

01 터틀크래프트 시작

- ◎ **학습목표** • 터틀크래프트에 접속하고, 간단한 조작법을 알 수 있다.
- 터틀크래프트에서 무엇을 하는지 알 수 있다.

■ 터틀크래프트 시작

생각열기

코딩(Coding)은 ‘Code’와 ‘ing’가 결합하여 만들어진 단어이다. 여기서 Code는 컴퓨터의 언어를 뜻한다. 즉, 코딩은 컴퓨팅 기기와 대화를 하는 것이다.



영어를 배우는 가장 좋은 방법은 영어를 쓰는 사람들과 대화하는 것이라고 한다. 즉, 코딩을 배우는 가장 좋은 방법도 컴퓨터와 대화를 시도해 보는 것이다. 간단한 코드로 그들과 대화를 시작해 보자.

1. 터틀크래프트 접속하기

우리는 ‘터틀크래프트’ 사이트에서 코딩을 시작할 것이다. 우선 다음 사이트 주소를 인터넷 브라우저 주소창에 입력하자.

<http://snucode.org/>

아래와 같은 화면이 나오면 빨간 테두리의 “스누콤중등 코딩나라”를 클릭하자.

알아봅시다!

■ 터틀말 버튼

‘터틀말’ 버튼을 누르면 ‘코딩수학.zip’을 다운로드 할 수 있다.

다운로드한 프로그램을 통해 ‘터틀말’을 사용할 수 있으나, 본 프로그램에서 다루지 않는다.

 창의코딩



시흥초등 창의코딩

스누콤중등 코딩나라

SNU시흥 코딩수학



코딩수학

알아봅시다!

■ 입력, 처리, 출력

프로그램의 작동은 크게 입력, 처리, 출력으로 나눌 수 있다. 사람이 컴퓨터에게 데이터를 주는 것이 **입력**, 주어진 데이터를 목적에 맞게 다루는 것을 **처리**, 처리된 결과를 다시 사람에게 돌려주는 것을 **출력**이라고 한다.

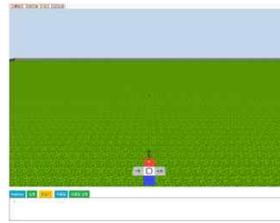
스누콤중등 코딩나라 실습

공지사항
공지사항이 없습니다.

사진



Making, Playing & Sharing



유튜브

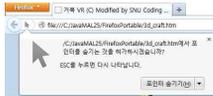


2020년 6월

일	월	화	수	목	금	토
31일	1일	2일	3일	4일	5일	6일
7일	8일	9일	10일	11일	12일	13일
14일	15일	16일	17일	18일	19일	20일
21일	22일	23일	24일	25일	26일	27일
28일	29일	30일	1일	2일	3일	4일
5일	6일	7일	8일	9일	10일	11일



TIP!



[코드 출력창]에 '마우스 포인터 숨기기'를 '항상 숨김'으로 바꾸면 편리하다.

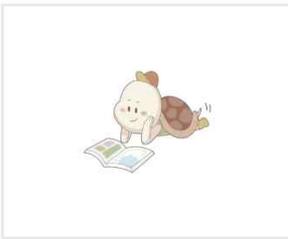
스누콤중등 코딩나라에서 앞으로 과제를 제출하거나 작품을 발표할 때 사용될 아이디어가 필요하니 회원가입을 하고 로그인을 하자. 로그인을 하고 나면 아래와 같은 메뉴창이 나오고 우리는 주로 ①과 ②를 사용하게 될 것이다. (나머지에 대해서는 영상을 통해 설명하겠다.)

스누콤중등 코딩나라 공지사항 코딩나라 교실 **과제 게시판** 실습 사진

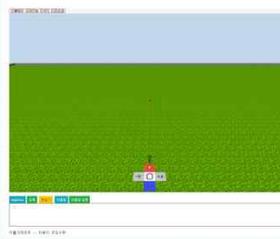
공지사항
공지사항이 없습니다.

②

사진



Making, Playing & Sharing



유튜브



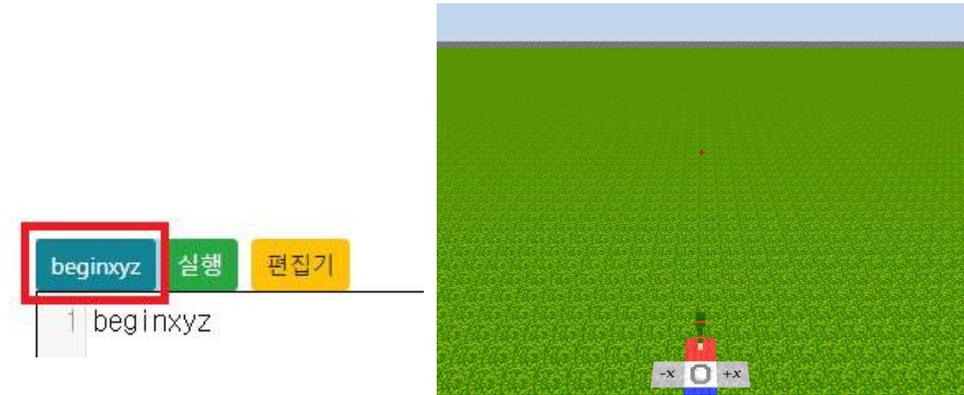
2020년 6월

일	월	화	수	목	금	토
31일	1일	2일	3일	4일	5일	6일
7일	8일	9일	10일	11일	12일	13일
14일	15일	16일	17일	18일	19일	20일
21일	22일	23일	24일	25일	26일	27일
28일	29일	30일	1일	2일	3일	4일
5일	6일	7일	8일	9일	10일	11일

여러분은 주로 과제게시판을 이용할 것이기에 ②번을 눌러 과제게시판으로 가서 본격적으로 터틀크래프트의 세계를 탐험해보자.

코딩으로 만나는 창의나라

과제계시판을 누르면 하얀 화면과 함께 왼쪽 아래에 다음과 같이 몇가지 버튼들이 등장한다. 입력창에 beginxyz라 치고 아래의 빨간 테두리의 beginxyz버튼을 누르면 오른쪽 그림과 같이 나타난다.



출력 창을 클릭하면 출력창 화면에 들어간다. ‘출력 창’에 들어가면 그 공간이 게임 화면이 된다. 이는 마치 Mojang의 마인크래프트(Minecraft)와 유사한 형식이다. 게임을 하듯이 화면 안을 이동하며 조작키를 발견해 보자.

알아봅시다!

■ 응답 없는 페이지

응답 없는 페이지
페이지가 응답할 때까지 기다리거나 페이지를 종료할 수 있습니다.

Tutorials Point (C) (모든 권리 보유)

인터넷 브라우저는 스크립트를 빠르게 처리하지 못할 경우 위와 같은 경고를 알려준다.

‘터틀크래프트’의 코드가 길어서 위와 같은 코드가 나타나면, ‘대기’를 클릭하고 기다리면 된다.

w, a, s, d	이동	마우스 좌클릭	
마우스 이동	시점 변환	ctrl+좌클릭	
z, c		alt+좌클릭	
shift	중력 만들기	숫자키	
h	현재 위치 지정	g	지정한 위치로 이동

‘ctrl+좌클릭’과 ‘alt+좌클릭’을 이용해서 간단한 작품을 만들어 보자. 그리고 이러한 방식이 가지는 불편함을 금방 느낄 수 있을 것이다. 일일이 하나씩 클릭하며 만드려면 소위 “노가다(?)”를 해야하므로 많은 시간과 노력이 들 것이다. 이제 우리는 터틀크래프트에서의 코드를 살펴보고 손쉽게 코딩하는 것을 하나하나 배워나가 보려한다.

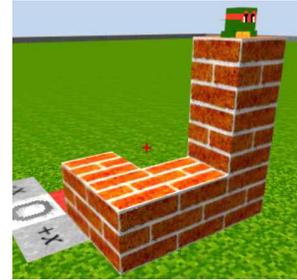
2. Playing Code - doit 명령어



출력창에는 거북이 한 마리가 외롭게 서 있다.
거북이는 이동하며 큐브를 만들 수 있다.

‘입력 창’에 다음과 같은 코드를 입력하고 오른쪽의 실행 결과와 비교하자.

```
1 doit (2s 2l 2u)
2
3
```



우리는 ‘doit()’을 통해 거북이에게 명령을 내렸다. 이는 말 그대로 ‘괄호 안의 행동을 해라!’라는 명령어이다.

※ 6개 방향으로 이동하는 코드는 다음과 같다.

TIP!

자주 사용하는
코드는 기억해 두면
편합니다.

뜻을 잘 보세요~
3차원의 6방향
앞글자만 따른
것입니다.

<p><u>s</u> (step) 앞으로 가면서 큐브</p>		<p><u>t</u>(turn-back) 뒤로 가면서 큐브</p>	
<p><u>l</u>(left) 왼쪽으로 가면서 큐브</p>		<p><u>r</u>(right) 오른쪽으로 가면서 큐브</p>	
<p><u>u</u>(up) 위로 가면서 큐브</p>		<p><u>d</u>(down) 아래로 가면서 큐브</p>	

TIP!

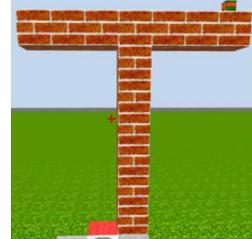
doit의 시작은 s부터 하는게 좋습니다. 기준이 되는 상자를 만들고 그곳에서 움직여 봅시다. * doit (s 6u...)를 doit(7u...)로 시작하면 어떻게 될까요?

아래는 큐브로 H를 써보는 계획, 코드 그리고 이를 실행한 결과이다.

계획	코드	결과
	<pre>1 doit (s 6u 3d 3s 3u 6d) 2 3</pre>	

위 예시와 같이 자신의 이름 한 글자를 쓰는 방법을 생각해 보고, 코딩 해서 결과를 살펴보자.

이제 두 번째 글자를 쓰려면 어떻게 해야 할까? 만약 T자를 추가하려고 하면 'doit (s 6u 3t 6s)' 코드를 실행하면 된다.



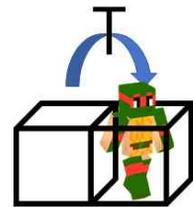
그러나 다음 코드를 실행하면 아래와 같이 의도치 않은 결과가 출력된다.

계획	코드	결과
	<pre>1 doit (s 6u 3d 3s 3u 6d) 2 doit (s 6u 3t 6s) 3</pre>	

이유는 간단하다. 오른쪽 그림에서 H를 만들고 나서 거북이의 위치에 주목하자. 오른쪽 아래에서 끝난 것을 확인할 수 있다.

이후 바로 T를 쓰라고 하면, 글씨가 겹칠 수밖에 없다.

여기서 우리는 거북이가 뛰어넘어갈 필요가 있다는 것을 알 수 있다. 이때, T를 입력하면 거북이가 상자를 만들지 않고 **튀(T)**어 넘는다!



계획	코드	결과
	<pre>1 doit (s 6u 3d 3s 3u 6d) 2 doit (5T) 3 doit (s 6u 3t 6s)</pre>	

알아봅시다!

■ 중복되면 안 좋은가?

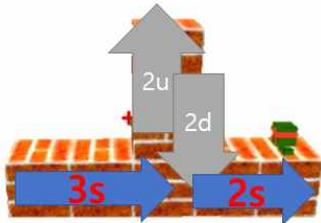
간단한 코드를 만들 때, 중복된 코드는 크게 문제가 안 된다.

그러나, 많은 양의 코드가 중복되면, 처리할 때 시간이 오래 소요된다.

3. Playing Code - 대괄호 [] 명령어

몇 번의 시도를 거쳐 우리는 거쳤던 길을 계속해서 중복하여 블록을 만들고 있음을 알 수 있다. 예를 들어 다음의 코드를 살펴보자.

```
1 doit (3s 2u 2d 2s)
2
3
```



이는 좋은 코드라고 보기 힘들다. 그 이유를 오른쪽 그림을 보면서 생각해 보자. 우선 3칸을 간 뒤에 2칸을 올라가서 윗부분을 만든다. 이후, 다시 2칸 내려온 뒤 앞으로 간다. 즉, 윗부분은 올라가고 내려가는 코드가 중복된 것을 알 수 있다.

이때 거북이의 위치를 우리가 지정할 수 있으면, 위와 같은 문제는 쉽게 해결된다. 오른쪽과 같이 위치를 저장하고, 그 위치로 돌아오는 코드를 입력하면 된다. 그 코드는 다음과 같다.



3s - 위치저장 - 2u -돌아오기- 2s

[위치 저장] 위치로 돌아오기

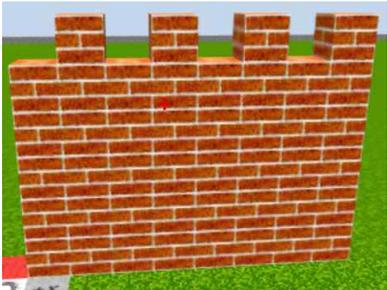
우리는 이 코드를 대괄호 라고 부른다. 즉, 위 코드는 다음과 바꿀 수 있다.

```
1 doit (3s 2u 2d 2s)
2
3
```

→

```
1 doit (3s [2u] 2s)
2
3
```

과제① 오른쪽은 5층, 6층의 큐브를 통해 만든 성벽의 일부이다. 가장 짧은 방법으로 코딩해보자. (층수에 주목하세요!)

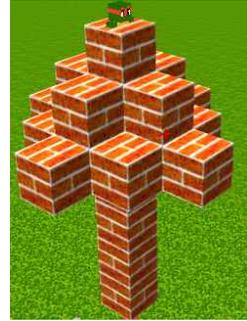


TIP!

각 큐브의 번호를 외을 필요는 없다. 다만, 필요할 때 찾아볼 수 있게 페이지 표시를 해두자.

4. Playing Code - item 설정과 goto 명령어

우리는 이제 우리가 원하는 작품에 대해 item을 설정하는 법을 배우려한다. 예를 들어 나무를 만들고 싶은데 나무가 벽돌로 구성되어 있다면 뭔가 미관상 좋지 않을 것 같다. 원하는 작품에 대해 원하는 큐브를 사용하기 위해 큐브의 종류를 바꾸는 방법에 대해 알아보자.



1	item=2
2	doit (s[3u])
3	item=9
4	doit (s[3u])

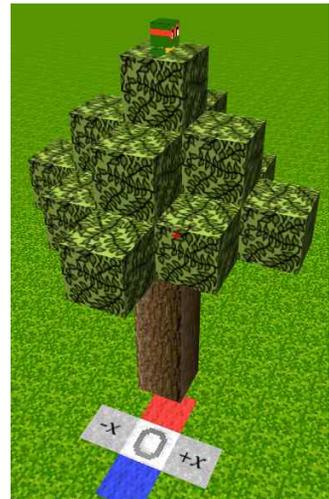
아이템을 줄기(2) 로 하기
 큐브 만들기
 아이템을 석재(9)로 하기
 큐브 만들기



즉, **item=(숫자)** 는 코드는 이어서 사용할 큐브의 종류를 결정하는 코드이다.

과제②

다음과 같이 본인만의 나무를 만들어보자. 대괄호 명령을 적극 활용하고 본인이 원하는 나무의 실제모습과 가깝게 적절한 큐브를 사용하자.

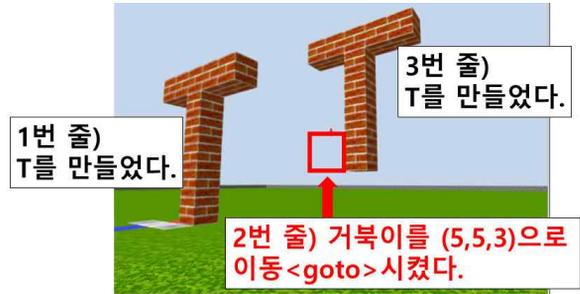


과제2에서 만든 나무의 위치를 바꾸어 심으려면 어떻게 해야 할까? 가까운 위치는 T를 이용하면 되겠지만, 이번에는 거북이를 내가 원하는 위치로 이동시켜 보는 법에 대해 알아보자. 우선 터틀크래프트 화면에서 바닥이나 큐브를 왼쪽 마우스로 클릭하면, 입력창 밑에 그 위치가 표시되는 것을 알 수 있다. 이를 통해, 적당히 떨어진 위치를 찾아서 다음과 같이 코딩하고 확인해 보자.



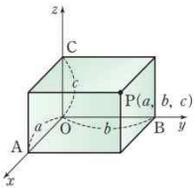
큐브 위치 (2, 1, 0)

```
1 doit (s [5u [2t] [2s] ])
2 goto (5,5,3)
3 doit (s [5u [2t] [2s] ])
```



알아봅시다!

■ 공간 좌표



중1때 배우는 평면 좌표가, 평면에서 점의 위치를 표현할 때 사용되었다면,

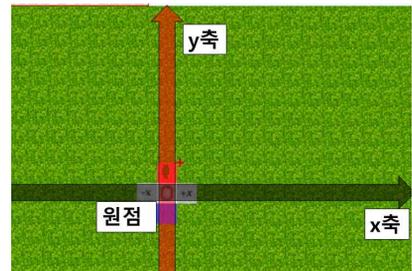
우리는 공간좌표를 통해 3차원에서 점의 위치를 표현할 수 있다.

위의 그림과 같이 x,y좌표로 평면의 위치를 정한 뒤, z를 이용해 높이를 정하면 공간의 점의 위치를 표현할 수 있다.

2번째 줄에 나온 goto(5,5,3)이라는 것은 거북이를 (5,5,3)으로 이동시키라는 것이다.

즉, `goto(x,y,z)` 는 (x,y,z)로 거북이를 이동시키라는 뜻이다.

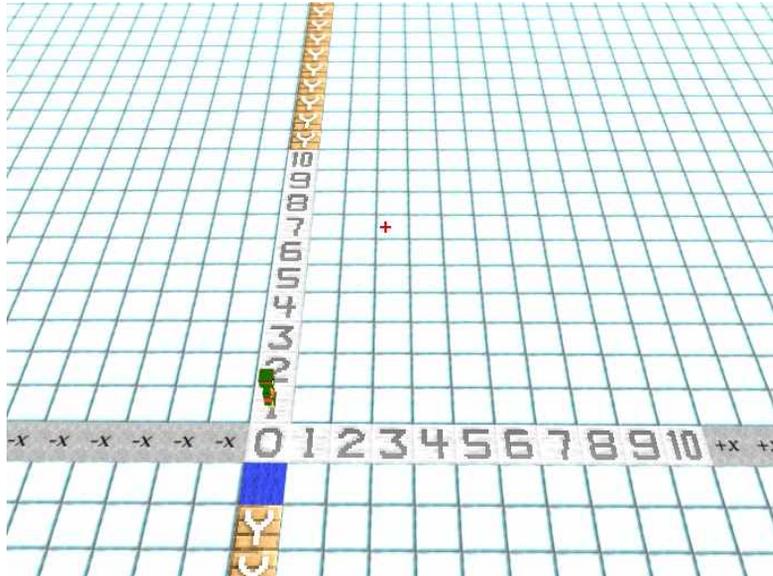
이때, (x,y,z)는 무슨 의미를 가질까? 오른쪽의 그림을 보면 쉽게 이해가 갈 것이다. z키를 눌러서 공중에서 바닥을 볼 때, x-와 x+가 표시된 쪽이 x축이다. 또한, 붉은색과 파란색으로 표시된 부분이 y축이다.



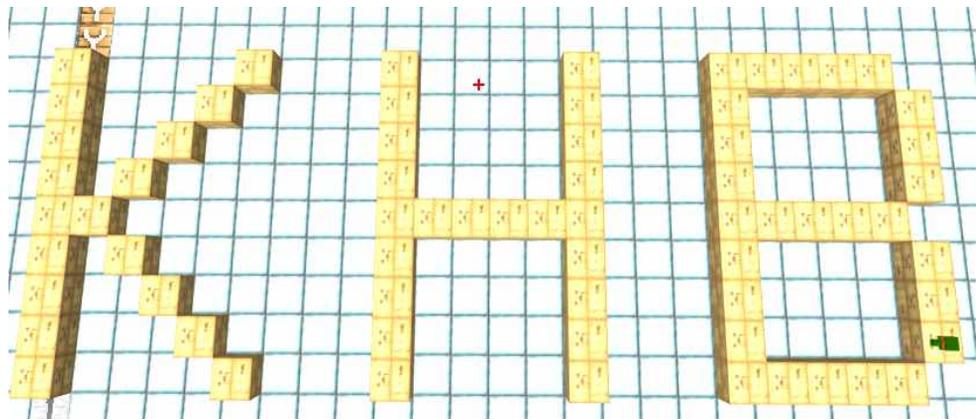
마지막으로 z는 높이를 나타낸다. 즉, 왼쪽과 같이 goto(5,5,1)의 경우 (5,5) 좌표의 1층으로 이동하라는 뜻이다. 따라서, x, y, z는 각각 좌우, 앞뒤, 위아래를 뜻하는 좌표라고 생각하면 된다.

이제 좌표에 대해 어느정도 이해가 되었다면 그 좌표를 눈에 쉽게 보기 위해 `beginxyz` 명령어 옆에 ; (세미콜론)을 붙여보고 바닥에 큐브끼리 구분이 가도록 바꿔보자.

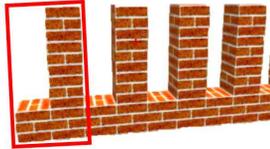
`beginxyz; item=6` 을 입력한 뒤 출력창을 살펴보면 우리가 원하는 위치를 쉽게 볼 수 있도록 격자점처럼 출력된 것을 볼 수 있다!



과제③ 지금까지 배운 `doit`, 대괄호, `goto`명령어 등을 모두 사용하여 자신의 이니셜을 표현해보자.



5. 치환과 회전



반복되는 부분

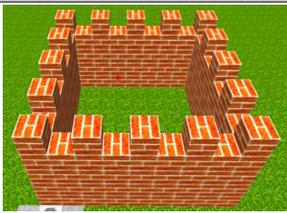
왼쪽 그림과 같이 반복되는 부분이 있을 때, 일일이 모든 부분에 대해 코드를 입력하려면 매우 성가신 일이 될 것이다. 만약, 이 부분을 “한 덩어리”의 레고의 조각처럼 활용할 수 있다면 작품을 만들기 편해질 것이다. 발견한 규칙을 하나의 기호로 표현하는 것을 우리는 치환이라고 표현한다. 이때, 우리는 다양한 치환문자로(ex> X, Y, Z)반복된 부분을 치환할 수 있다. 치환 코드는 다음과 같다.

- | | |
|---|------------------------|
| 1 | 치환문자 = ‘같은 규칙을 가지는 코드’ |
| 2 | doit(치환문자) |

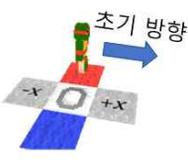
즉, 위 문제는 다음과 같이 치환을 통해 간단히 표현 가능하다.

1	doit (2s [3u] 2s [3u] 2s [3u] 2s [3u] 2s [3u])	→	1	X = ‘2s [3u]’
2			2	do(5X)

다음으로 성벽을 만드는 코드를 살펴보자. 코드를 보면, 거북이가 앞으로 가고, 왼쪽으로 가고, 뒤로 가고,

작품	코드
	<pre> 1 X = 's[3u] s[4u]' 2 Y = 'l[3u] l[4u]' 3 Z = 't[3u] t[4u]' 4 A = 'r[4u] r[5u]' 5 doit (4X 4Y 4Z 4A) </pre>

오른쪽으로 간다. 좀 더 Simple 하게 표현할 수는 없을까?

초기 방향	왼쪽 90도 회전	오른쪽 90도 회전
		<pre> 1 doit(L) </pre> 

과제 4 자, 이제 X만 가지고도 표현을 위의 성벽을 표할 수 있지 않을까? 치환한 X를 한 덩어리로 본다면 X만큼 앞으로 가고 왼쪽으로 돌아 또 앞으로 가는 식의 ‘앞으로가고 왼쪽으로 도는 것’을 4번 반복한 것이다!!

Hint :

1	X = ‘s[3u] s[4u]’
2	doit(?? ? ?? ? ?? ? ?? ?)