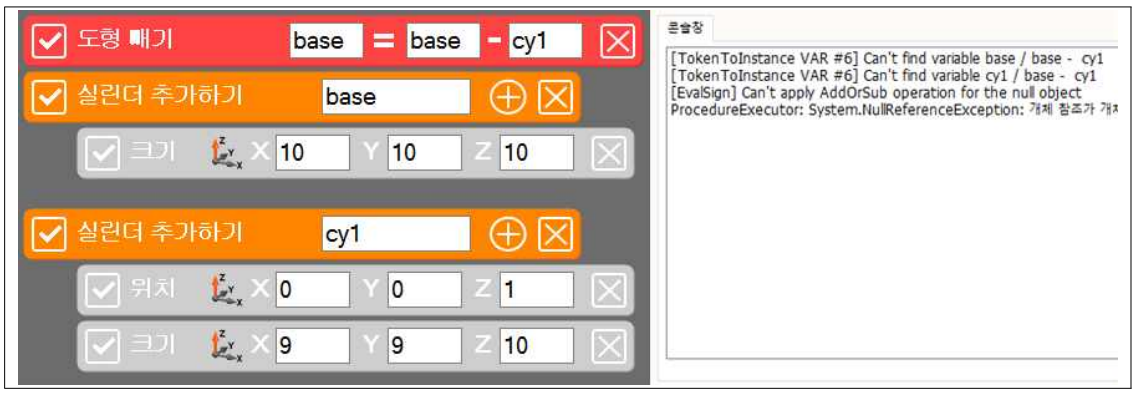
실습 중간에 손잡이와 큰 실린더를 도형 더하기를 해주었습니다.
만일 손잡이(cy1)와 큰 실린더(base)를 더하지 않고 컵을 만들기 위해 작은 실린더(cy3)를 뺄 시 위 그림과 같이 컵 안쪽의 손잡이 부분이 없어지지 않는 현상이 발생합니다.



더하기를 하지 않을 경우 큰 실린더(base)와 손잡이(cy1)이 서로 다른 개체가 됩니다.
만일 여기서 도형 더하기 "base = base + cy1"의 명령어를 사용하게 되면 큰 실린더와 손잡이는 새로운 하나의 도형 base가 되게 됩니다.

※중요2

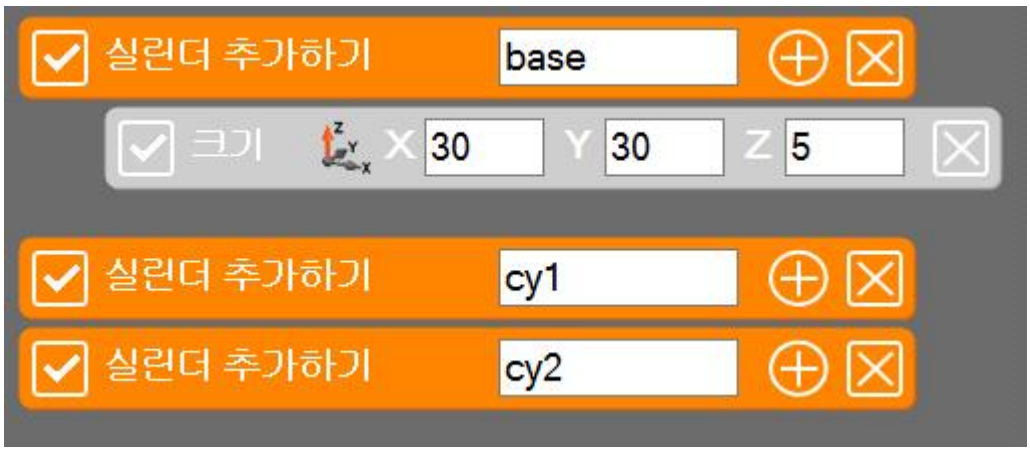
SPL For 3D 프린팅은 SW프로그래밍과 마찬가지로 위에서 아래로 진행하게 됩니다.
아래 그림과 같이 실행을 하게 되면 오류가 나옵니다.



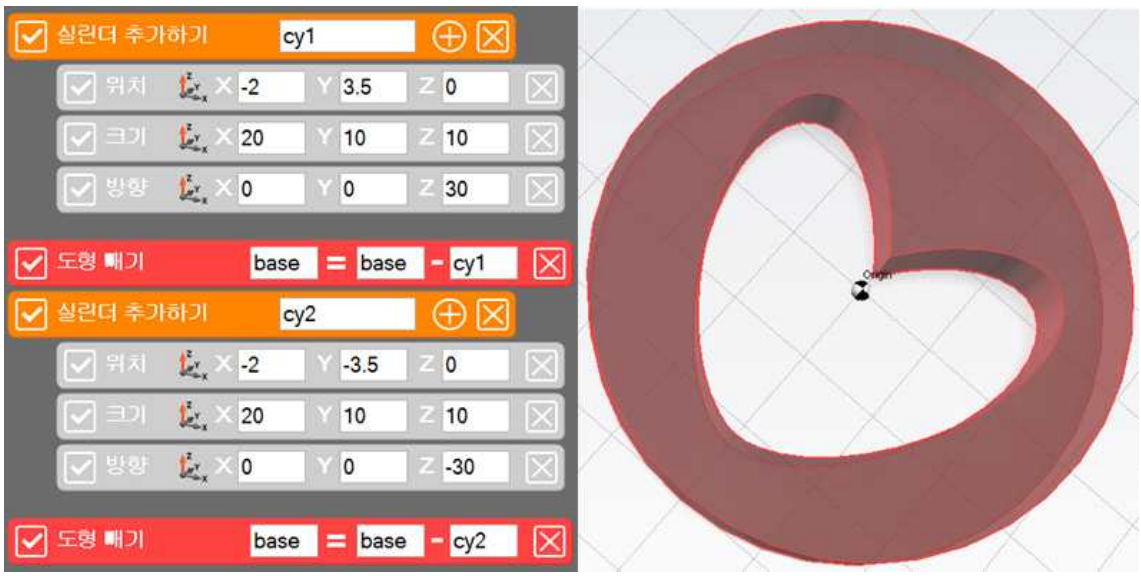
도형 빼기 명령어를 수행하기 위해 base에서 cy1을 빼려합니다.
하지만 프로그래밍의 원칙상 위에서 아래로 진행되기 때문에 도형 빼기 명령어 위에 base와 cy1이라는 도형이 없기 때문에 오류가 발생하게 됩니다.
명령어를 선언할시 사용할 도형이 명령어가 위에 있어야 합니다.

3) 하트 열쇠고리 만들기

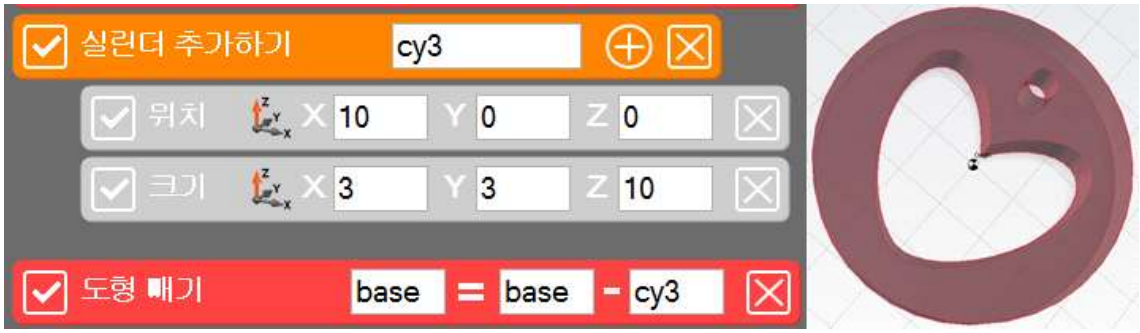
우선 열쇠고리를 만들 넓고 얇은 동그란 판을 하나 만들어 줍니다.



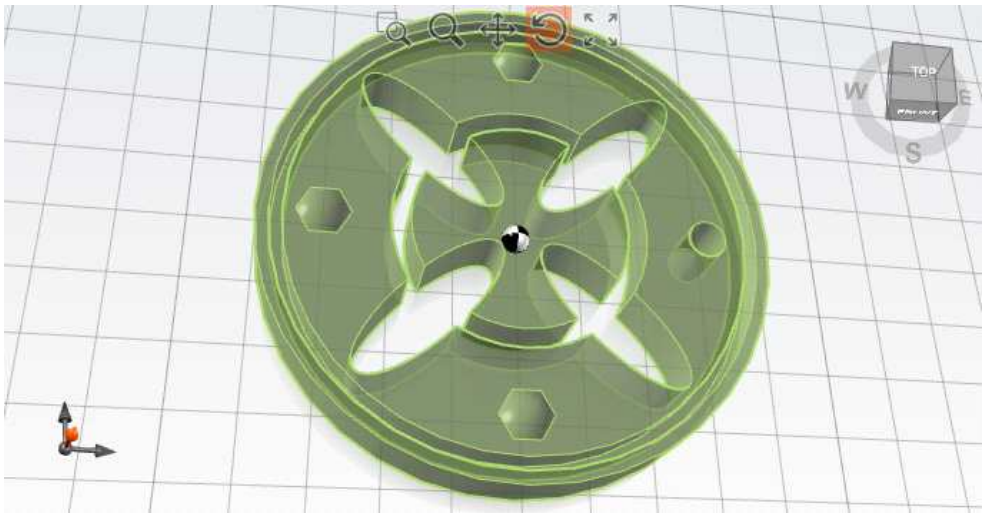
실린더 두 개를 추가해줍니다.
이 두 실린더는 위에서 배웠던 하트 모양을 만들기 위해 사용하게 될 실린더입니다.

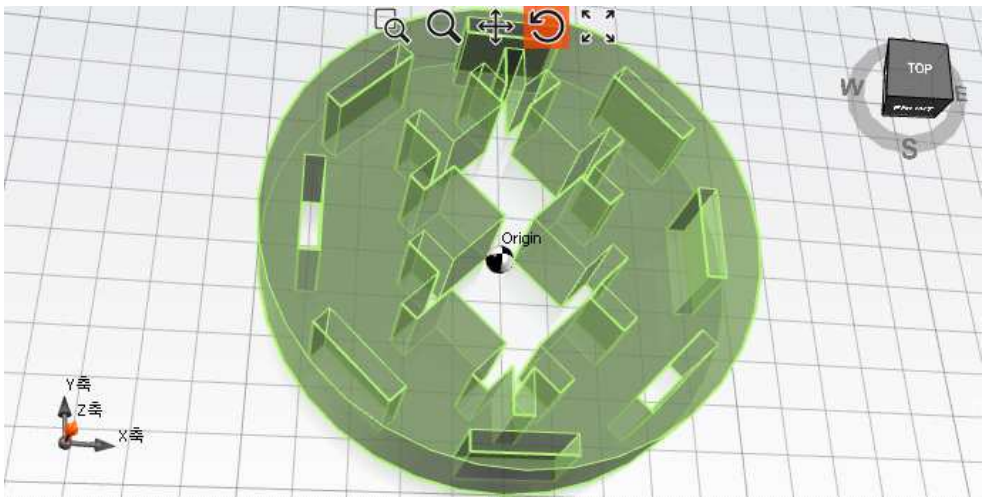


위 그림과 같이 위치, 크기, 방향을 이용하여 하트 모양을 만들어주고 도형 빼기를 이용하여 빼기를 해줍니다.



마지막으로 실린더를 하나 더 추가해줍니다.
이 실린더는 고리가 들어갈 구멍으로 구멍을 파줍니다.
이로써 하트모양 열쇠고리가 완성됩니다.

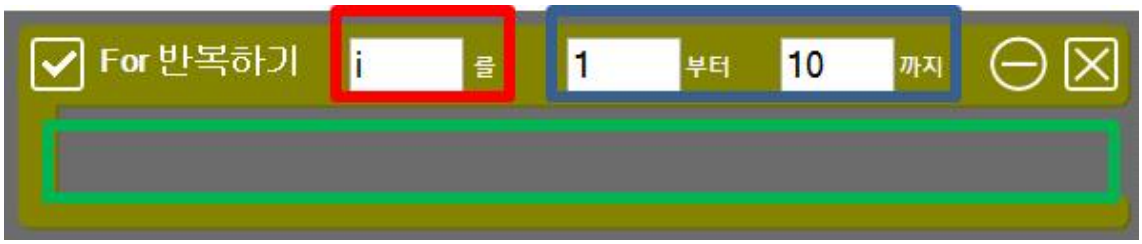




그 외에도 여러 모양으로 열쇠고리를 만드는 것이 가능합니다.

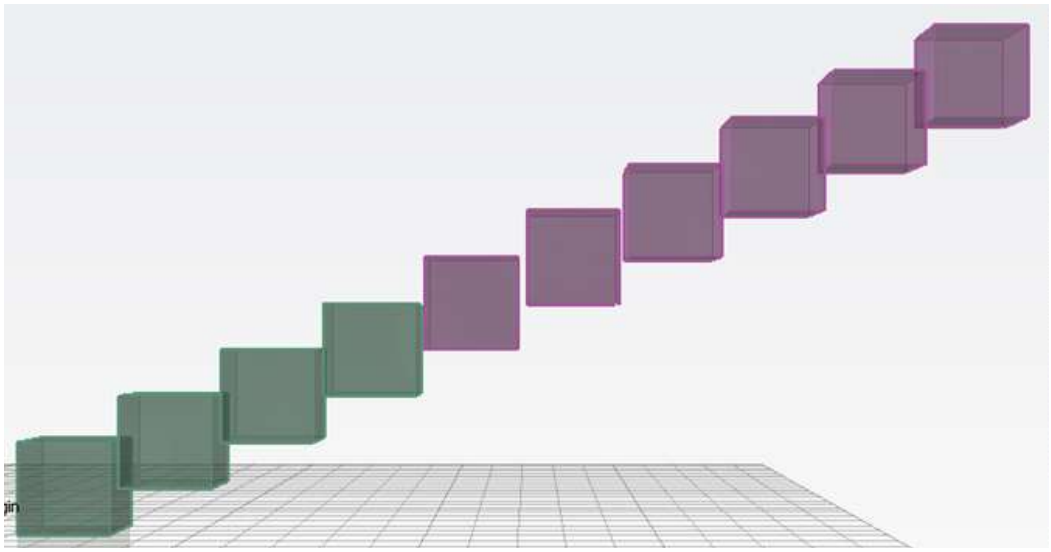
5. SPL For 3D 프린팅 로직 명령어 For 반복문 익히기

5.1 For 반복문 알아보기

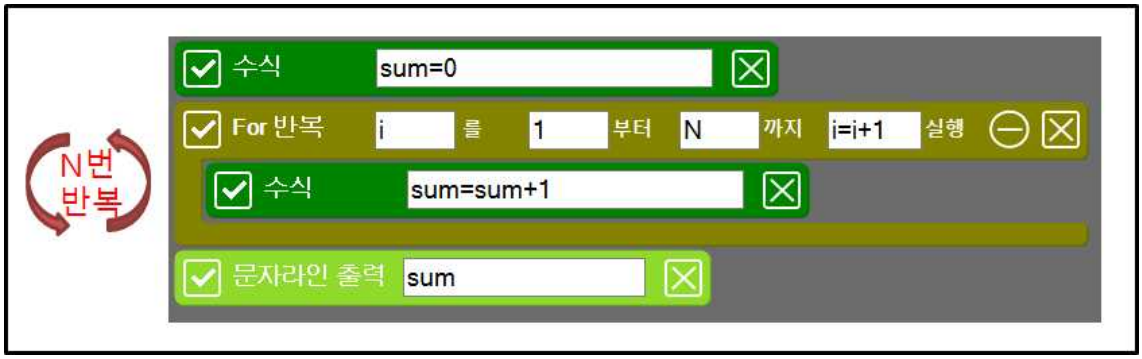


i라는 임의의 이름(변수)을 지정하여
i가 지정한 숫자 1부터 시작하여 10까지
For의 안쪽에 있는 내용을 반복하는 로직 명령어입니다.

For 반복문을 이용하는 이유는?
- 다수의 도형을 생성가능
- 생성된 도형들에게 일정한 규칙을 지정가능

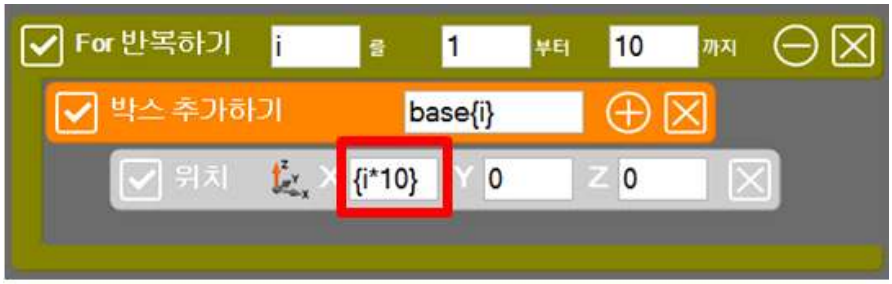


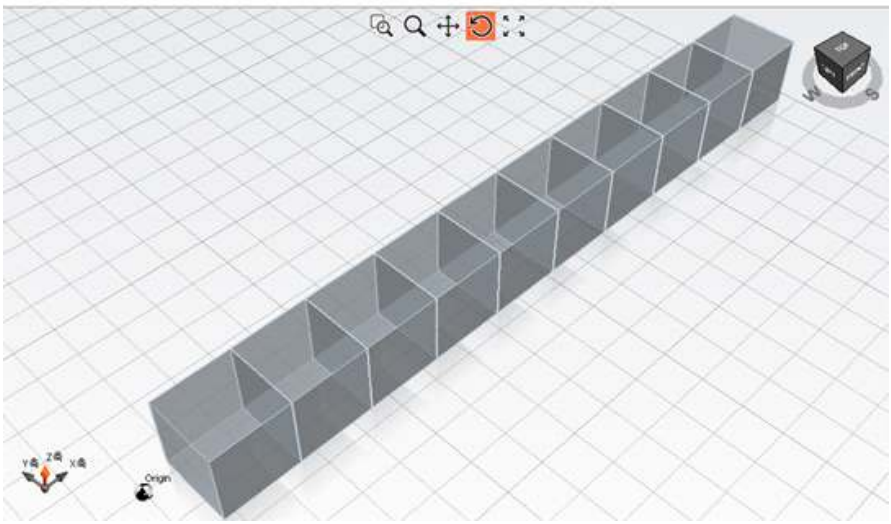
위 그림과 같은 도형을 생성하기 위해선 박스 생성하기를 10번을 해야 하지만 For문을 이용하면 간단히 만들 수 있습니다.



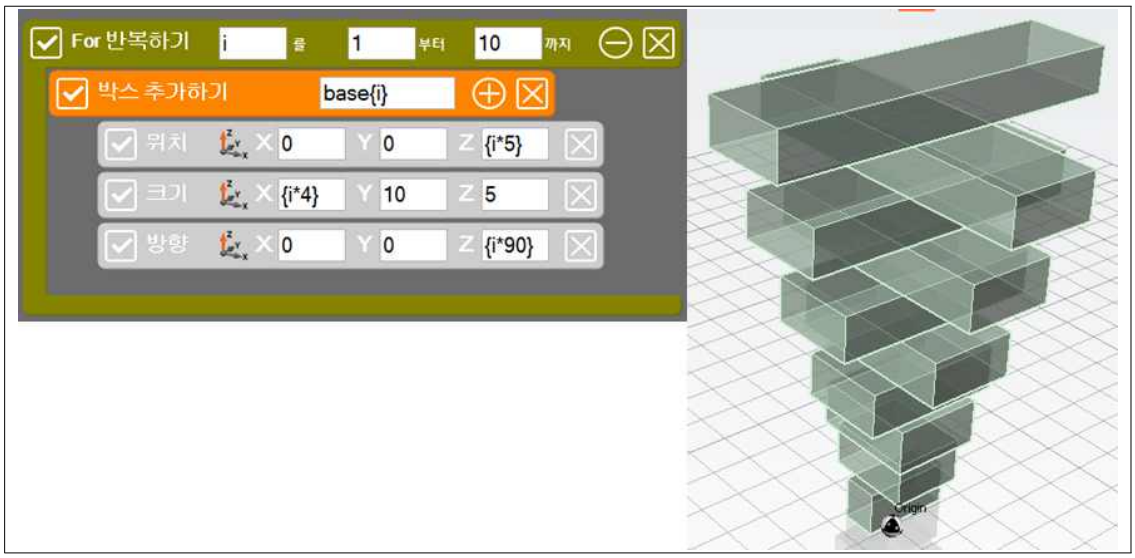
위 그림과 같이 명령어를 실행하게 되면 For문으로 들어갈 시 i가 1부터 시작하여 N번까지 (임의의 지정한 숫자) For 반복 안의 있는 명령어를 i가 N이 될 때까지 실행하게 됩니다.
모든 반복을 수행한 이후 다시 밑 명령어를 수행하게 됩니다.

For 반복문을 도형에 적용시키기 위해선 도형 이름 뒤에 {i}를(i는 For에서 지정한 변수 이름) 붙여줘야 합니다.
붙이지 않을시 똑같은 이름의 도형(base)가 중복되기만 할뿐입니다.
For 반복하기 i가 1부터 시작하여 base1이라는 박스를 만들고 i가 2가되어 base2라는 박스를 다시 만들어 base10까지 박스를 생성합니다.
위 그림은 아무런 이동명령을 내리지 않았기에 10개의 박스가 겹친 상태입니다.





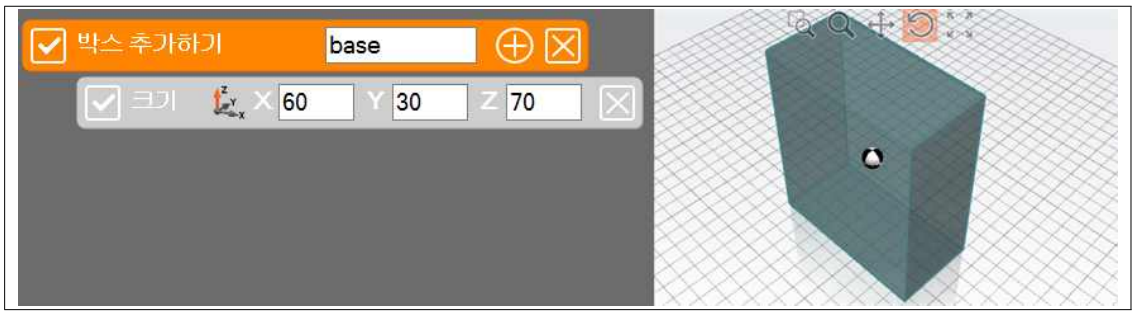
위 그림과 같이 위치를 만들어 X좌표에 {i*10}을 입력하였습니다.
그 결과 겹쳐있던 10개의 박스가 X축 방향으로 일제히 놓이게 됩니다.
For 반복하기 i가 1일 때 base1를 생성하여 X축 위치 10을 지정하게 되고 i가 2일 때 base2를 생성하여 X축으로 위치값 20이 들어가게 됩니다.
마지막엔 박스 base10가 X축 100까지 이동하게 됩니다.



위 그림과 같이 Z축에 값을 지정하여 위로도 올릴 수 있습니다.
또한 크기도 변환이 가능하며 회전을 할 수 있습니다.

5.2 For 반복문 활용

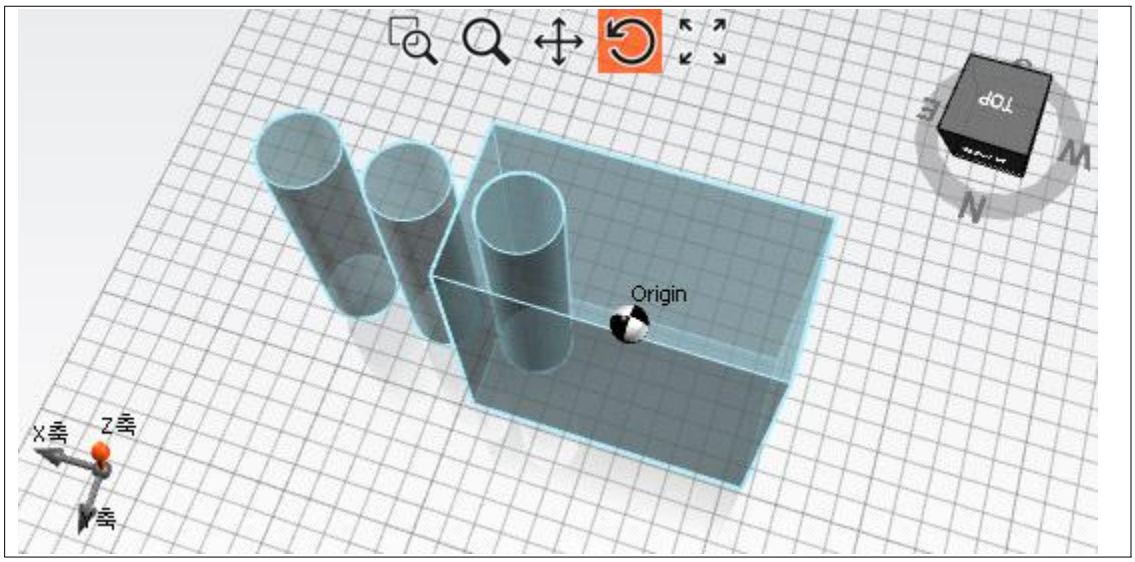
1) 칫솔 보관함 만들기



임의의 크기 박스를 하나 생성해 줍니다.

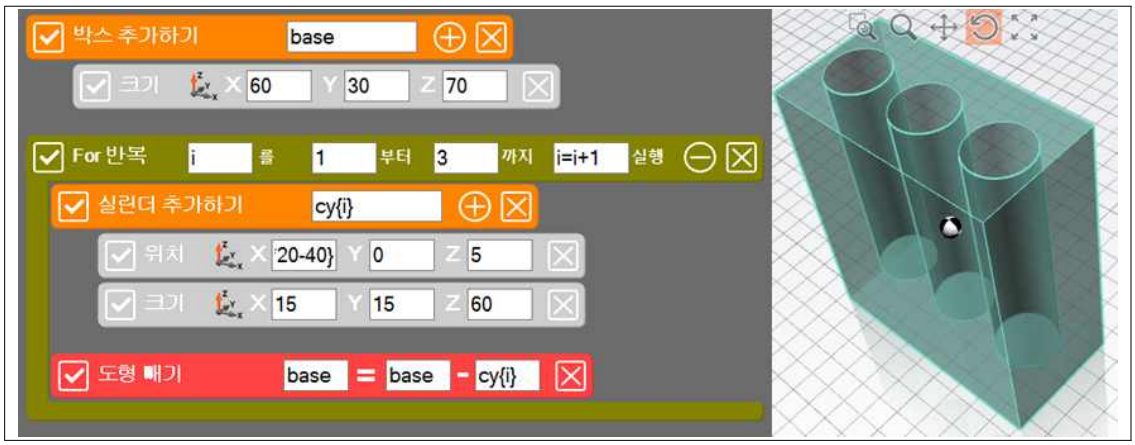
For 반복을 생성해줍니다.
이 반복문으로 칫솔들이 들어갈 구멍을 만들 예정입니다.

칫솔이 들어갈 공간을 만들기 위해 실린더를 생성하였습니다.
구멍은 3개를 만들기 때문에 반복에서 1부터 3까지로 수정해줍니다.
이제 위치 X를 {i*20} 씩 이동 시켜줍니다.



하지만 결과창을 보시면 박스의 위치와 실린더의 위치가 어긋나 있는걸 확인 할 수 있습니다.
해결 방법은 박스(base)를 이동을 시키던지 실린더(cy1, cy2, cy3)를 이동 시켜야 합니다.
여기선 실린더를 이동 시키도록 하겠습니다.

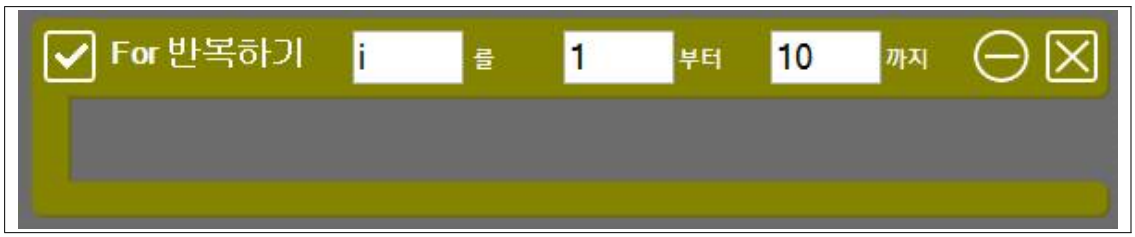
위치 X를 {i*20-40}으로 수정해줍니다.
다른 방법은 수직을 이용하는 방법인데 수식은 다른 페이지에서 설명하도록 하겠습니다.



마지막으로 빼기를 해줍니다.
For 반복문을 이용하면 한 번에 연산도 가능합니다.

2) 톱니바퀴 만들기

For 반복하기를 이용하여 톱니바퀴를 만들어 줄 수 있습니다.



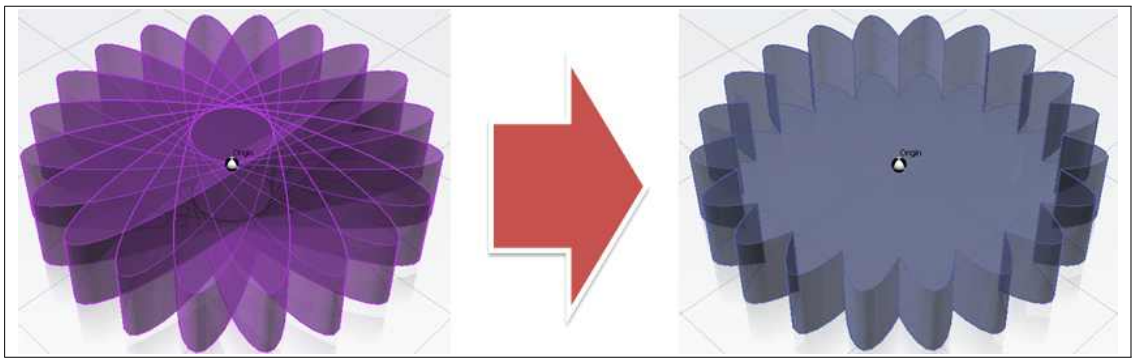
For 반복하기를 하나 생성합니다.

실린더를 하나 생성해줍니다.
생성한 실린더에 {i}를 추가해 줍니다.
현재 실린더는 10개의 실린더가 (0,0,0)에 위치에 있습니다.

실린더에 방향을 추가해줍니다.
방향 Z를 {i*18}로 입력하면 18배수만큼씩 실린더(base1~10)가 회전을 하게 되어 그림과 같은 톱니바퀴를 얻을 수 있습니다.

For 반복 위에 박스 하나를 추가해줍니다.
이 박스는 실린더보다 작게 만들어줍니다.

For 반복 안에 도형 더하기 명령어를 추가합니다.
실린더와 마찬가지로 더하기 명령어 역시 For 반복문 안에 있으면 반복하게 됩니다.
박스 도형 bx1를 기준으로 모든 실린더를 더하게 됩니다.



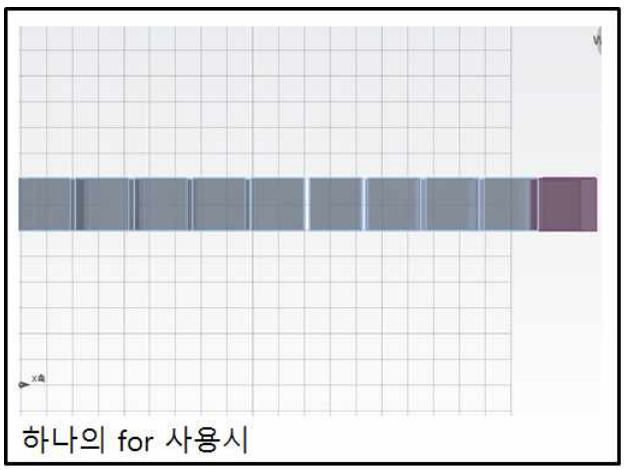
한 번의 명령어로 수많은 도형들이 하나의 개체가 되었습니다.
For 반복문을 이용하면 위와 같이 여러 도형도 한 번에 합칠 수 있습니다.

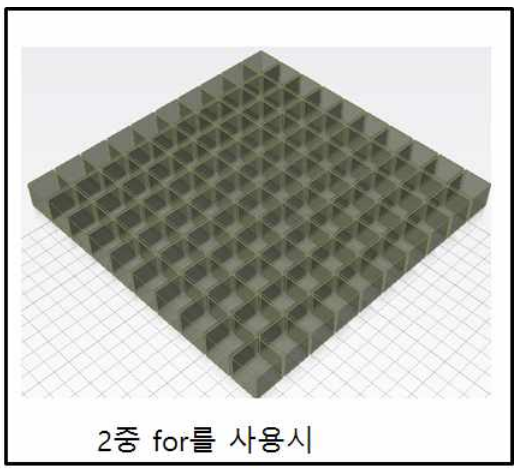
6. SPL For 3D 프린팅 로직 명령어 2중 For 반복문 익히기

6.1 2중 For 반복문 알아보기

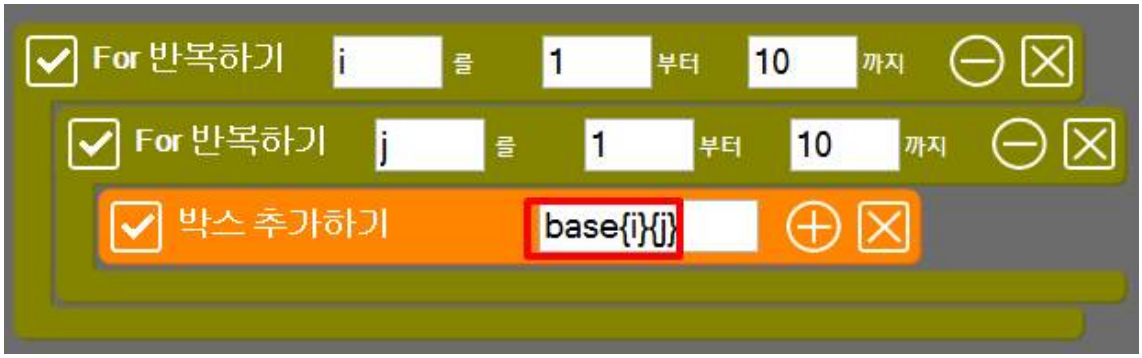
2중 For문이란?

For 반복문안에 또 다른 For문을 넣어주는 구조입니다.

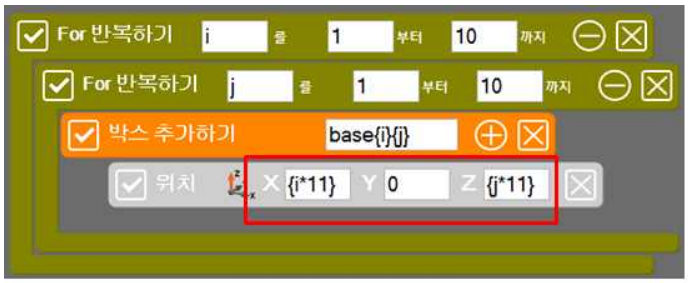




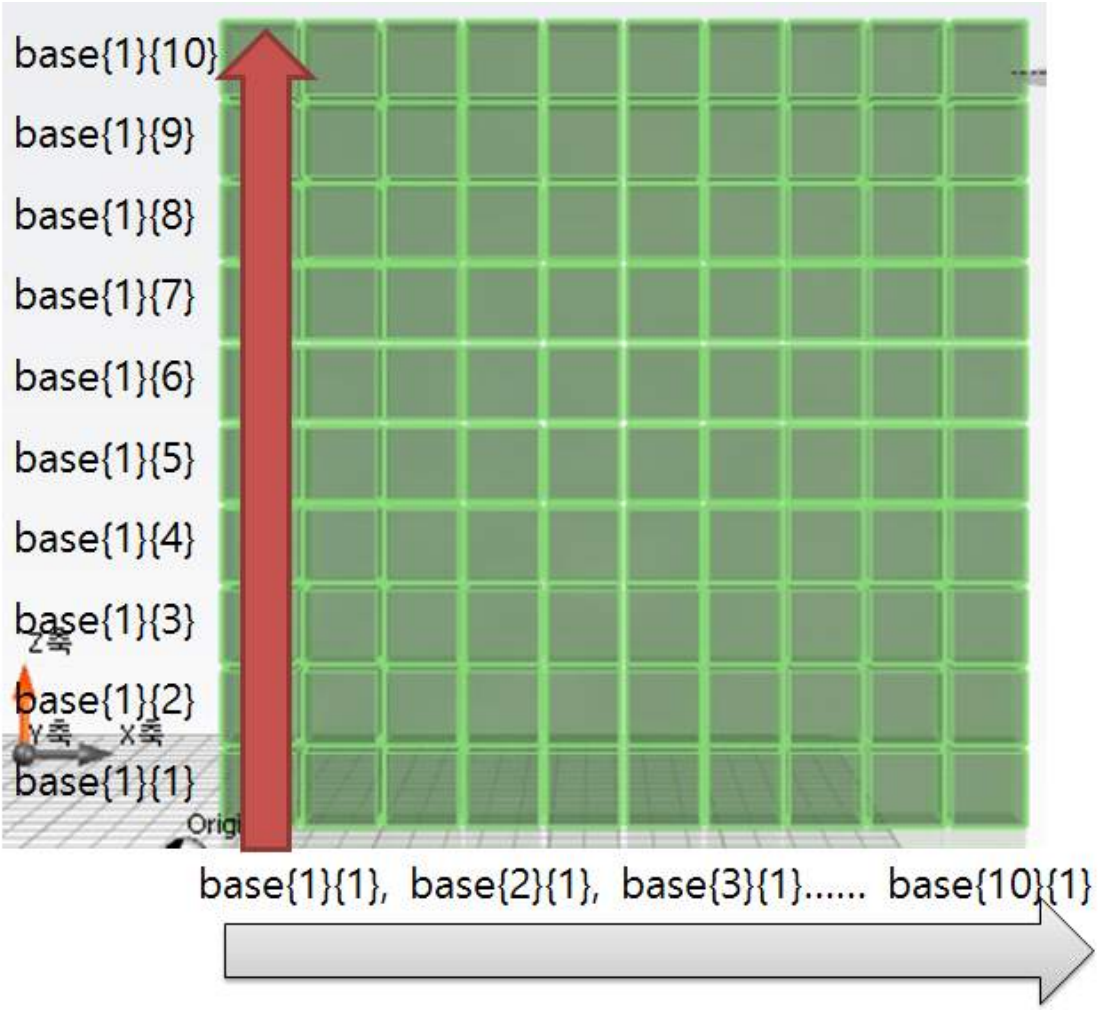
For 반복문 하나만 있으면 선(1차원)이지만 2중 For문을 구사하면 평면(2차원)이 됩니다.
두 개의 For 반복문을 이용하여 더 많은 도형과 명령을 수행이 가능합니다.



For 반복문과 마찬가지로 2중 For 반복문에선 두 개의 For문이 존재하므로 각각의 For 반복문의 변수들인 {i}{j}를 사용합니다.
2중 For 반복문의 경우 우선 밖에 있는 For 반복문 i가 한번 실행되고 안쪽에 있는 For 반복문 j가 1부터 10까지 실행을 한 뒤 다시 i로 올라와서 i가 2가 되고 다시 j로 들어가서 다시 1부터 10까지 반복합니다.
이것을 i가 10이 될 때까지 반복하게 됩니다.



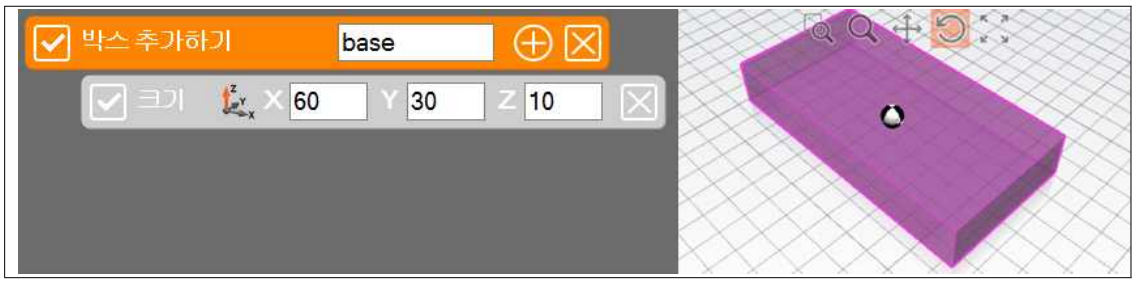
2중 For 반복문 안에 박스 base{i}{j}를 만들어 박스의 위치 X를 {i*11}로 지정하여 i가 증가할 때마다 base의 박스의 X위치가 11배수만큼 이동하게 됩니다.
위치 Z에 {j*11}를 입력하여 j가 증가 할 때 마다 Z축으로 11배수만큼 박스가 이동하게 됩니다.
하나의 For 반복문으로는 한 줄만 만들 수 있지만 2중 For 반복문을 사용함으로써 한 차원 더 이용이 가능하게 되어 집니다.



2중 For 반복문안에 박스 base{i}{j}의 위치 X : {i*11}, Y : 0, Z : {j*11}로 지정하면 위 그림과 같이 진행되게 된다.

6.2 2중 For 반복문 활용

1) 레고 블록 만들기



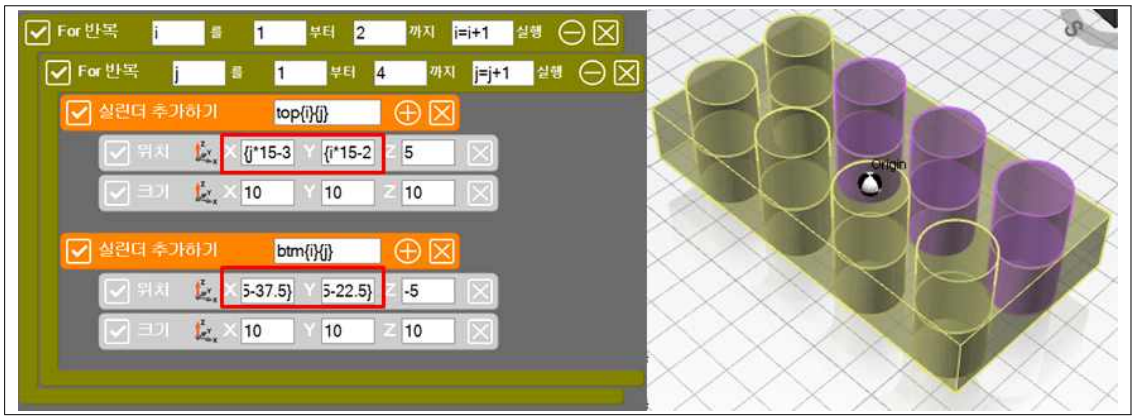
기준이 될 박스를 만듭니다.
이 박스 위아래에 블록을 결합 할 수 있는 결합부를 만들어야 합니다.

2중 For 반복문을 생성합니다.

결합부의 개수를 정해줍니다.
1부터 시작 시 i와 j의 값의 최대값을 곱하면 총 반복되는 횟수입니다.
6개의 결합부를 만들어야 하니 i와 j는 각각 2와 4으로 선언합니다.

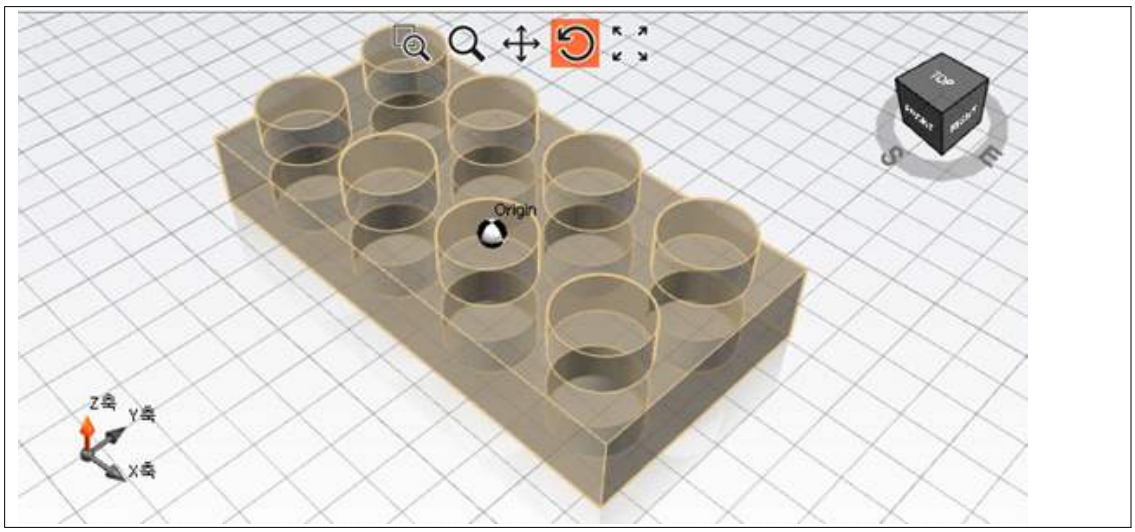
두 개의 실린더를 생성합니다.
하나의 실린더는 상단의 돌기를 생성하여 더할 도형이며 하나는 하단에 들어갈 공간을 만들어 줄 도형입니다.

실린더의 이름을 바꾸어 줍니다.
두 도형이 cy{i}{j}이면 중복이 되니 서로 다른 이름으로 바꾸어 주셔야 합니다.



각각의 실린더의 위치를 지정해줍니다.
X축은 {j*15-37.5} Y축은 {i*15-22.5} Z축은 5와 -5를 입력합니다.

마지막으로 top{i}{j} 실린더는 더하기를 btm{i}{j} 실린더는 도형 빼기를 해줍니다.



레고 블록이 완성된 모습입니다.
상단에는 실린더가 나와 있고 하단에는 실린더가 뚫려 있습니다.
2중 For문을 이용하여 하나하나 실린더를 추가하고 더하고 빼지 않고 간단한 명령으로 여러 도형을 생성하고 연산 할 수 있습니다.

2) 컵케익 만들기

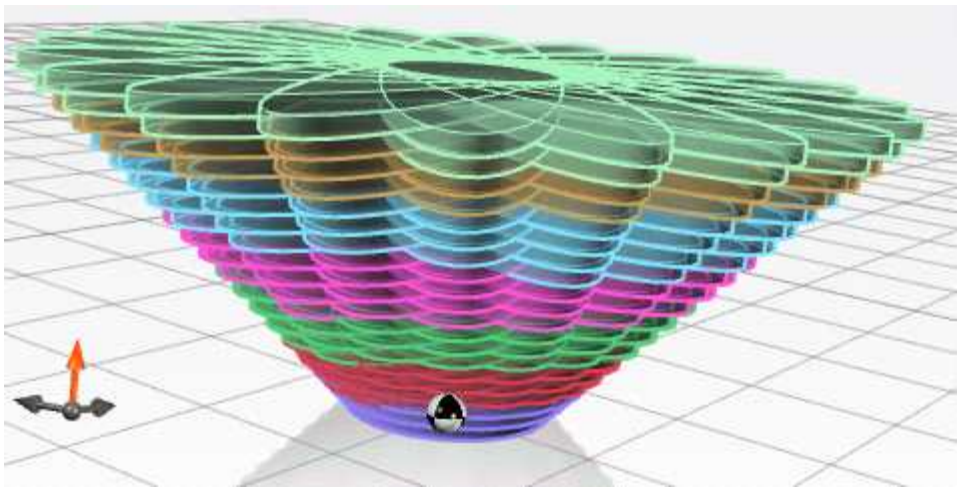
2중 For문을 사용하여 컵케익 모양을 만들 수 있습니다.

일단 2중 For 반복문을 생성합니다.

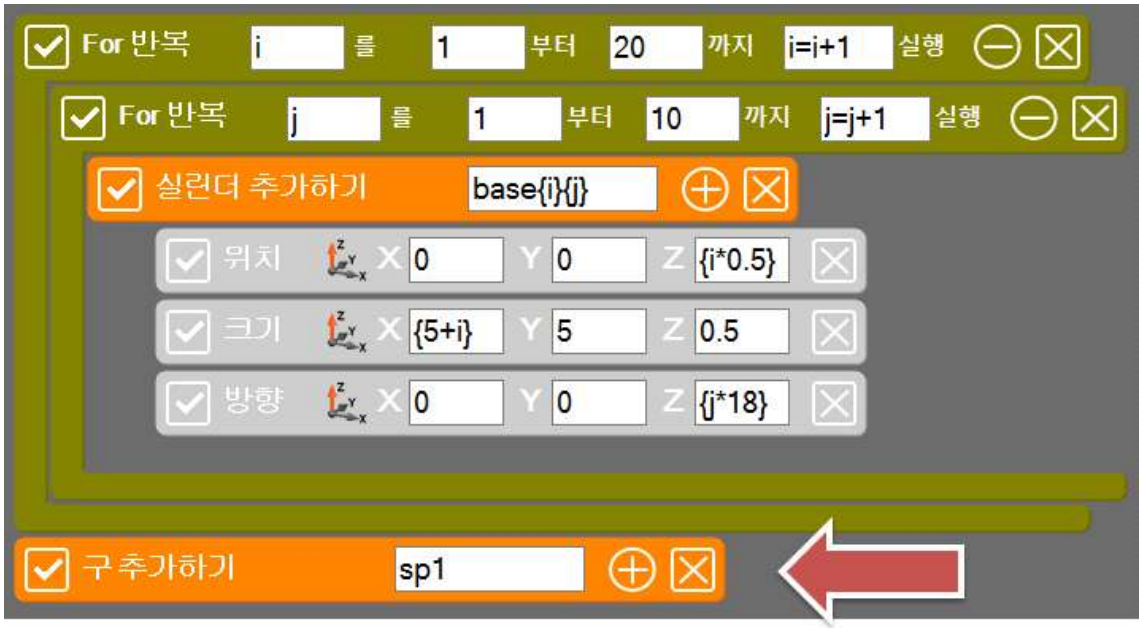
반복문 안쪽에 실린더를 추가하여 위치, 크기, 방향을 생성해줍니다.

2중 For문은 위 그림과 같은 성질을 기자고 있는 것을 이용해야합니다.

반복문 i의 최대값을 20까지 늘려줍니다.
실린더의 이름에 {i}{j}를 추가해줍니다.
위치는 {i*0.5}로 지정하여 i가 증가할 때마다 Z축으로 올라가게 됩니다.
실린더의 크기 X는 {5+i}로 Y는 5 Z는 0.5로 입력하여 주고 방향은 Z 축으로 j가 증가할 때마다 {j*18}씩 늘어나게 됩니다.

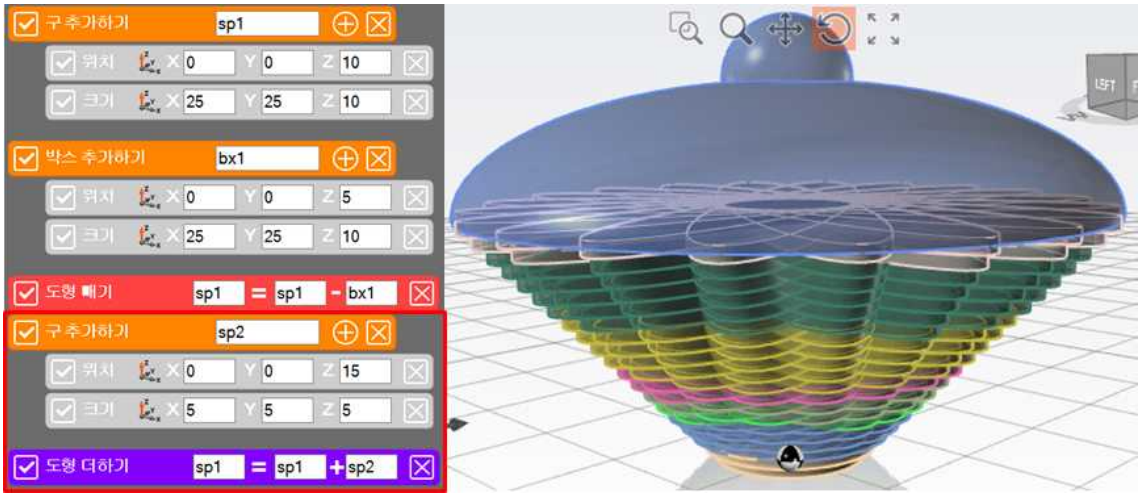


결과 창에서 위 그림과 같은 모습을 확인 할 수 있습니다.



For문 밖에 구를 추가해줍니다.

구를 2중 For문에서 만든 톱니바퀴의 최종적인 크기만큼 키워주고 반만 걸칠 수 있도록 높이를 높여줍니다.
박스는 구를 반원으로 만들기 위해 만든 박스로 크기와 위치를 조정한 뒤 도형 빼기 sp1 = sp1 - bx1을 하여 구를 반원으로 만들어 줍니다.

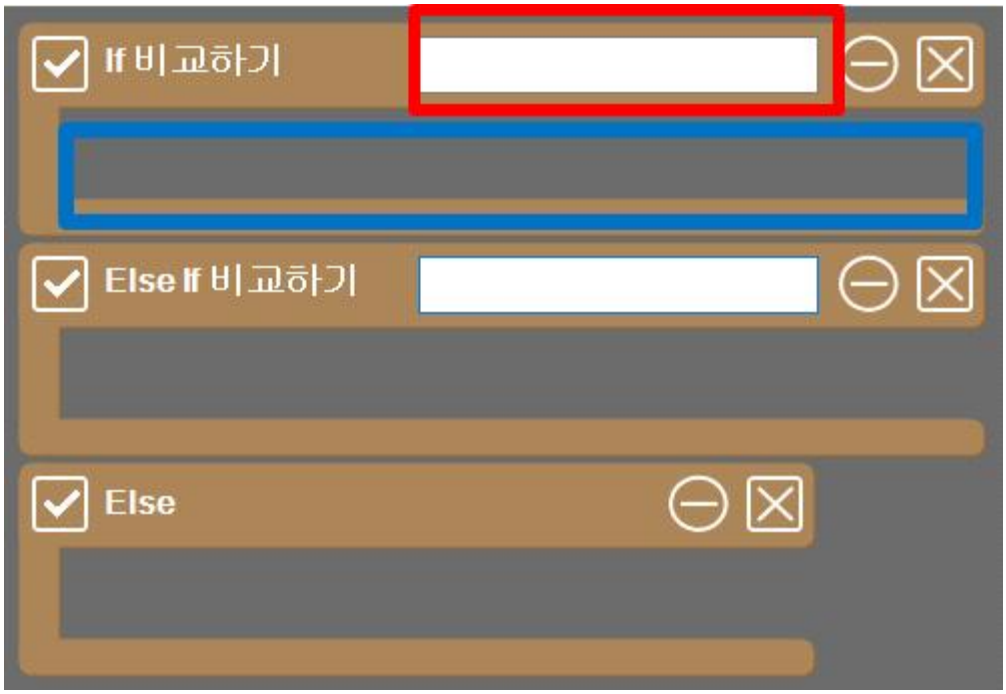


마지막으로 구를 하나 더 만들어 줍니다.
이 구는 컵케익에 맨 위에 올라가는 장식물의 역할을 합니다.
완성이 되면 위 그림과 같은 컵케익 모형을 만들 수 있습니다.

7. SPL For 3D 프린팅 로직 명령어 If 조건문 익히기

7.1 If 조건문 알아보기

If 조건문이란?



특정 조건을 지정하여 그 조건이 맞으면 If안에 있는 명령어들을 실행 하는 명령어 Else If는 첫 번째 If와 다른 조건을 사용할 경우 추가하여 사용합니다.
Else는 조건에 해당하지 않는 남은 경우의 수들을 Else에서 실행하게 됩니다.

조건문에 사용하는 연산자

(위 그림은 예시 입니다)

- i <= 3 : i가 3보다 작고 같을 경우
- i >= 7 : i가 7보다 크고 같을 경우
- i != 5 : i가 5가 아닐 경우
- i == 5 : i가 5일 경우

조건문에 사용하는 연산자

(위 그림은 예시 입니다)

- i > 3 && i < 6 : &&는 and를 의미 i가 3보다 크고 6보다 작을 경우
- i > 8 || i < 2 : ||는 or를 의미 i가 8보다 크거나 2보다 작을 경우
- i % n : %는 n만큼 나누어 나머지를 구하는 수식으로 홀수, 짝수, 배수를 구할 시 사용

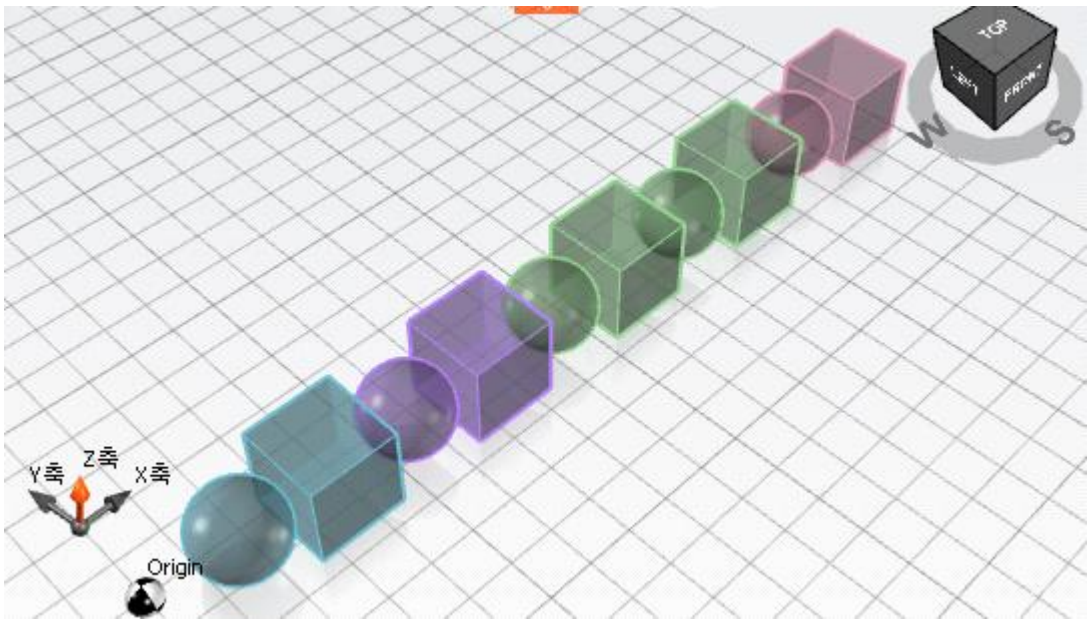
간단한 사용법입니다.

For 반복문을 하나 만들어 줍니다.

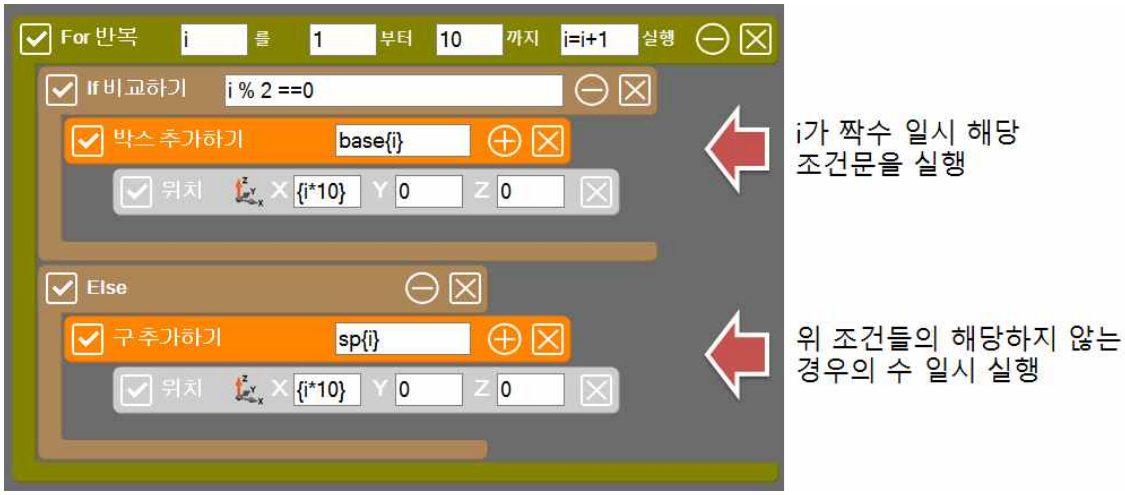
For 반복문 안에 If비교하기와 Else를 추가해줍니다.
i의 홀수 짝수를 구하는 조건문을 만들도록 하겠습니다.

If비교하기 박스에 i % 2 == 0을 입력합니다.
i를 2로 나누어 0이면(짝수) 해당 조건문을 실행시킵니다.
그렇지 않을시 전부 Else로 가게 됩니다.

조건문에 들어가는 확인하기위해 각 조건문에 다른 도형을 생성합니다.



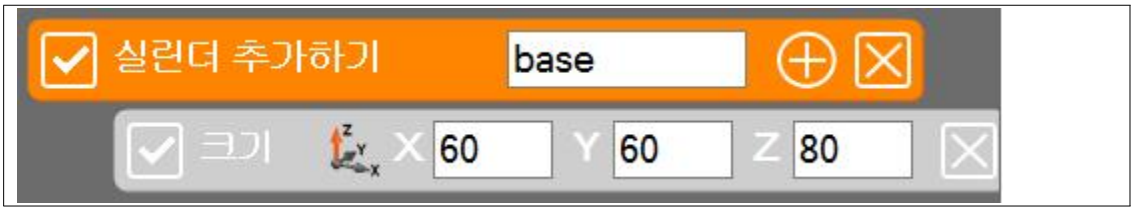
실행을 하면 위 그림과 같은 결과를 얻을 수 있습니다.
하나하나 간격으로 다른 도형들이 배열된 모습입니다.



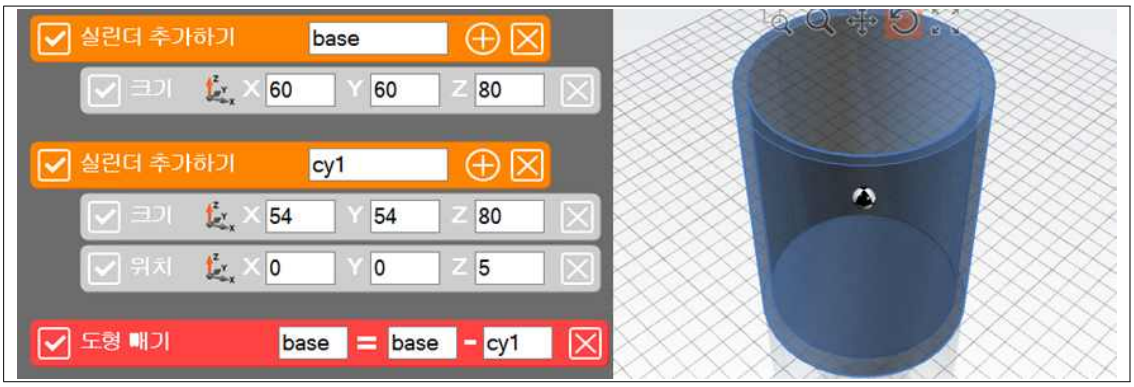
If비교하기 'i%2==0'에선 i가 2, 4, 6, 8, 10일 때 조건문에 들어가서 안에 있는 내용을 실행하게 됩니다.
그 외의 수는 Else로 가게 됩니다.

7.2 If 조건문 활용

1) 그물 모양 연필꽂이 만들기



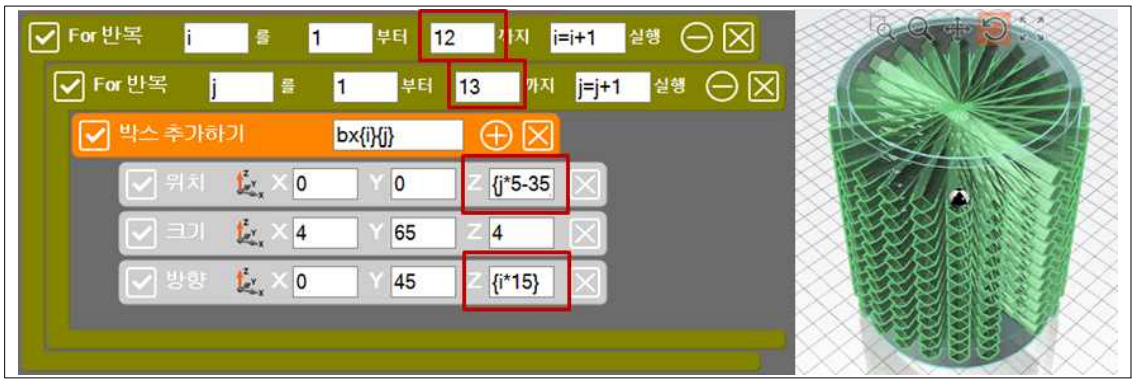
임의의 크기의 실린더를 하나 만들어줍니다.
이 실린더는 연필꽂이의 원통이 될 실린더입니다.



큰 실린더(base)에서 작은 실린더(cy1)을 빼줍니다.
이제 이 원통(base)에 옆면에 그물 모양을 만들어 줘야합니다.

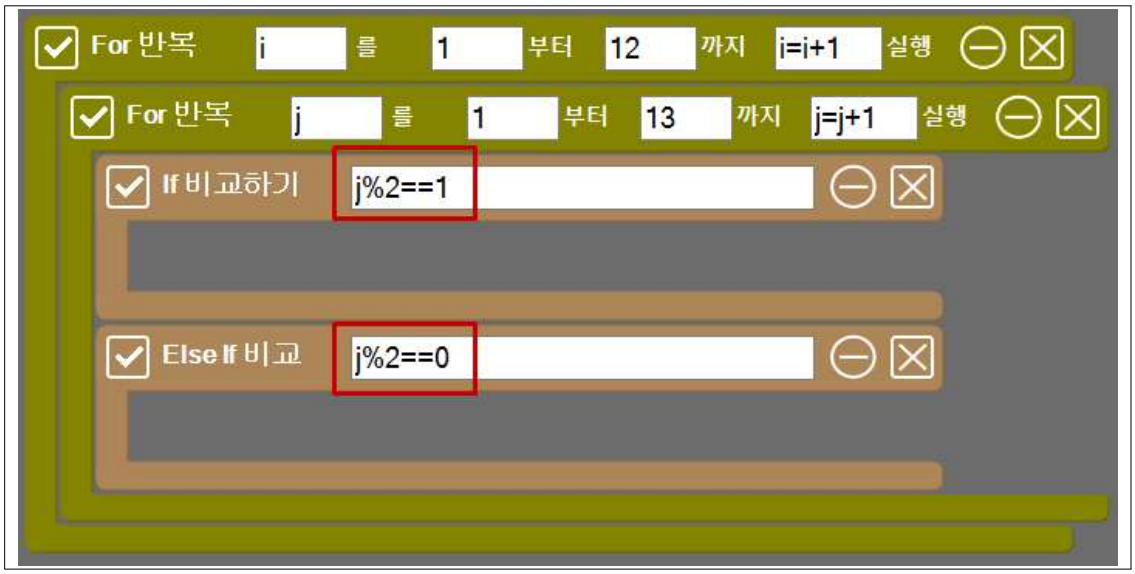
2중 For 반복문을 만들어줍니다.
여러층에 거쳐 모양을 만들어줘야 하기 때문에 2중 For문을 사용합니다.

2중 For문안에 얇고 긴 막대를 만들어 줍니다.
방향을 이용하여 살짝 틀어주어 다이아몬드 모양으로 만듭니다.
이 막대기를 반복문을 이용하여 다수의 막대기를 만들어 원통의 외각을 깎아내야 합니다.

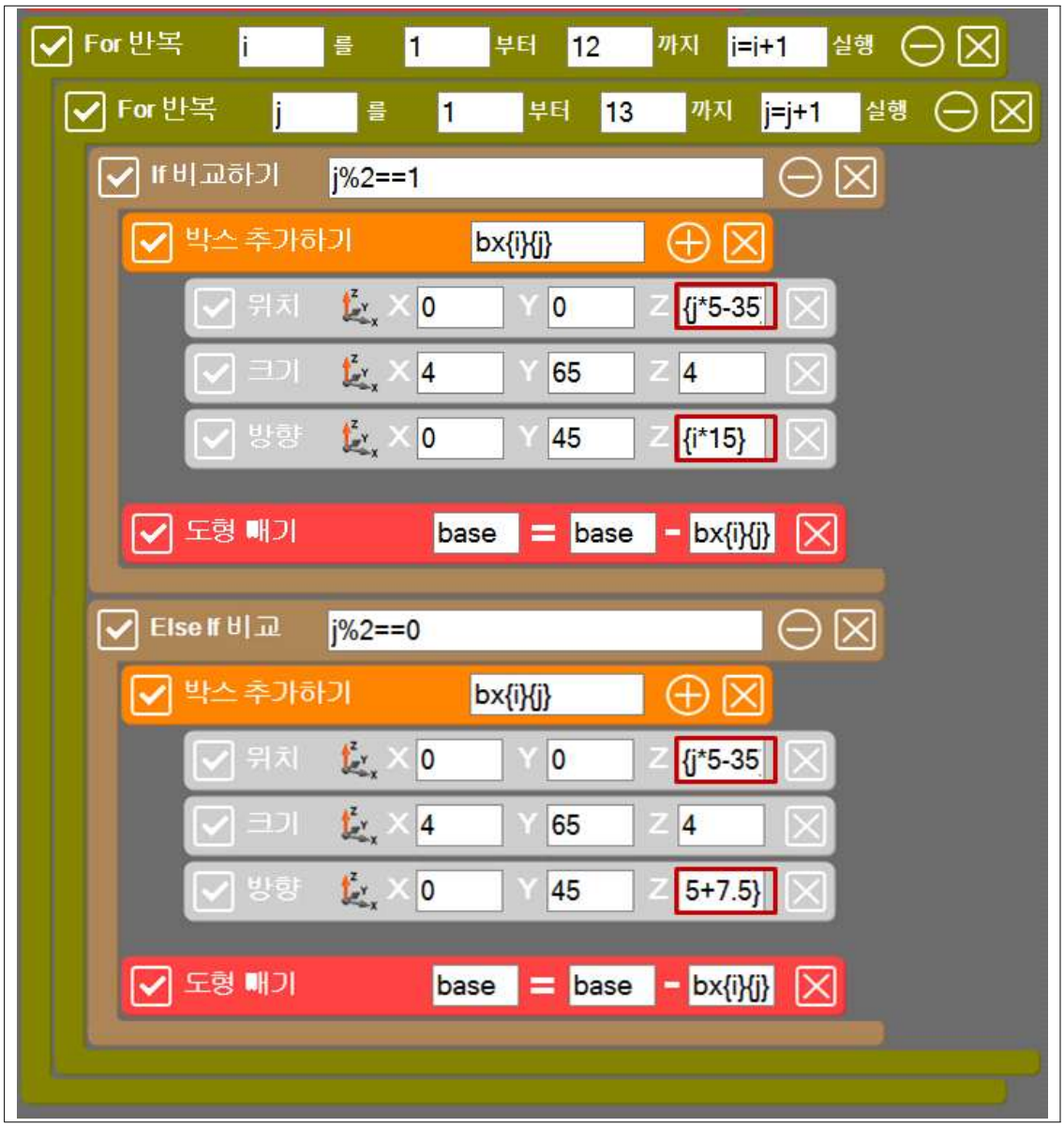


For반복 의 i는 12 j는 13으로 바꾸어 줍니다.
위치 Z는 {j*5-35}, 방향 Z는 {i*15}로 지정해줍니다.
원통의 크기와 박스의 크기를 고려하여 값을 수정해줘야 합니다.

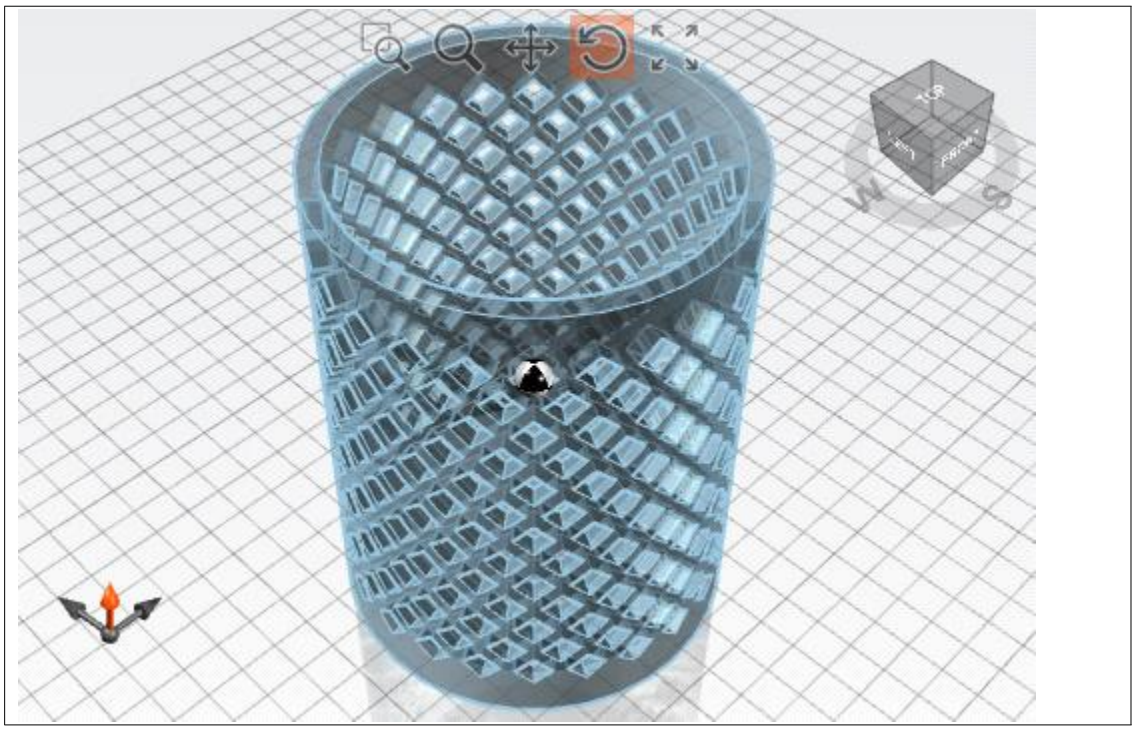
도형 빼기를 해줍니다.
하지만 결과창에서의 모습은 그물모양이 아닙니다.
지그재그의 모양을 하기 위해선 다른 방법을 사용해야합니다.
여기서 If 조건문을 이용해보도록 합니다.



If조건문을 두 개 만들어 줍니다.
본 실습에서 j의 값에 따라 높이가 변화하니 j의 값으로 조건을 줍니다.
j가 홀수 일 때와 짝수일 경우로 나누어 줍니다.



처음 만들었던 박스를 If 조건문 안에 다시 만들어 넣어줍니다.
위치 Z는 {j*5-35}로 똑같이 지정해줍니다.
두 박스의 차이점은 방향에서 나오는데 방향 Z에서 한쪽은 {i*15}로 하시고 한쪽은 {i*15+7.5}로 지정함으로써 살짝 틀어줄 수 있습니다.



아까와 달리 지그재그 모양이 되었습니다.
이와 같이 조건문 If를 이용하여 특정 규칙에 따라 다른 명령을 실행 할 수 있습니다.

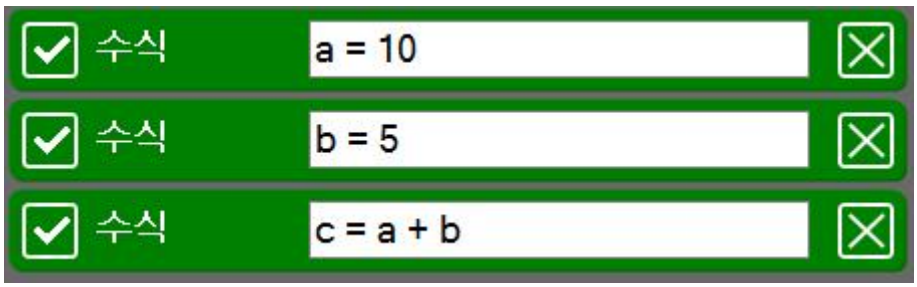
8. SPL For 3D 프린팅 로직 명령어 수식 익히기

8.1 수식 알아보기

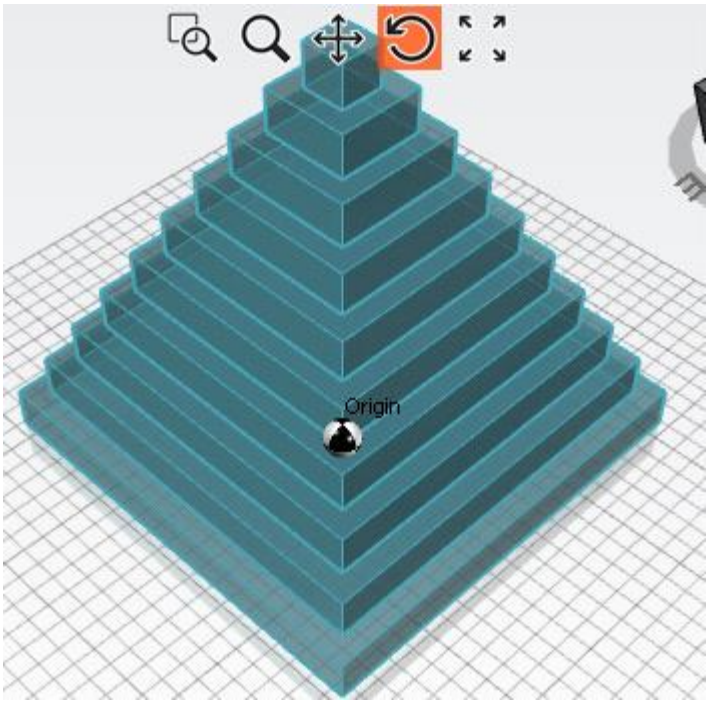
수식이란 ?

로직 명령어에 있는 가장 위에 있는 메뉴입니다.

수식을 사용하면 다음과 같은 블록이 생성됩니다.
C언어의 코딩과 같은 형식으로 입력이 가능합니다.
SPL 3D 프린팅에서는 주로 계산식을 넣을 때 사용합니다.



위 그림과 같이 계산식으로 사용하여
모델링에 적용시킬 수 있습니다.



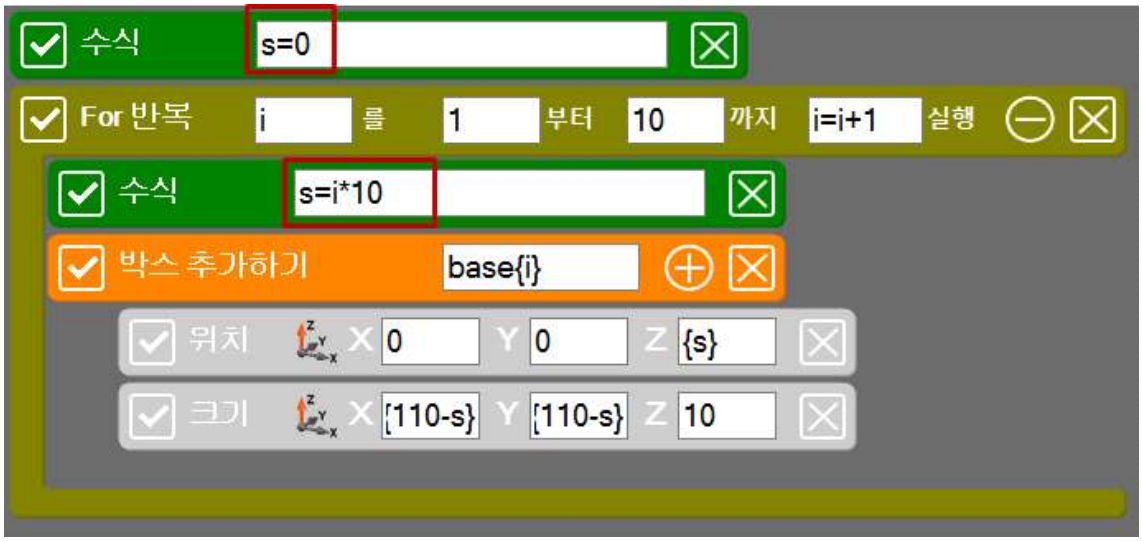
간단한 예를 하나 들어보겠습니다.
위 그림과 같은 모양을 만들어 보도록 하겠습니다.

위 그림과 같이 위치 Z에 {i*10} 크기 X, Y에 {110-i*10}을 입력하면 만들 수 있습니다.
이 과정을 수식을 이용하도록 하겠습니다.
수식을 이용하면 반복된 문장의 입력을 간단히 할 수 있습니다.

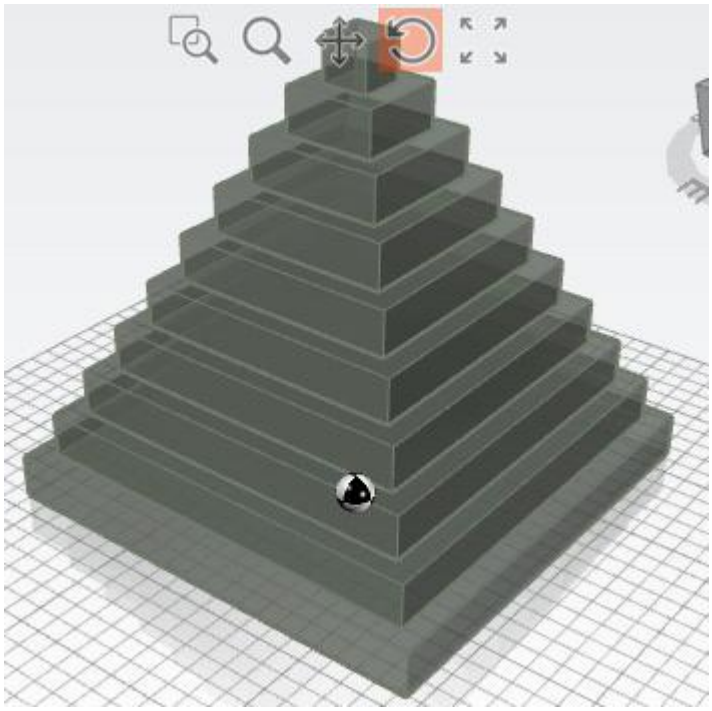
새파일을 만들어 수식하나를 추가합니다.

그 아래에 For 반복을 만들어 안에 수식과 박스 하나를 추가해줍니다.

박스의 이름을 base{i}로 바꾸고 위치와 크기를 추가해줍니다.
아까 수식을 쓰지 않는 방식에선 위치 Z는 {i*10}, 크기 X, Y에선 {110-i*10}을 입력하였습니다.
여기서 'i*10'이 3번이나 나옵니다.
수식을 이용하면 입력 시간을 단축 할 수 있습니다.

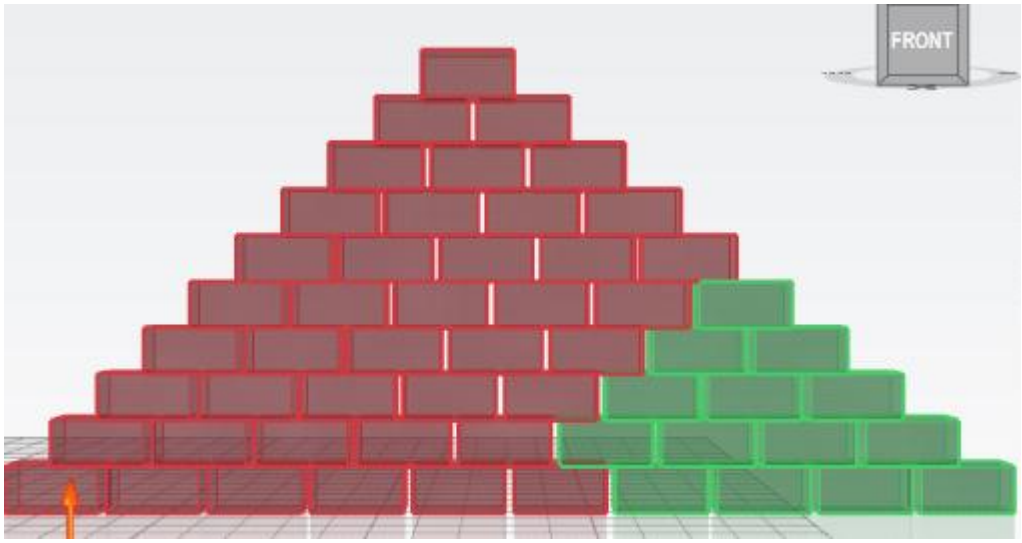


맨 처음에 만들었던 수식에 s=0을 입력합니다.
우선 s라는 변수를 만들어서 이 s에 0이 되게 해주는 명령어입니다.
두 번째 수식에 s=i*10을 입력합니다.
그러면 s는 처음에 0이였지만 For반복문에서 반복문이 반복될수록 10,20,30.....100까지의 값을 가지게 됩니다.
위치 Z에는 {s}를 크기 X, Y에는 {110-s}를 입력하고 실행을 합니다.



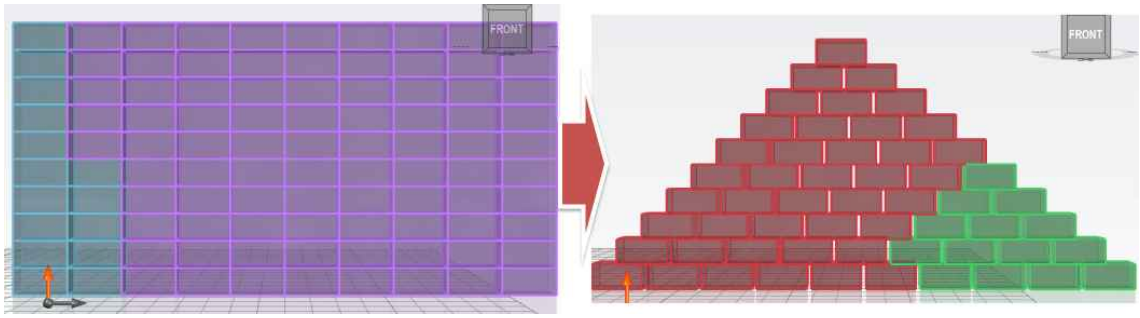
마찬가지로 완성이 됩니다.

이번엔 다른 방법으로 응용하는 방법을 알아보도록 하겠습니다.



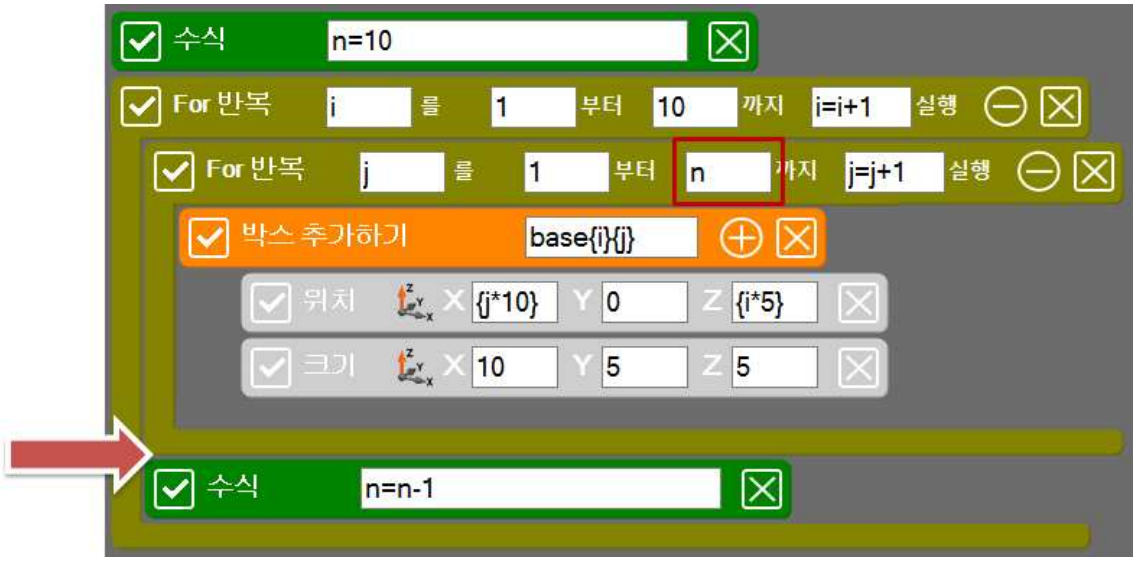
다음과 같이 박스의 개수가 위로 올라갈 때 마다 하나씩 줄어드는 방식입니다.

2중 For문을 만들어 박스를 만들도록 합니다.
위치 X는 {j*10} Z는 {i*5}를 입력합니다.

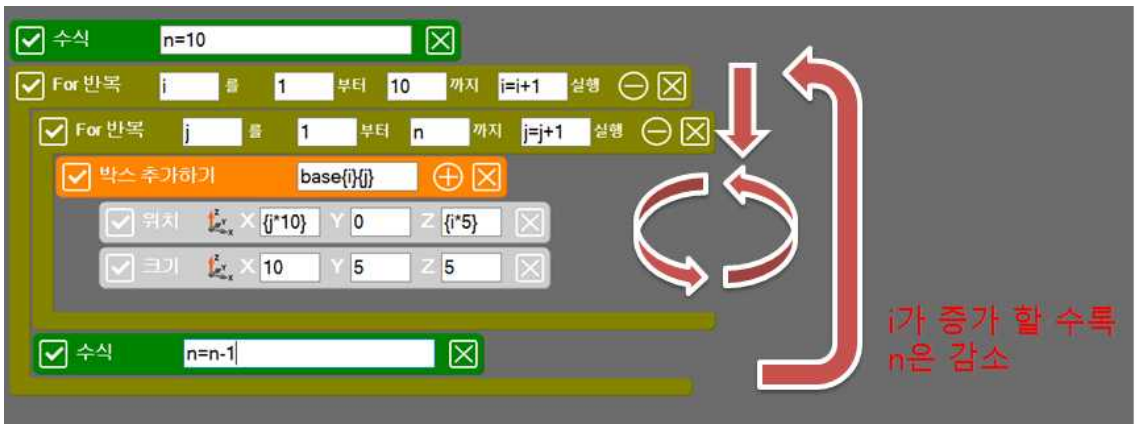


그러면 위 그림의 왼쪽과 같은 상태가 됩니다.
이 상태에서 수식을 이용하여 오른쪽과 같은 형태를 만들어 줄 수 있습니다.

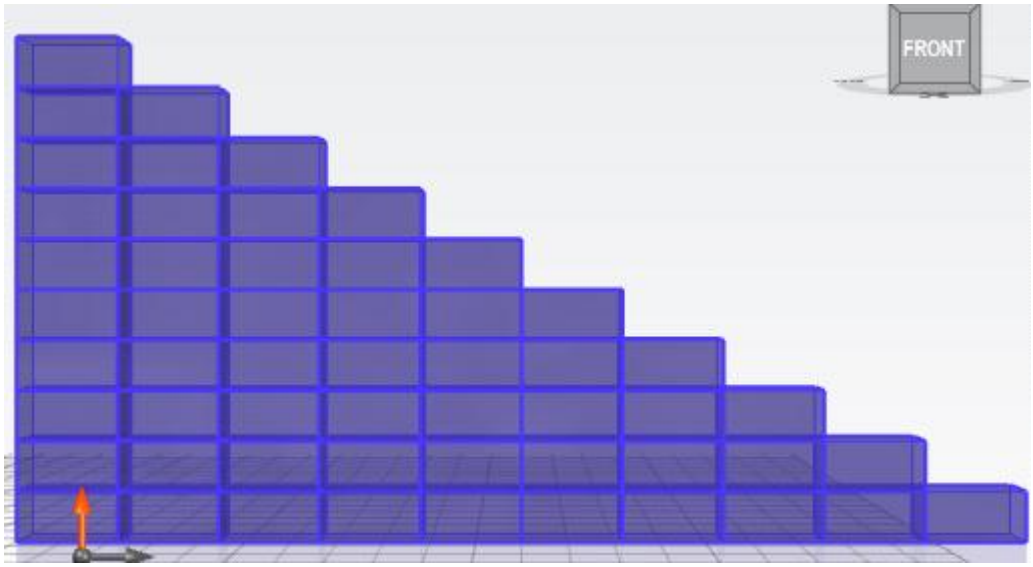
아까의 작업한 2중 For문 위에 수식을 삽입합니다.
SW 코딩은 위에서 아래로 진행되기 때문에 먼저 실행해야할 명령어는 위에 있어야 합니다.
그리고 수식에 n=10을 입력합니다.
n은 다른 이름으로도 지정 할 수 있습니다.
임의로 지정한 변수의 이름 n은 박스가 위로 올라갈수록 하나하나 없어지는 역할을 하게 됩니다.
10은 첫 번째 줄의 박스의 개수입니다.



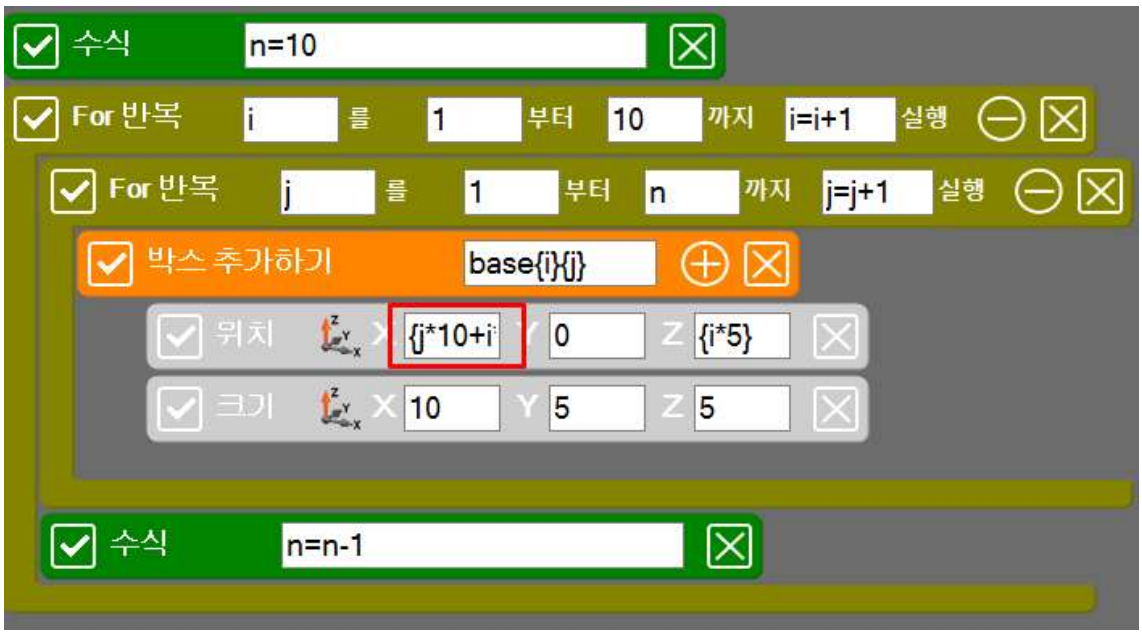
수식을 하나 더 추가합니다.
추가한 수식은 For반복문 j의 안쪽이 아닌 For반복문 i의안에 즉 For반복문 j의 밑에 넣어주셔야 합니다.
해당 수식은 n=n-1로 입력합니다(n--로 입력하셔도 됩니다.)
그리고 For반복문 j의 반복문에서 1부터 10까지의 문장에서 1부터 n까지로 수정합니다.



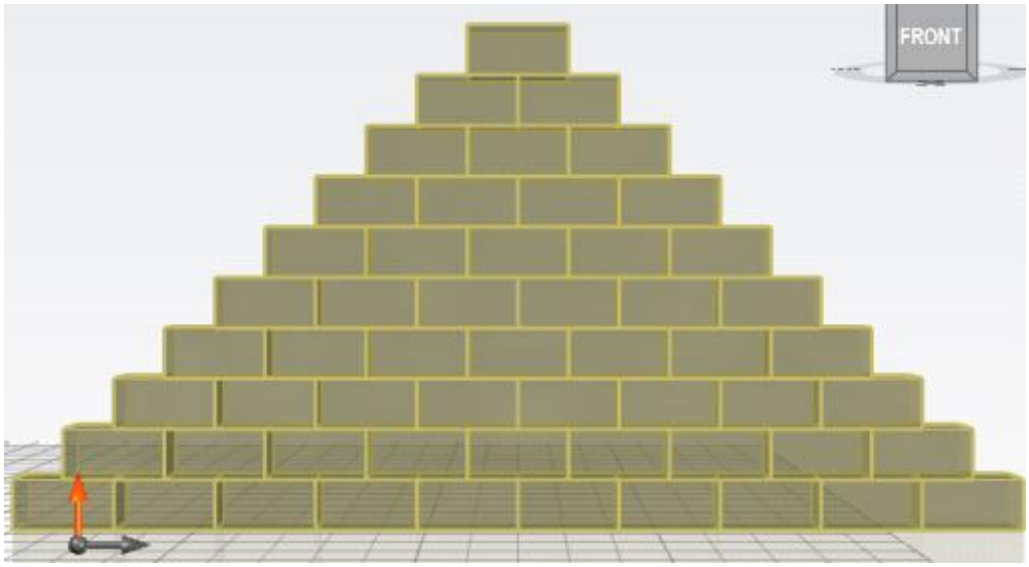
우선 수식에서 n이 10이라는 숫자로 선언됩니다.
그 뒤 For반복문에서 i부터 시작하여 i가 1일 경우 For반복문 j로 들어가서 j가 1부터 n까지(현재 n은 10) 반복문 안의 명령어 박스를 실행하게 됩니다.
For반복문 j의 반복이 끝나면 수식 n=n-1을 실행하게 됩니다.
10이였던 n은 수식 n=n-1에 의하여 9가 됩니다.
다시 위로 올라가 For반복문 i로 가서 i가 2가 되고 다시 밑문장을 i가 10이 될 때까지 반복하고 n은 점점 줄어들게 됩니다.
이 수식을 이용함으로써 박스가 한층 올라갈 때 마다 옆으로 만들어지는 박스의 수가 줄어들게 됩니다.



실행을 하게 되면 위 그림과 같은 형태로 만들어 집니다.
이제 값을 조금 수정하여 삼각형태로 만들어 주면 완료가 됩니다.



반복문 안에 있는 박스의 위치 X를 {j*10+i*5}로 수정만 해줍니다.



실행을 하면 위 그림과 같은 형태로 출력됩니다.
수식을 이용하면 새로운 식을 추가하여 다른 규칙을 만들 수 있습니다.

9. SPL For 3D 프린팅 로직 명령어 문자 출력 및 문자 라인 출력

9.1 문자 출력 알아보기

문자출력이란?

위 그림이 문자 출력 명령어의 모습입니다.
문자 출력은 모델링에 영향은 주지 않습니다.
문자 출력을 이용하면 화면 오른쪽 하단의 콘솔창에 메시지를 띄울 수 있습니다.

“SPL 3D 프린팅 !” 이라고 입력해 보았습니다.

콘솔창에 입력한 문자가 띄워지게 됩니다.

9.2 문자라인 출력 알아보기

문자라인 출력이란?

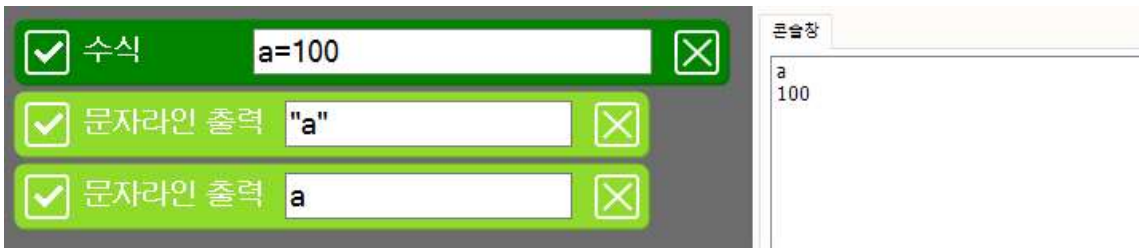
위 그림이 문자라인 출력 명령어의 모습입니다.
문자 출력과 마찬가지로 모델링에 영향은 주지 않습니다.
문자라인 출력 역시 이용하면 화면 오른쪽 하단의 콘솔창에 메시지를 띄울 수 있습니다.

문자출력의 경우 위와 같이 입력하면 콘솔창에 123으로 붙어서 나옵니다.

반면 문자라인 출력의 경우 한줄 한줄 입력됩니다.

9.3 문자 출력으로 수식 입력하기

문자 출력(혹은 문자라인 출력)은 주로 수식과 연관되어 쓰게 됩니다.
문자 자체를 콘솔창에 띄우고자 할 때는 큰 따옴표(“”)를 사용해야합니다.
수식의 변수를 출력하고자 할 때는 따옴표를 사용하지 않습니다.

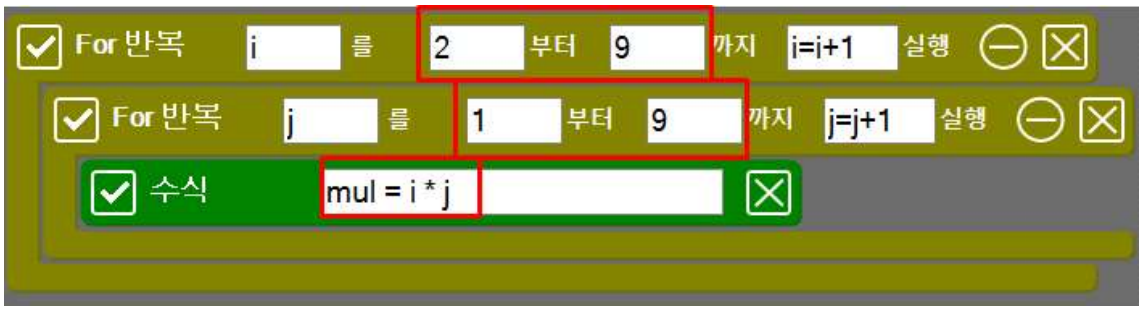


위 그림과 같이 수식 a=100을 입력을 합니다.
a라는 이름의 변수에 100의 값을 넣는 수식입니다.
그리고 문자라인 출력 “a"와 문자라인 출력 a를 입력합니다.
그리하면 오른쪽 콘솔창과 같이 첫 번째 줄은 a가 나오고 두 번째 줄은 100이 나오게 됩니다.
문자 출력의 주 사용 법은 수식의 변수의 변화를 확인하고자 할 때 사용합니다.

9.4 문자 출력 & 문자라인 출력 응용하기

1) 구구단 출력하기

우선 2중 For문을 만들어줍니다.

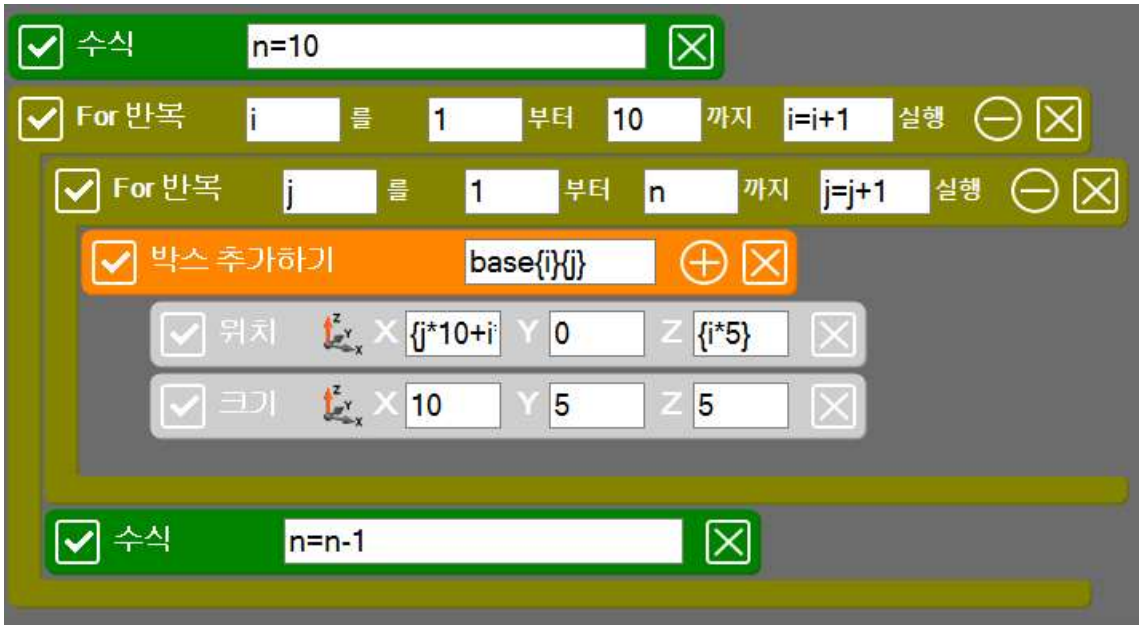


구구단을 출력하기 위한 2중 For문으로 i는 2부터 9로 수정하고 j는 1부터 9로 수정해줍니다. 그리고 For반복문 안에 수식 mul = i * j를 입력합니다.

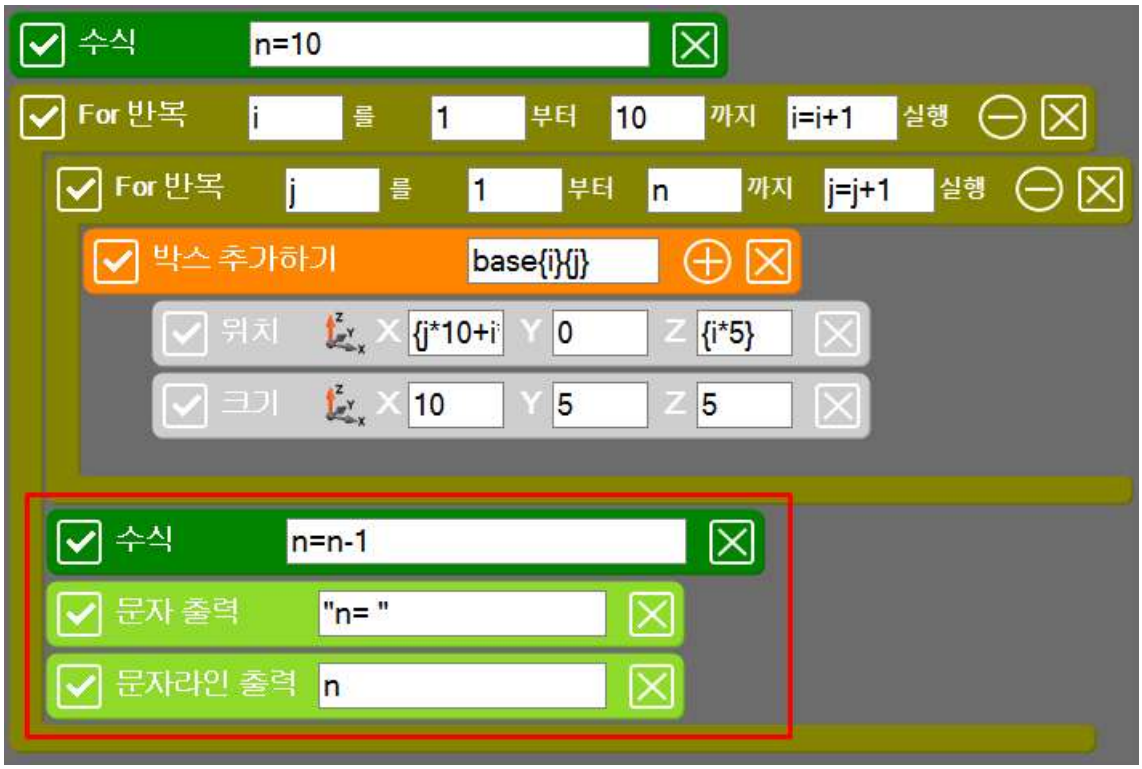
문자 출력과 문자라인 출력을 다음과 같이 입력합니다.

콘솔창에 그 그림 같은 결과를 얻을 수 있습니다.

2) 실제 변수 확인하기



수식 페이지에서 사용한 소스를 불러옵니다.



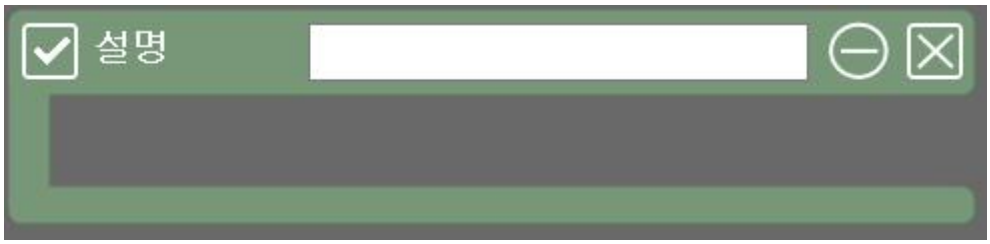
문자 출력과 문자라인 출력을 추가 해주도록 합니다

결과창에서 변수n의 수의 변화를 확인 할 수 있습니다.
실제 SPL 3D 프린팅에서 문자출력과 문자라인 출력은 변수의 움직임 확인하고자 쓰입니다.

10. SPL For 3D 프린팅 로직 명령어 주석과 공백

10.1 주석과 공백

1) 설명(주석달기)



설명(주석달기)는 위 그림과 같은 형태를 가지고 있습니다.
위 명령어는 모델링에 영향을 주지 않습니다.

위 그림과 같이 주로 명령어들을 구분하고 그에 대한 설명을 해주기 위해 사용합니다.
너무 코딩이 길어져서 해당 구간을 설명하여 나중에 보더라도 확인 할 수 있습니다.

2) 공백라인추가

위 그림은 공백라인의 블록입니다.
3D모델링에 영향은 주지 않습니다.

위 그림과 같이 붙어있는 명령어들을 구분지어 주거나 너무 붙어있는 블록아이콘들이 보기 불편하거나 헷갈릴 때 사용합니다.
문서로 보면 엔터를 한 번 더 친 것 이라 생각하시면 됩니다.

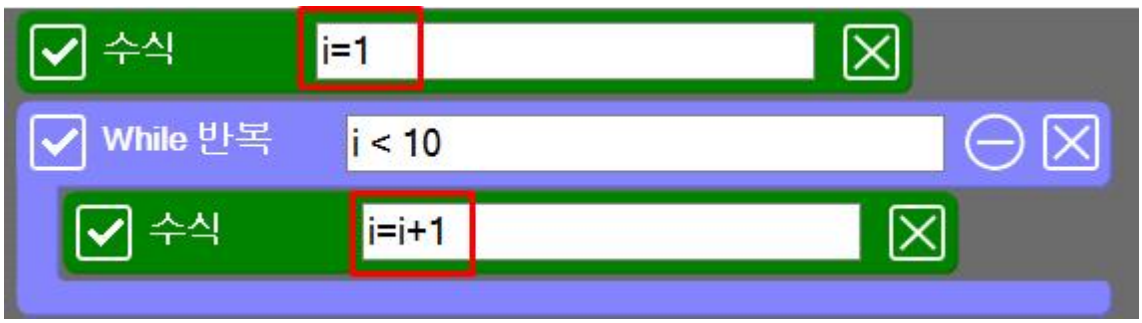
11. SPL For 3D 프린팅 로직 명령어 while문

11.1 while문 사용하기

while문이란 ?

For 반복문과 마찬가지로 while문 역시 조건에 따라 반복하는 명령어입니다.

For 반복문은 For 반복문 자체에 조건이 ①1부터 ②10까지 ③i+i+1이라는 3가지의 조건이 한 번에 들어 있는 반면 ① i < 10 이라는 조건 하나만 존재합니다.



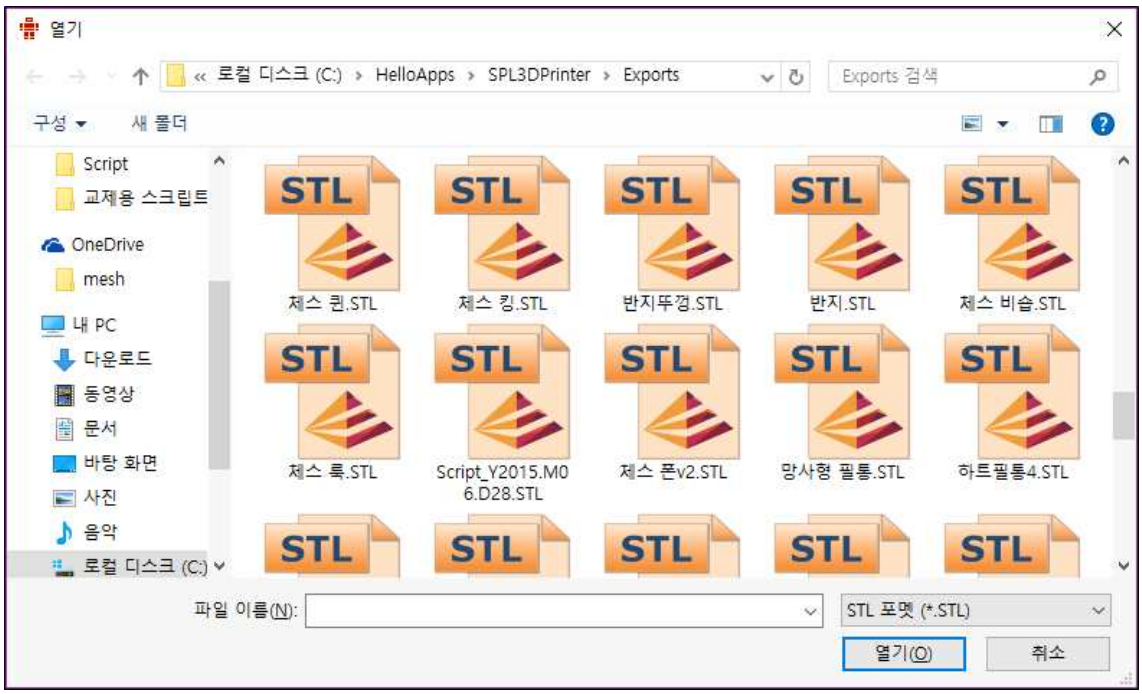
While 반복문을 For문과 동일 하게 쓸려면 위 그림과 같이 수식을 두 개 추가해줘야 합니다.
While 반복문은 For반복문과 달리 좀 더 유동적으로 조건을 줄 수 있다는 점이 있습니다.

12. SPL For 3D 프린팅 도형 불러오기 & 도형 변환 명령어

12.1 도형 불러오기 사용하기

도형 불러오기란?

기존에 있는 STL파일을 불러오는 기능입니다.



도형 불러오기를 사용하면 위 그림과 같이 모델링 파일을 불러올 수 있습니다.
불러온 파일은 편집이 가능합니다.

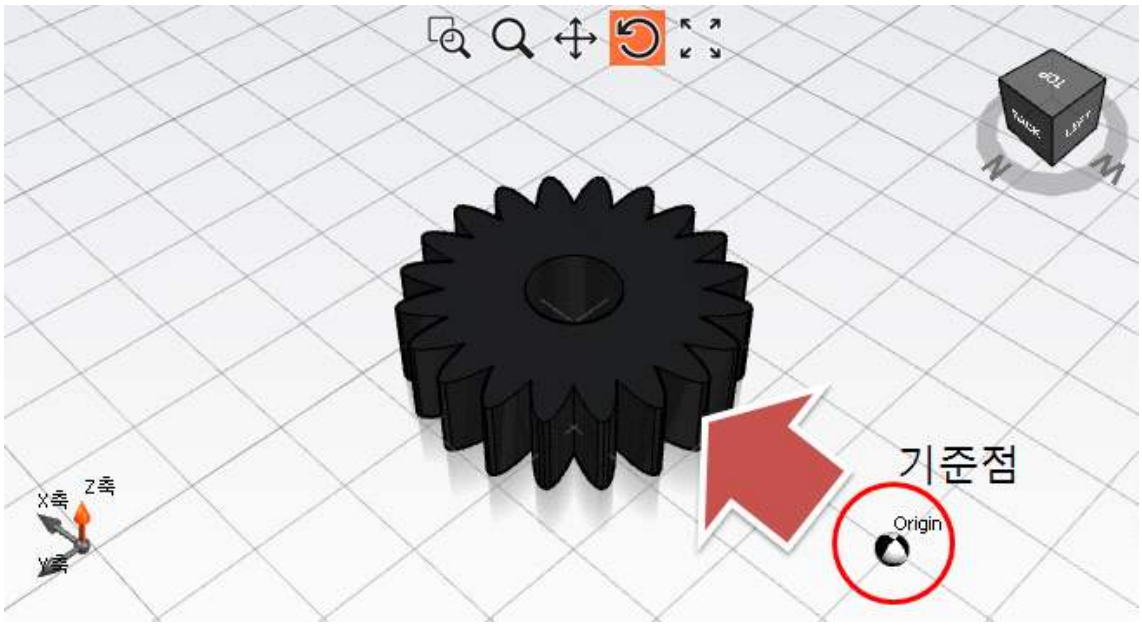
도형 불러오기를 이용하여 기존에 만들었던 톱니바퀴를 불러온 모습입니다.
만일 모델링의 작업이 길어지면 “모델링 파일 내보내기”를 이용하여 새 창에 불러와서 작업을 이어가는 방법이 있습니다.
이 불러온 도형을 가지고 이동 및 크기 변환 등 편집이 가능합니다.
다음에 설명할 명령어들을 가지고 편집을 할 수 있습니다.

12.2 도형 변환 명령어 사용하기

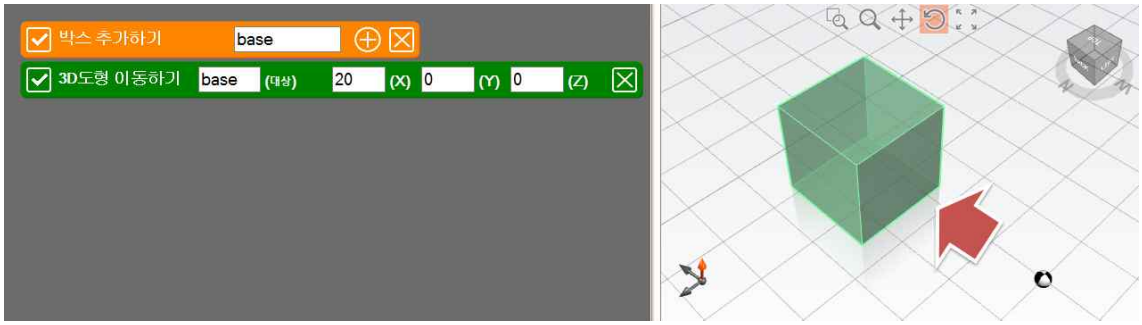
1) 3D 도형 이동하기

3D도형 이동하기는 빨간 네모 박스에 표시된 곳에 도형의 이름을 입력하여 옆의 X, Y, Z의 칸에 값을 이동하여 이동 시키는 명령어입니다.

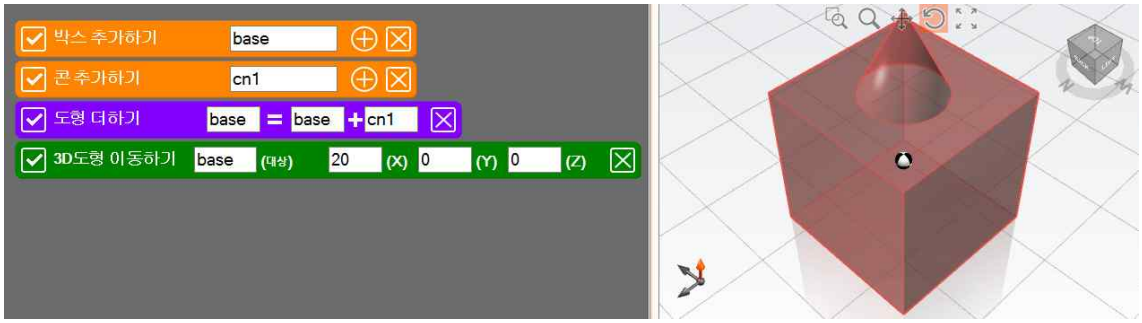
파일 불러오기를 이용해서 불러온 모델링 파일도 이동이 가능합니다.
파일 불러오기로 불러온 모델링 파일의 이름을 st1이라 지정하고 3D도형 이동하기 명령어를 사용하여 (대상)의 이름을 이동할 도형의 이름을 입력하고 X, Y, Z에 값을 입력하면 이동이 가능합니다.



X의 값을 20을 입력하여 불러오기로 불러온 모델링파일이 (0,0,0)에서 X축으로 20만큼 이동하였습니다.

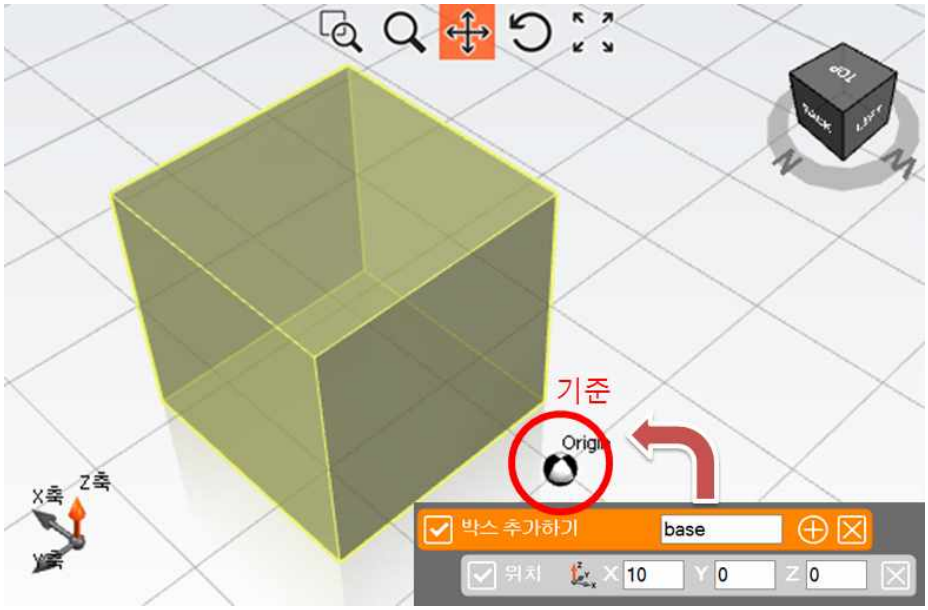


위 그림과 같이 기본 도형도 3D도형 이동하기로 이동이 가능합니다.



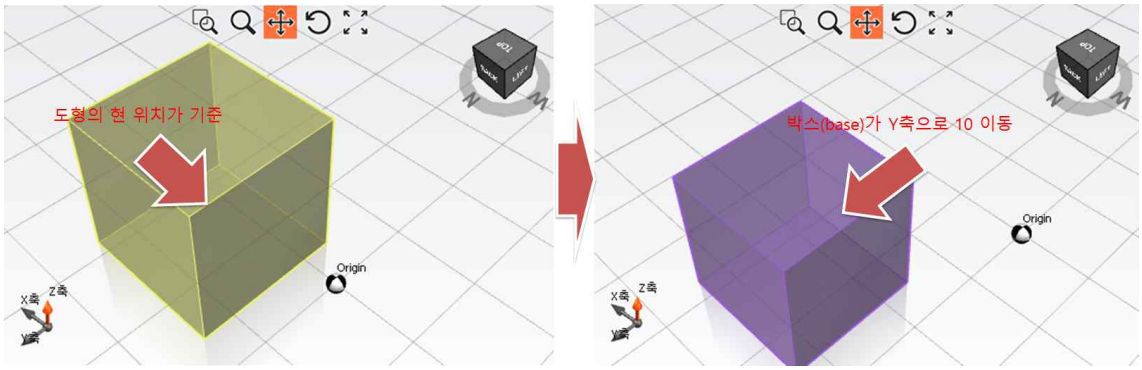
하지만 위 그림과 같이 도형 더하기를 작업을 한 도형은 3D도형 이동하기로 이동이 되지 않습니다.
도형 연산 명령어로 변하게 된 기본 도형의 경우 “모델링 파일로 내보내기”로 내보내기를 하여 도형 불러오기 명령어로 불러와서 작업해야 사용이 가능합니다.
이는 뒤에서 배울 모든 도형 변환 명령어가 해당하는 사항입니다.

기본 도형의 위치와의 차이점은 ?

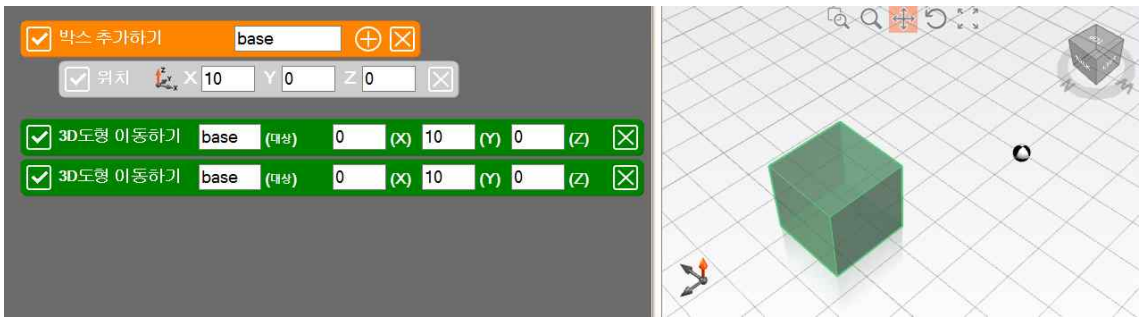


위치의 경우 기준점인 (0,0,0)에서 위치의 입력한 값만큼 이동하게 됩니다.
반면 3D도형 이동하기의 경우 명령을 내리는 도형의 현 위치가 기준이 되어 입력한 값만큼 이동하게 됩니다.

박스(base)의 위치를 X축 10을 입력하고 3D도형 이동하기로 Y축으로 10을 입력하였습니다.



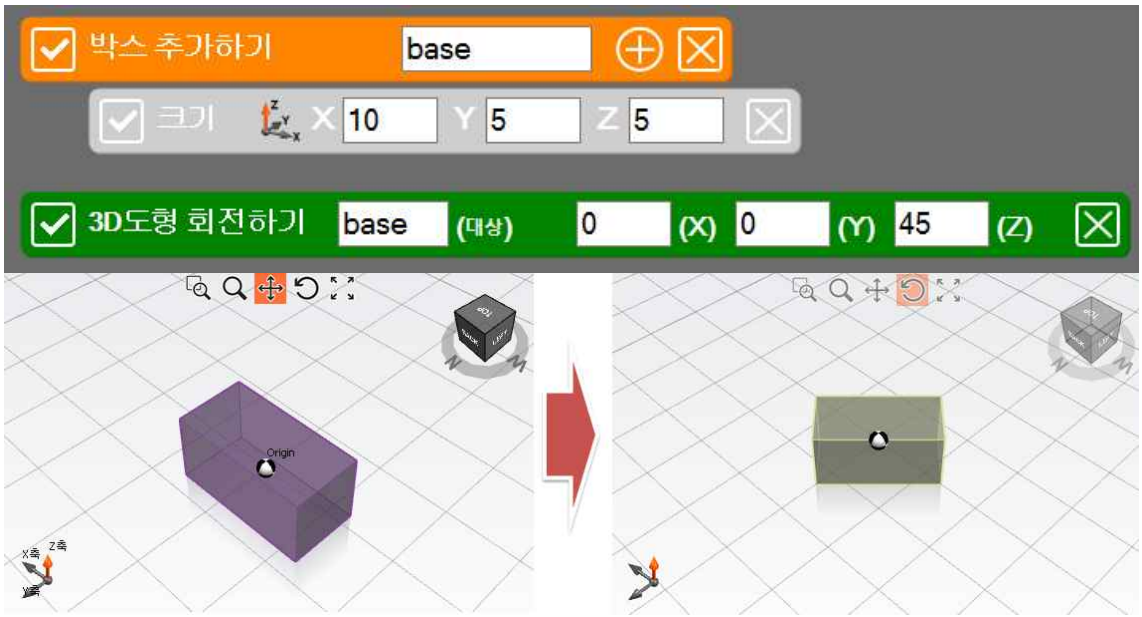
위 그림과 같이 박스(base)가 위치 X축 10을 입력한 상태에서 3D도형 이동하기 명령을 사용하여 Y축에 10을 입력하였습니다.
그 결과 박스(base)의 위치는 최종적으로 (10,10,0)이 되었습니다.



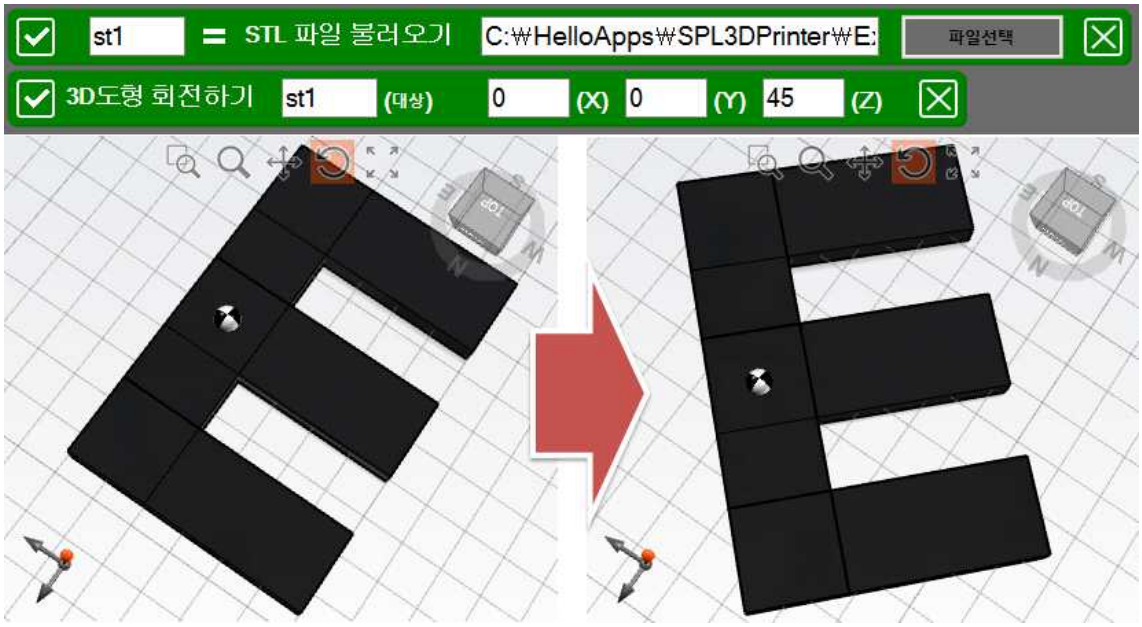
그리고 다시 3D도형 이동하기 명령어를 사용하여 Y축 10을 다시 입력하였습니다.
위 그림과 같이 기본의 위치였던 (10,10,0)의 위치가 기준이 되어 다시 Y축으로 10 이동한 모습을 확인 할 수 있습니다.

2) 3D 도형 회전하기

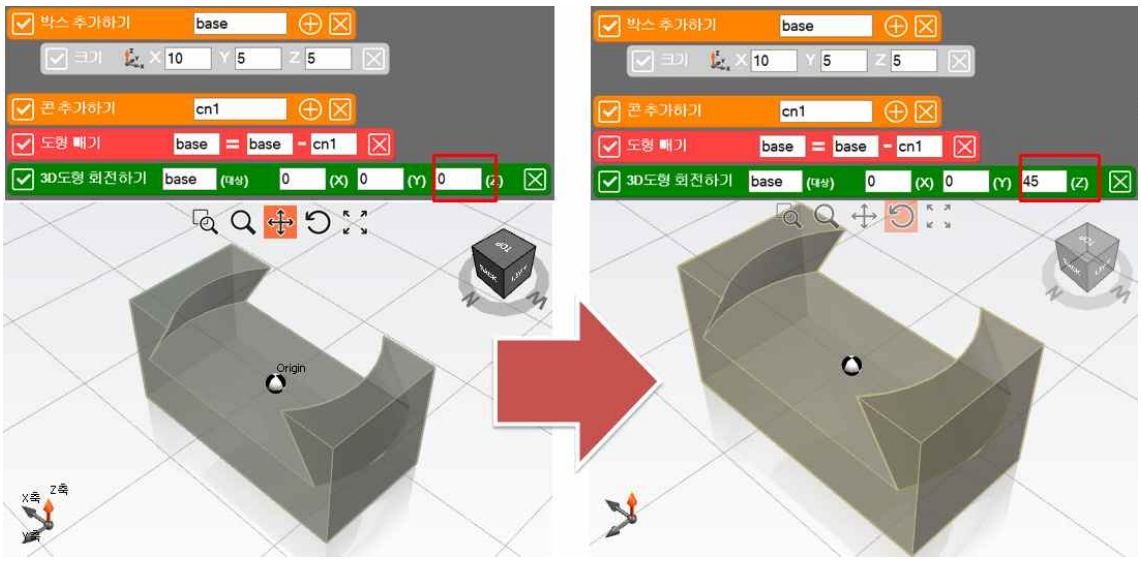
3D도형 회전하기는 3D도형 이동하기와 마찬가지로 (대상)에 도형의 이름을 입력하고 X, Y, Z에 값을 입력하여 회전을 할 수 있는 명령어입니다.



박스를 하나 생성하여 회전의 변화를 쉽게 관찰하기 위해 크기를 약간 변형합니다.
3D도형 회전하기를 이용하여 (대상)을 base로 하고 Z축의 회전값을 45를 주면 위 그림과 같이 변화한 모습을 볼 수 있습니다.



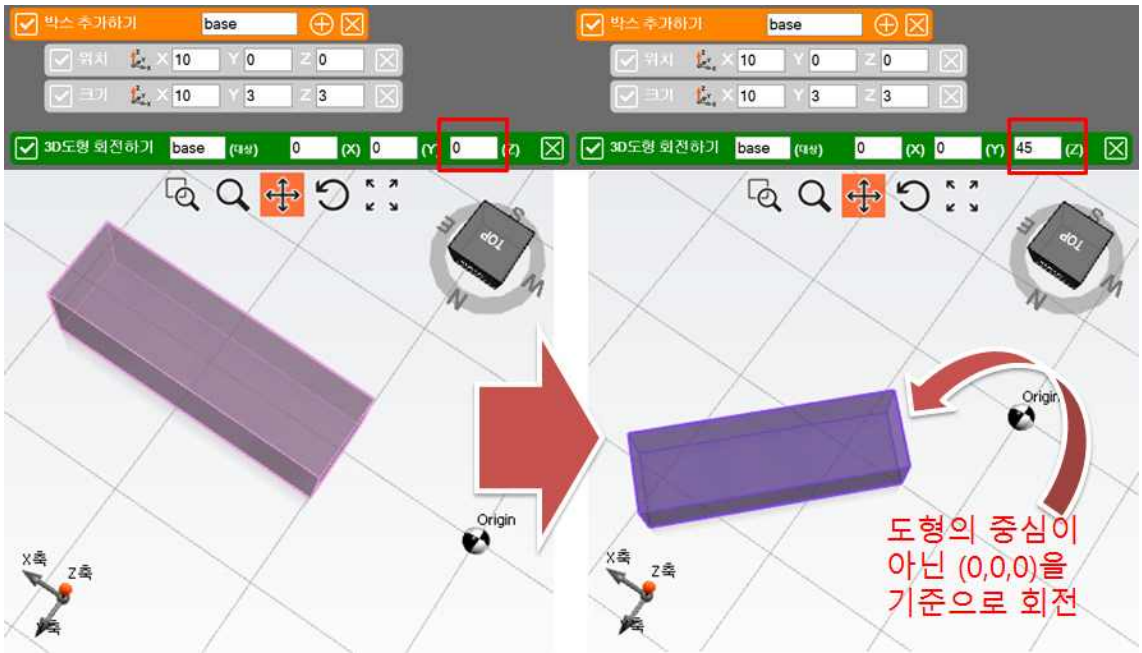
도형 불러오기로 불러온 도형도 3D도형 회전하기로 회전이 가능합니다.



기본 도형에서 연산이 이루어진 도형의 경우 3D도형 회전하기를 사용할 수 없습니다.
3D도형 이동하기와 마찬가지로 “모델링 파일로 내보내기”를 하여 도형 불러오기로 불러와서 작업이 가능합니다.

기본도형의 방향과 3D도형 회전하기의 차이점은?

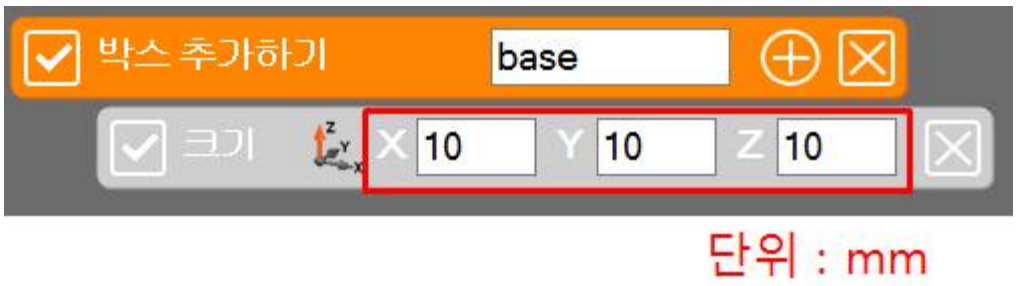
위 그림은 기본도형의 방향을 이용한 모습입니다.
일단 변화를 알아보기 위해 위치를 기준점에서 차이를 두고 크기 역시 조절합니다.
방향 값을 주면 도형을 기준으로 회전하는 모습을 볼 수 있습니다.



같은 위치, 크기의 박스를 방향이 아닌 3D도형 회전하기로 회전을 시켰습니다.
방향과 달리 기준점(0,0,0)을 기준으로 회전 하는 모습을 확인 할 수 있습니다.

3) 3D도형 크기비율

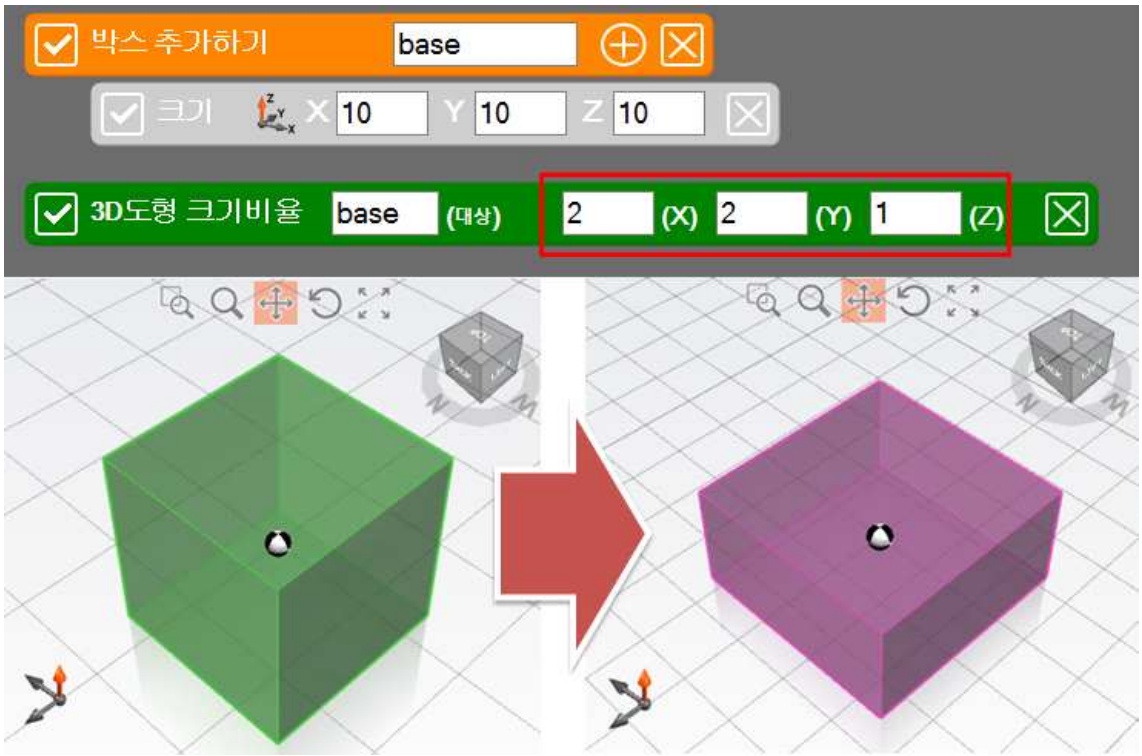
3D도형 크기비율은 (대상)에 도형의 이름을 입력하고 X, Y, Z의 값을 입력하여 크기를 변화시킵니다.



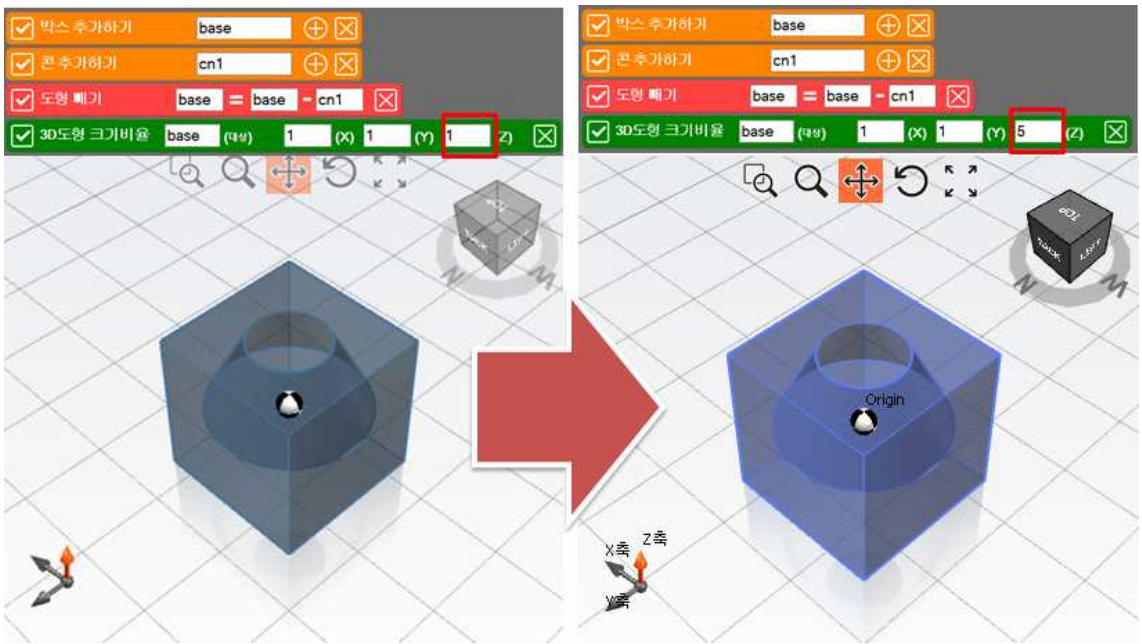
기본 도형에서의 크기는 X, Y, Z의 방향으로 입력한 만큼 증가하는 방식입니다.

반면 3D도형 크기비율의 경우 이름 그대로의 '비율'을 증가시키는 원리입니다.

3D도형 크기비율은 기본 도형보단 주로 도형 불러오기로 불러온 도형에 주로 쓰입니다.



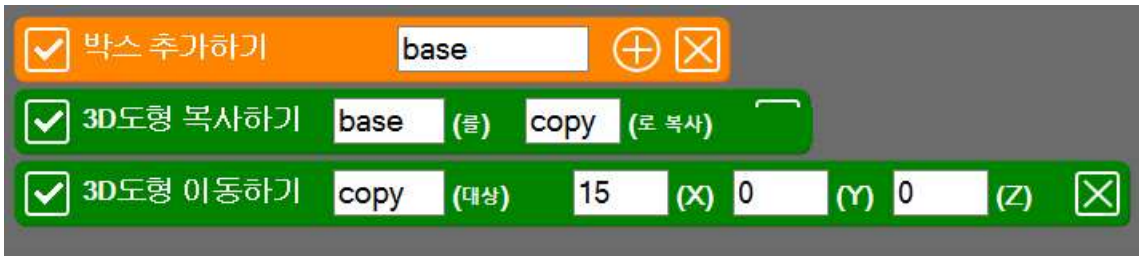
박스의 기본 크기는 상태에서 3D도형 크기비율을 적용시켜보았습니다.
3D도형 크기비율에서 X, Y의 값을 2로 지정하고 실행을 시키면 박스(base)가 X, Y 방향으로 두 배 늘어난 것을 확인 할 수 있습니다.



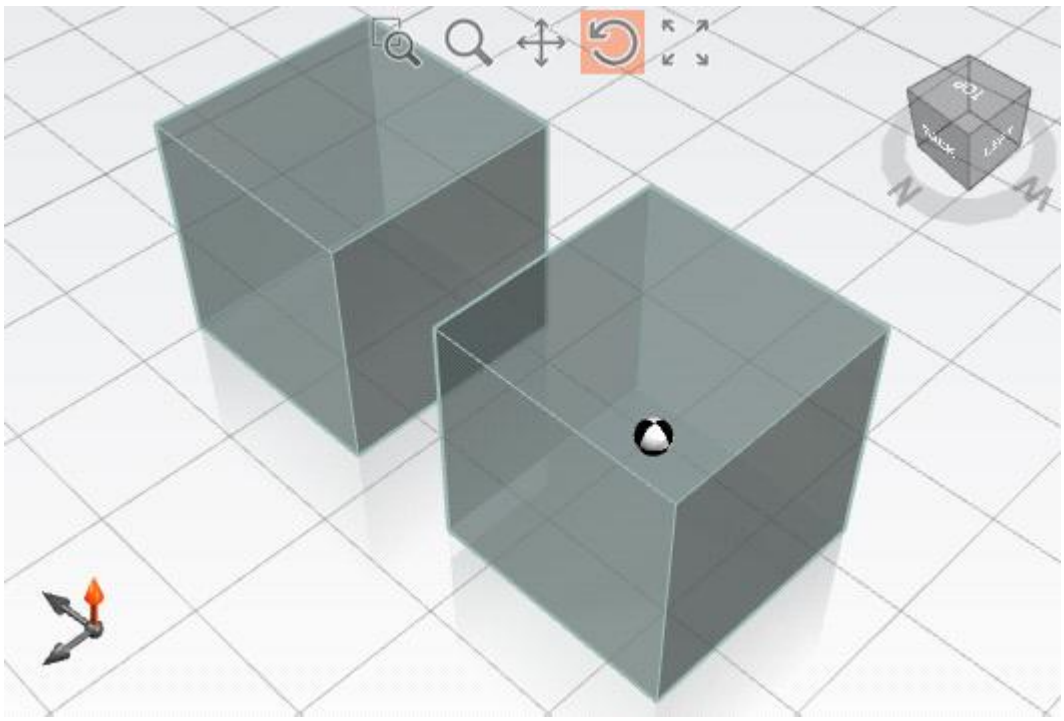
마찬가지로 도형 더하기나 빼기 같은 수식어를 사용해서 적용이된 도형은 3D도형 크기비율이 적용되지 않습니다.

4) 3D도형 복사하기

3D도형 복사하기는 이름 그대로 복사하는 기능입니다.
복사할 대상을 첫 번째 네모박스에 입력하고 새로운 이름을 두 번째 네모박스에 입력하면 복사가 됩니다.

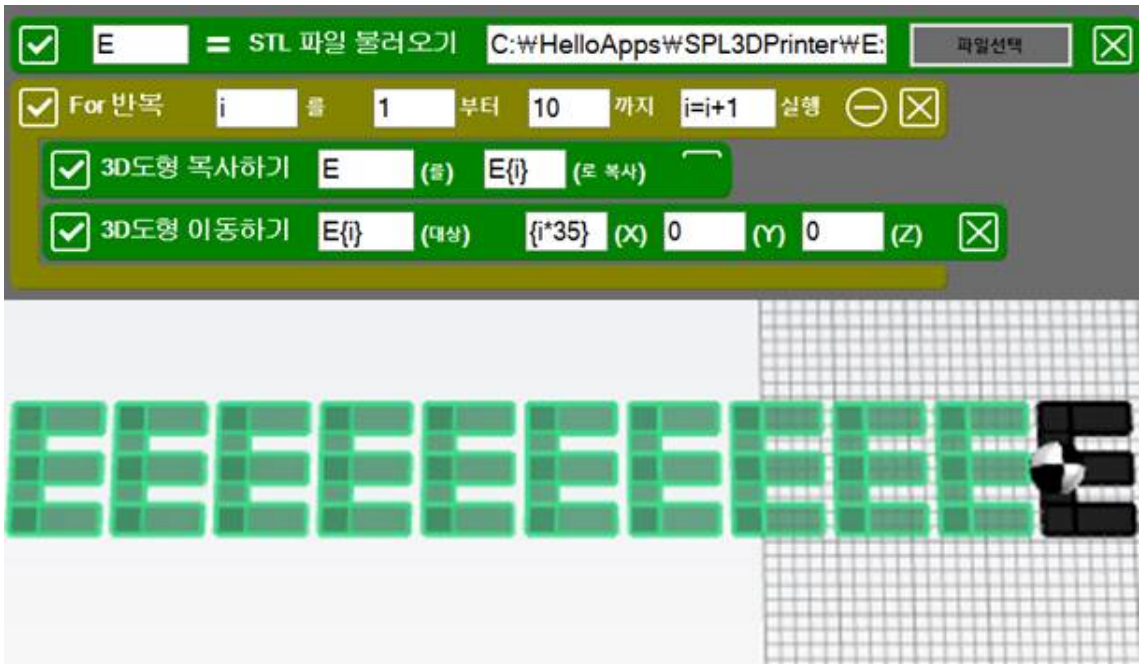


위 그림과 같이 박스(base)를 생성합니다
그리고 3D도형 복사하기를 명령어를 입력하여 복사 대상인 박스(base)의 이름을 입력합니다.
이름이 중복이 되선 안되니 새로운 이름으로 복사를 해줍니다
그리고 복사한 대상을 확인 하기위해 3D도형 이동하기로 이동을 시켜줍니다.



실행결과 두 개의 같은 크기의 도형이 복사된 것을 확인 할 수 있습니다.
도형 복사도 마찬가지로 기본도형을 연산으로 변형시킨 도형은 복사 할 수 없습니다.

3D도형 복사하기는 3D도형 크기비율과 마찬가지로 3D도형불러오기로 불러온 도형에 주로 쓰입니다.

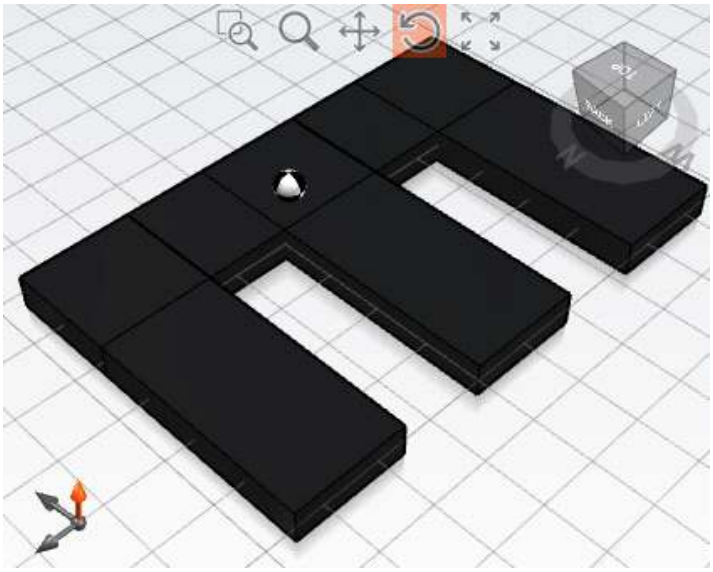


위 그림과 같이 같은 도형을 불러올시 For반복문과 3D도형 복사하기를 사용하면 알고리즘을 간단히 줄일 수 있습니다.

5) 3D색상 변경하기

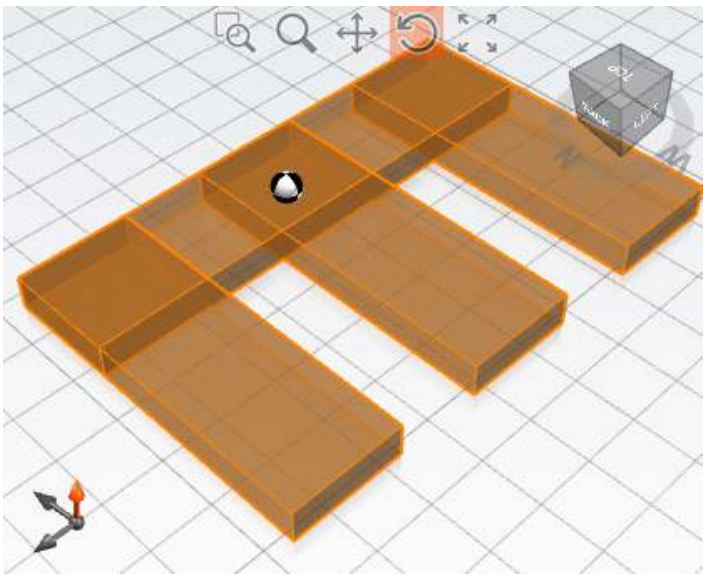
3D색상 변경하기는 결과창에 뜨는 3D도형의 색상을 변경 시킬 수 있는 기능입니다.
첫 번째 흰색 네모박스에 색상을 변경시킬 도형의 이름을 입력합니다
그 옆에서 투명도를 조절 할 수 있습니다.

색상변경 버튼을 누르면 위 그림과 같은 색상으로 변경이 가능합니다.



보통 STL파일 불러오기를 실행하면 위 그림과 같은 검은색 도형으로 나타납니다.
구분하기가 어려운 관계로 색상을 변경시켜줍니다.

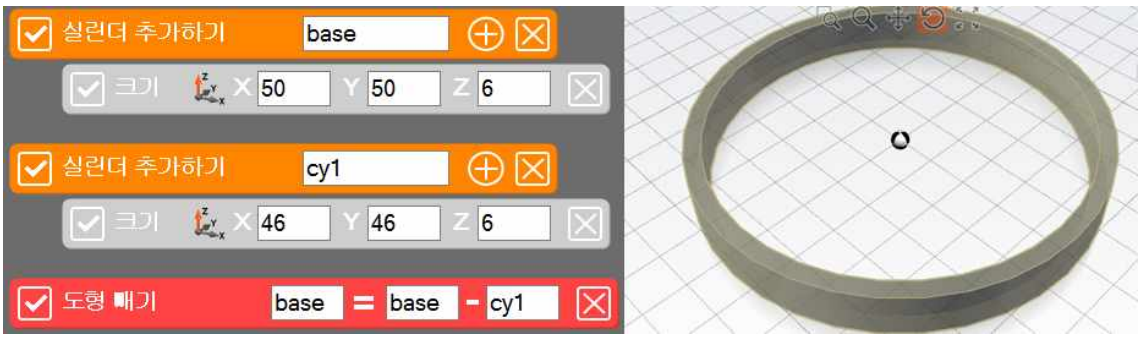
불러온 도형의 이름은 E이고 3D색상 변경하기에 E를 입력한 뒤 색상은 선택하고 실행을 해줍니다.



다시 투명한 색상의 도형이 되었습니다
※ 프로그램에서 도형의 색상을 변경한다고 해서 3D프린터에 원하는 색상이 나오지는 않습니다.

12.3 도형 불러오기 & 도형 변환 명령어 활용하기

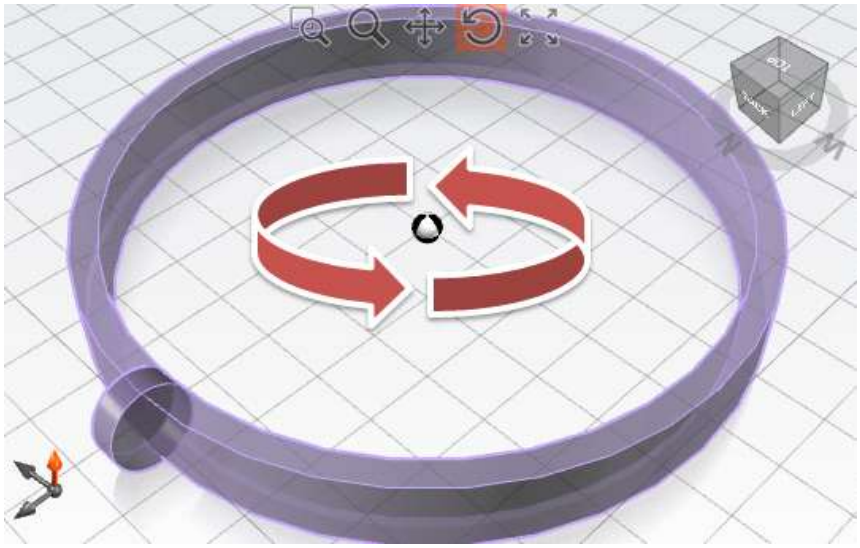
1) 비눗방울 막대 만들기



임의의 크기의 실린더를 추가해줍니다.
넓고 얇은 실린더를 만들고 그리고 그것보다 작은 실린더를 만들어 빼줍니다.

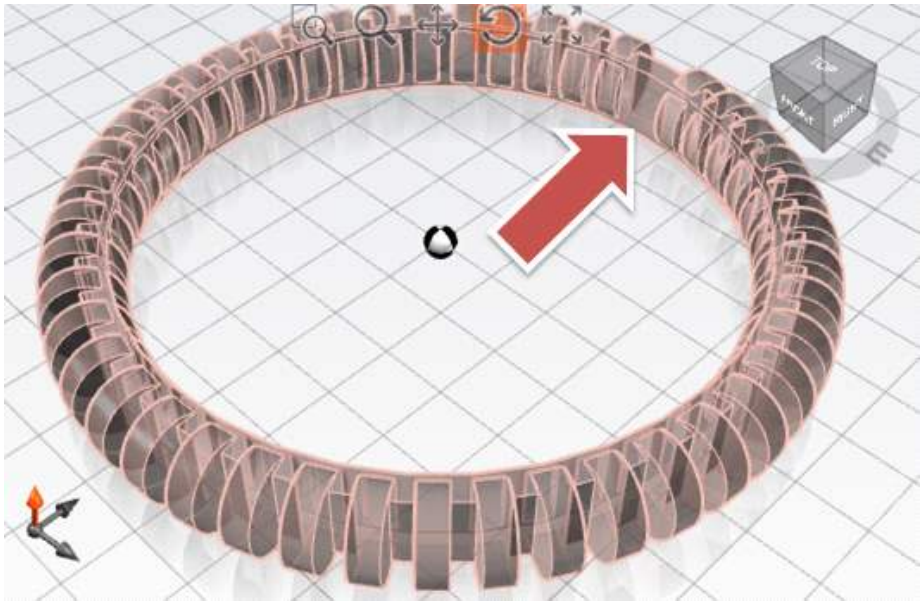
비눗방울 막대를 보면 방울이 나오는 곳에 돌기가 존재합니다.
먼저 만든 실린더의 외각을 돌기를 만들어주기 위해 For 반복문을 사용합니다.

돌기로 사용될 실린더의 위치는 기존에 만든 실린더(base)에 걸칠 수 있도록 이동 시킵니다.
크기와 방향의 값을 입력해줍니다.

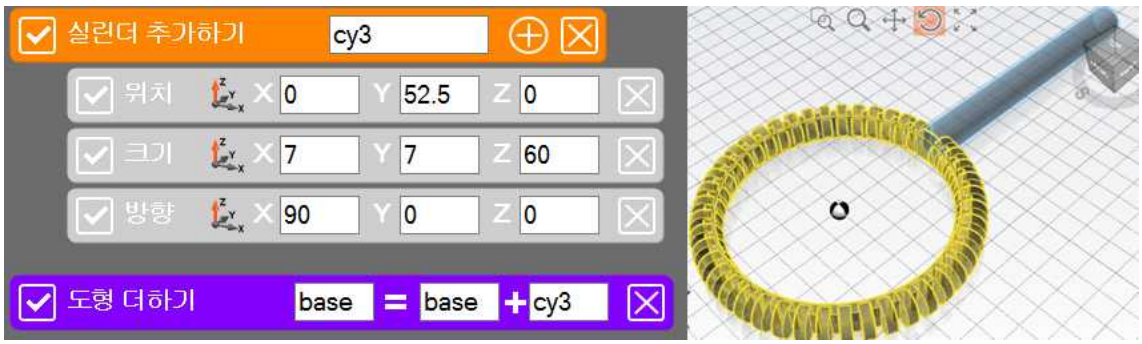


작은 실린더를 큰 실린더 외각으로 돌리기 위해선 방향이 아닌 3D도형 회전하기를 사용해야 합니다.
방향은 도형의 중심에서 회전하지만 3D도형 회전하기는 (0,0,0)을 기준으로 회전합니다.

3D도형 회전하기를 만들어줍니다.
대상은 cy{i}이고 촘촘히 돌려주기 위해 Z의 값은 {i*8}로 입력합니다.
For 반복은 1부터 44까지 지정합니다.
도형 더하기를 하여 모든 실린더를 더해줍니다

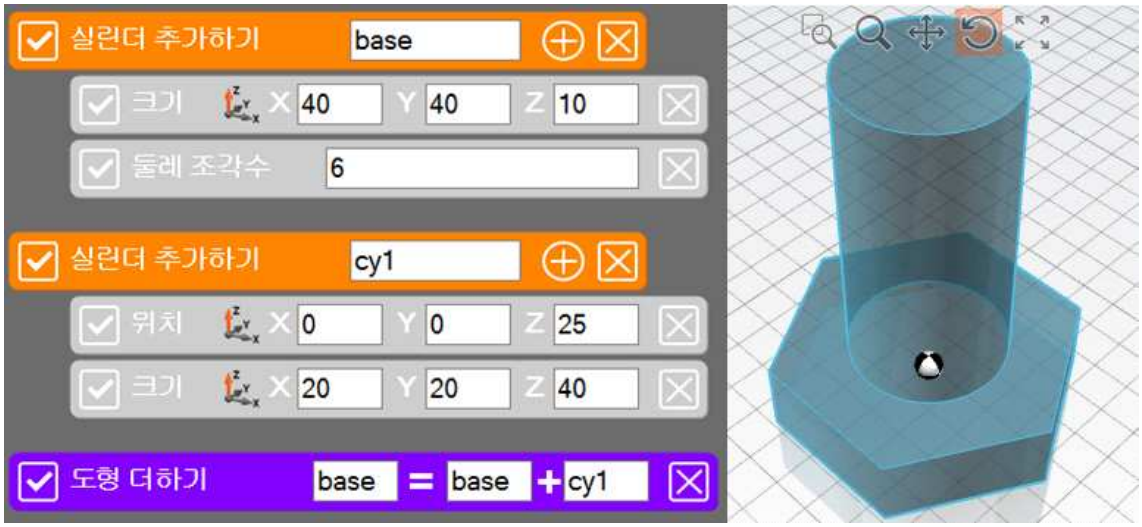


남는 부분은 손잡이를 만들어줄 부분입니다.



임의의 크기와 모양의 손잡이를 만들어줍니다.
그리고 마지막에 더해줌으로써 비눗방울 막대가 완성됩니다.

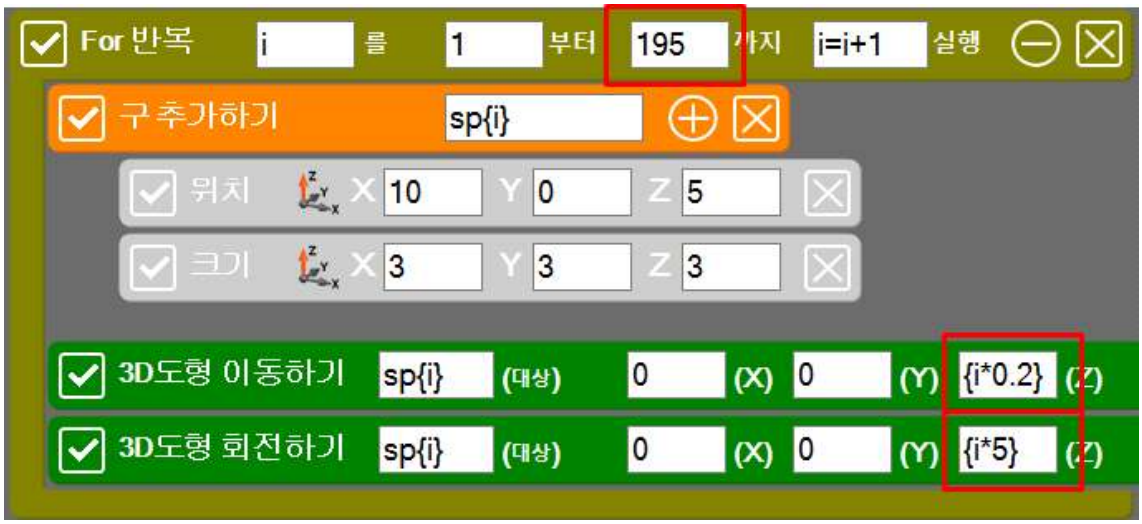
2) 나사 만들기



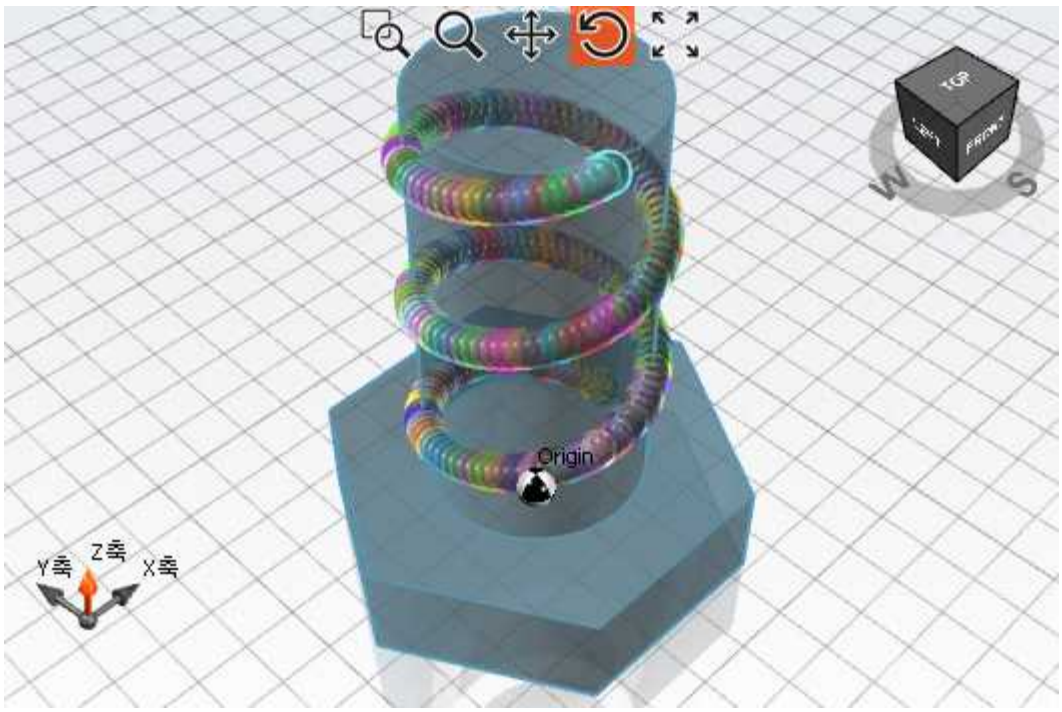
나사를 만들기 위해 우선 실린더를 하나 만들어 둘레 조각수를 6으로 지정하여 육각 헤드를 만듭니다.
그 뒤 실린더를 하나 더 만들어 기둥으로 만들어 더해줍니다.
크기는 임의로 지정합니다.

SPL 3D 프린팅에서는 스프링 도형이 지원이 되지 않음으로 직접 제작해야 합니다.
For 반복문을 이용하여 스프링 모양을 만들 수 있습니다.

구를 하나 생성합니다.
이 구의 위치는 기둥의 외각으로 빼시고 크기는 적당한 크기로 지정해줍니다.
무수히 많은 구를 이용해 기둥을 나선모양으로 덮는 방식을 이용하도록 하겠습니다.

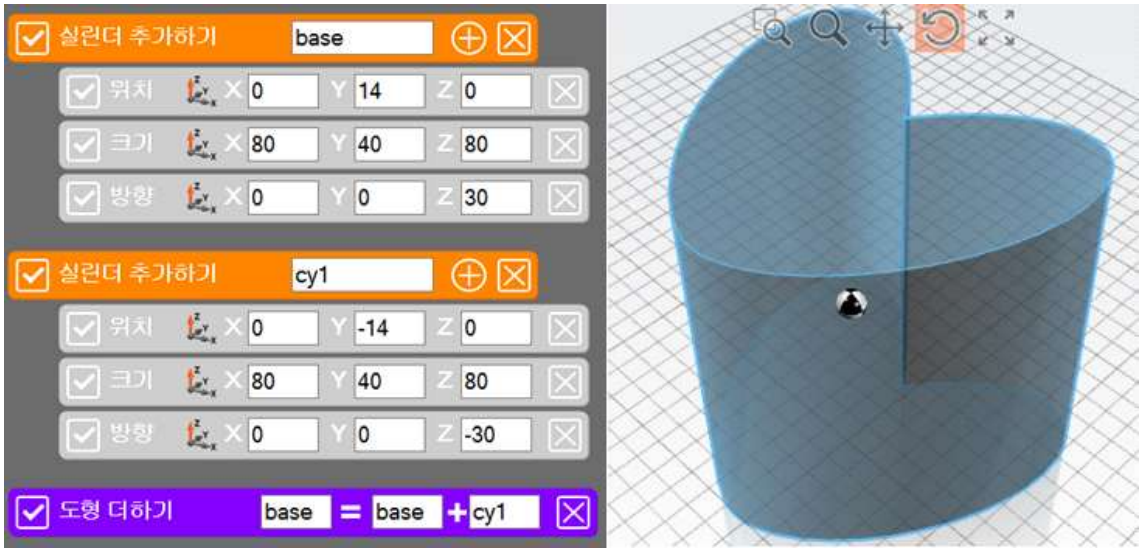


3D도형 이동하기 명령어를 추가합니다.
추가한 이동명령은 {i*0.2}로 지정하여줍니다.
반복문 i가 증가 할 때마다 0.2씩 Z축으로 올라갑니다.
그리고 3D도형 회전하기 명령어를 추가해 Z에 {i*5}를 입력 합니다
i가 증가 할 때 마다 sp{i}는 5도씩 돌아가게 됩니다.
For 반복문의 최대값을 195까지 지정해줍니다.



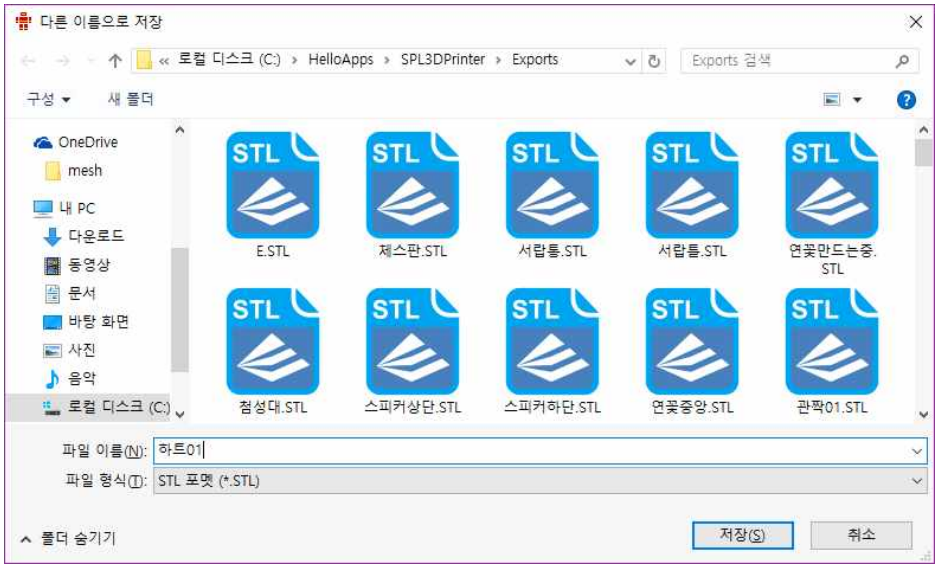
나선모양이 만들어진 것을 확인 할 수 있습니다.

3) 하트 모양 화분 만들기



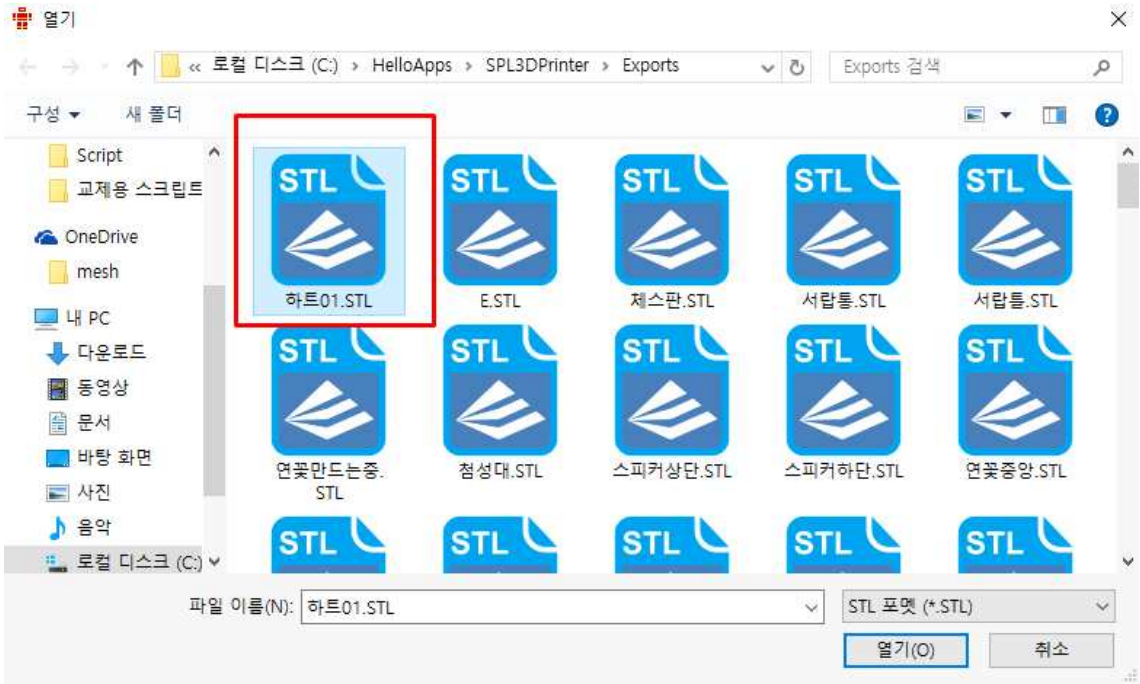
우선 거대한 하트를 만들도록 합니다.

실행을 한 뒤 모델링 파일로 내보내기를 합니다.

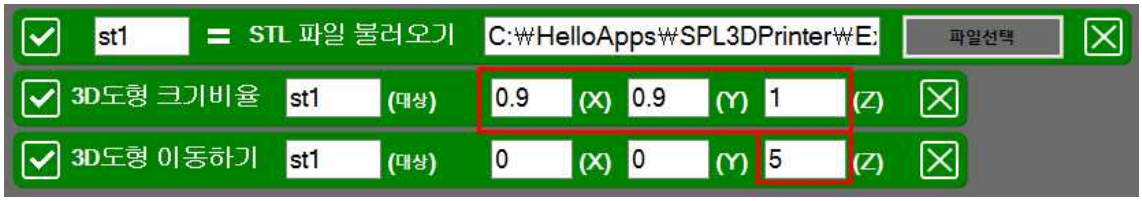


임의의 이름 지정하여 내보내기 합니다.

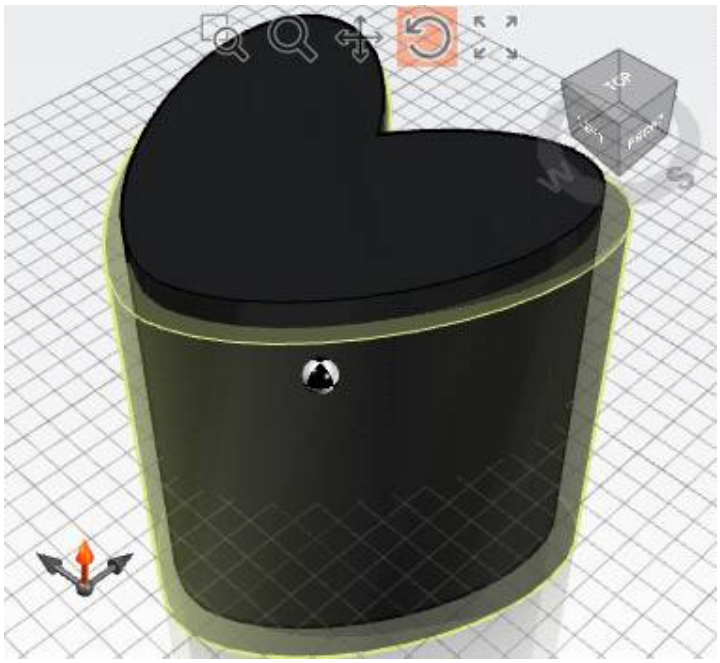
도형 불러오기 명령어를 사용합니다.



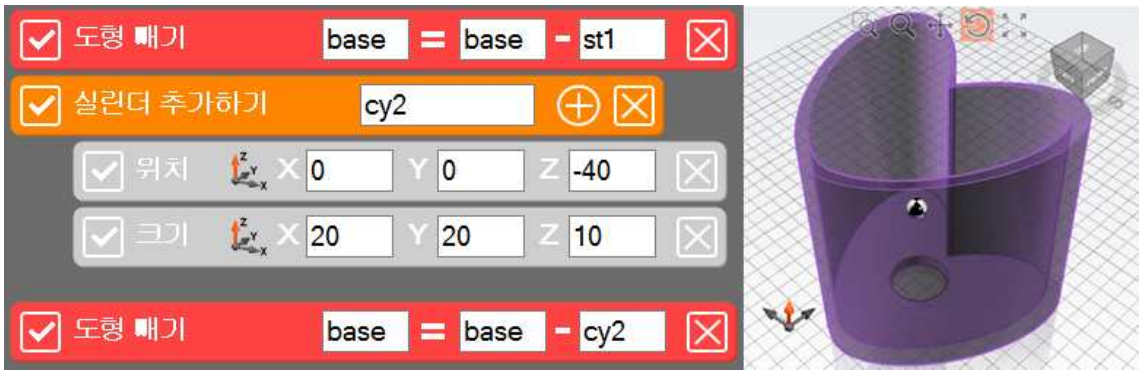
아까 내보내기 했던 파일을 다시 불러옵니다.



불러오기 한 파일의 이름은 st1이 되었고 3D도형 크기비율로 이 불러온 하트를 줄여줍니다.
다음으로 이 줄어든 하트를 3D도형 이동하기를 이용하여 Z축으로 5만큼 이동시킵니다.



위 그림과 같은 모습을 하게 됩니다.
이제 이 둘을 도형 빼기를 합니다.

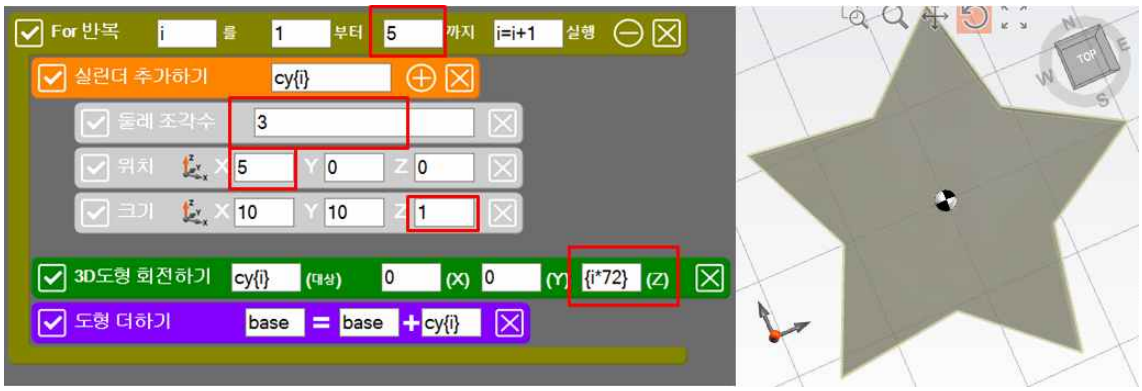


그리고 실린더 하나를 더 추가해줍니다
추가해준 실린더는 화분 바닥의 구멍을 뚫어줄 실린더입니다.
실린더를 하트통 하단까지 내린 뒤 도형 빼기를 해주면 완성입니다.

4) 별모양 연필꽂이 만들기

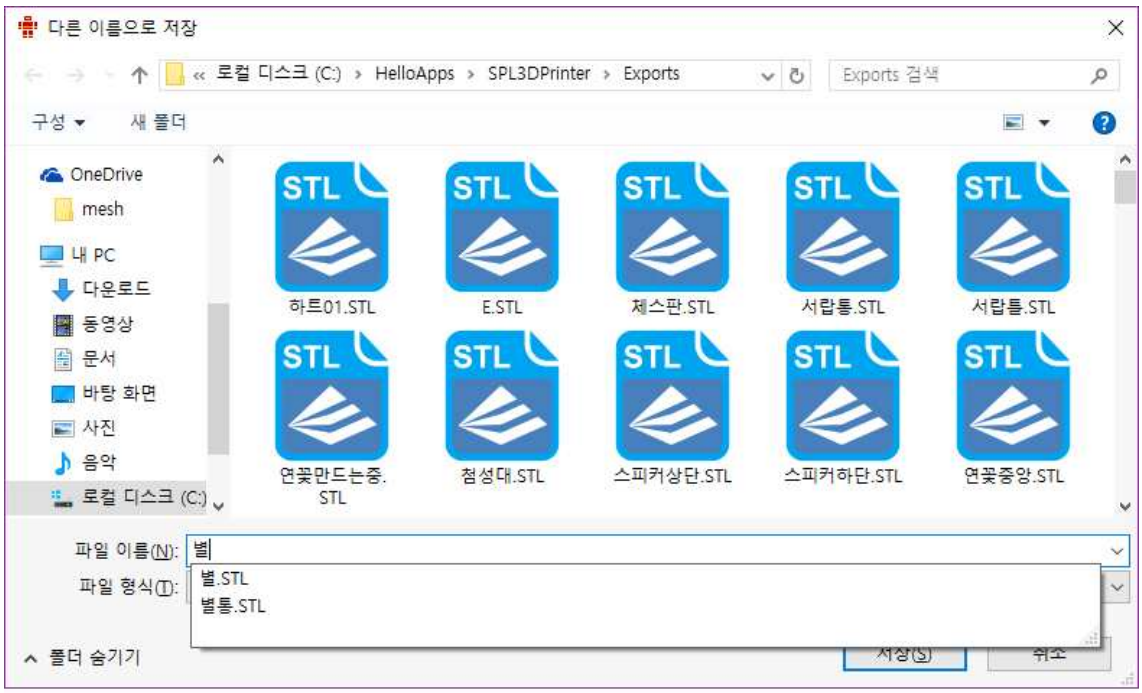
별모양은 하트와 마찬가지로 직접 만들어야 합니다.
일단 실린더를 만들어 얇은 크기로 지정합니다.

For 반복문을 하나 만들어줍니다.
위에서 만든 실린더에 총 5개의 삼각형을 합쳐줄 For문입니다.



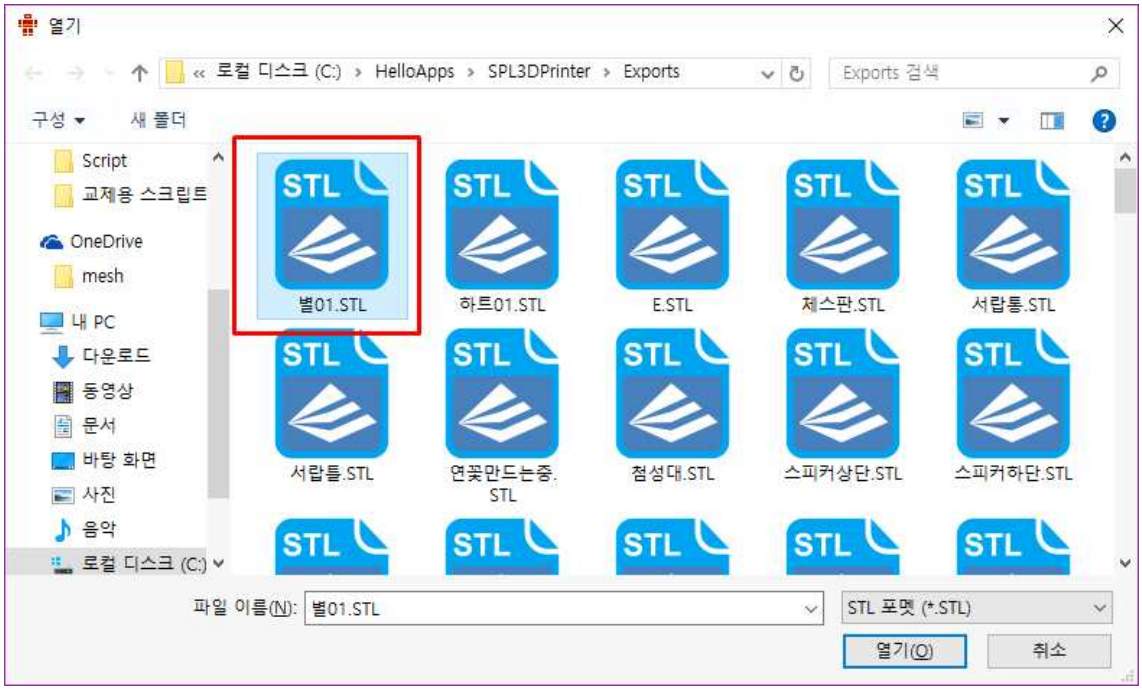
For 반복문에서 5개의 삼각형을 사용 할 것이니 1부터 5까지로 수정 합니다
실린더를 추가해서 cy{i} 이름을 변경하고 둘레조각수를 3개로 하여 삼각형을 만듭니다.
위치를 실린더(base)로 빼시고 크기를 실린더와 같은 높이로 수정해줍니다.
이 삼각형들을 회전을 시키기 위한 3D도형 회전하기 명령어를 추가 합니다.
회전값 Z에 {i*72}를 입력합니다.
마지막으로 실린더(base)와 삼각형들을 더합니다.

별모양을 내보내기를 합니다.

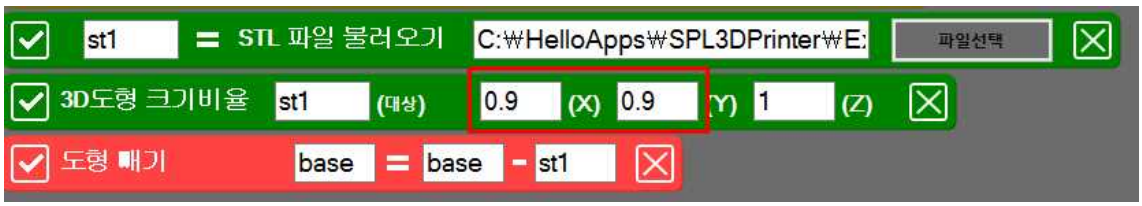


STL파일로 저장합니다.

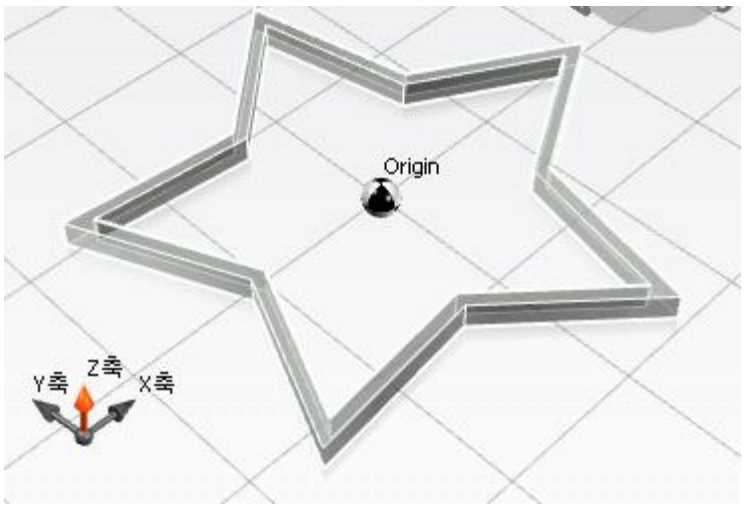
도형 불러오기 명령어를 사용합니다.



저장했던 별을 불러옵니다.

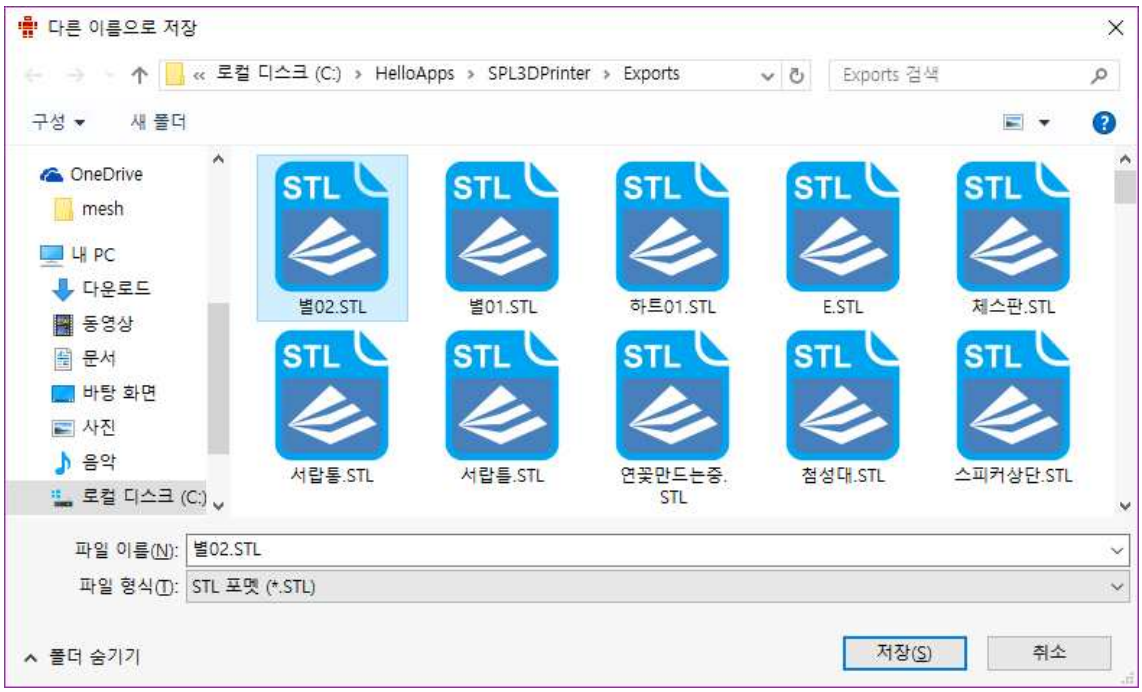


불러온 별의 크기비율을 줄여줘서 도형빼기를 해줍니다.

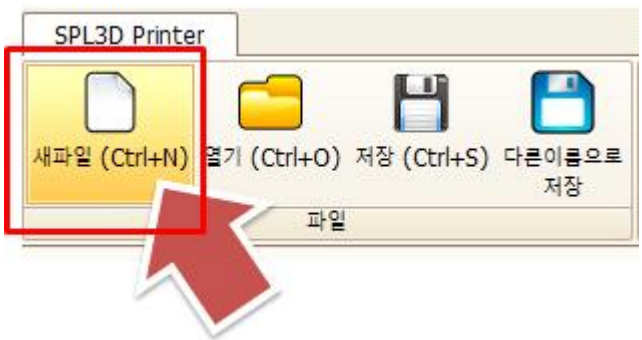


위 그림과 같은 모양을 지니게 됩니다.

다시 이 모델링파일을 내보내기를 해줍니다.

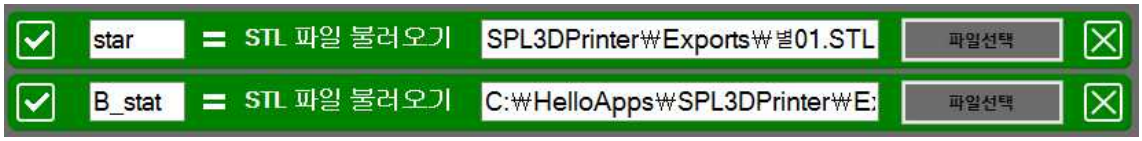


새로운 이름으로 저장해줍니다.



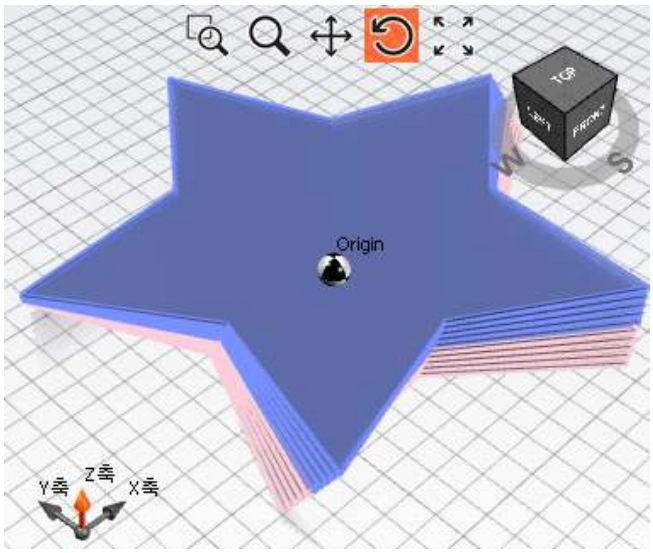
새파일을 실행합니다.

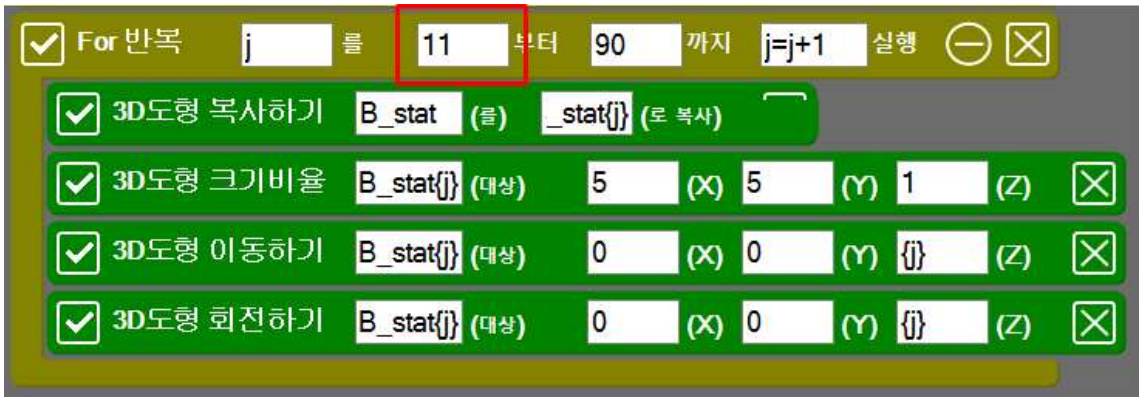
아까 내보내기를 했던 별들을 이용하여 꽈배기 모양의 연필꽂이를 만들고자 합니다.



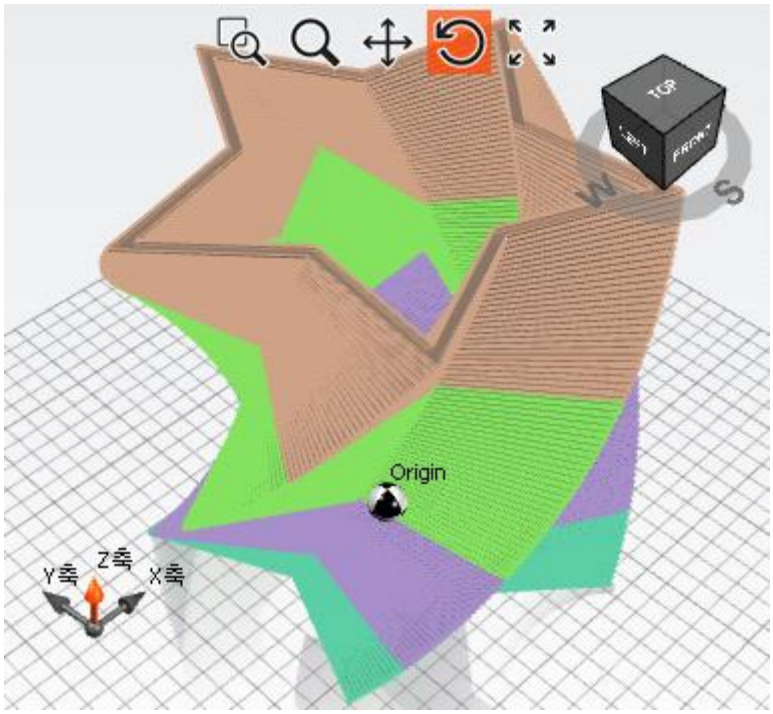
아까 내보내기를 했던 두 개의 STL파일을 불러옵니다.
둘의 이름을 다르게 변경합니다.
이 별들로 한층한층 쌓아 올려야 하기 때문에 For반복문을 사용하도록 합니다.
star는 하단 바닥을 B_stat는 그 위를 덮어줄 도형들입니다.

For 반복문을 생성하여 그 안에 3D도형 복사하기를 먼저 생성합니다.
복사할 대상의 이름을 입력하고 새로운 이름을 입력하고 그 뒤에 {i}를 붙여줍니다.
도형의 크기가 만족스럽지 못하다면 크기비율로 도형의 크기를 늘려줍니다.
그리고 3D도형 이동하기와 회전하기의 Z값에 각각 {i}를 입력합니다.
이러면 i가 증가 할 때마다 도형 star는 1씩 올라가도 1도씩 회전하게 됩니다.





마찬가지로 다시 For 반복문을 밑에 생성해줍니다.
나머지 위로 올라갈 별을 만들기 위한 For문입니다(※주 : 2중 For문이 아닙니다.)
반복문 j의 최소값을 11로 변경해줍니다.
반복문 i의 최대값이 10이였기에 반복문 i위에서부터 쌓이기 위해선 j에선 i의 최대값 + 1부터 시작합니다.
3D도형 복사하기의 대상을 복사하고 새로운 이름을 주어 뒤에 {j}를 입력합니다.
마찬가지로 3D도형 크기와 이동, 회전의 명령어를 주어 연필꽂이를 완성하도록 합니다.



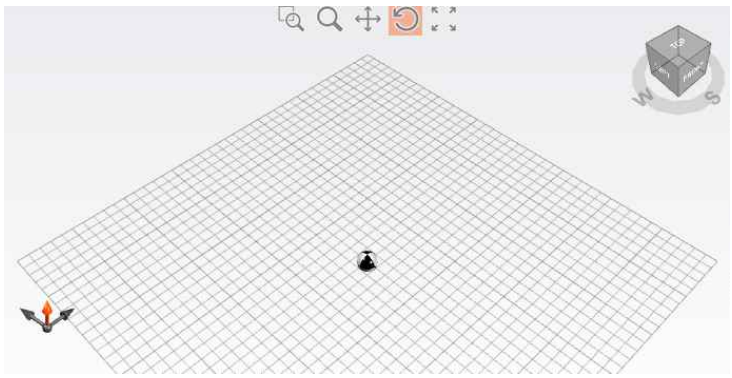
위 그림과 같은 모습이 완성되었습니다.

13. SPL For 3D 프린팅 로직 명령어 함수

13.1 함수 정의

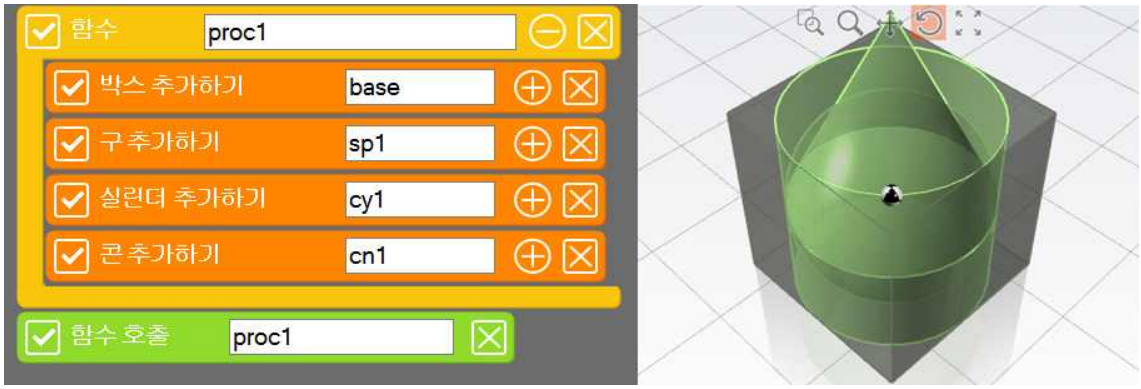
위 그림은 함수의 블록박스입니다.
함수 명령어를 사용하면 기본적으로 proc1, proc2 이름이 붙지만 자율적으로 변경이 가능합니다.

함수 안에 도형들을 추가해보았습니다.

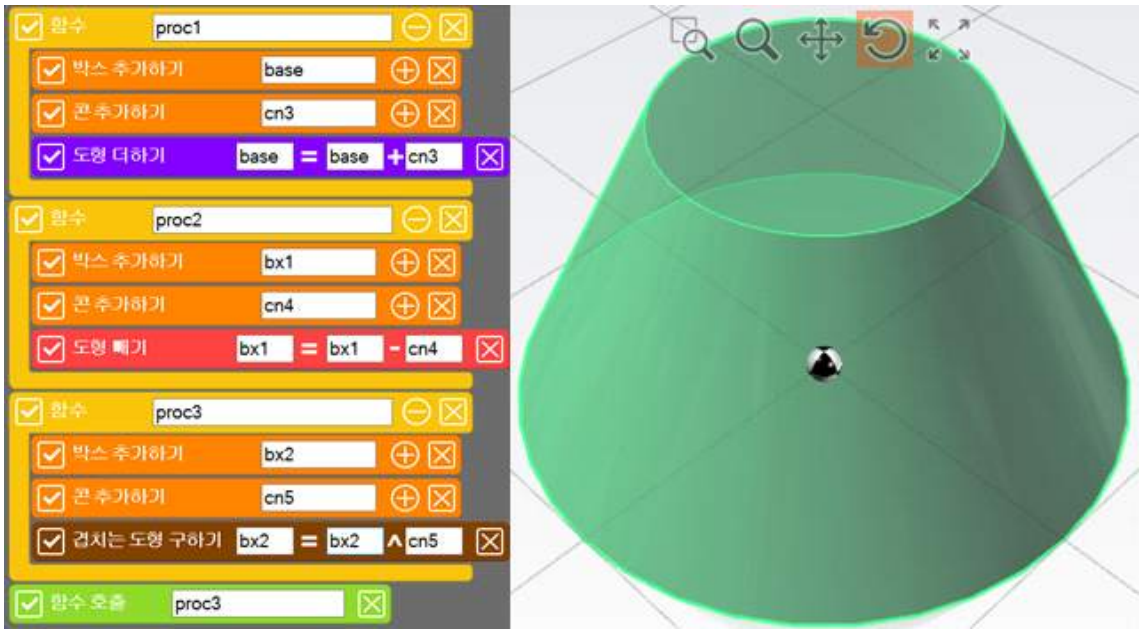


하지만 실행을 하여도 결과창엔 아무런 변화가 없습니다.

함수 안에 있는 내용을 사용하기 위해선 함수 호출이라는 명령어를 사용해야합니다.
함수 호출의 흰색 네모박스에 사용할 함수의 이름을 정확히 입력해주어야 합니다.



proc1라는 함수에 기본 도형들을 추가하였고 그 밑에 함수 호출을 이용해 proc1의 함수를 호출 하였습니다.
그 결과 실행하면 결과창에 기본도형들이 나타나게 됩니다.

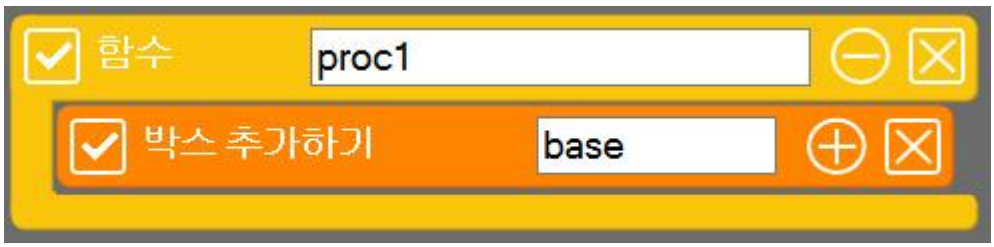


위 그림과 같이 그룹을 지어 원하는 함수를 호출하여 사용할 수 있습니다.

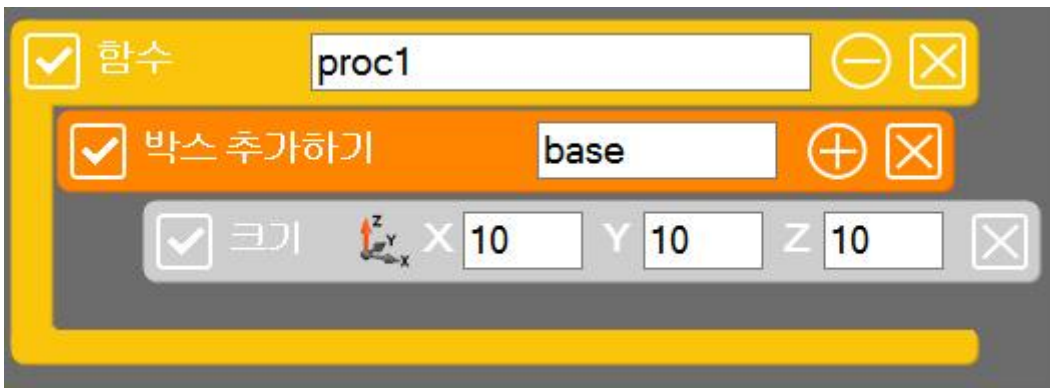
13.2 함수의 활용

다음은 함수의 실질적인 활용법에 대해 알아보고자 합니다.

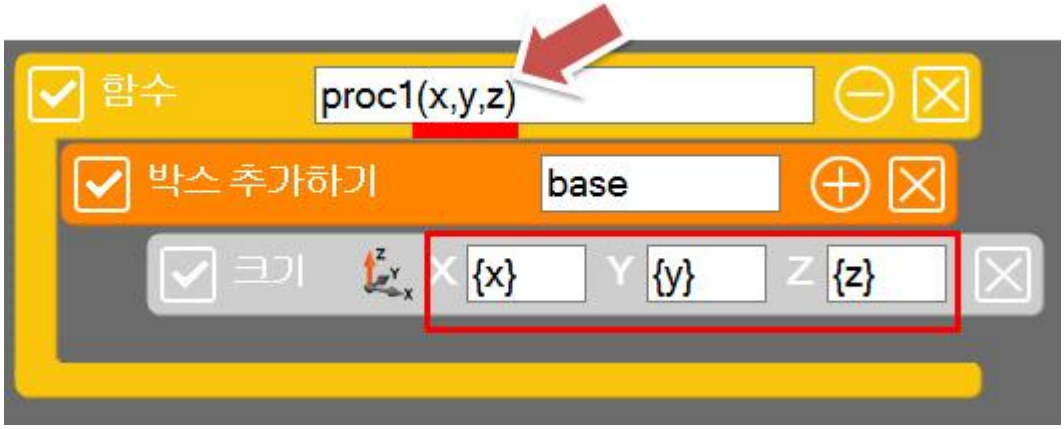
함수 하나를 만들어 줍니다.



함수 안에 박스 하나를 추가해줍니다.



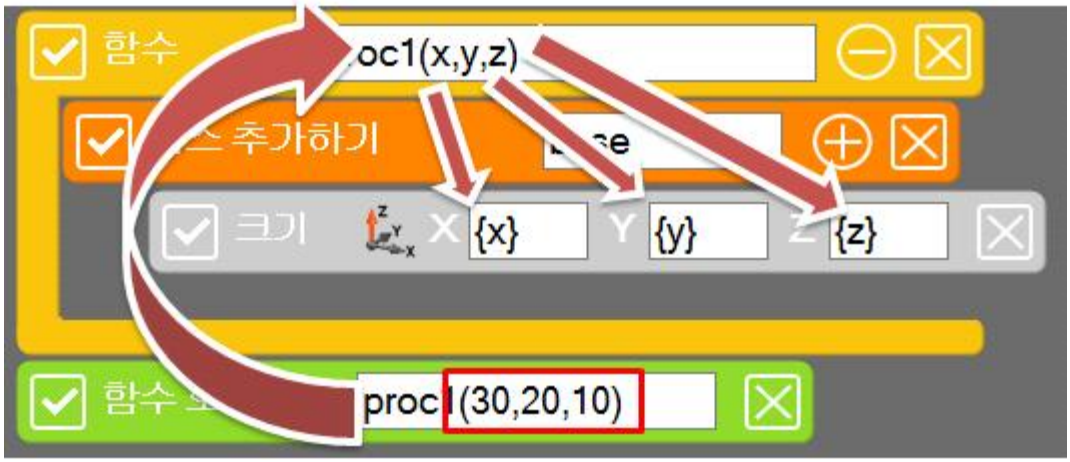
박스의 크기 메뉴를 하나 만들어줍니다.



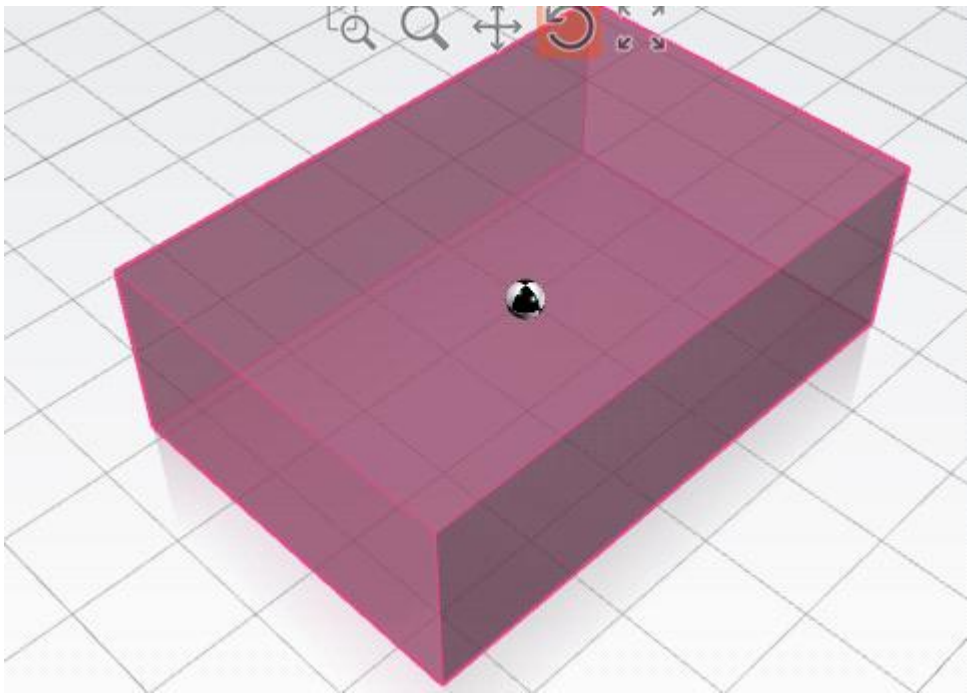
함수의 이름 뒤에 괄호 ()를 만들어줍니다.
그리고 그 괄호 안에 x, y, z를 입력합니다.
이렇게 함으로써 함수 proc1은 x, y, z라는 입력할 수 있는 변수를 3개를 만들었습니다.
함수 이름에 x, y, z를 만들었으니 함수 본문 안에서도 x, y, z가 사용될 곳을 만들어 줘야합니다.
아까 만들었던 박스의 크기에 {x}, {y}, [z}를 입력해줍니다.

함수 호출 명령어를 사용합니다.
위에서 만든 함수 proc1(x,y,z)를 만들었으니
함수 호출 또한 proc1(x,y,z)로 호출하시면 안됩니다.
만들었던 괄호 ()안에는 숫자 혹은 다른 변수를 입력해야합니다.
위 그림과 같이 괄호 ()안에 숫자를 입력해주셔야 합니다.

함수를 만들 때 x, y, z 3개의 입력 변수를 만들었기에 함수 호출시에도 3개의 입력을 줘야합니다.

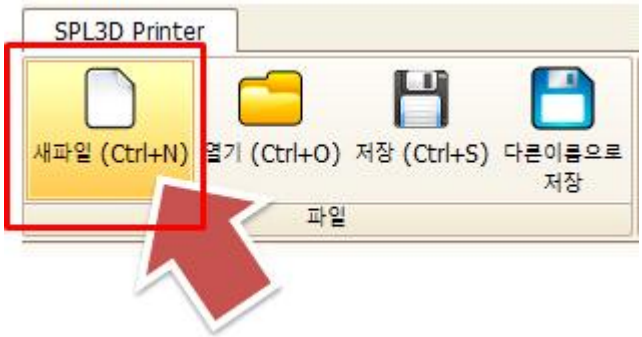


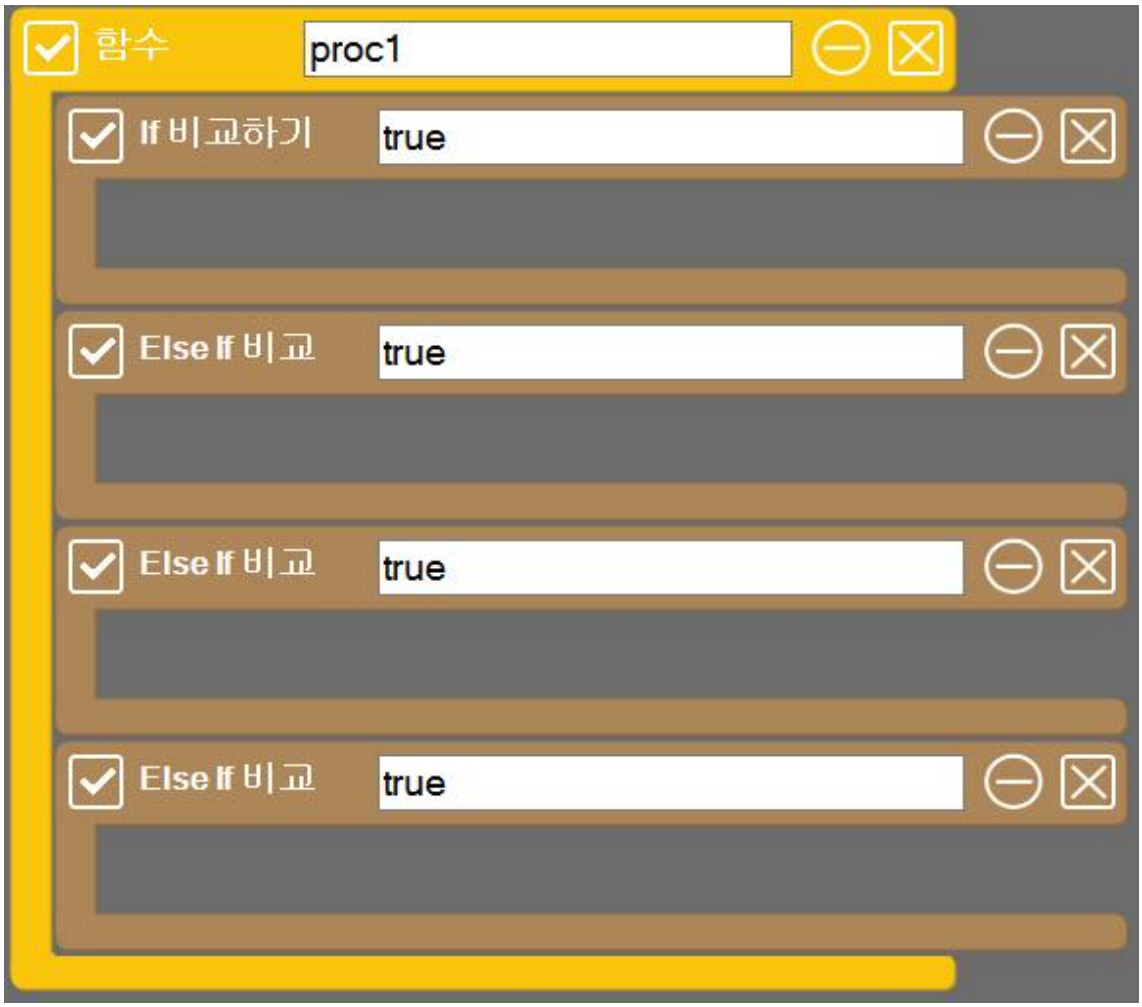
함수 호출에서 30, 20, 10 순으로 입력을 했습니다.
그렇게 되면 proc1의 함수에 x에는 30, y에는 20, z에는 10이 적용되어
함수 안에 지정해준 크기 X : 30, Y : 20, Z : 10이 들어가게됩니다.



실제 적용이 된 모습입니다.
크기 이외에도 숫자가 적용되는 모든 곳에 변수를 생성해주면 변화를 시킬 수 있습니다.

이번엔 If문으로 사용해보도록 하겠습니다.





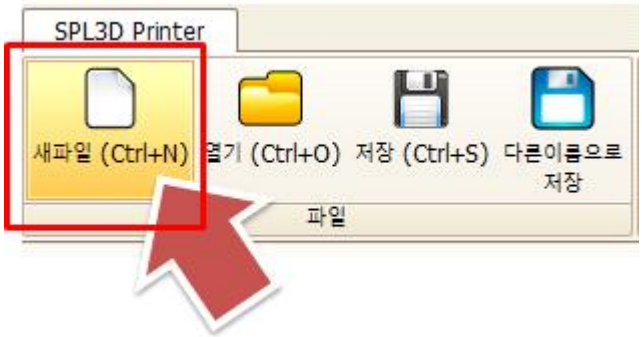
새파일을 열어 함수를 만들어줍니다.
만든 함수 안에 If 비교하기 명령어와 Else If 3개를 만들어 줍니다.

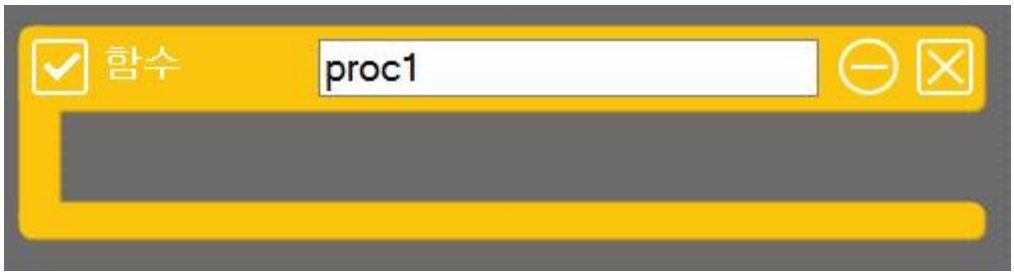
비교하기 문장들 안에 각각 다른 도형들로 넣어줍니다.
그리고 함수 이름 뒤에 (n)을 입력해줍니다.
비교하기 문장에 각각 n == 1, n == 2, n == 3, n == 4를 넣어줍니다.

함수 호출을 사용해보도록 합니다.

함수 호출을 하여 함수 proc1()에 괄호 안에 숫자를 각각 1~4씩 넣었을 때의 결과입니다.
함수와 함수 호출을 이용하여 특정 명령을 가지고 올 수도 있습니다.

이번엔 For 반복하기에 사용해보도록 하겠습니다.





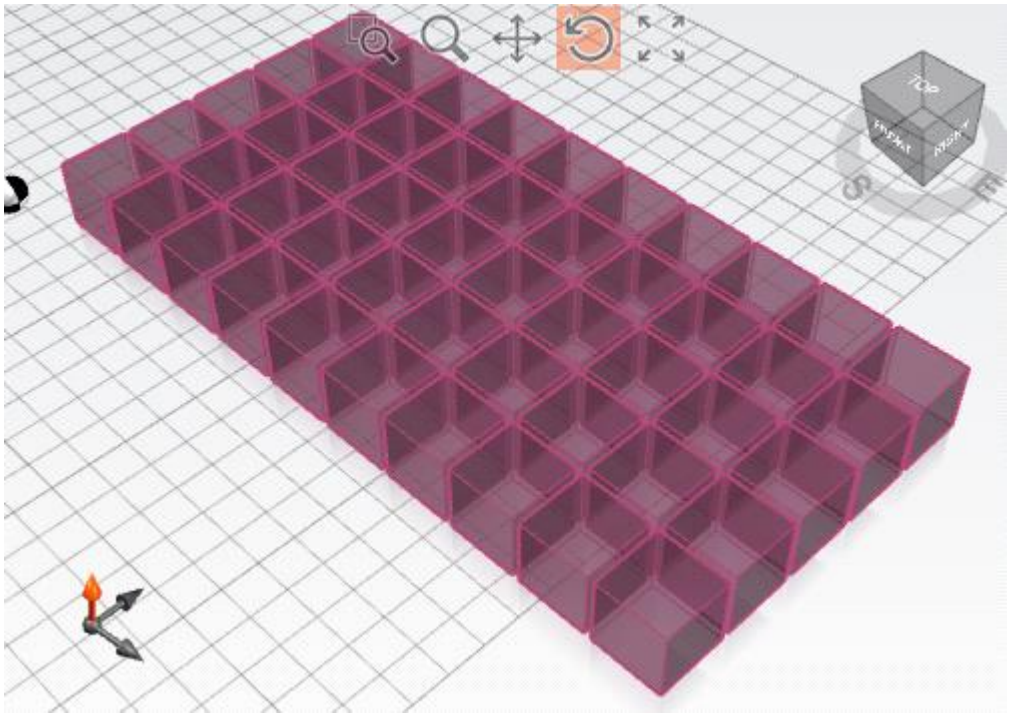
새파일을 열어 함수를 만들어 줍니다.

함수 안에 2중 For문을 작성해줍니다.

For반복문 j안에 박스를 추가하여 base{i}{j}로 이름을 바꾸어 For문을 사용할 수 있게 합니다.
박스를 x, y 방향으로 10개씩 배치하는 일반적인 2중 For문입니다.

함수이름 proc1에 (x,y)를 추가로 입력해줍니다.
그리고 For 반복문들의 최대값에 x와 y를 입력해줍니다.
이렇게 함으로써 박스 배치의 최대값을 임의로 수정해줄 수 있습니다.

이제 원하는 만큼 숫자를 입력하면 박스를 채워 넣을 수 있습니다.



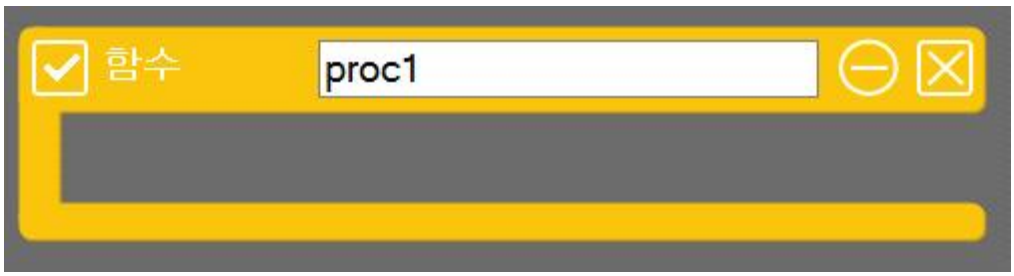
위 그림과 같이 proc1(10,5)를 입력하여 10칸 5줄로 채워진 박스를 만들었습니다.

이렇게 함수를 이용하여 입력 변수를 만들어 함수 호출을 이용해 수치 조정이 가능해집니다.

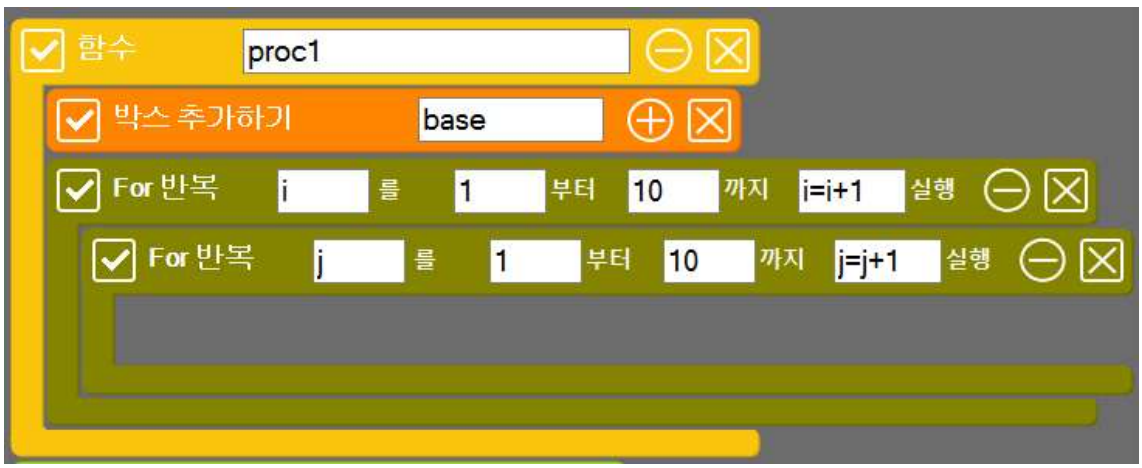
13.3 함수를 이용하여 모델링 제작

1) 레고 블록 1

For반복문 시간에 만들었던 레고를 이번엔 함수를 이용하여 언제든 자신이 크기 조절을 할 수 있게 만들어 보도록 하겠습니다.



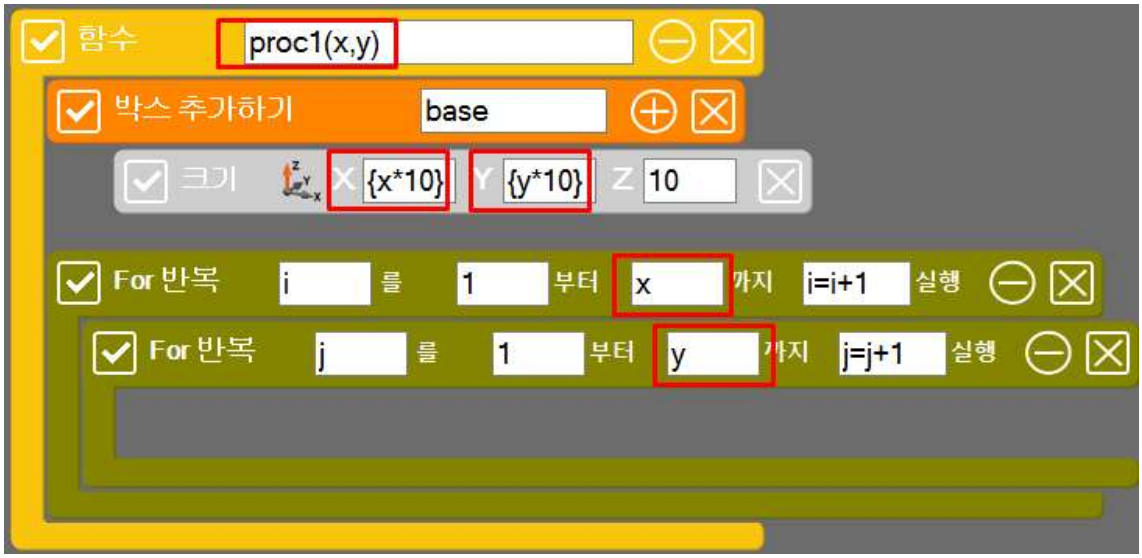
함수를 하나 만들어 줍니다.



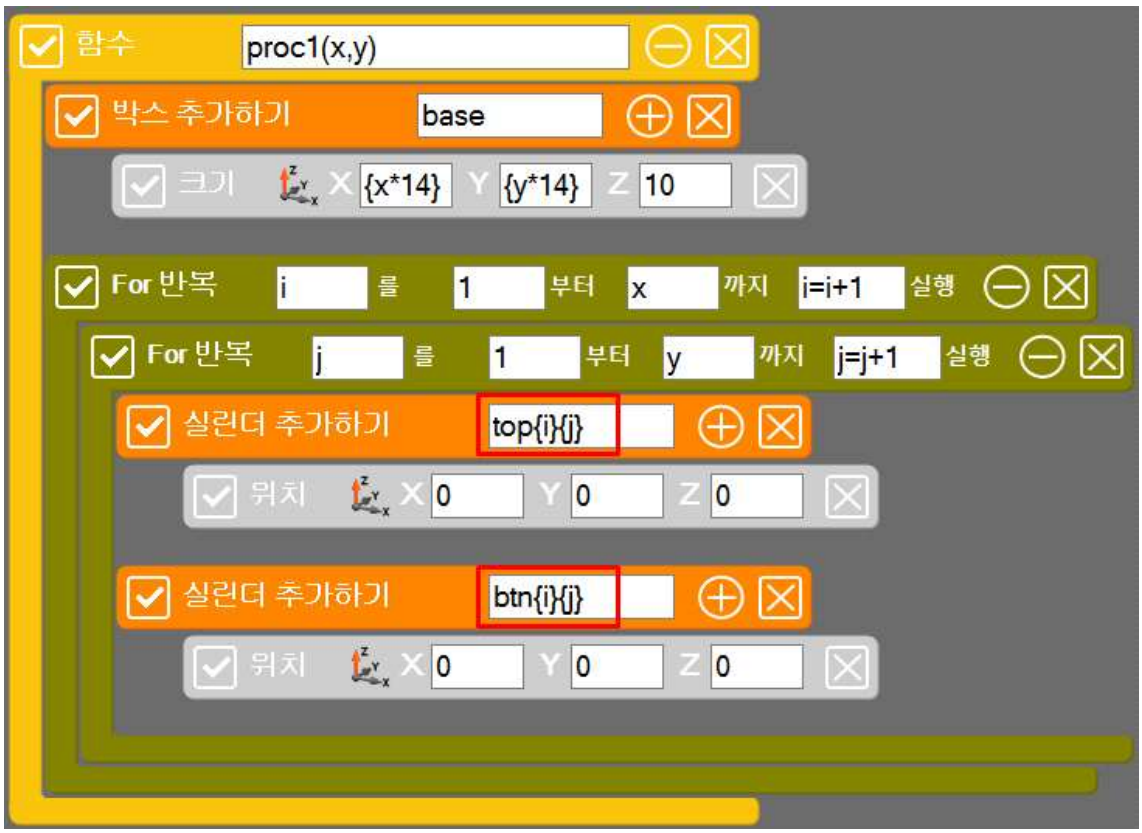
함수안에 박스 하나와 2중 For문을 하나 만들어줍니다.

※tip :

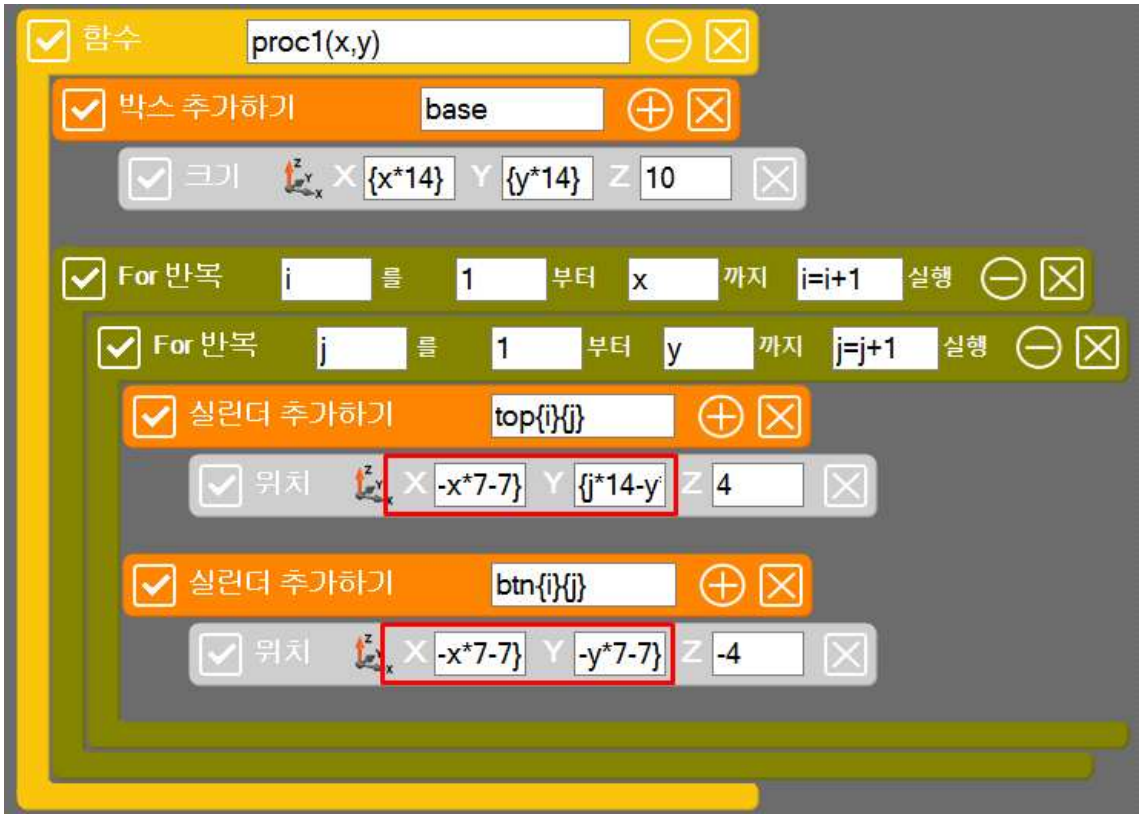
함수안에 진행되고 있는 모델링의 상황은 결과창에 나타나지 않으니 함수 밑에 함수 호출 명령어를 사용하여 결과창에 나타날 수 있게 할 수 있습니다.



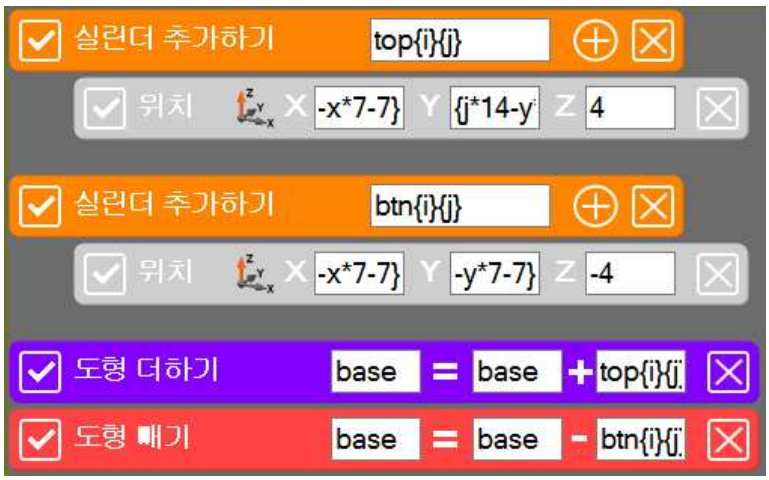
함수의 이름 뒤에 (x,y)를 추가해줍니다.
박스 크기를 만들어 크기 X : {x*10}, Y : {y*10}을 입력해줍니다.
For 반복문의 최대값에 각각 x, y로 바꿔줍니다.
이제 함수 호출시 입력한 x, y의 값에 따라 박스의 크기와 For 반복문의 반복 개수가 달라집니다.



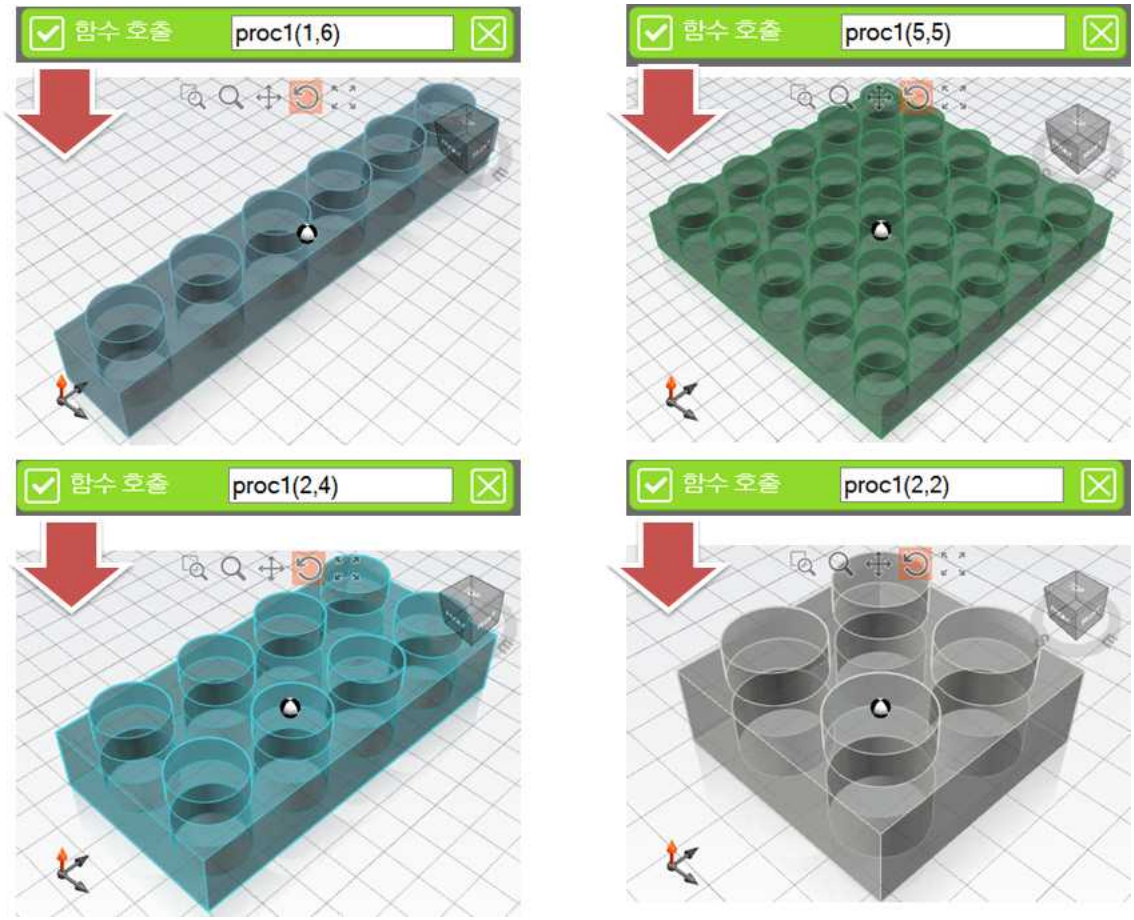
2중 For문에서 만든 레고와 마찬가지로 2개의 실린더를 만들어 이름을 다르게 변경해줍니다.



위치를 지정해줍니다.
두 실린더의 X에는 {i*14-x*7-7}, Y에는 {j*14-y*7-7}를 입력해줍니다.
상단 위치 Z는 4, 하단 실린더의 Z는 -4를 입력합니다.



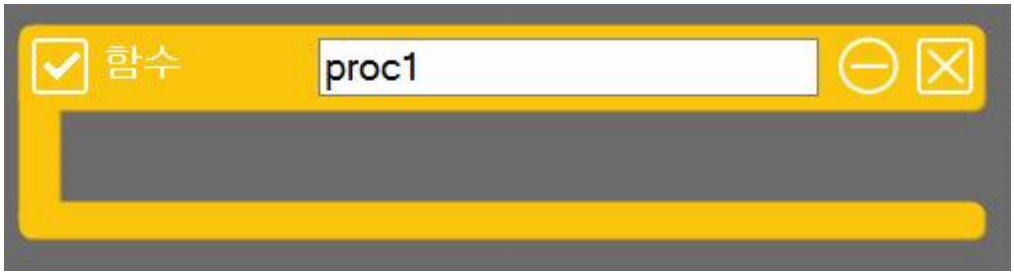
아래쪽에 도형 더하기와 도형 빼기를 추가해줍니다.



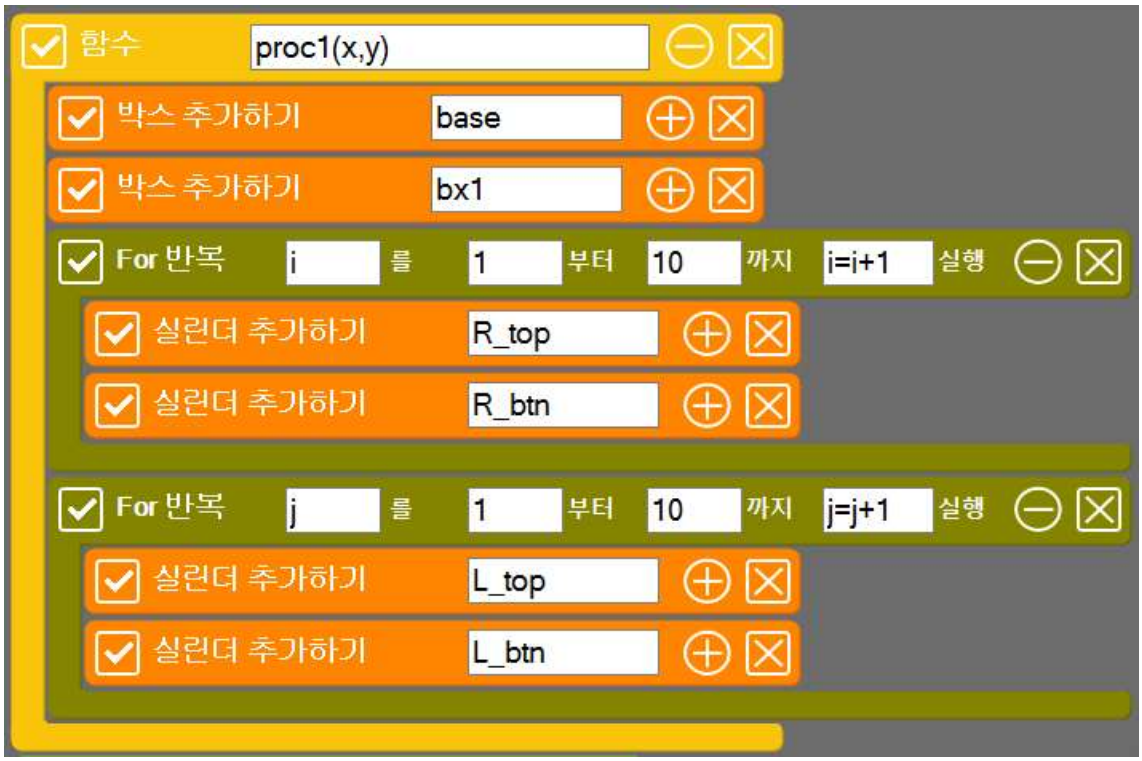
위 그림과 같이 함수 호출을 하여 입력 변수(x, y)에 임의의 숫자를 입력하여 원하는 만큼의 가로 세로를 가진 레고 블록을 만들 수 있습니다.

2) 레고 블록 2

이번엔 ㄱ모양의 레고 블록을 만들고자 합니다.

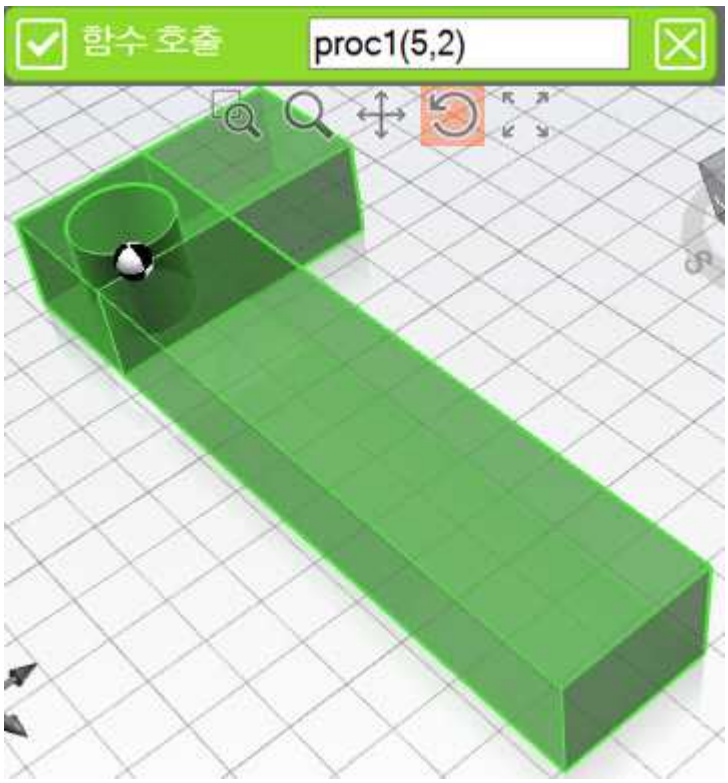


새파일을 열어 함수 명령어를 만들어 줍니다.

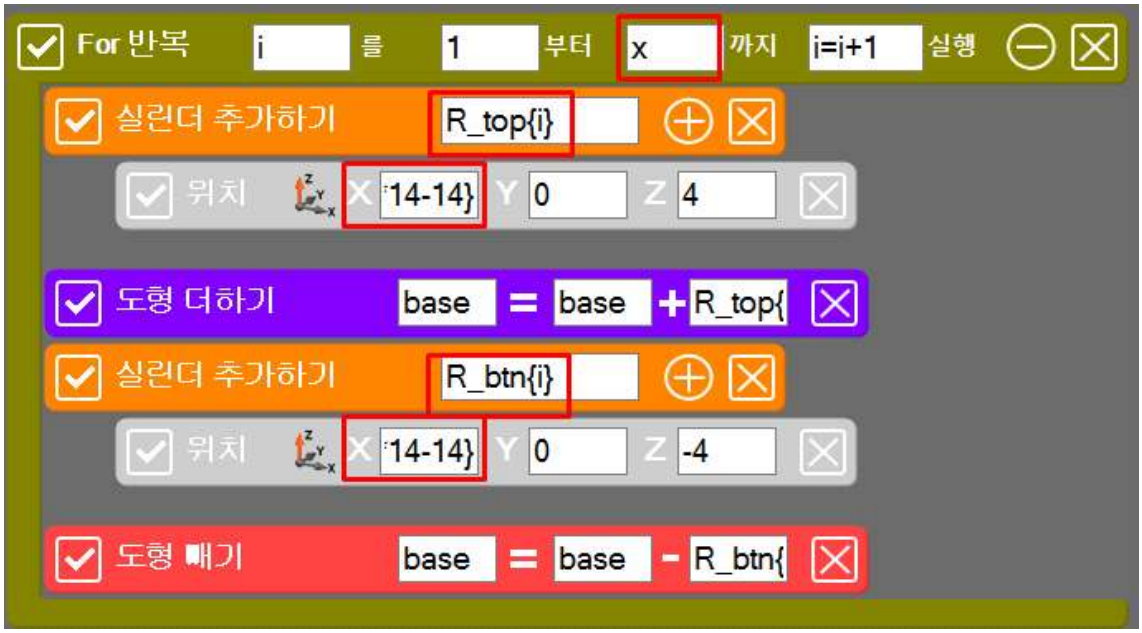


아까와 같이 박스와 For문을 사용합니다.
이번엔 박스를 두 개 만들어 줍니다.
For문 역시 2중 For문이 아닌 따로 사용합니다.
For문안에 두 개의 실린더를 생성합니다.
실린더들의 이름은 다 달리 해줍니다.

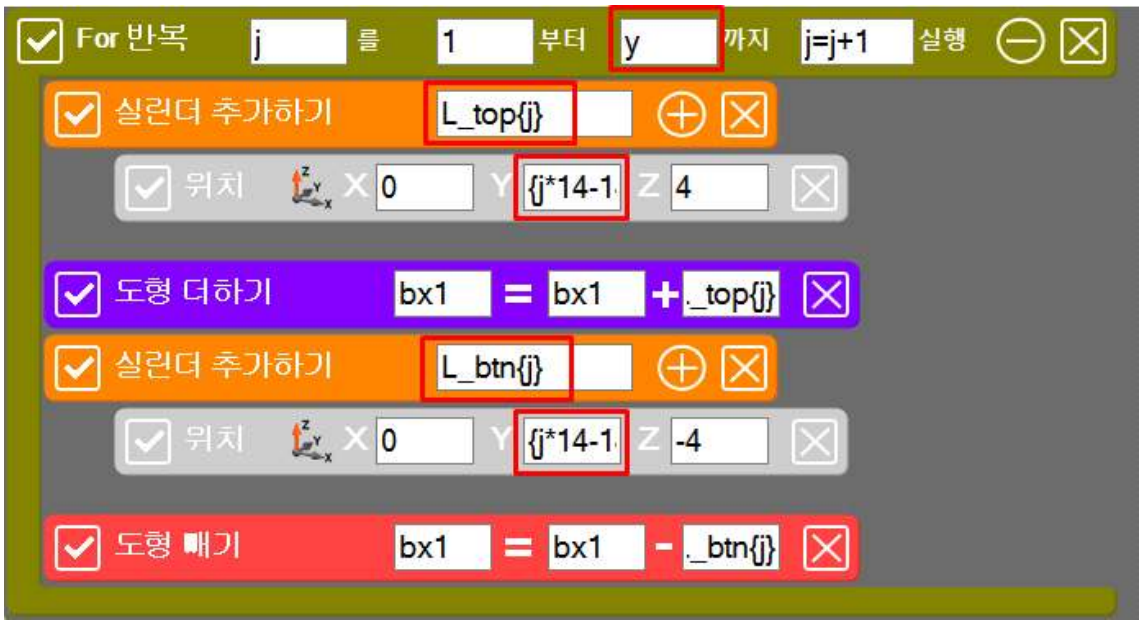
두 개의 박스에 위치와 크기를 만들어 줍니다.
ㄱ모양에서 각각 다른 방향으로 길어지는 박스들입니다.
base 박스의 위치 X : {x*7-7}, 크기 X : {x*14}, Y : 14로 입력하고 bx1 박스의 위치 Y : {y*7-7}, 크기 X : 14, Y : {7*14}를 입력합니다.



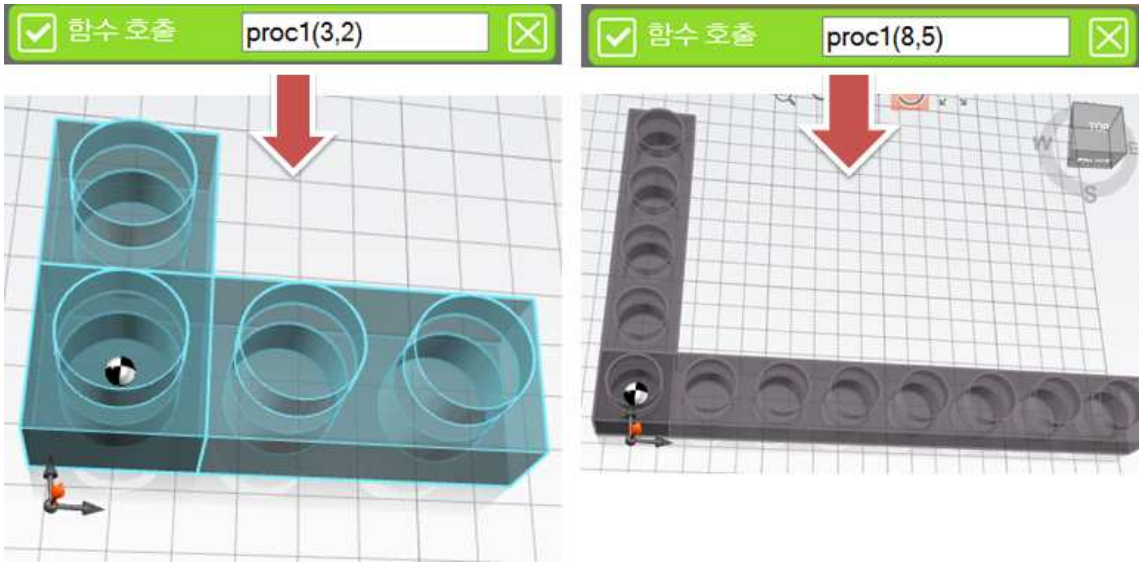
잠깐 함수 호출을 이용해서 적용되었는지 확인한 모습입니다.



첫 번째 For반복문에서 최대값을 x로 수정합니다.
첫 번째 실린더를 R_top{i}로 이름을 바꾸고 위치를 만들어 X : {i*14-14}, Z : 4를 입력하고 도형 더하기 base = base + R_top{i}를 해줍니다.
두 번째 실린더의 이름은 R_btn{i}에 위치를 만들어 X : {i*14-14}, Z : -4를 입력하고 도형 빼기를 하여 bx1 = bx1 - R_btn{i}를 해줍니다.



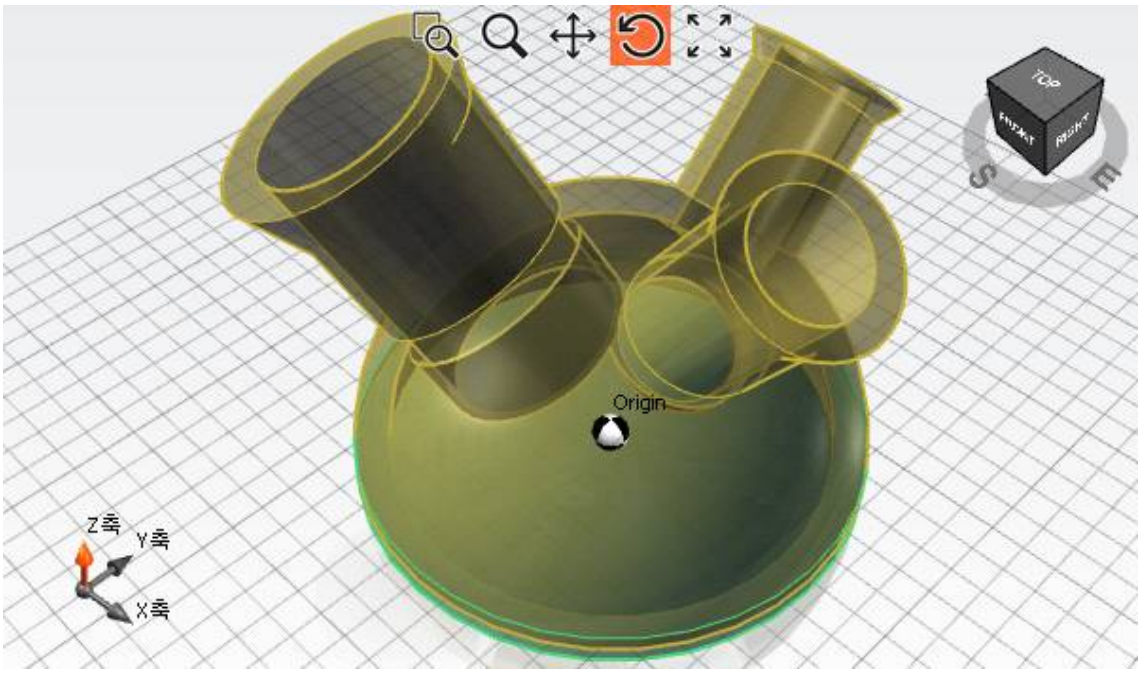
마찬가지로 이번엔 두 번째 For반복문에서 최대값을 y로 수정하여 L_top{j}로 이름을 바꾸고 위치는 X : {j*14-14}, Z : 4를 입력하고 도형 더하기 base = base + L_top{j}를 해줍니다.
두 번째 실린더의 이름은 L_btn{j}에 위치를 만들어 X : {j*14-14}, Z : -4를 입력하고 도형 빼기를 하여 bx1 = bx1 - L_btn{j}를 해줍니다.



함수 호출을 하여 임의의 값을 입력하여 확인해 봅니다.
위 사진과 같이 원하는 수치를 입력하여 ㄱ모양의 레고의 칸수를 조절할 수 있습니다.

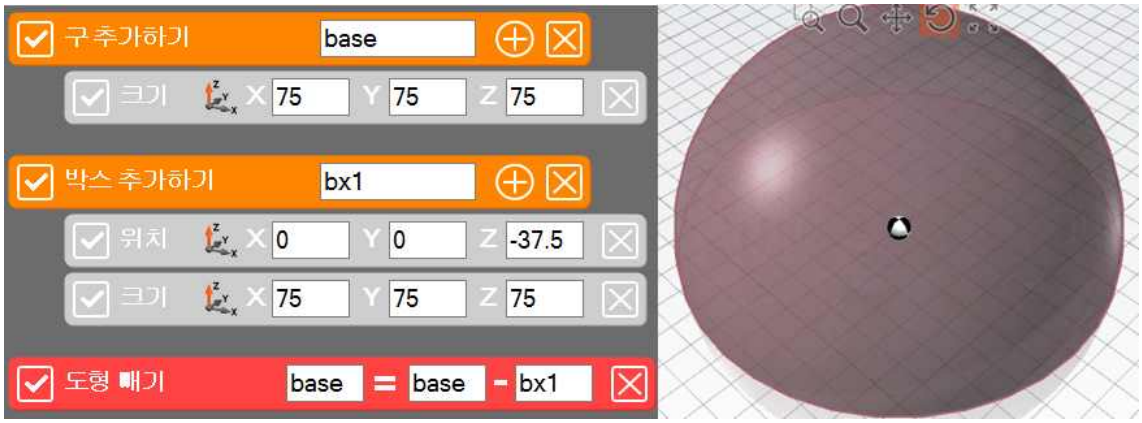
14. 다른 창작물 만들어 보기

14.1 굴뚝 모양 꽃이 만들기

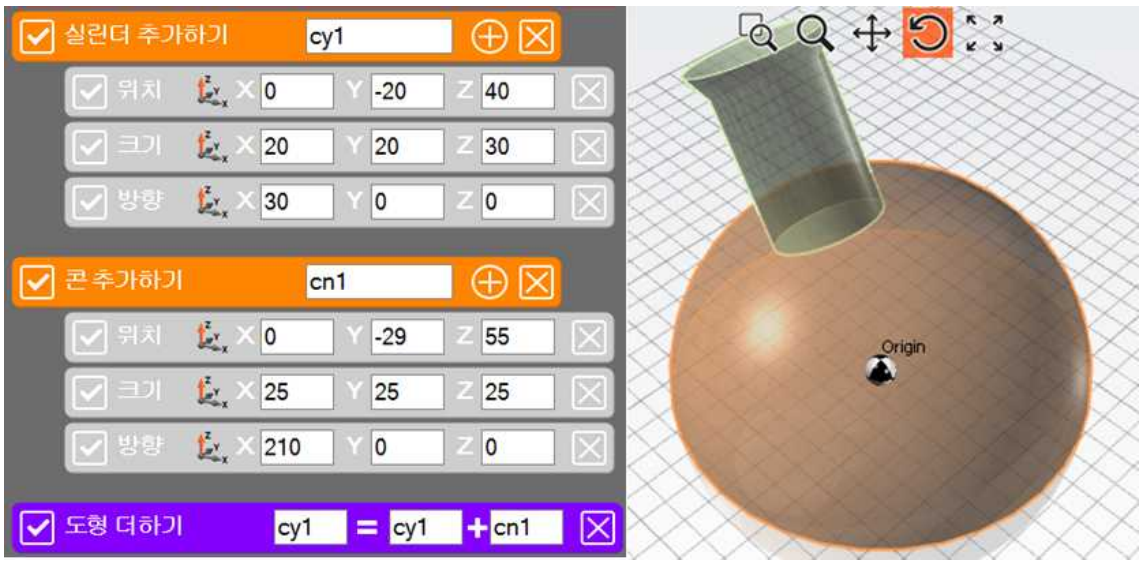


위 그림과 같은 굴뚝모양 꽃이를 만들어 보고자 합니다.

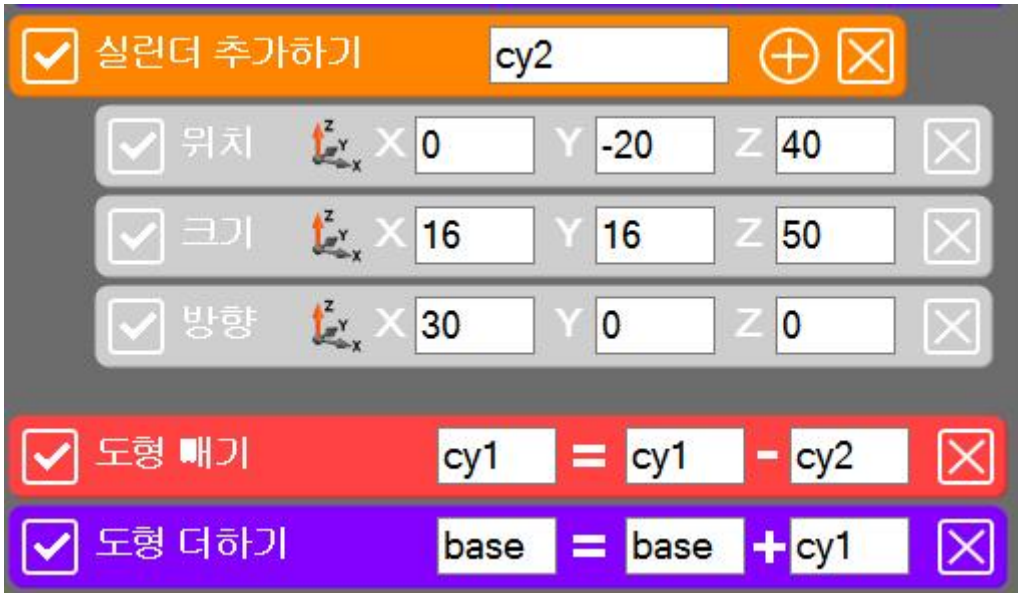
우선 적당한 크기의 구를 추가합니다.



박스를 하나 추가해 구와 같은 크기로 하여 구의 절반을 빼기 해줍니다.



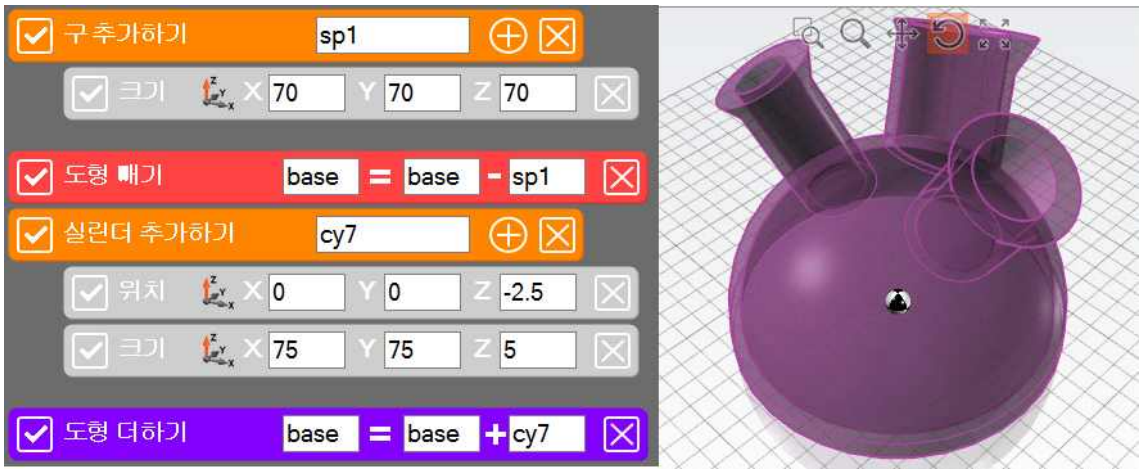
다음과 같이 실린더와 콘을 이용하여 굴뚝 모양을 만들어 줍니다.



실린더를 하나 더 추가해줍니다.
이 실린더는 방금 만든 굴뚝에 구멍을 뚫어줄 실린더입니다.
이 실린더와 굴뚝을 빼줍니다.
그리고 최종적으로 만들어진 이 굴뚝과 반원(base)와 더해줍니다.

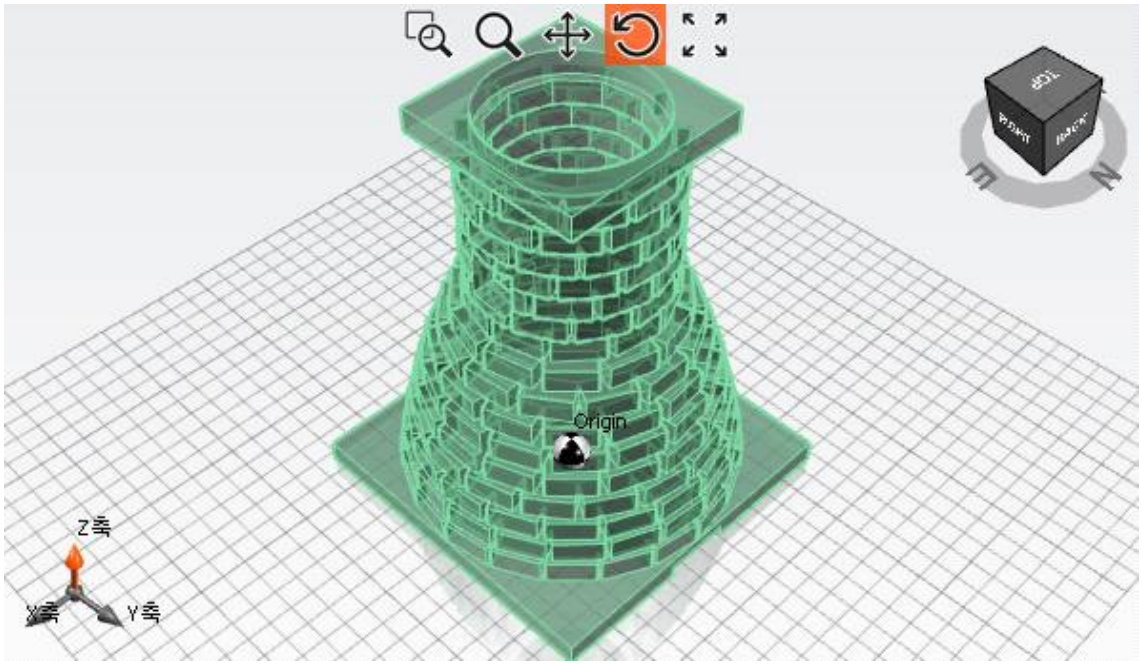
마찬가지로 다른 방향의 굴뚝을 만들어 줍니다.

나머지 한쪽 굴뚝도 만들어 줍니다.



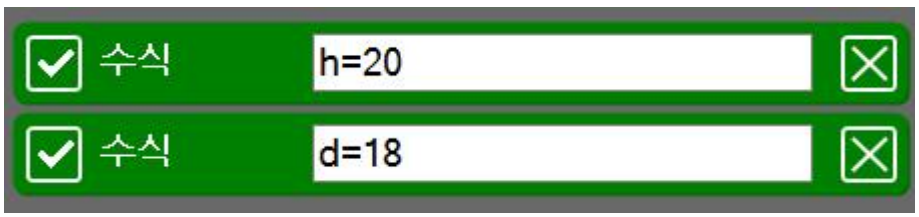
구를 하나 만들어 base의 구보다 약간 작게 하고 빼기를 해줍니다.
바닥이 뚫려 있으니 실린더를 만들어 구의 넓이는 구의 크기와 동일하게 해주고 높이는 적당한 크기로 하여 최종적으로 도형 더하기를 해줍니다.
위 그림 오른쪽과 같이 굴뚝모양의 꽃이를 만들 수 있습니다.

14.2 첨성대 만들기



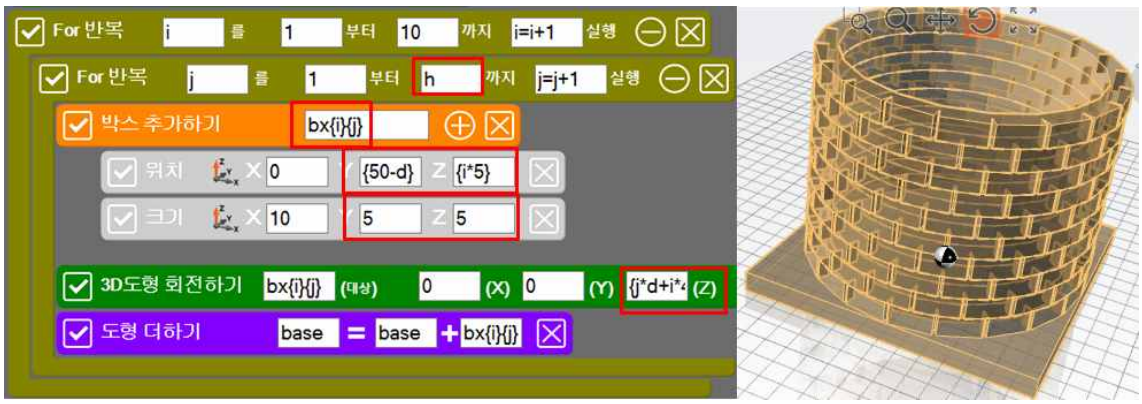
위 그림과 같은 첨성대를 만들고자 합니다.

우선 밑판이 될 박스를 하나 만들어 줍니다.

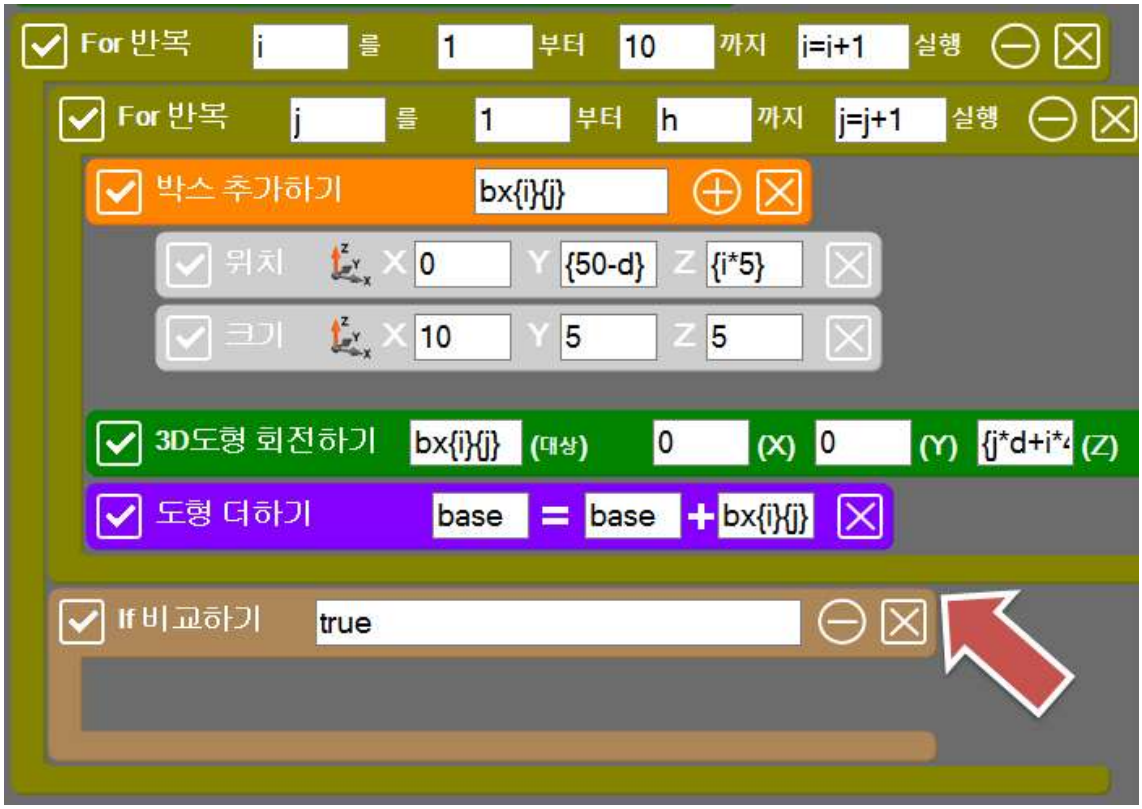


수식 두 개를 추가해줍니다.
이 수식에 대해선 밑에서 설명하겠습니다.

2중 For문을 만들어 줍니다.
그리고 j반복문안에 박스를 하나 추가해줍니다.
본 강의에선 이 박스를 벽돌로 가정하고 벽돌을 쌓아 올리는 형식으로 첨성대를 만들고자 합니다.

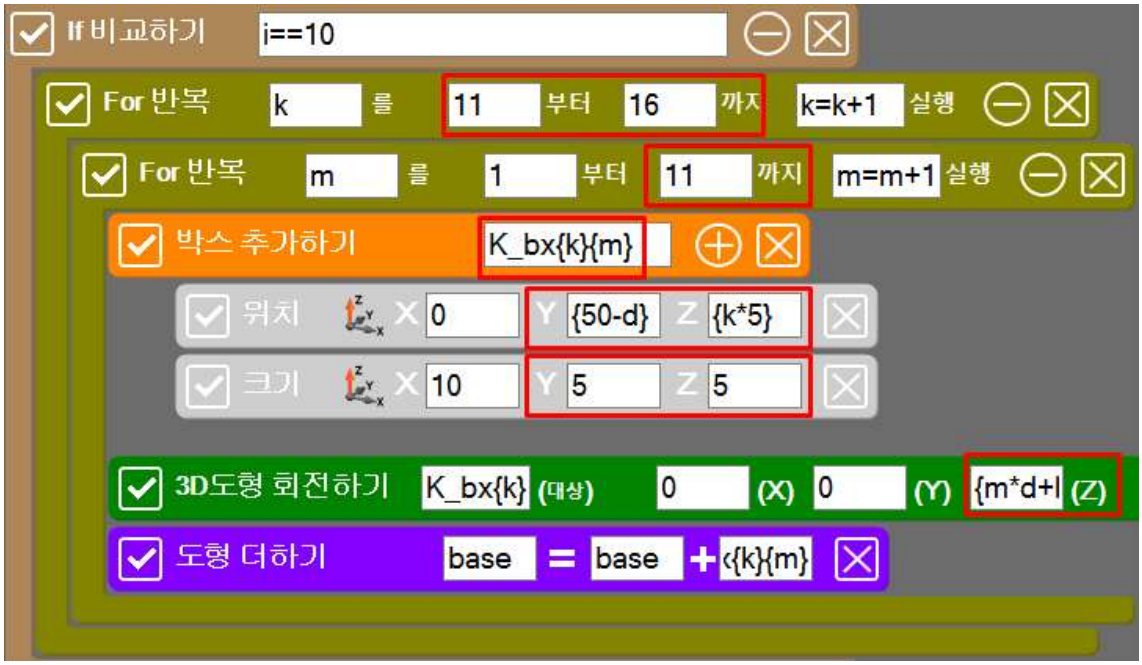


2중 For문에서 j의 반복문의 최대값을 h로 수정합니다.
박스의 이름을 반복문에 적용 시킬 수 있게 {i}{j}로 수정합니다.
위치를 만들어 Y : {50-d}, Z : {i*5}로 입력하고 크기를 만들어 Y : 5, Z : 5로 수정합니다.
3D도형 회전하기를 만들어 박스를 대상으로 하고 회전값 Z : {j*d+i*45}로 만들어 줍니다.
마지막으로 도형 더하기를 base = base + bx{i}{j}를 합니다.
이렇게 함으로써 벽돌(bx{i}{j})들을 반복문을 이용하여 회전시켜 둘러쌓고 한층 올라가 10층까지 둘러 쌓는 반복문을 만들게 되었습니다.

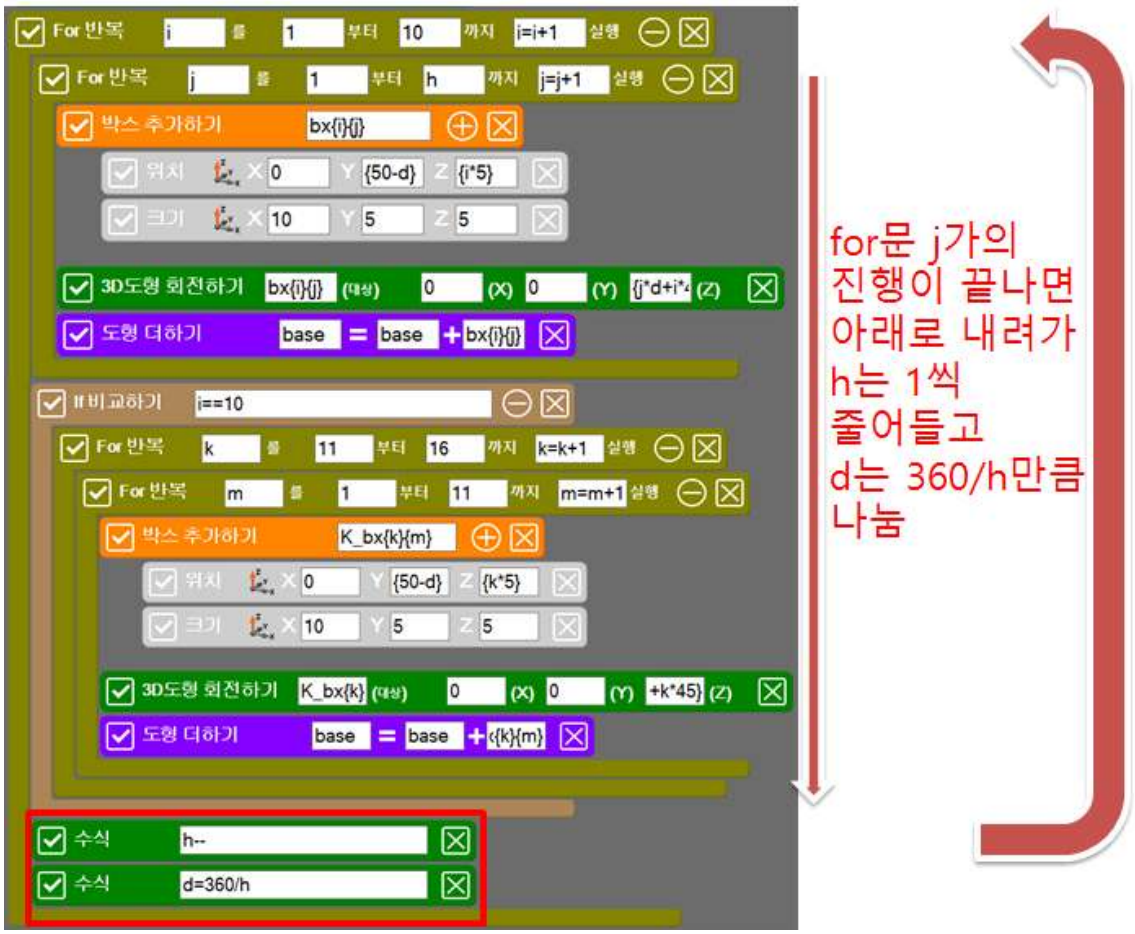


For 반복문 i에 If 비교하기 명령어를 추가해줍니다.
For 반복문 j 안쪽이 아닌 아래쪽에 있어야합니다.

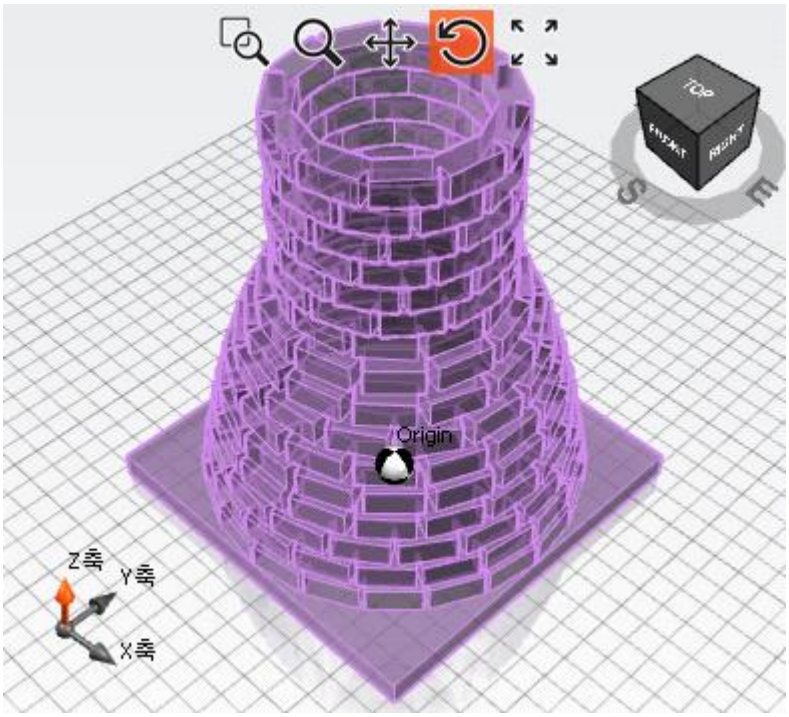
비교하기 안에 2중 For 반복문을 만들어 줍니다.
비교하기에서의 조건은 i == 10으로 조건을 만들어 줍니다.
위의 For 반복문 i가 10이 되면 해당 조건문에 들어오게 됩니다.



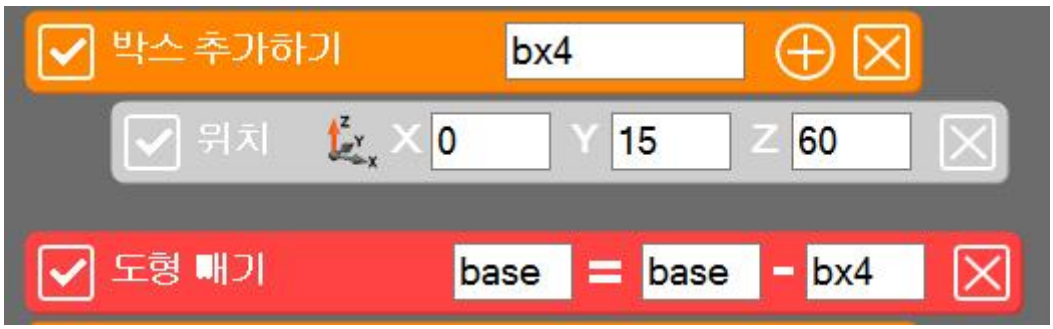
For 반복문에서 k는 11부터 16까지로 반복문 m의 최대값은 11로 수정합니다.
박스를 하나 추가해서 이름을 바꿔줍니다.
박스 이름은 위의 박스의 이름과 다른 것으로(본 강의에선 bx) 해주셔야합니다.
본 강의에서 이번에 추가한 박스의 이름은 K_bx{k}{m}으로 하였습니다.
박스의 위치를 만들어 Y : {50-d}, Z : {k*5}로 지정하였고 크기를 만들어 Y : 5, Z : 5로 입력하여줍니다.
3D 도형 회전하기 명령어를 만들어 대상은 이번에 만든 박스로 하고 회전값 Z : {m*d+k*45}로 입력합니다.
이번 반복문의 변수 이름이 k와 m이니 {i}{j}로 쓰는 실수를 범하면 안됩니다.



If비교하기 문장 아래에 수식 두 개를 넣어 줍니다.
h--와 d=360/h라는 수식입니다.
맨 처음에 만들었던 수식 h는 20이고 d는 18입니다.
h가 20일 때 첫 j의 반복문에 들어가 1부터 20까지 박스를 만들어 둘레를 칩니다.
3D도형 회전하기의 Z의 식은 박스의 크기와 개수에 맞춘 회전 값입니다.
해당 반복이 끝나면 내려와서 수식 h--로 들어가 h=h-1을 실행하여 h는 19가 됩니다.
d는 360에서 19만큼 나누게 됩니다.
다시 For i반복문으로 들어가 j 반복문을 실행합니다.
h가 줄어들었으니 둘러 쌓을 박스가 19개로 줄어들었고 회전의 폭도 줄어들었습니다.
이런 방법으로 첨성대의 점점 얇아지는 방법을 사용하였습니다.

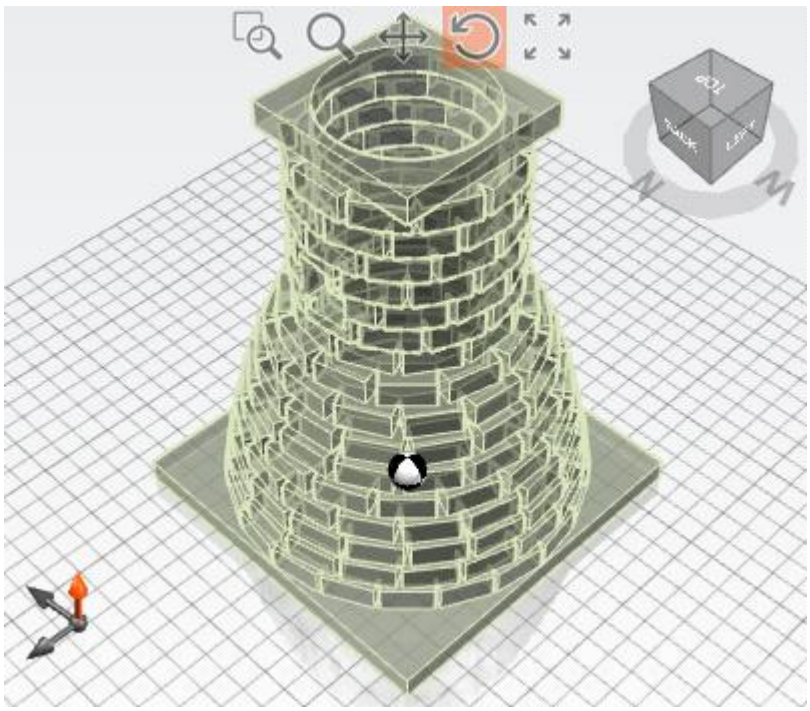


위 식까지 적용하게 되면 첨성대의 상태를 확인 할 수 있습니다.



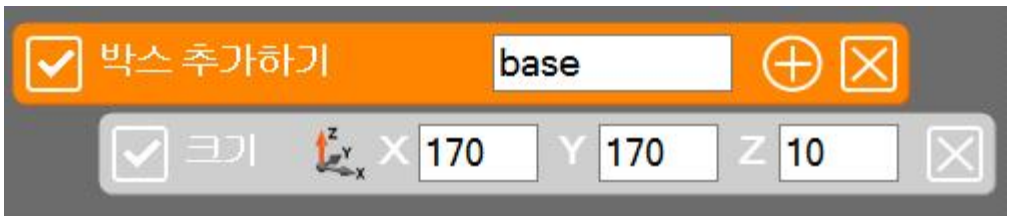
박스를 하나 만들어줘서 도형 빼기를 실행해줍니다.
이 박스는 첨성대 중간에 뚫려 있는 관측문 입니다.

마지막으로 첨성대 천장을 추가해줍니다.
그 박스를 실린더로 동그랗게 뚫고 마지막에 지금까지 작성했던 첨성대에 더하기를 해줍니다.



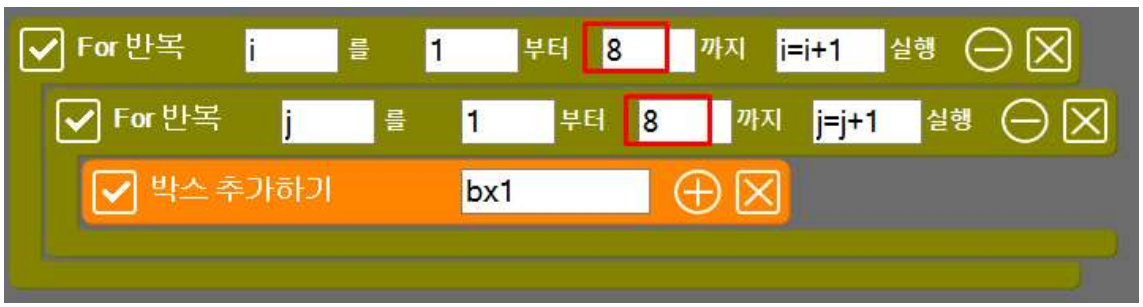
14.3 체스 세트 만들기

1) 체스판 만들기



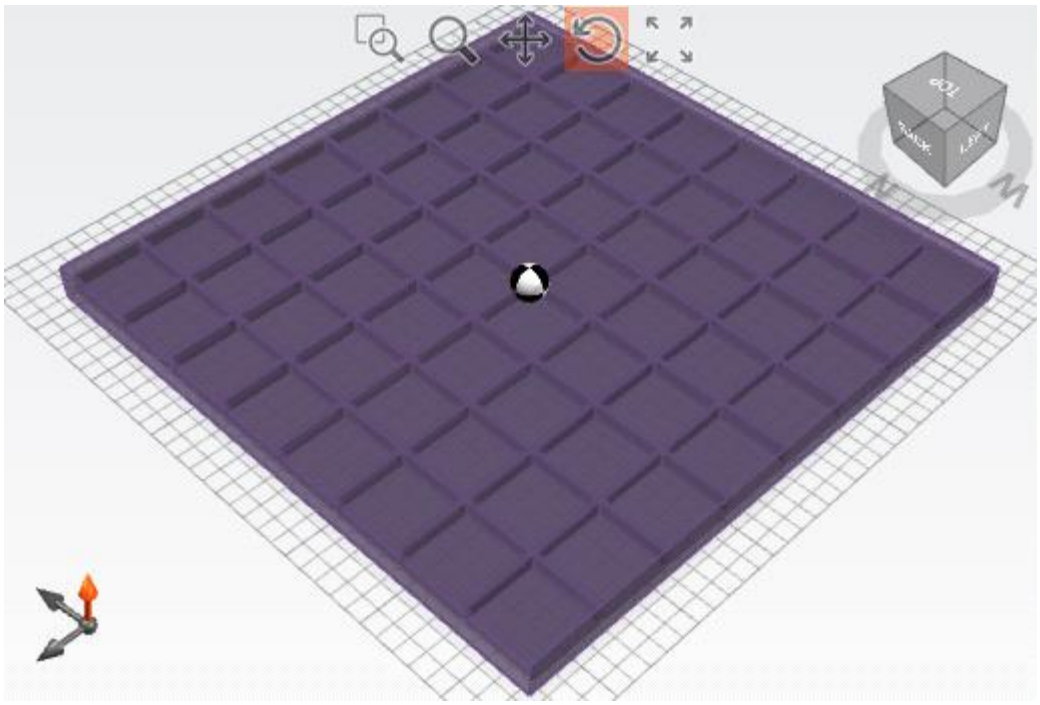
체스판을 만들기 위해 충분한 크기의 박스를 만들어 줍니다.

2중 For문을 만들어 줍니다.



체스판은 가로 세로 8칸씩 있으니 최대값의 수를 변경해줍니다.
그리고 반복문 안에 박스 하나를 추가해 줍니다.

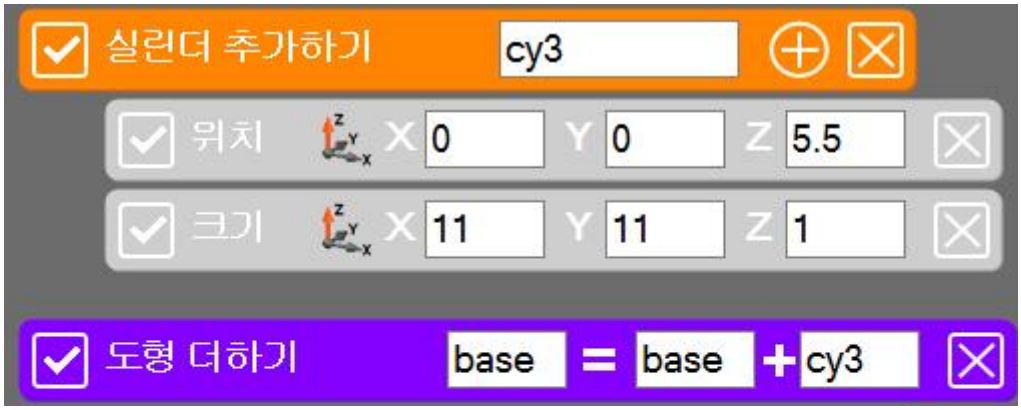
박스의 이름을 반복문에 맞춰 수정해줍니다
위치와 크기를 만들어 위치 X : {i*21-95}, Y : {j*21-95}로 지정하고 박스의 크기는 X : 20, Y : 20, Z : 5로 지정합니다.
마지막으로 도형 빼기를 해 반복문 전체의 박스와(bx{i}{j}) 큰 판(base)를 빼줍니다.



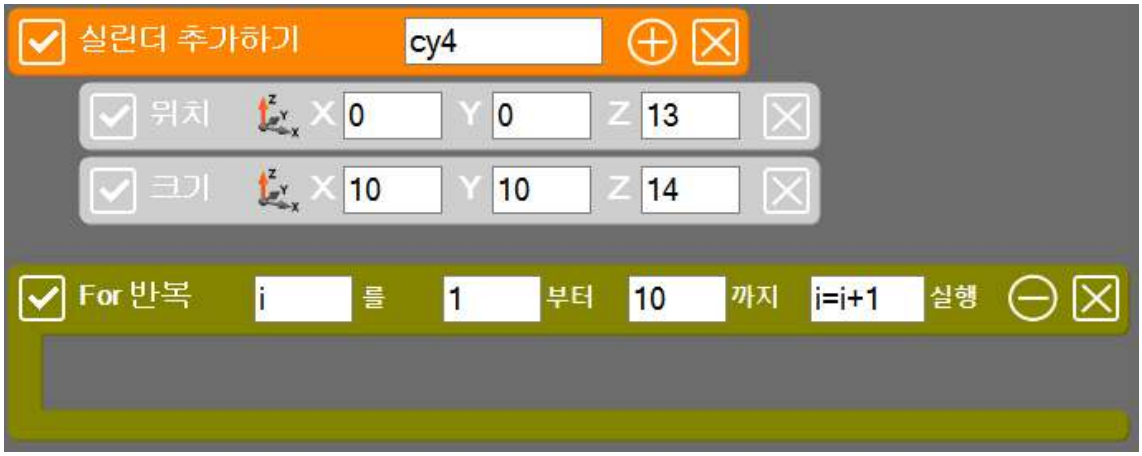
체스판이 완성되었습니다.
이제 이 위에 올라갈 체스말들을 만들고자 합니다.

2) 체스 폰 만들기

우선 실린더를 추가해줍니다.
체스말의 하단 부분을 만들게 될 실린더들입니다.

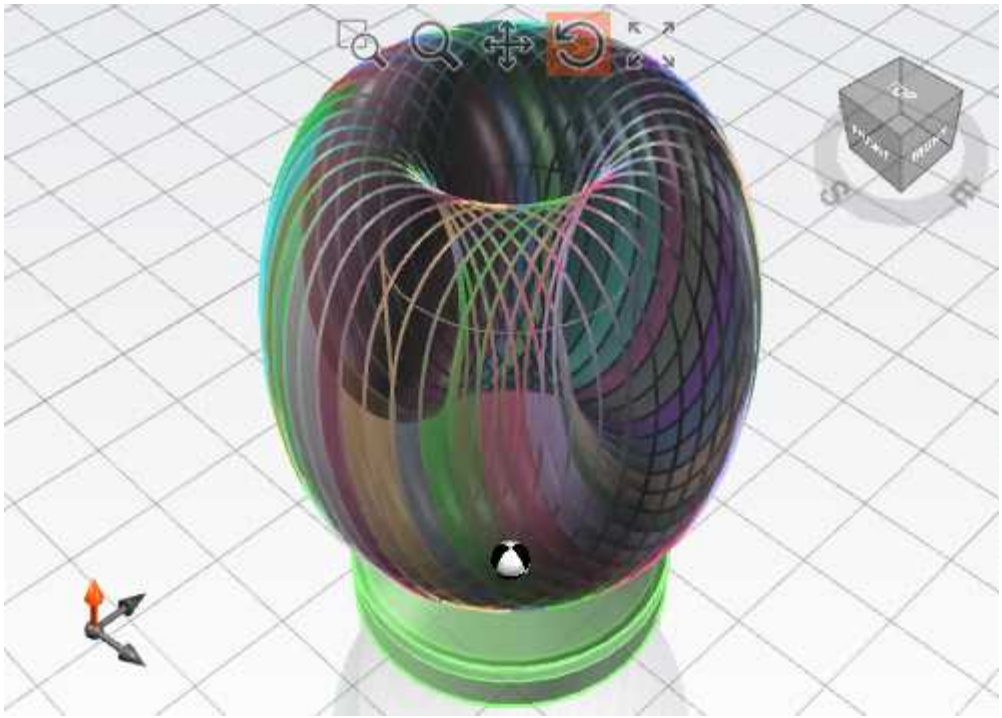


여기까지 완성하면 위 그림과 같은 밑모양이 완성됩니다.
이제 이 위로 하여 폰을 만들고자 합니다.



실린더를 하나 더 만들어 줍니다.
이 실린더는 폰의 허리 부분입니다.
그리고 For 반복문을 만들어 줍니다.

반복문을 수정해줍니다.
i 반복문의 최대값을 36까지 수정해주고
추가한 구를 반복문에 사용할 수 있게 이름뒤에 {i}를 추가해줍니다.
위치, 크기, 방향을 수정해줍니다.
그리고 3D 도형 회전하기를 사용해서 Z에 {i*10}을 입력해줍니다.

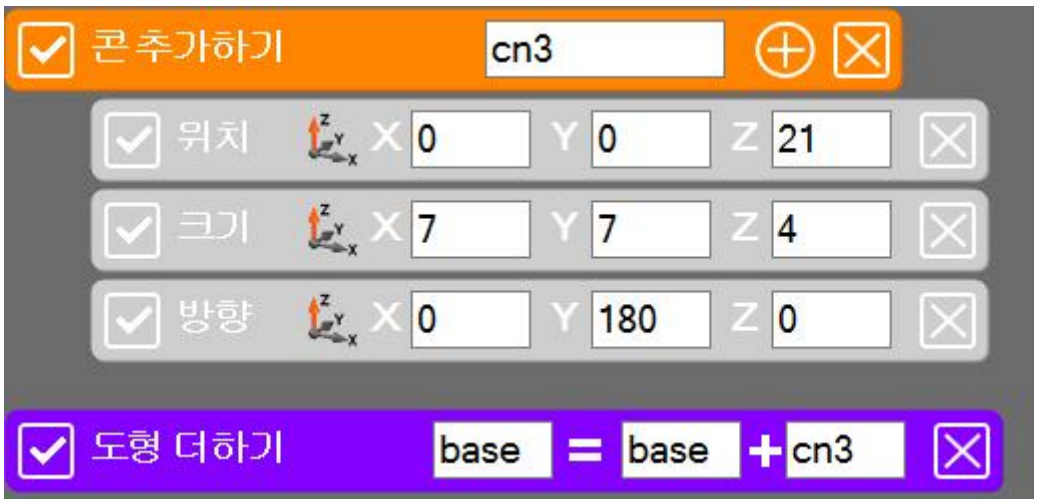


작업이 완료되면 위 그림과 같은 모양을 하게됩니다.
이제 이 무수히 많은 구를 실린더에 빼서 잘록한 허리를 만들게 될것입니다.

For안에 도형 빼기를 추가해줍니다.
주의 하실 것은 뺄 대상이 base가 아닌 For반목문 위에 만들었던 실린더(cy4)와 빼셔야 합니다.

빼기가 완료되면 위 그림과 같은 잘록하게 실린더를 깍았습니다.

이 잘록한 실린더와 base를 도형 더하기로 더해줍니다.
도형 더하기는 For문 밖에서 선언해줍니다.



나머지 폰의 머리를 꾸밀 작업들만 남았습니다.
위 그림과 같은 명령어를 사용하여 나머지 작업을 해줍니다.

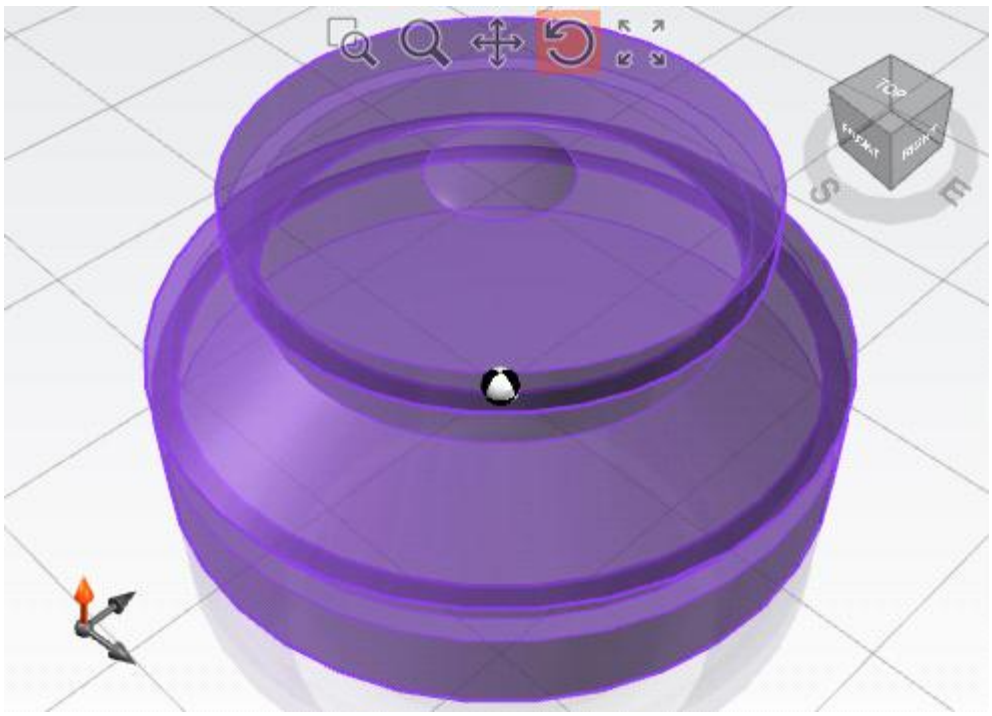
위 그림과 같이 끝내시면 작업이 완료입니다.

3) 체스 나이트 만들기

이번엔 나이트를 만들고자 합니다.
나이트는 다른 작업들과는 달리 더하기 빼기만 이용하게됩니다.

폰 때와 마찬가지로 하단부를 만들어 주는 작업입니다.

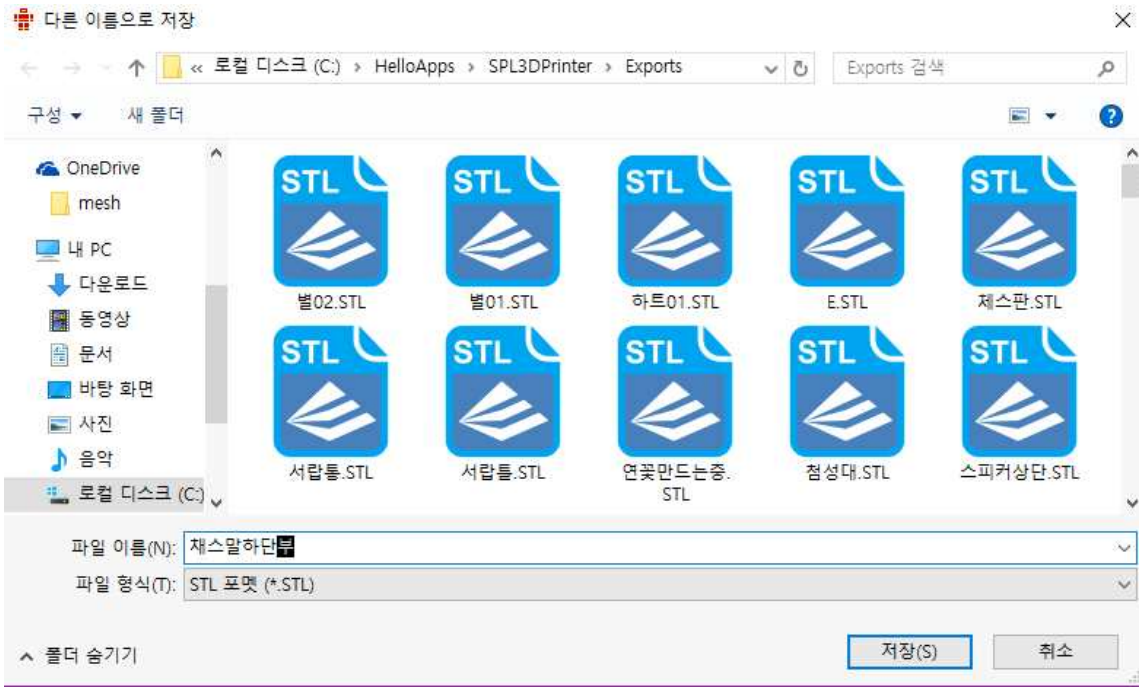
위 그림까지가 하단부 작업입니다.



완료되면 결과창에 위 그림과 같은 모양을 하게 됩니다

앞으로 다른 체스 말들 또한 이 작업이 반복될 예정입니다.
작업을 줄이기 위한 방법으로 **모델링 파일로 내보내기**를 사용하도록 합시다.

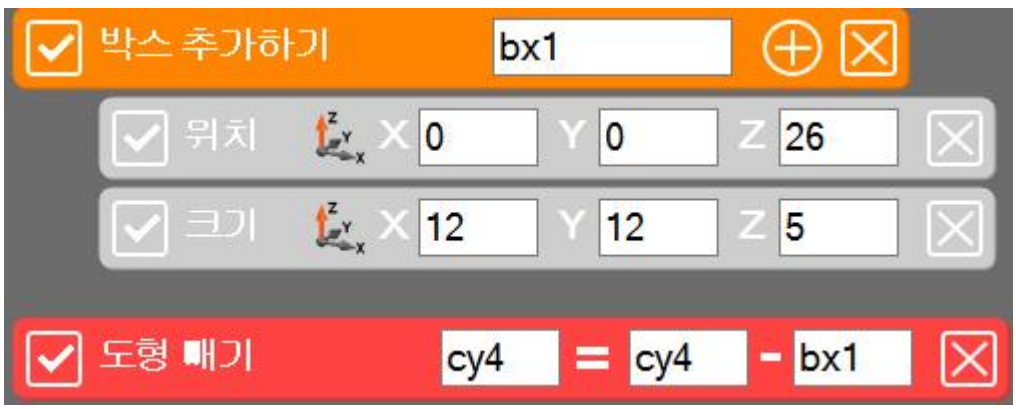
하단부 작업한 내용을 실행한 뒤 모델링 파일로 내보내기를 해줍니다.

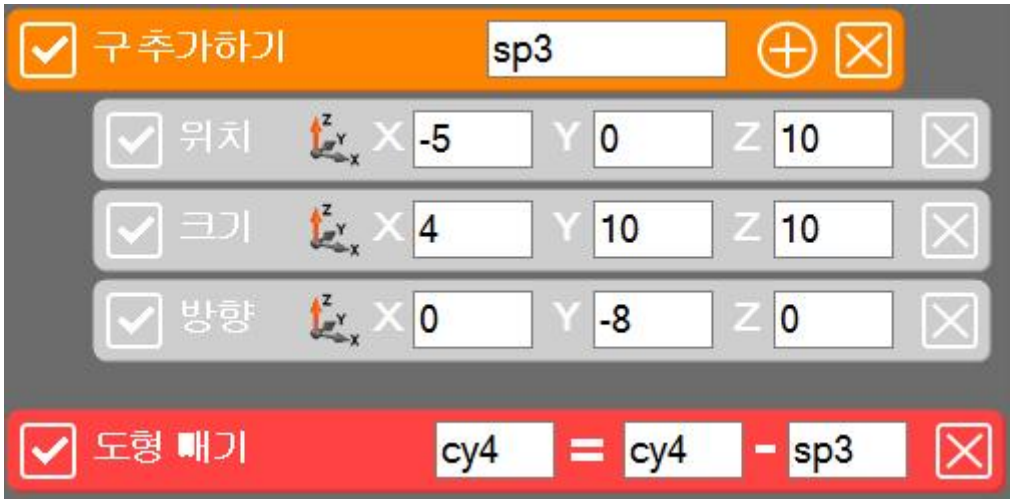


임의의 이름으로 지정해줍니다.

다시 나이트 작업으로 돌아서 실린더를 만들어 작업을 시작합니다.

구 추가하기
sp1
위치 X 2 Y 7 Z 10
크기 X 10 Y 14 Z 16
방향 X 10 Y 0 Z -120
도형 빼기
cy4 = cy4 - sp1
구 추가하기
sp2
위치 X 2 Y -7 Z 10
크기 X 10 Y 14 Z 16
방향 X -10 Y 0 Z 120
도형 빼기
cy4 = cy4 - sp2


박스 추가하기
bx1
위치 X 0 Y 0 Z 26
크기 X 12 Y 12 Z 5
도형 빼기
cy4 = cy4 - bx1

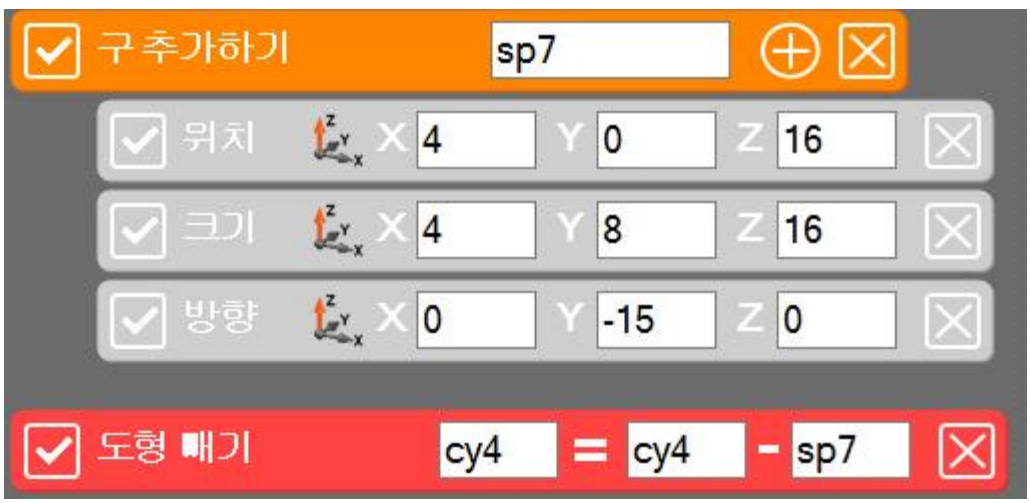

구 추가하기
sp3
위치
X -5
Y 0
Z 10
크기
X 4
Y 10
Z 10
방향
X 0
Y -8
Z 0
도형 빼기
cy4 = cy4 - sp3

박스 추가하기
bx6
위치
X -4
Y 0
Z 12.5
크기
X 4
Y 3
Z 10
방향
X 0
Y -10
Z 0
박스 추가하기
bx7
위치
X -4
Y 0
Z 7
크기
X 3
Y 3
Z 3
방향
X 0
Y 25
Z 0
도형 더하기
bx6 = bx6 + bx7

구 추가하기
sp5
위치
X -7
Y 1.5
Z 10
크기
X 8
Y 2
Z 16
구 추가하기
sp6
위치
X -7
Y -1.5
Z 10
크기
X 8
Y 2
Z 16
도형 빼기
bx6 = bx6 - sp5
도형 빼기
bx6 = bx6 - sp6
도형 더하기
cy4 = cy4 + bx6

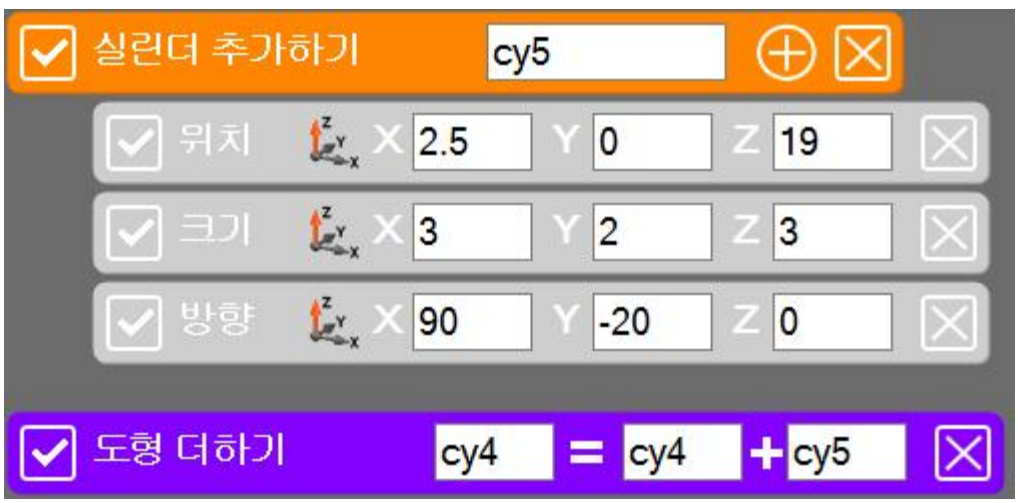
박스 추가하기
bx2
위치
X 0
Y 5
Z 16
크기
X 10
Y 5
Z 20
도형 빼기
cy4 = cy4 - bx2
박스 추가하기
bx3
위치
X 0
Y -5
Z 16
크기
X 10
Y 5
Z 20
도형 빼기
cy4 = cy4 - bx3

박스 추가하기
bx4
위치 X 4 Y 0 Z 22
크기 X 7 Y 7 Z 5
방향 X 0 Y 33 Z 0
도형 빼기 cy4 = cy4 - bx4
박스 추가하기
bx5
위치 X -5 Y 0 Z 22
크기 X 10 Y 10 Z 5
방향 X 0 Y -20 Z 0
도형 빼기 cy4 = cy4 - bx5


구 추가하기
sp7
위치 X 4 Y 0 Z 16
크기 X 4 Y 8 Z 16
방향 X 0 Y -15 Z 0
도형 빼기 cy4 = cy4 - sp7

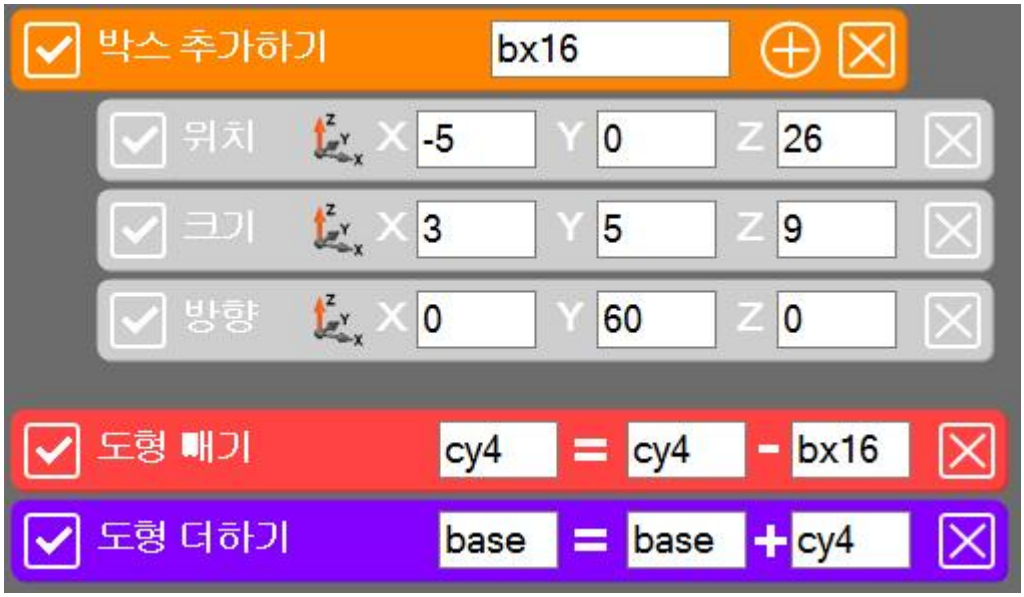
박스 추가하기
bx8
위치
X 0
Y 0
Z 22
크기
X 8
Y 5
Z 5
도형 더하기
cy4 = cy4 + bx8
실린더 추가하기
cy3
위치
X 1.5
Y 0
Z 24
크기
X 6
Y 3
Z 5
방향
X 90
Y 50
Z 0
도형 빼기
cy4 = cy4 - cy3
박스 추가하기
bx10
위치
X 4.5
Y 0
Z 25
크기
X 5
Y 5
Z 6.5
도형 빼기
cy4 = cy4 - bx10

박스 추가하기
bx11
위치 X 0 Y 2.5 Z 22
크기 X 10 Y 1 Z 5
방향 X 0 Y 0 Z -5
도형 빼기 cy4 = cy4 - bx11
박스 추가하기
bx12
위치 X 0 Y -2.5 Z 22
크기 X 10 Y 1 Z 5
방향 X 0 Y 0 Z 5
도형 빼기 cy4 = cy4 - bx12

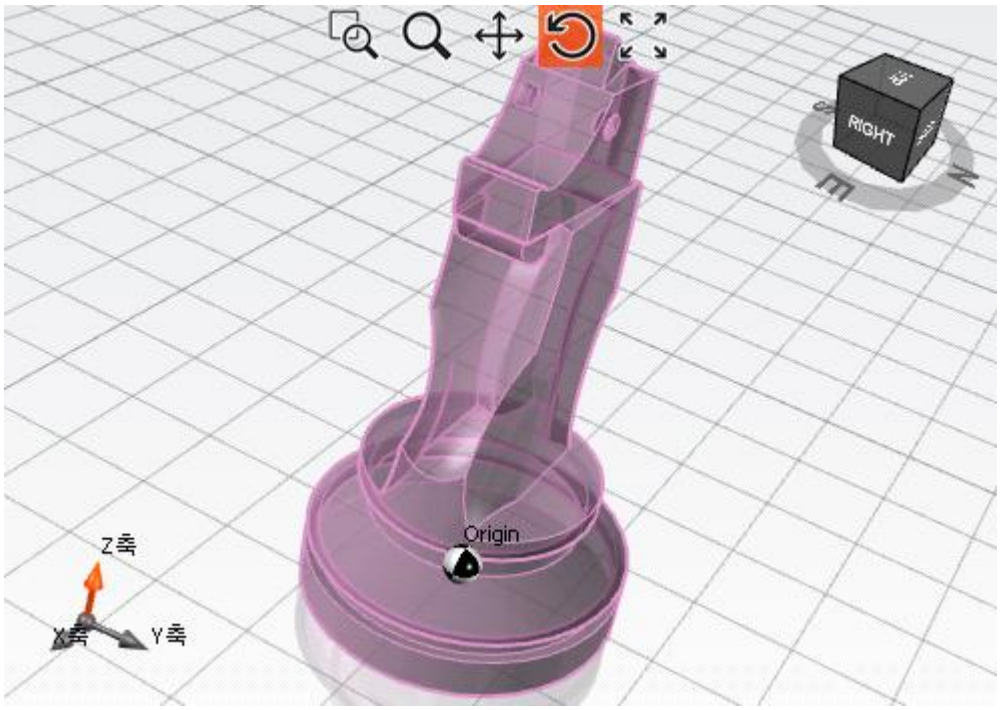

실린더 추가하기
cy5
위치 X 2.5 Y 0 Z 19
크기 X 3 Y 2 Z 3
방향 X 90 Y -20 Z 0
도형 더하기 cy4 = cy4 + cy5

실린더 추가하기
cy6
위치
X -1
Y 2
Z 23
크기
X 1
Y 1
Z 1
방향
X 90
Y 0
Z 0
도형 빼기
cy4 = cy4 - cy6
실린더 추가하기
cy6
위치
X -1
Y -2
Z 23
크기
X 1
Y 1
Z 1
방향
X 90
Y 0
Z 0
도형 빼기
cy4 = cy4 - cy6

박스 추가하기
bx13
위치
X -4
Y 0
Z 23
크기
X 6
Y 3
Z 7.5
방향
X 0
Y 70
Z 0
박스 추가하기
bx14
위치
X -4
Y 2
Z 23
크기
X 7
Y 2
Z 8
방향
X 0
Y 70
Z 9
도형 빼기
bx13 = bx13 - bx14
박스 추가하기
bx15
위치
X -4
Y -2
Z 23
크기
X 7
Y 2
Z 8
방향
X 0
Y 70
Z -9
도형 빼기
bx13 = bx13 - bx15
도형 더하기
cy4 = cy4 + bx13



위 작업까지가 체스 나이트 작업입니다.
완료 하고 실행하면 아래 그림과 같은 모습을 하게됩니다.

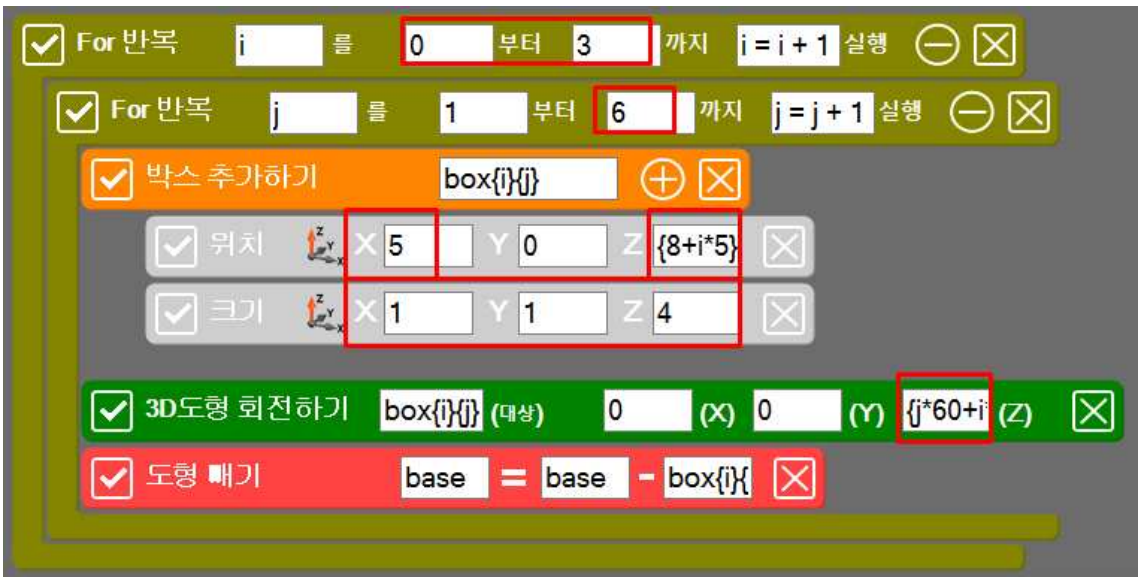


4) 체스 룩 만들기

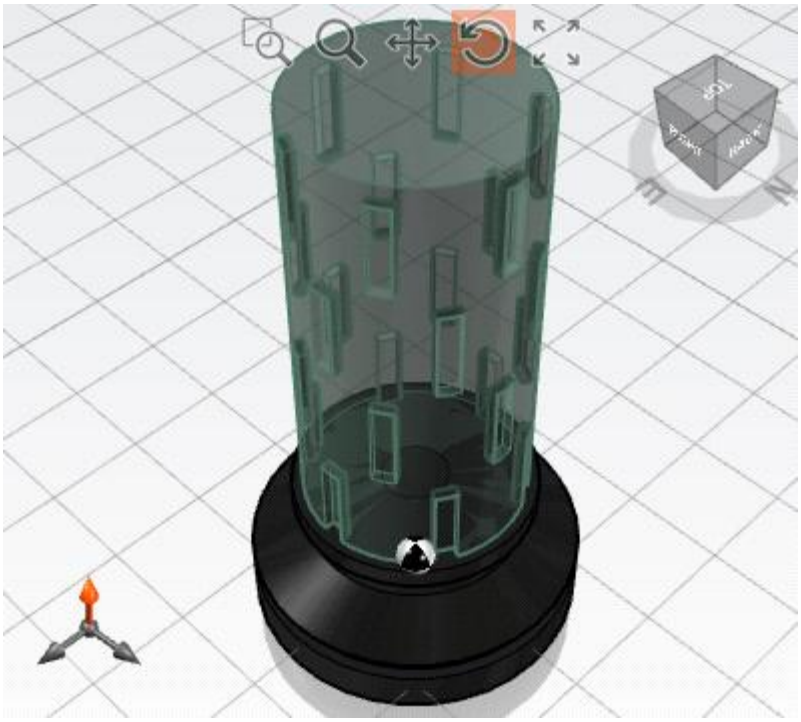
이번엔 룩을 만들고자 합니다.
아까 내보내기로 내보냈던 파일을 불러오도록 합니다.

도형 불러오기 명령어를 이용하여 체스 하단부를 가지고 오도록 합니다.
이제 이 위에서 체스 작업을 하게 됩니다.

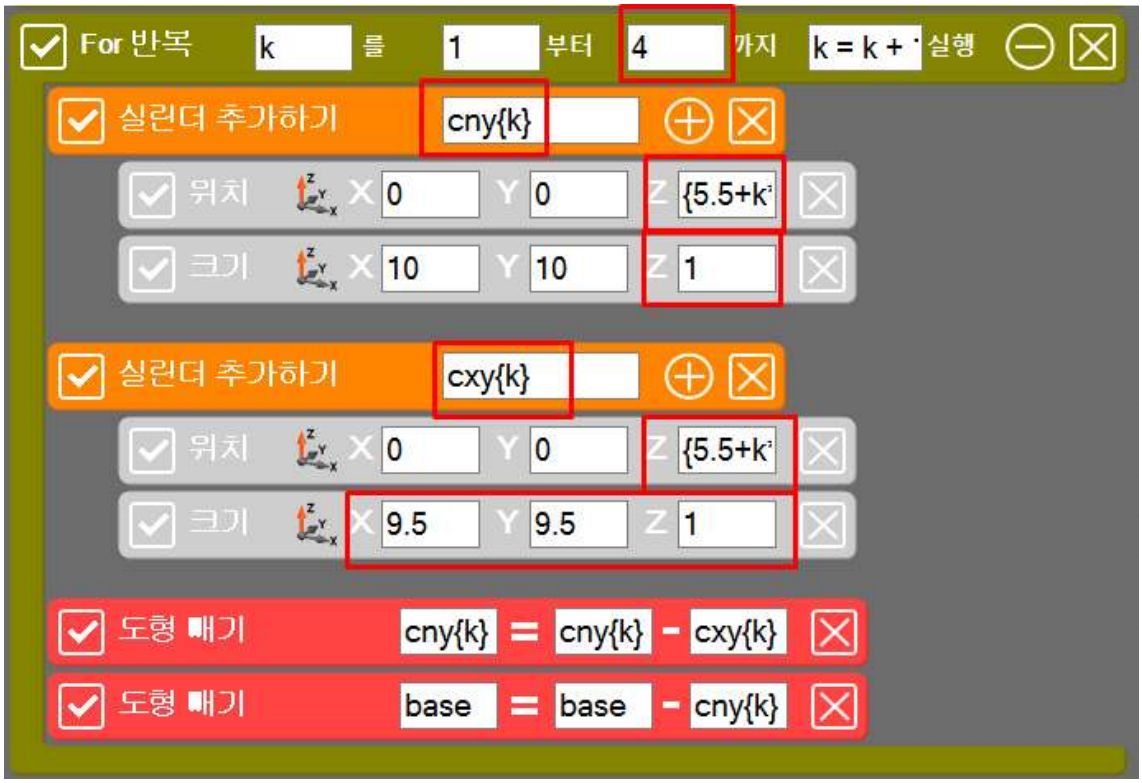
실린더를 하나 만들어 줍니다.
이 실린더가 룩의 허리부분이 될 것이며 이 실린더에 벽돌 무늬를 만들도록 하겠습니다.



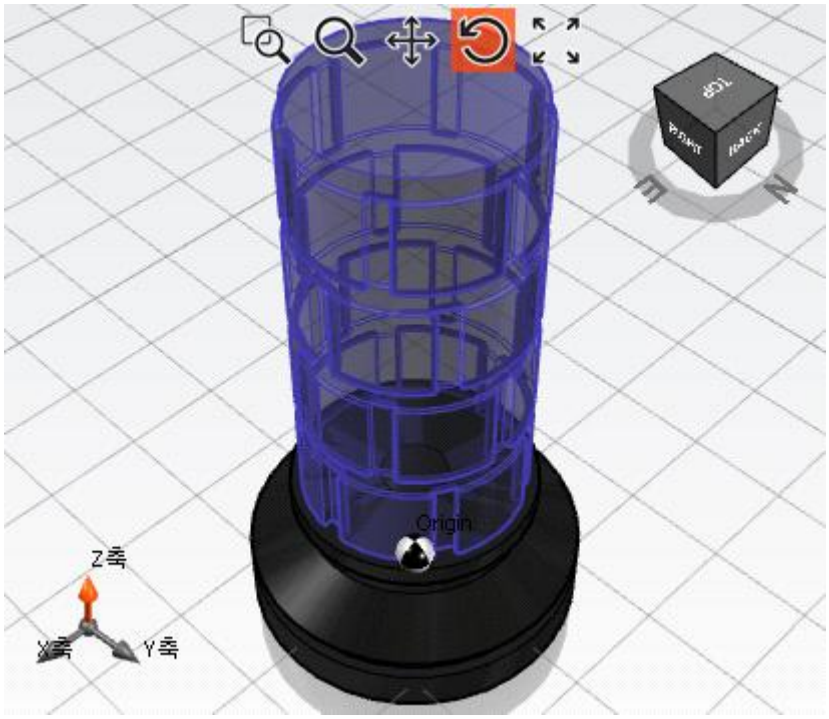
아래 2중 For문을 만들어 주도록 합니다.
For 반복문 i의 최소값은 0 최대값은 3으로 수정하고 j의 최대값은 6으로 수정합니다.
박스 하나를 만들어 이름을 반복문에 사용할 수 있게 하고 위치를 만들어 X : 5, Z : {8+i*5},
크기는 X : 1, Y : 1, Z : 4로 만들어 줍니다.
3D도형 회전하기 명령어를 사용하여 Z : {j*60+i*30}를 입력하고 도형 빼기를 해줍니다.



결과는 위 그림과 같이 나옵니다.
이제 무늬들 사이사이에 부족한 무늬를 만들어 줄 차례입니다.

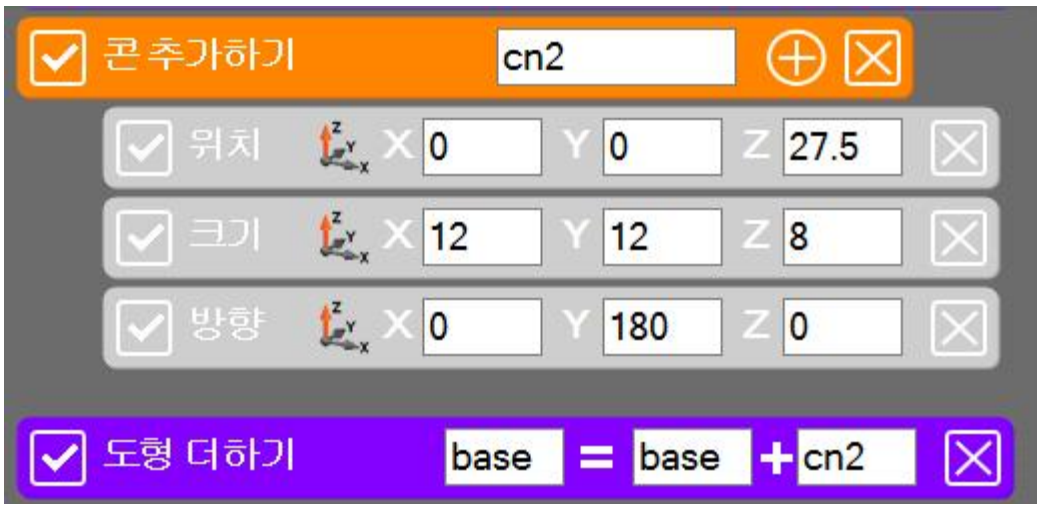


2중 For문 밖에 새로운 반복문을 만들어줍니다.
k의 최대값을 4로 수정합니다.
실린더를 두 개 만들어 줍니다.
실린더의 이름이 중복되기에 다른 이름으로 설정해줍니다.
첫 번째 실린더의 위치 Z : {5.5+k*5}로 설정하고 크기를 Z : 1로 수정합니다.
두 번째 실린더의 위치 Z : {5.5+k*5}로 동일하게 해주고 크기를 X : 9.5, Y : 9.5, Z : 1로 첫 번째 실린더보다 작게 만들어줍니다.
그 뒤 도형 빼기를 하여 첫 번째 실린더와 두 번째 실런더를 우선 빼줍니다.
첫 번째 빼기 작업을 해서 고리를 만들게 됩니다.
만들어진 고리를 허리부분(base)에 빼줍니다.

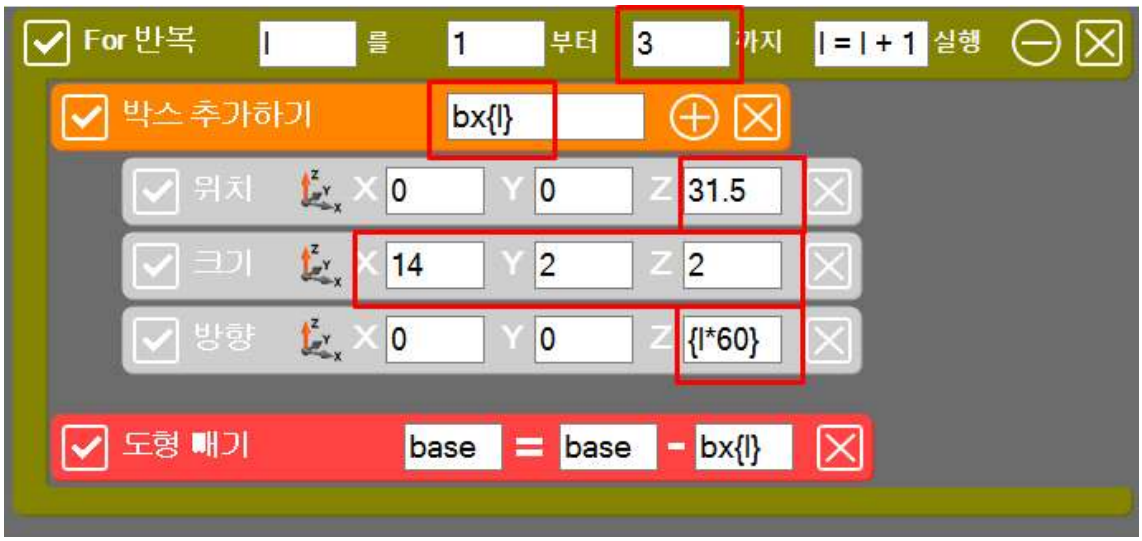


여기까지 작업하면 위 그림과 같은 모습을 하게 됩니다.

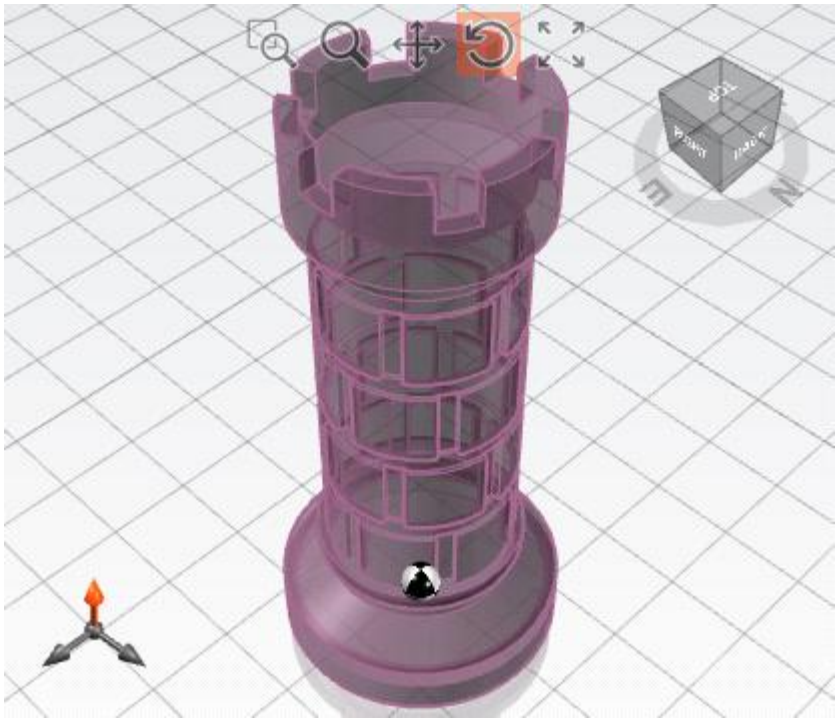
허리 부분(base)과 맨 처음 도형 불러오기로 가지고온 하단부를(st1) 도형 더하기를 해줍니다.
이제 룩의 머리 부분을 만들어줄 차례입니다.



위 그림들과 같이 작업을 해줍니다.



마지막으로 For문을 하나 더 만듭니다.
이 작업은 룩의 머리의 성 꼭대기를 표현하기 위한 작업입니다.
l For문의 최대값은 3으로 수정해줍니다.
박스 하나를 만들어 이름을 변경해주고 위치를 만들어 Z : 31.5로 지정해줍니다.
크기는 X : 14, Y : 2, Z : 2로 변경하고 방향은 Z : {l*60}으로 해줍니다
마지막에 도형 빼기를 해줍니다.

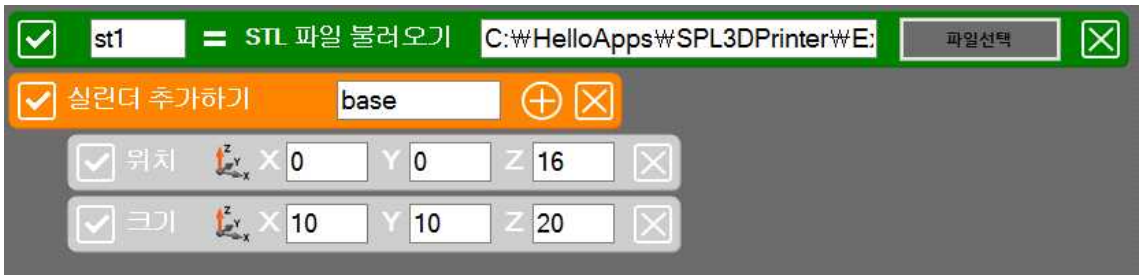


완료되면 위 그림과 같은 모습을 하게 됩니다.

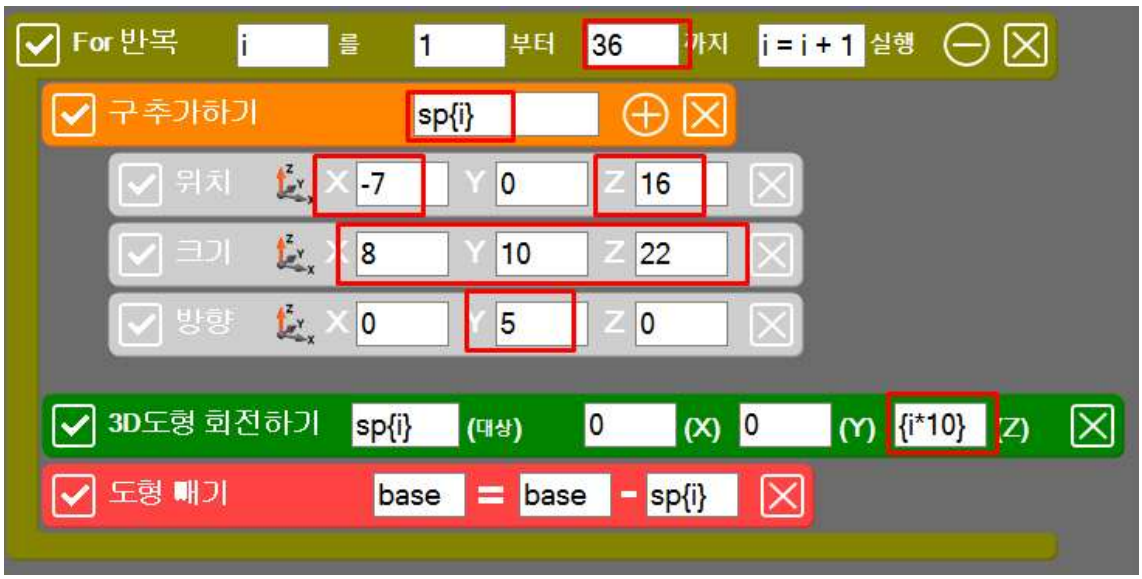
5) 체스 비숍 만들기

이번엔 비숍을 만들고자합니다.

룩때와 마찬가지로 하단부를 도형 불러오기를 해줍니다.



불러오기를 하신 뒤 몸통이 될 실린더를 만들어줍니다.

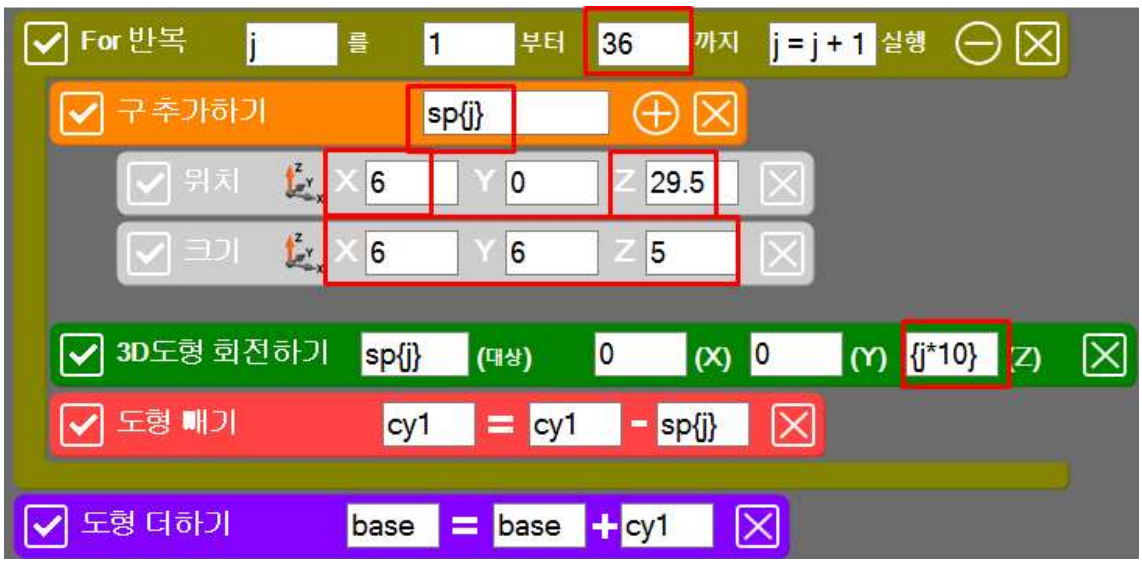


For 반복문 i의 최대값을 36까지 올립니다.
폰의 몸통을 만들 때와 같은 작업입니다.
구를 추가하여 주고 반복문에 사용 할 수 있는 이름으로 바꿔줍니다.
위치, 크기, 방향을 수정해줍니다.
3D도형 회전하기 명령어를 만들어 구를 대상으로 하여 Z : {i*10}로 지정해줍니다.
마지막으로 도형 빼기를 하여 base에서 전체 구를 빼줍니다.

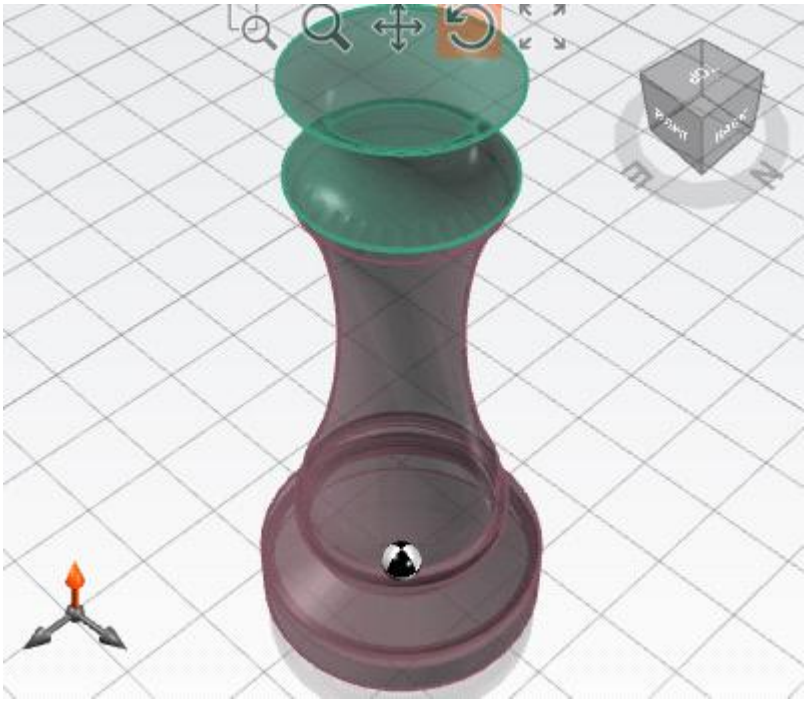
여기까지 하면 위와 같은 모양이 나오게 됩니다.

몸통 부분과 도형 불러오기를 한 하단부와 도형 더하기를 해줍니다.

뒤를 이은 작업을 해줍니다.



반복문을 다시 만들어 줍니다.
비숍의 머리쪽을 꾸며줄 반복문입니다.
For반복 j의 최대값을 36까지하고 구를 추가하여 위치와 크기를 주절해 줍니다.
3D도형 회전하기 명령어를 추가하여 Z : {j*10}을 입력해줍니다
위에서 만든 실린더(cy1)과 모든 구를 빼기를 해주고 For문에서 나와 도형 더하기를 해줍니다.

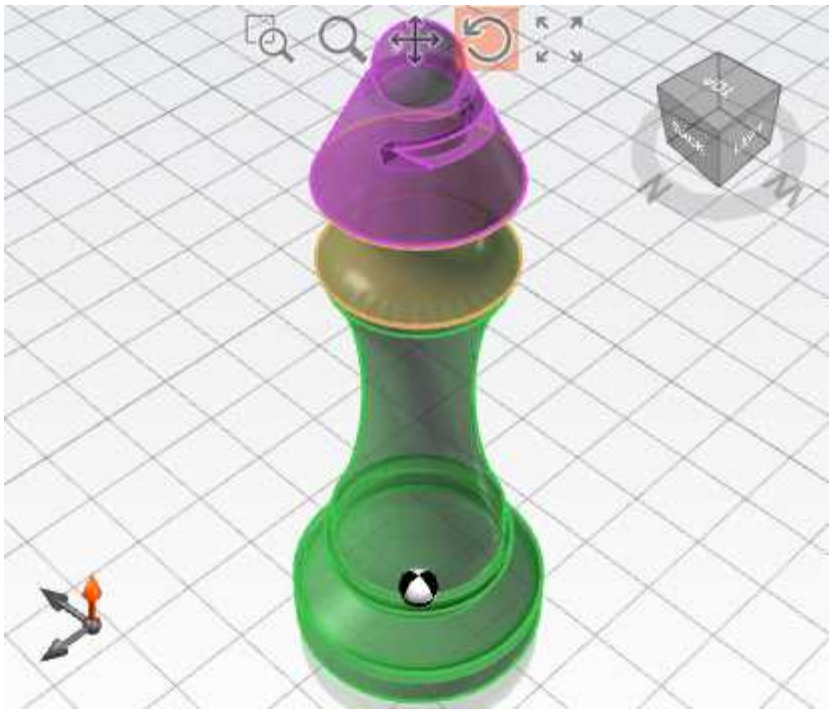


실행을 하면 다음과 같은 모습이 된 것을 확인 할 수 있습니다.

이제 나머지 머리 부분을 작업해줍니다.

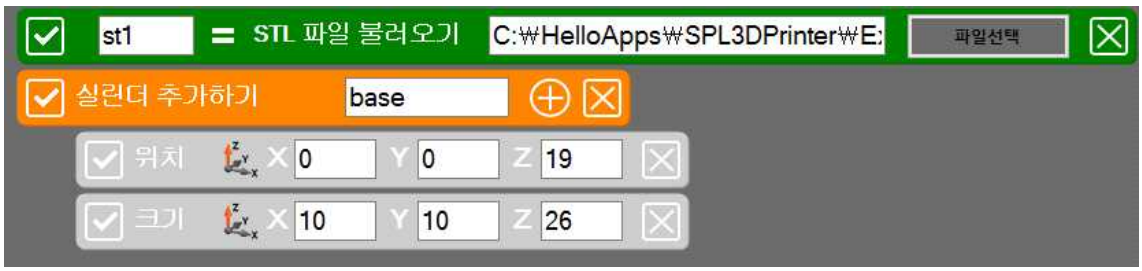
구 추가하기
sp1
위치 X 0 Y 0 Z 38
크기 X 4 Y 4 Z 4
실린더 추가하기
cy1
위치 X -0.5 Y 0 Z 35
크기 X 6 Y 6 Z 1
방향 X 0 Y 15 Z 0
실린더 추가하기
cy2
위치 X -0.5 Y 0 Z 35
크기 X 4 Y 4 Z 1
방향 X 0 Y 15 Z 0
도형 빼기
cy1 = cy1 - cy2

위 작업은 비숍의 나머지 머리 부분을 꾸미게 되는 부분들입니다.

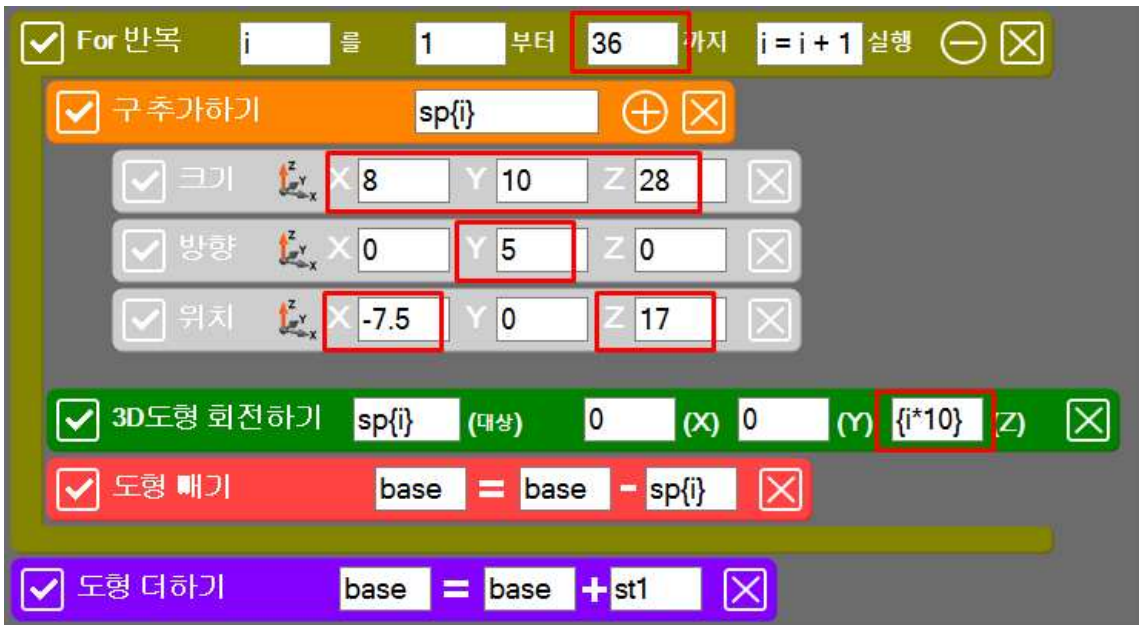


완료하면 위 그림과 같은 모습을 하게 됩니다.

6) 체스 킹 만들기



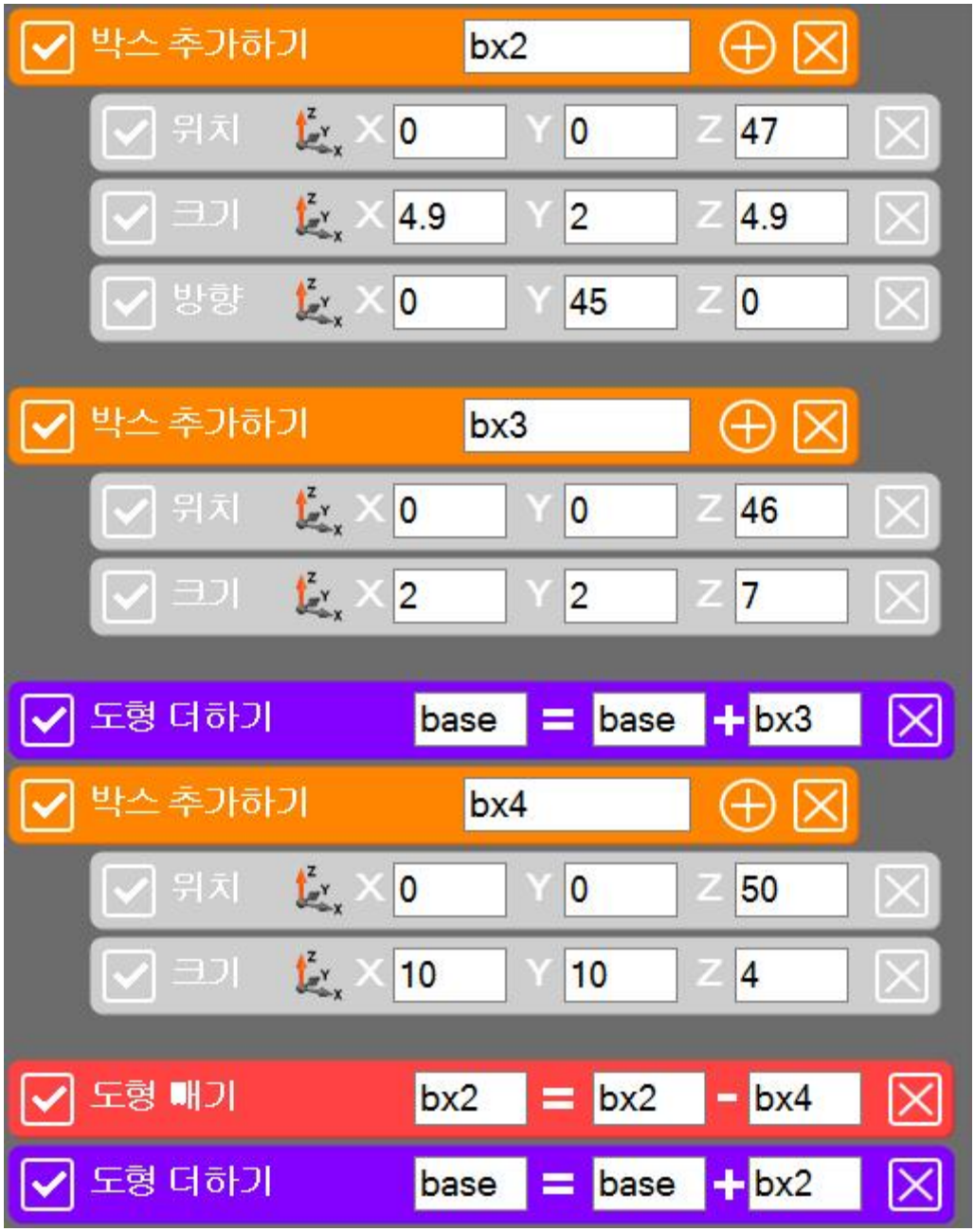
하단부를 도형 불러오기로 불러오고 킹의 몸통이 될 실린더를 만들어 줍니다.

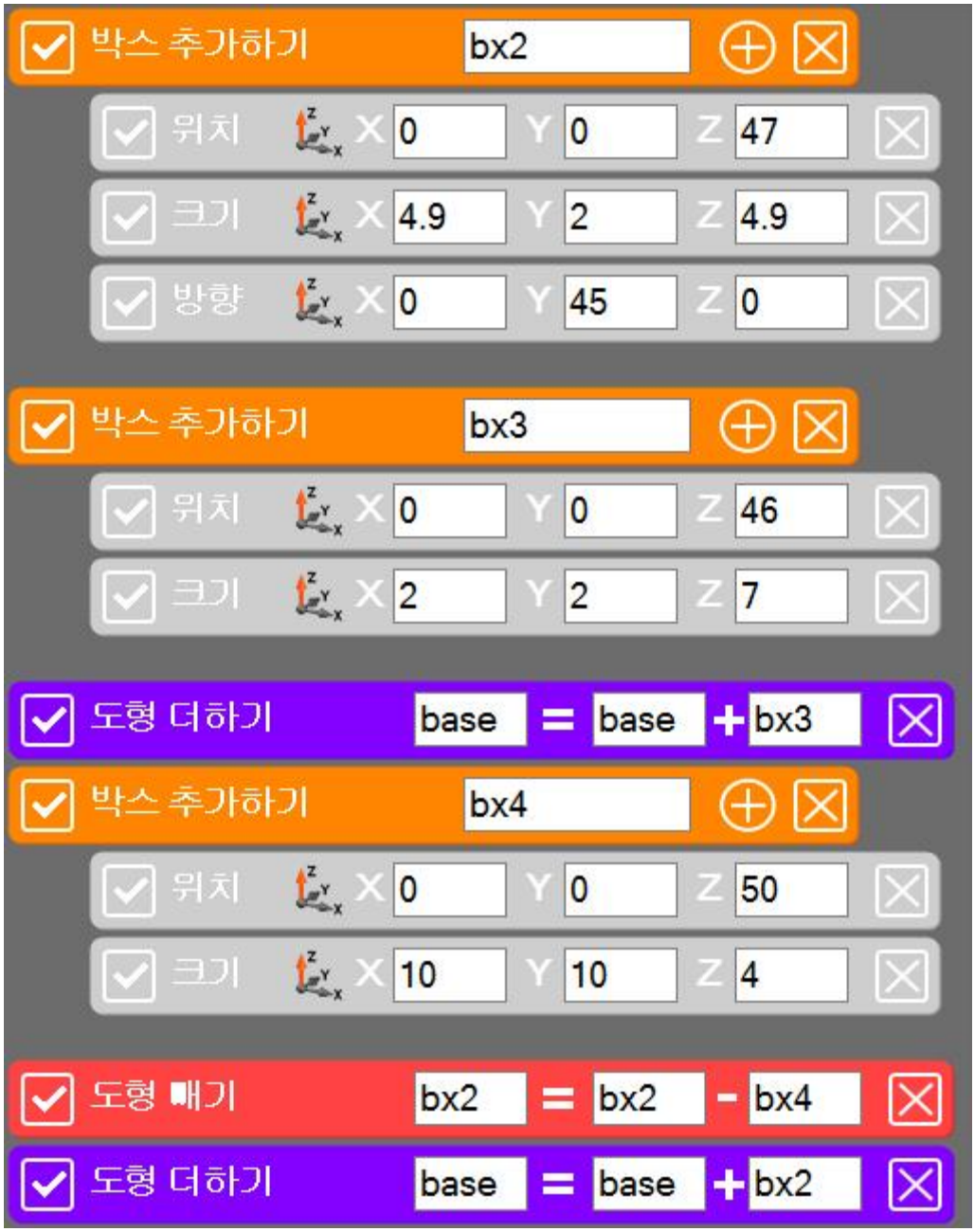


지금까지 해왔던 작업들과 마찬가지로 반복문 i의 최대값을 36으로 변경하고 구를 추가하여 구의 크기, 방향, 위치를 위 그림과 같이 바꾸어 줍니다.
3D도형 회전하기를 만들어 구를 대상으로 하고 Z : {i*10}으로 값을 입력해줍니다.
그리고 도형 빼기를 하여 지금까지의 구들(sp{i})을 몸통(base)에 빼줍니다.
마지막으로 For문 밖에서 몸통과 불러오기를 한 하단부와 도형 더하기를 해줍니다.
이제 체스 킹의 상단 머리를 만들어줄 차례입니다.

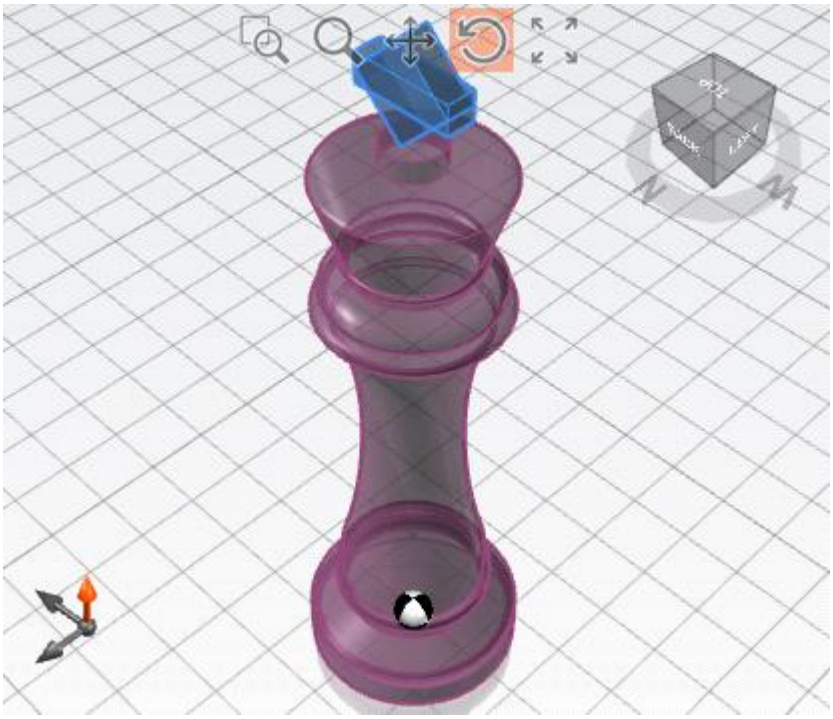
구 추가하기
sp1
위치 X 0 Y 0 Z 31.5
크기 X 12.5 Y 12.5 Z 3
도형 더하기 base = base + sp1
구 추가하기
sp2
위치 X 0 Y 0 Z 33
크기 X 11.5 Y 11.5 Z 4
도형 더하기 base = base + sp2
실린더 추가하기
cy1
위치 X 0 Y 0 Z 34.5
크기 X 8 Y 8 Z 2
도형 더하기 base = base + cy1
구 추가하기
sp3
위치 X 0 Y 0 Z 35.5
크기 X 9 Y 9 Z 2
도형 더하기 base = base + sp3

콘 추가하기
cn1
위치 X 0 Y 0 Z 41
크기 X 12 Y 12 Z 18
방향 X 0 Y 180 Z 0
도형 더하기 base = base + cn1
구 추가하기
sp1
위치 X 0 Y 0 Z 41
크기 X 12 Y 12 Z 2
도형 더하기 base = base + sp1
실린더 추가하기
cy2
위치 X 0 Y 0 Z 42.5
크기 X 4 Y 4 Z 2
도형 더하기 base = base + cy2
박스 추가하기
bx1
위치 X 0 Y 0 Z 47
크기 X 7 Y 2 Z 2
도형 더하기 base = base + bx1


박스 추가하기
bx2
위치 X 0 Y 0 Z 47
크기 X 4.9 Y 2 Z 4.9
방향 X 0 Y 45 Z 0
박스 추가하기
bx3
위치 X 0 Y 0 Z 46
크기 X 2 Y 2 Z 7
도형 더하기 base = base + bx3
박스 추가하기
bx4
위치 X 0 Y 0 Z 50
크기 X 10 Y 10 Z 4
도형 빼기 bx2 = bx2 - bx4
도형 더하기 base = base + bx2

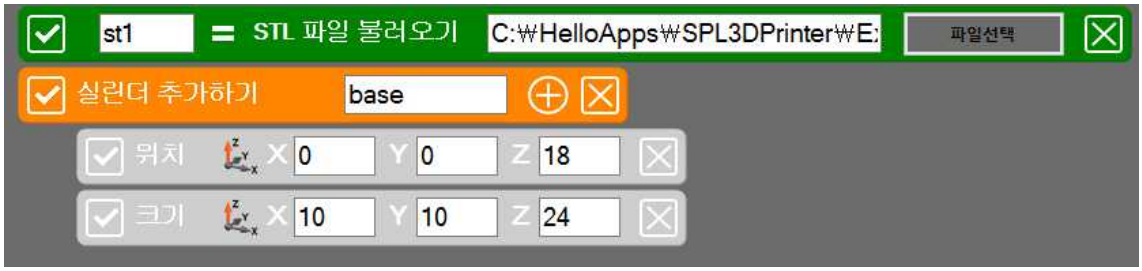


위 작업까지 하시면 킹의 상단을 만드는 작업이 끝나게 됩니다.

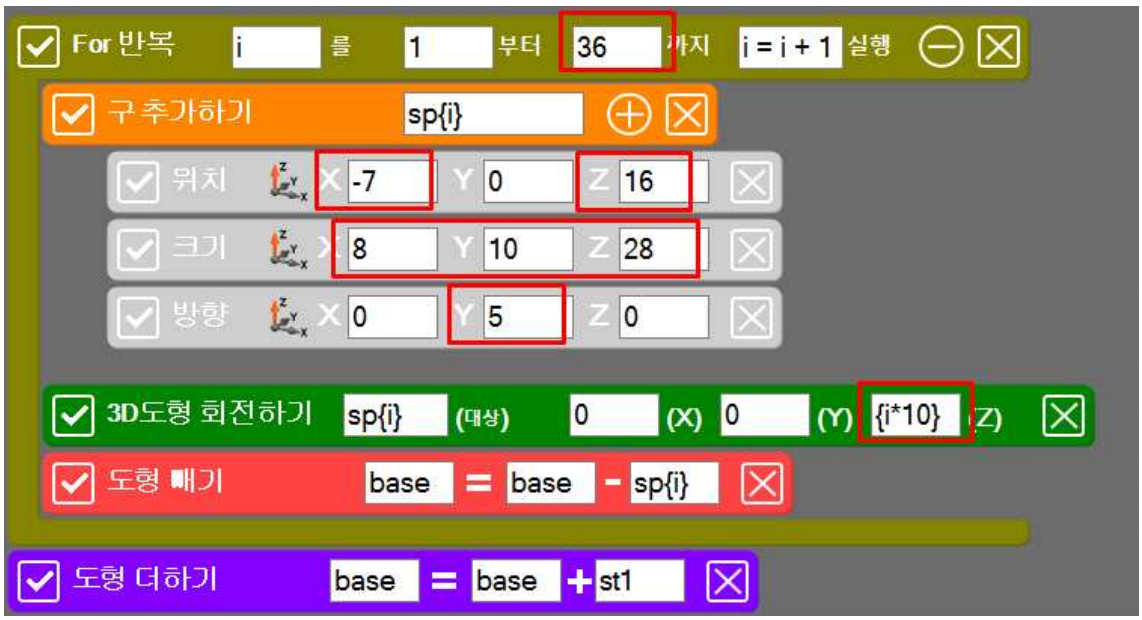


작업을 완료하면 위와 같은 모습을 하게됩니다.

7) 체스 퀸 만들기

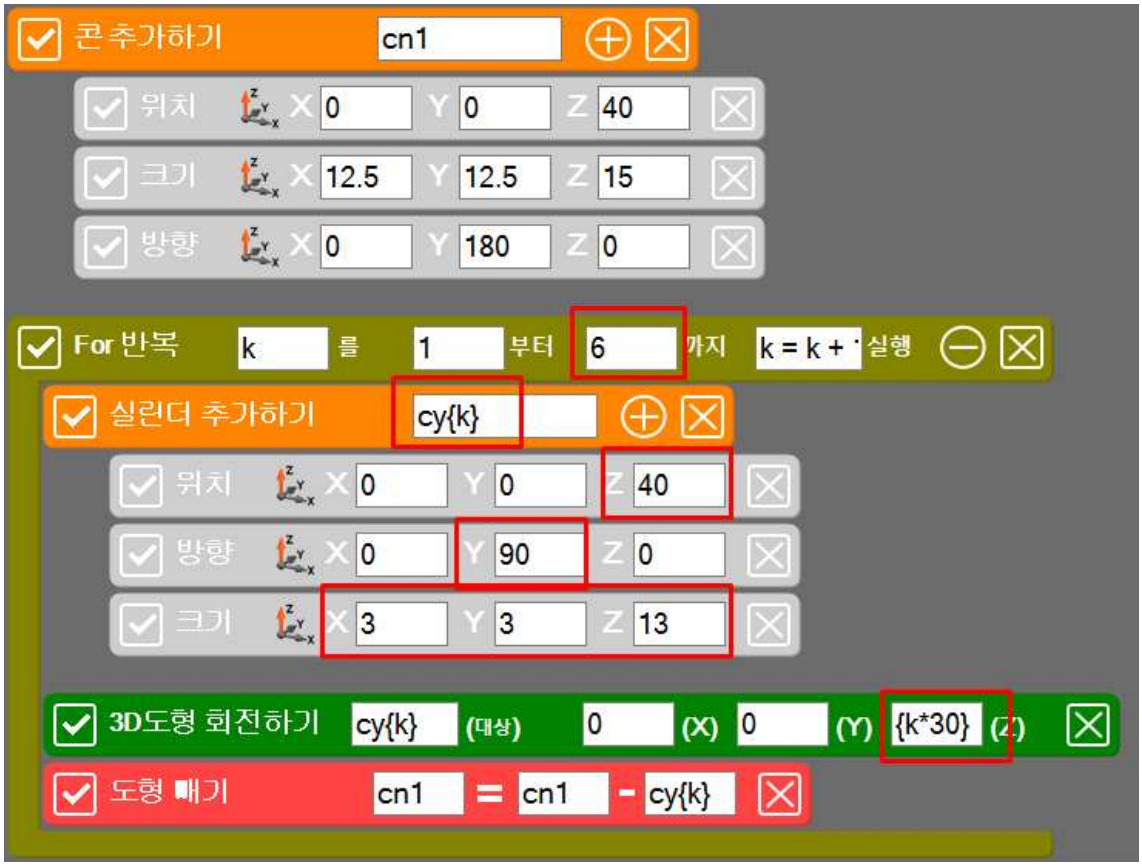


하단부를 도형 불러오기로 부르고 몸통이 될 실린더를 만들어 줍니다.

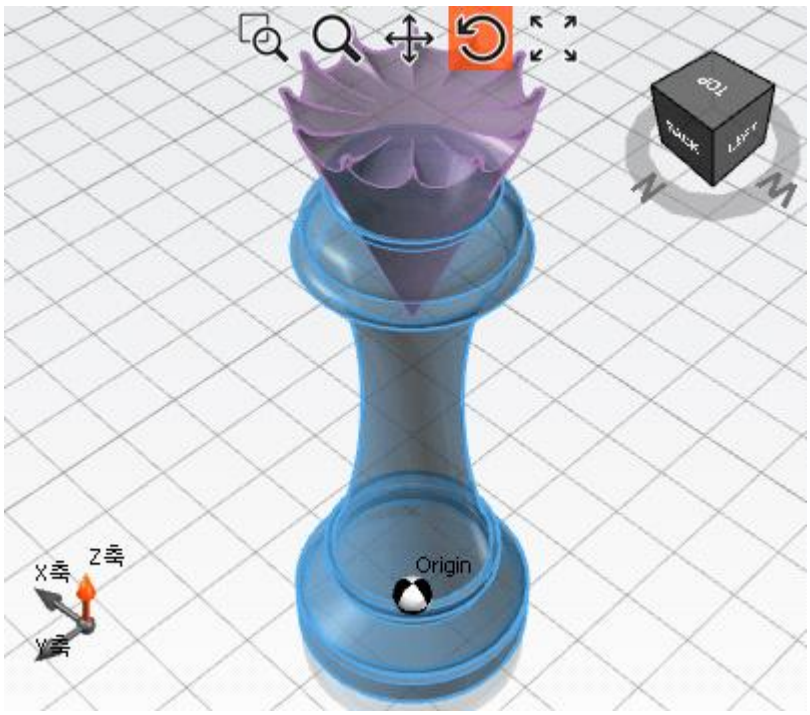


지금까지와 했던 작업을 마찬가지로 해줍니다.
For의 최대값을 36으로 바꾸고 구를 추가하여 위치, 크기, 방향을 바꾸어 3D도형 회전하기 명령어를 만들어 Z : {i*10}을 해주고 도형 빼기를 해줍니다.
For문을 닫고 도형 더하기로 몸통과 하단부를 더해줍니다.
퀸의 나머지 부분을 꾸밀 단계입니다.

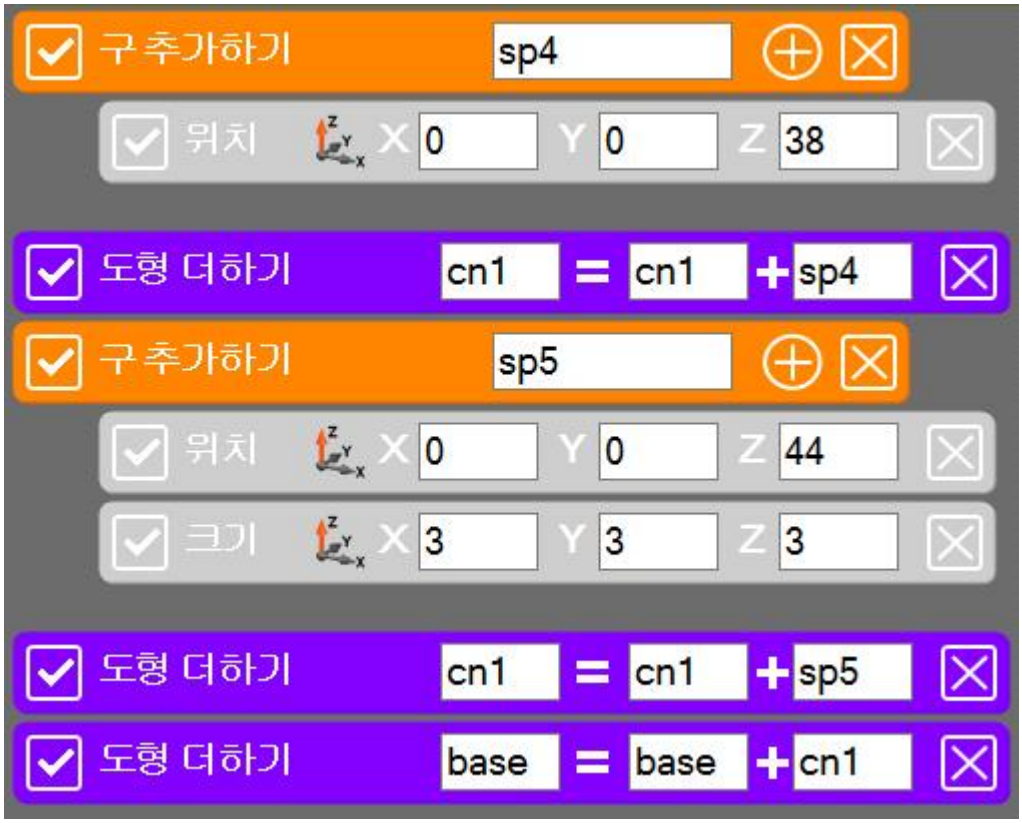
구 추가하기
sp1
위치
X 0
Y 0
Z 30.5
크기
X 12.5
Y 12.5
Z 3
도형 더하기
base = base + sp1
구 추가하기
sp2
위치
X 0
Y 0
Z 32
크기
X 11.5
Y 11.5
Z 4
도형 더하기
base = base + sp2
실린더 추가하기
cy1
위치
X 0
Y 0
Z 33.5
크기
X 8
Y 8
Z 2
도형 더하기
base = base + cy1
구 추가하기
sp3
위치
X 0
Y 0
Z 34.5
크기
X 9
Y 9
Z 2
도형 더하기
base = base + sp3



이 부분은 퀸의 크라운 부분입니다.
콘을 하나 만들어 뒤집어서 크라운으로 만들기 위해 For문을 만들어 작업을 해줍니다.
실린더를 이용해 뒤집은 콘의 윗부분들을 깎아 냅니다.



이제 나머지 부분을 채워서 완성을 해주면 됩니다.



여기까지 작업하면 체스 퀸도 완료가 됩니다.