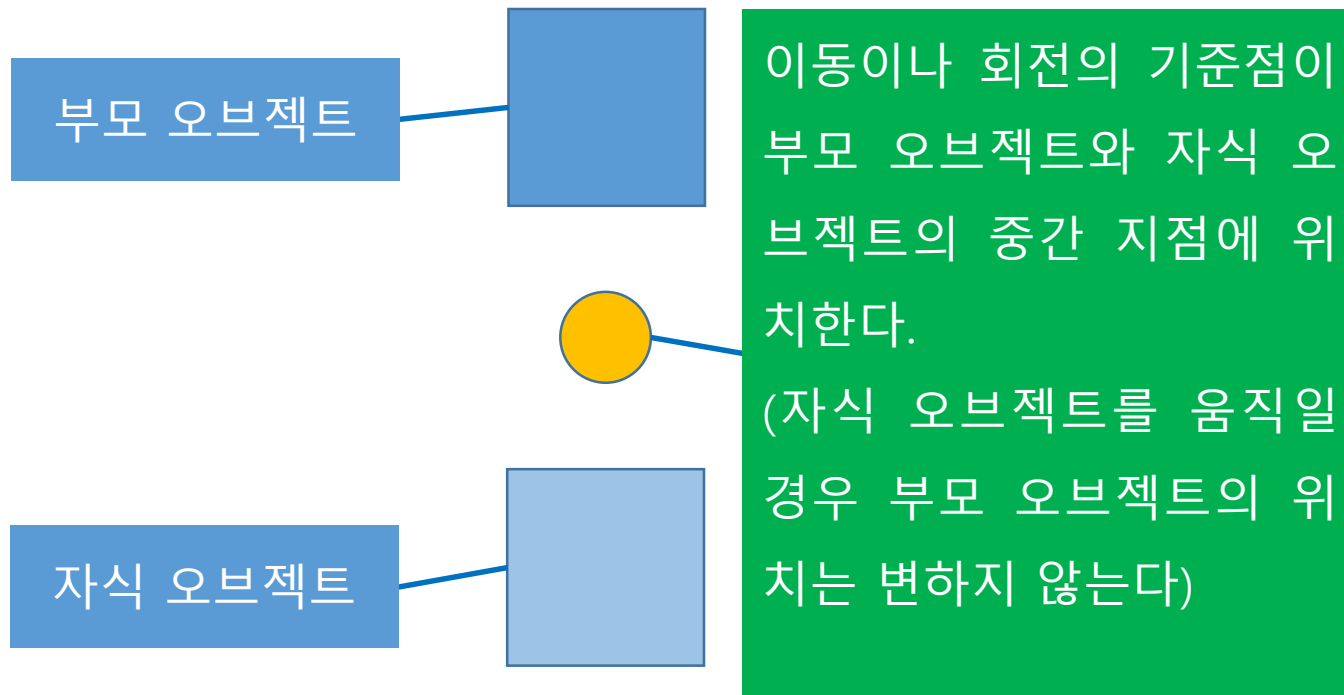
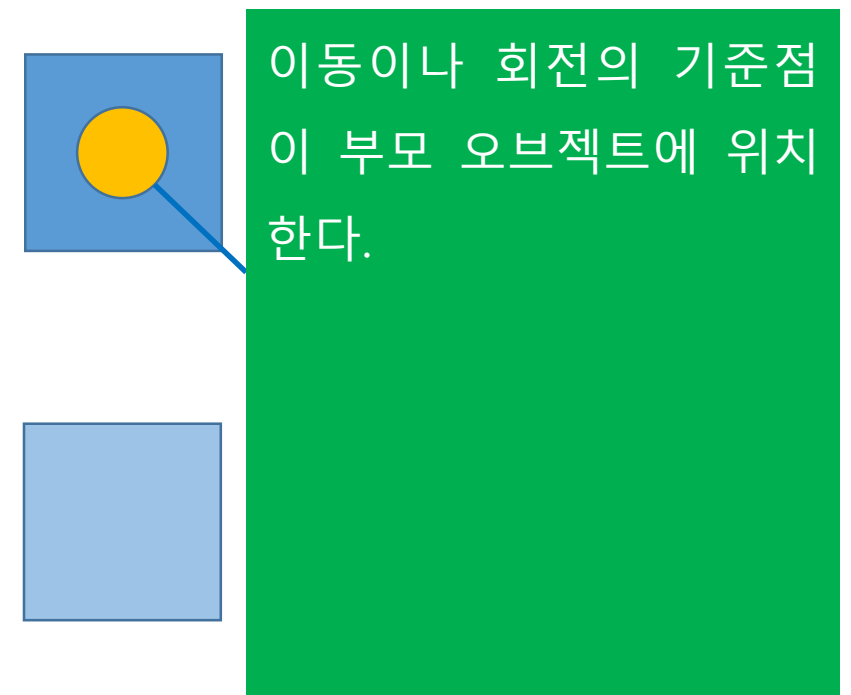


글로벌 좌표 vs. 로컬 좌표

글로벌 좌표

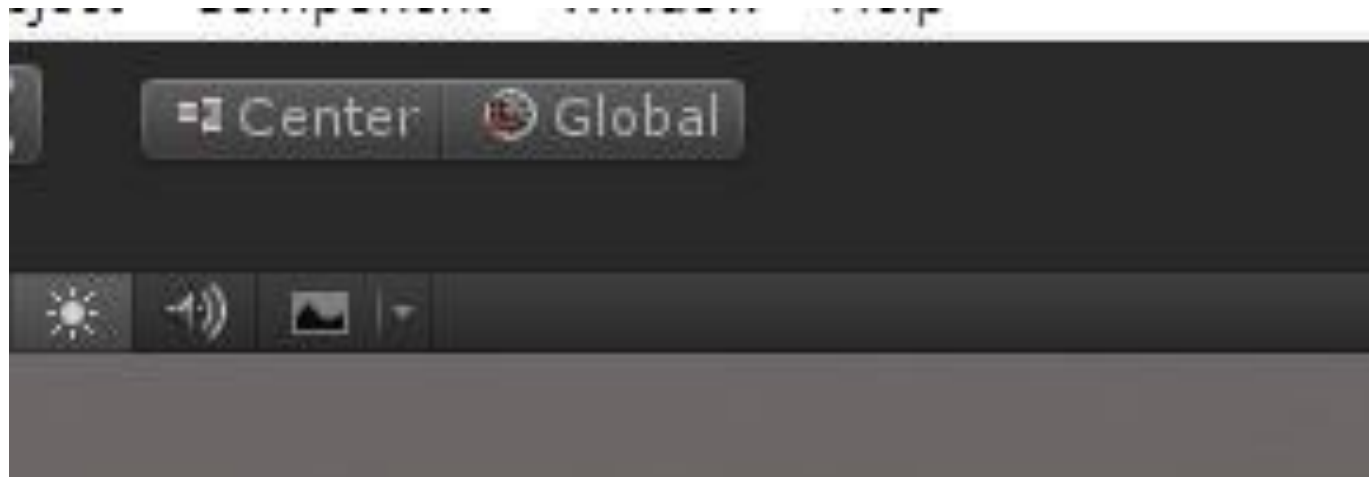


로컬 좌표

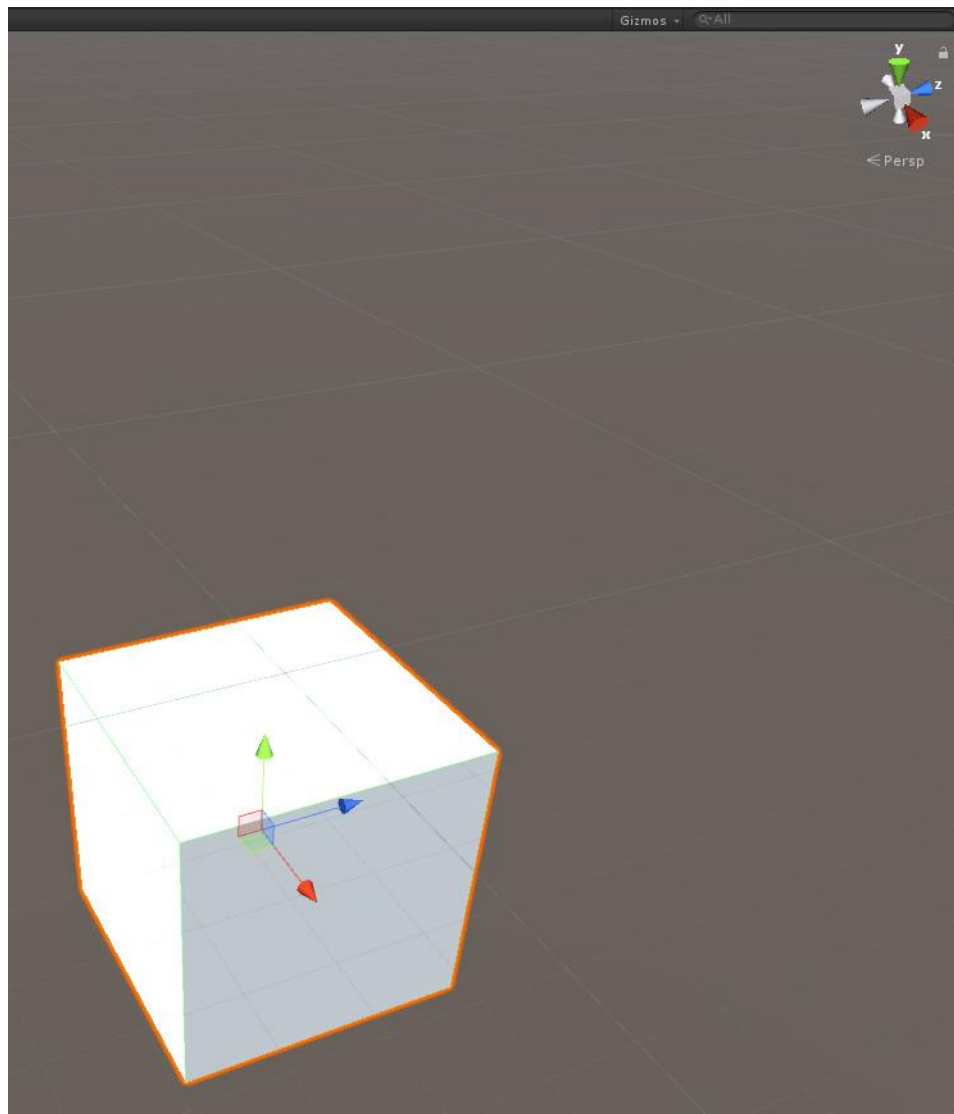


글로벌 좌표 vs. 로컬 좌표

- 글로벌 좌표
 - 게임 화면을 기준으로 하는 절대 좌표
 - 게임 오브젝트의 기즈모는 오브젝트가 바라보는 방향과는 상관없이 씬 기즈모와 같은 방향으로 표시된다.

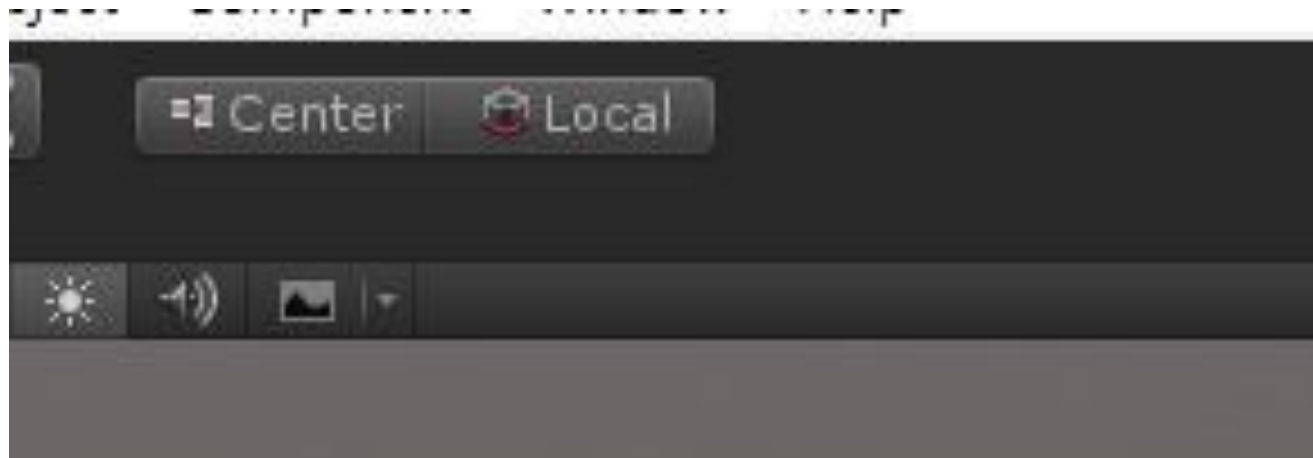


글로벌 좌표

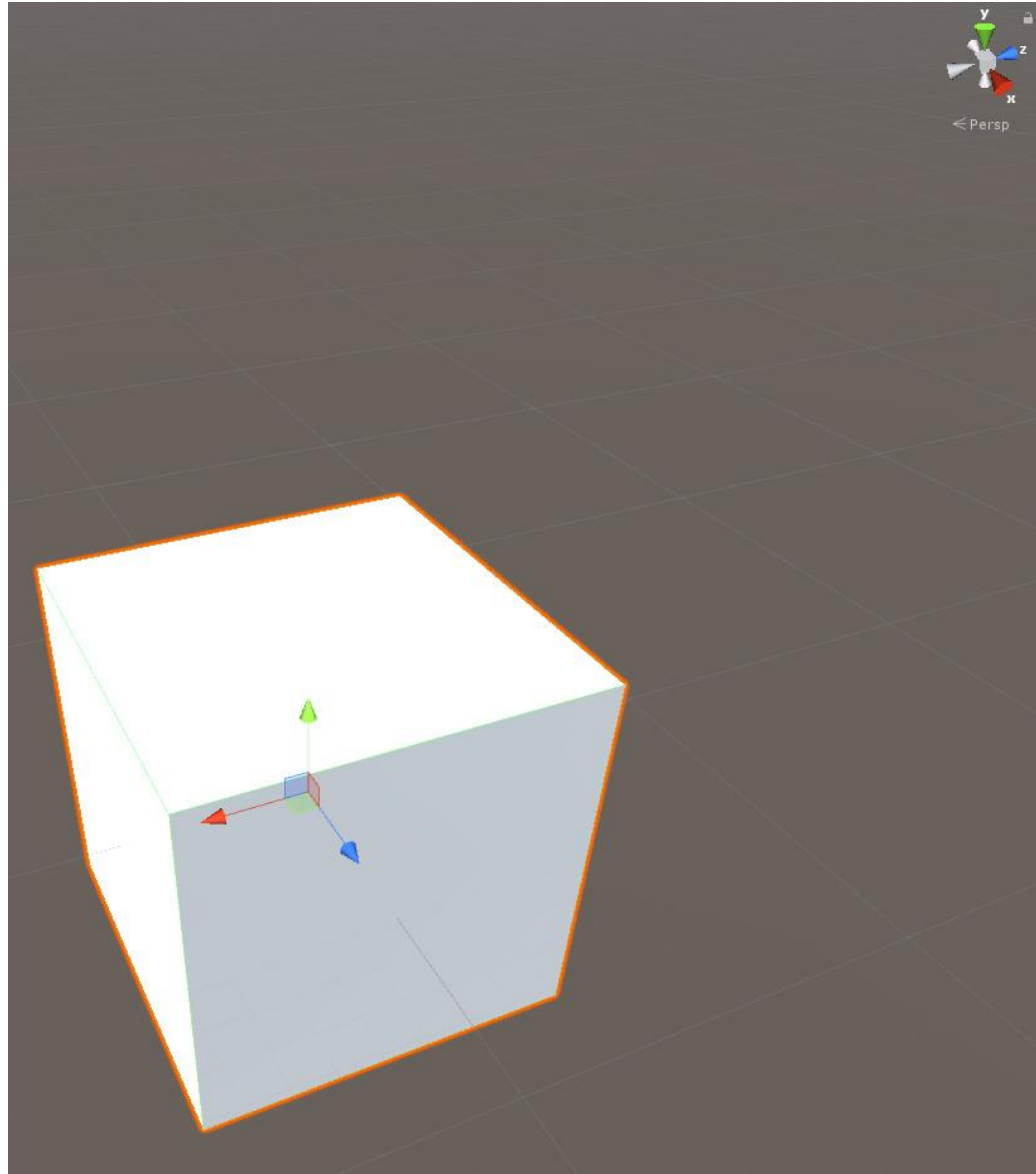


로컬 좌표

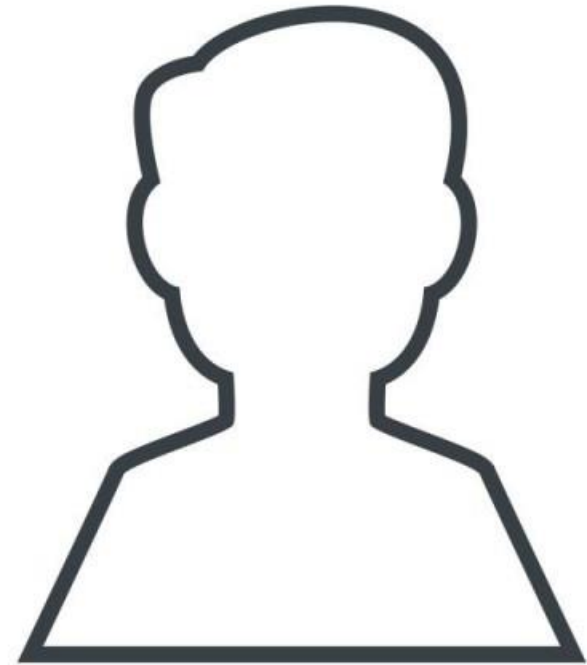
- 로컬 좌표
 - 게임 오브젝트를 기준으로 한 좌표
 - 게임 오브젝트가 바라보는 방향이 z축이 됨



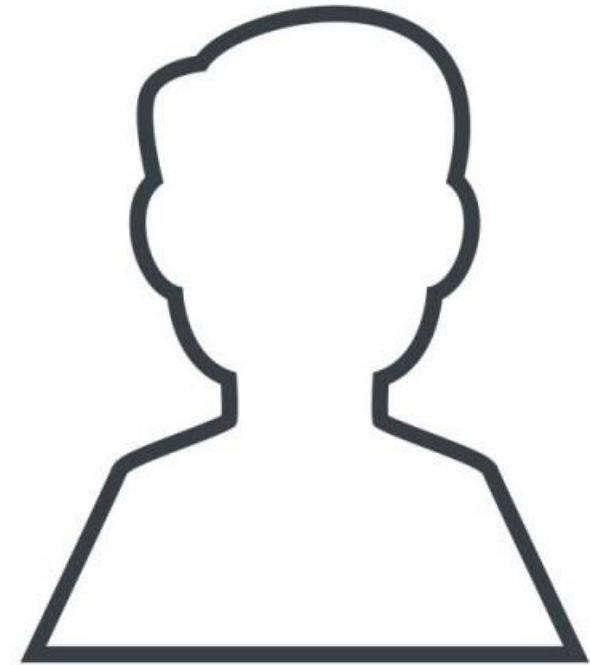
로컬 좌표



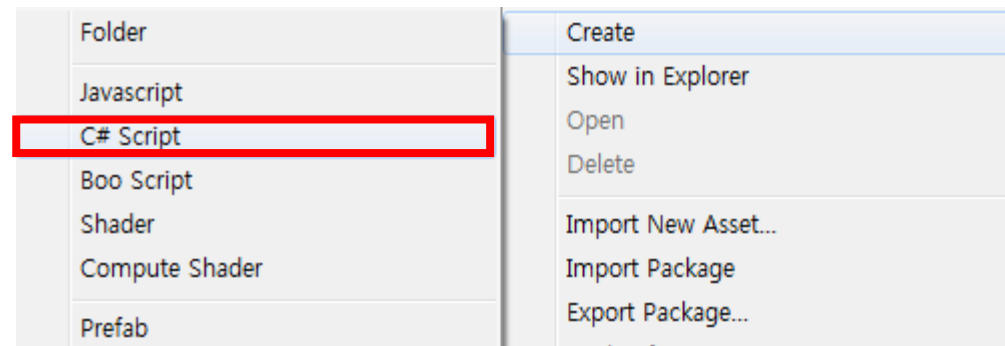
Parent vs. Child



Parent vs. Child



Script를 담는 그릇, Class(클래스)



Script를 담는 그릇, Class(클래스)

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11     // Update is called once per frame
12     void Update () {
13
14     }
15 }
16
```

Class의 이름

Class의 내용

Script의 이름과 class의 이름은 일치되어야 한다.

스크립트의 핵심요소, 변수와 함수

사람들끼리 대화할 때 명사와 동사가 사용되듯이, 유니티에서 기본 되는 개념이 변수와 함수이다.

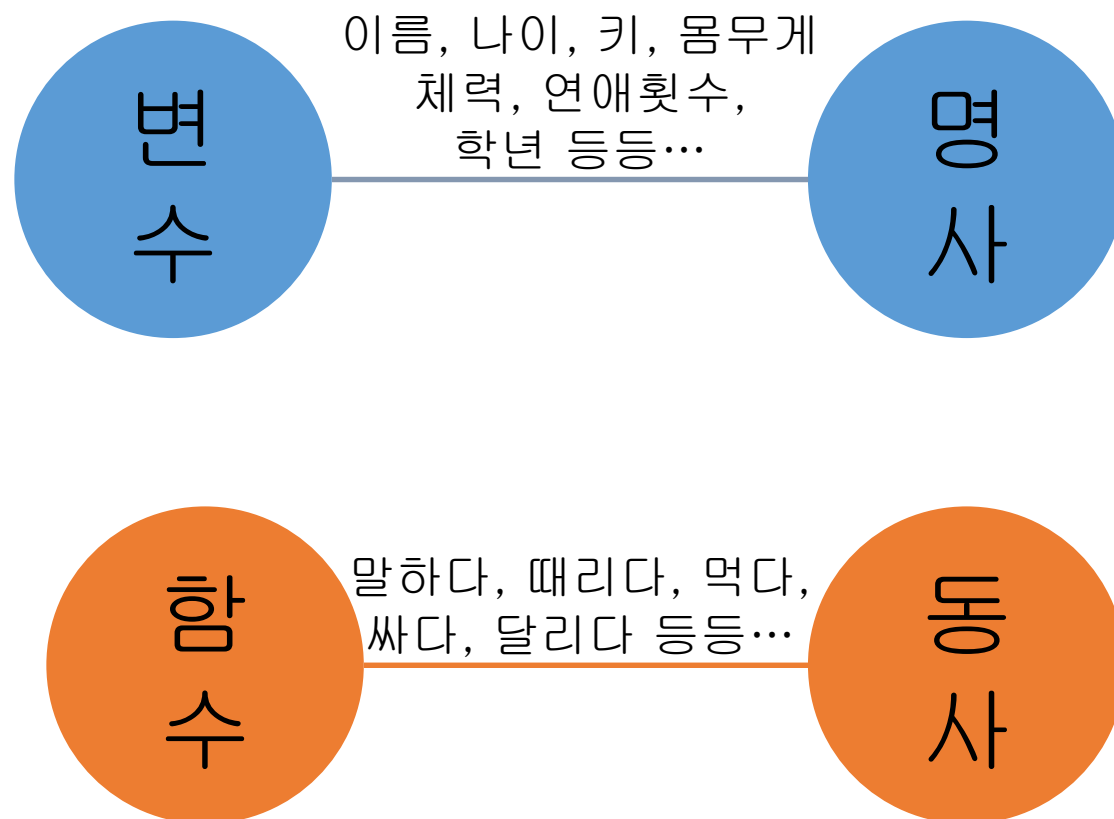


수치, 혹은 문자 등
데이터를 넣어놓기 위
해 확보해 둔 공간
Int A = 17;



변수를 변화시키기 위
해
컴퓨터와 사용자가 미
리 정해놓은 규칙
void Update(){ }

스크립트의 핵심요소, 변수와 함수



스크립트의 핵심요소, 변수와 함수

오른쪽 키가 눌리면 마법사는 오른쪽으로 10m 움직인다.

오른쪽 키 눌리면 마법사 오른쪽 10m 움직인다.

개발의 기본 요소, 변수

데이터 타입	예 시	설 명
Int	...-3, -2, -1, -, 1, 2, 3...	정수타입의 변수. 1, 2, 3과 같이 정수로 떨어지는 수들을 위해 사용하는 변수의 종류이다. 숫자이기에 변수 간의 크고 작음의 구분이 가능하다.
float	...-0.2f, -0.1f, 0f, 0.1f, 0.2f...	소수 타입의 변수. 0.1, 0.001 등 소수점 단위까지 사용해야 하는 변수의 종류이다. 숫자라서 변수 간의 크고 작음의 구분이 가능하다.
bool	True, False	참과 거짓. 2가지 중 하나의 값이 들어가는 스위치와 같은 역할을 하는 변수. 숫자가 아니기에 변수 간의 크고 작음의 구분은 불가능하지만 지금 활성화/비활성화에 대해서 체크를 할 수 있다.
string	null, "abc", "안녕"...	숫자가 아닌 문자열을 저장하기 위한 변수 타입. 화면에 텍스트를 출력하거나 텍스트가 맞는지 틀리는지 체크하는 용도로 사용한다.
enum		새로운 데이터 타입을 선언해서 관리하고 사용할 수 있도록 변수의 종류를 새롭게 만들어 주는 변수.

개발의 기본 요소, 변수

데이터 타입	예 시	설 명
GameObject	Scene뷰에 있는 오브젝트, 프로젝트	유니티 C#에만 존재하는 변수 타입. Scene 뷰에 존재하는 오브젝트, 혹은 Project 뷰에 있는 Prefab을 스크립트에서 사용하고 싶을 때 GameObject 타입으로 변수를 선언해서 사용할 수 있다.
Transform	Scene 뷰에 좌표 값을 가지고 있는 오브젝트	3D를 기반으로 한 유니티 엔진에서 Scene에 존재하는 오브젝트의 좌표, 회전값, 크기를 저장하기 위한 변수 타입이다.
Vector2	(0, 0), (134,234), (-34,2)...(x,y)	(x, y)처럼 2가지의 값을 묶어서 갖고 있는 변수 타입. 가로, 세로로 된 이미지의 사이즈나 마우스의 위치 (x, y)의 좌표값을 표기하기 위해 사용되는 변수 타입
Vector3	(0,0,0), (1,2,3)....(x,y,z)	(x, y, z)처럼 3가지의 값을 갖고 있는 변수 타입. 3D오브젝트들의 좌표/스케일 값을 표기하는데 자주 사용되지만, 어떤 경우에는 (현재의 수치, 바뀔 수치, 바뀔 시간)으로 사용되기도 하는 등 3가지의 값을 다양하게 묶어서 사용되는 변수 타입이다.

개발의 기본 요소, 변수

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript : MonoBehaviour {
5
6     int A;
7     float B;
8     bool C;
9     string D;
10    GameObject E;
11
12
13    // Use this for initialization
14    void Start () {
15
```

왜? 문장의 끝에 ; 을 넣어야 하는가?

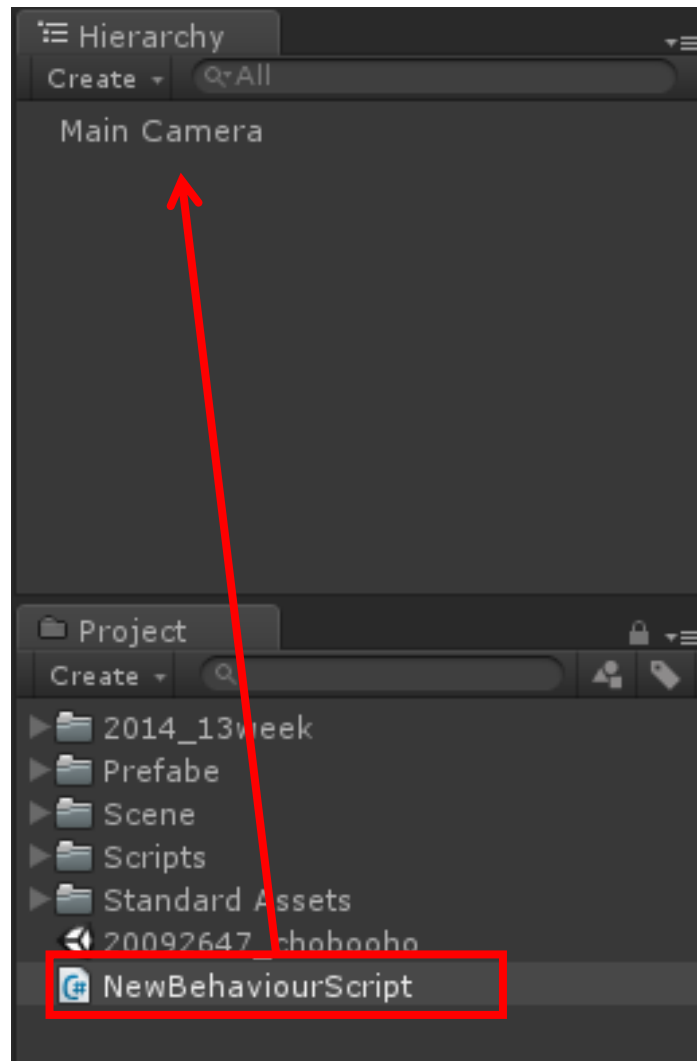
컴퓨터는 Script 내의 글을 일렬한 줄로 쭉 연결하고 한 글자씩 읽다가 마침표가 보일 때마다 한 줄씩 끊고, 각각 한 줄씩을 순차적으로 실행한다.

; 부호는 한 줄이 끝났구나 라는 것을 컴퓨터에게 알리는 마침표이다.

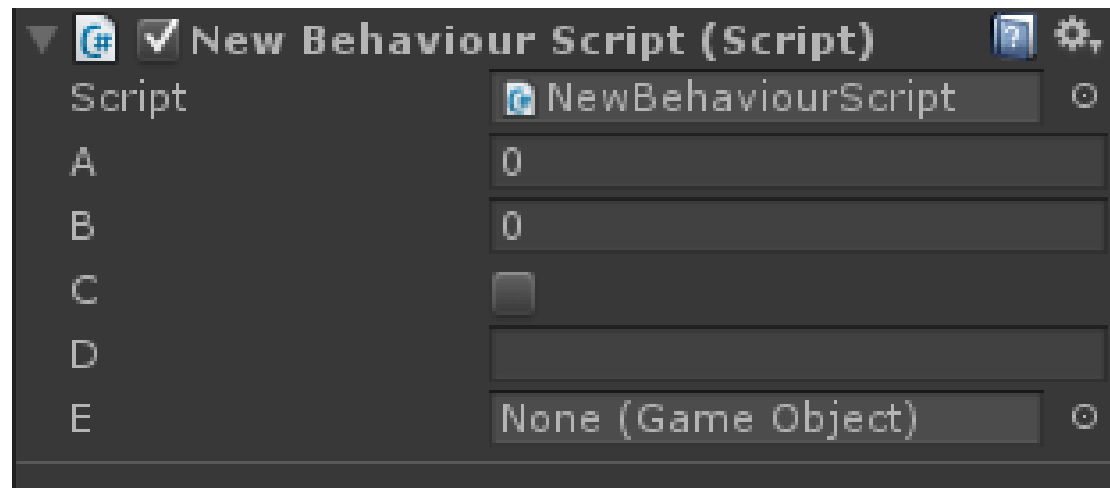
개발의 기본 요소, 변수

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript :
5
6     public int A;
7     public float B;|
8     public bool C;
9     public string D;
0     public GameObject E;
1
2
3     // Use this for initialization
```


개발의 기본 요소, 변수



개발의 기본 요소, 변수



※ 변수의 선언방식

Public	int	A	;
사용권한	타입	이름	마침표

부품을 움직이게 하는 동력, 함수

※ 함수의 역할과 구조, 그리고 함수의 종류

Public void Move () { }

사용권한 타입 이름 전달값 실행 구문

	종류	선언	실행조건
함수	시점함수	필요함	특정 시점에 따라 자동으로 실행
	일반함수	필요함	실행문으로 실행
	기본제공 함수	필요 없음	실행문으로 실행

부품을 움직이게 하는 동력, 함수

※ 선언만 해놓으면 자동실행되는 시점함수

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11     // Update is called once per frame
12     void Update () {
13
14     }
15 }
16
```

부품을 움직이게 하는 동력, 함수

시점함수	필요 전달값	발동 시점
Start	없음	이 Script가 시작되는 순간, 단 한 번 실행 구문 안의 내용을 한 번씩 실행시킨다.
Update	없음	이 Script가 활성화되어 있는 동안, 매 프레임마다 실행 구문 안의 내용을 한 번씩 실행시킨다.
OnEnable	없음	이 Script가 부착된 오브젝트가 활성화되는 순간마다, 실행 구문 안의 내용을 한 번씩 실행시킨다.
OnDisable	없음	이 Script가 부착된 오브젝트가 비활성화되는 순간마다, 실행 구문 안의 내용을 한 번씩 실행시킨다.
OnTriggerEnter	충돌체	Trigger 속성의 충돌체가 특정 오브젝트와 부딪힌 순간, 실행 구문 안의 내용을 한 번씩 실행시킨다.
OnTriggerStay	충돌체	Trigger 속성의 충돌체의 경우 충돌체를 가진 오브젝트가 계속 겹쳐있는 동안, 매 프레임마다 실행 구문 안의 내용을 한 번씩 실행시킨다.
OnTriggerExit	충돌체	Trigger 속성의 충돌체의 경우 자신으로부터 특정 오브젝트가 자신의 충돌영역을 벗어나는 순간, 실행 구문 안의 내용을 한 번씩 실행시킨다.

사운드

- 사운드를 듣는 오디오 리스너(Audio Listener)
- 사운드를 발생시키는 오디오 소스(Audio Source)
- 오디오 소스에서 사용하는 오디오 데이터(Audio Clip)

사운드 적용하기

- Plane 오브젝트 생성
- Sphere 오브젝트 생성
- Sphere 오브젝트의 Position을 Plane 오브젝트 위로 설정한다.
- Sphere 오브젝트 이름 변경 : **Ball**
- Ball을 선택

사운드 적용하기#1

- **Ball을 선택한 후** 다음과 같이 **컴포넌트를 추가한다.**
 - Component → Audio → Audio Source
- Play On Awake 옵션에 체크를 해제한다.
 - 이 옵션이 활성화 되어있다면 게임이 시작되었을 때 오디오 클립을 무조건 한 번 실행시킨다.
- Ding 사운드를 Audio Clip에 연결한다.

사운드 적용: 스크립트#2

- C# 스크립트를 추가한다.

```
void OnCollisionEnter(Collision collision)
{
    GetComponent<AudioSource>().Play();
}
```

사운드 적용: PlayOneShot

- C# 스크립트를 추가한다.

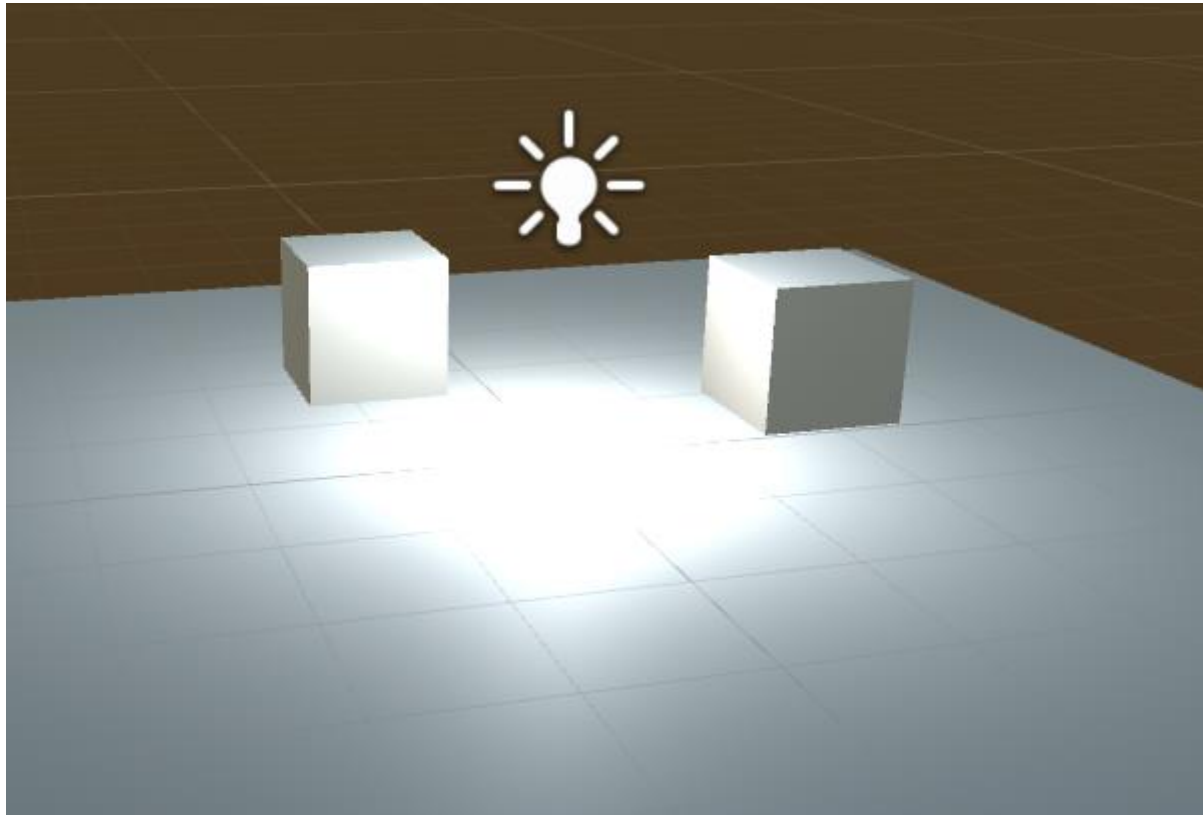
```
public AudioClip clip;
```

```
void OnCollisionEnter(Collision collision) {
```

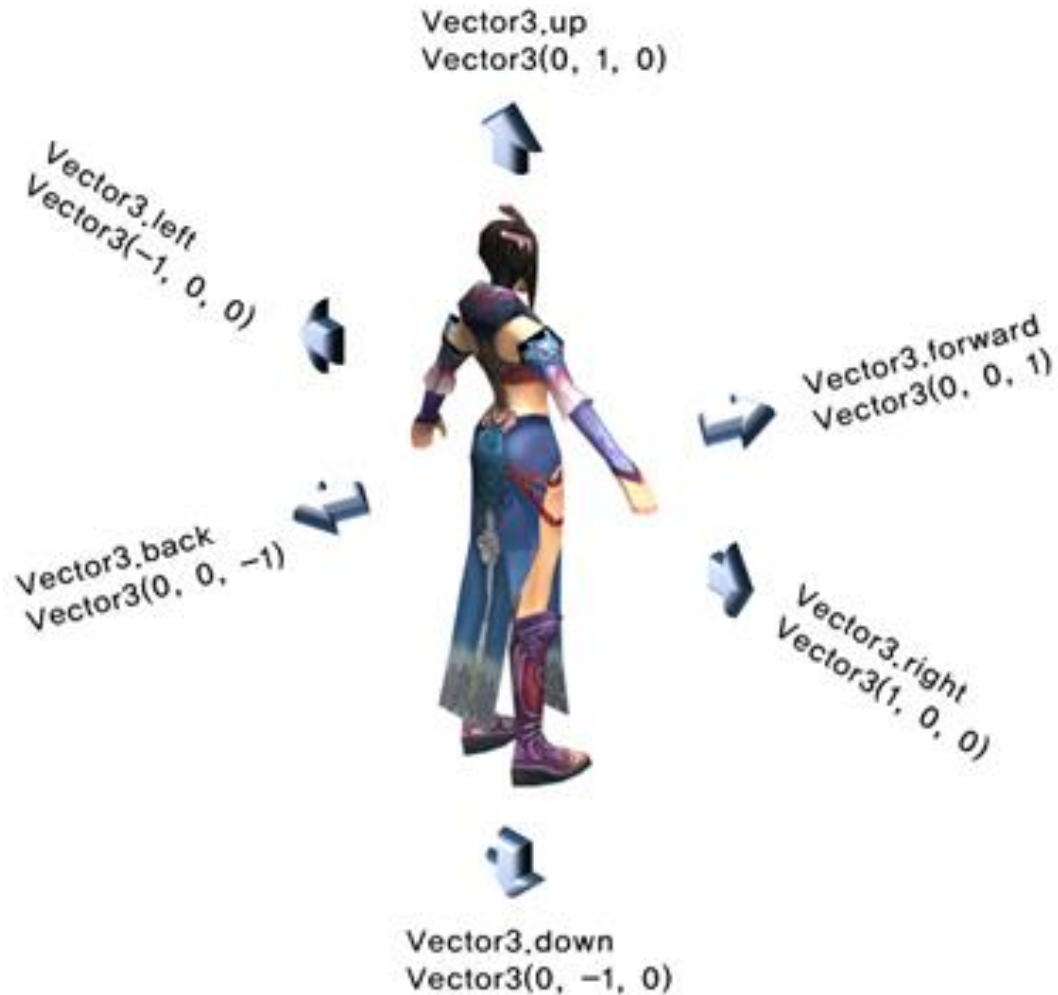
```
    GetComponent<AudioSource>( ).PlayOneShot(clip, 0.8f);
```

```
}
```

스크립트 활용하기 - MOVE



스크립트 작성하기



- 방향을 지시하는 용도로 사용할 때는 (x, y, z) 값을 0과 1로 표시한다.
- 거리를 지시하는 용도로 사용할 때는 (x, y, z) 값을 해당 크기만큼 표시한다.

스크립트 활용하기 – MOVE

- Scene을 저장한다. [Move]
- Cube가 전후좌우로 움직이게 만든다.
- 스크립트 생성하기 – Move

스크립트 작성하기

```
public class Move : MonoBehaviour {  
    float speed = 20.0f;  
  
    void Update () {  
        // 키보드 입력 스크립트  
  
        float h = Input.GetAxis("Hori");  
  
        float v = Input.GetAxis("Vertical");  
  
  
        // 이동 거리 보정 스크립트  
  
        h = h * speed * Time.deltaTime;  
  
        v = v * speed * Time.deltaTime;
```

스크립트 작성하기

```
// 실제 이동 스크립트
```

```
transform.Translate(Vector3.right * h);
```

```
transform.Translate(Vector3.forward * v);
```

```
}
```

```
}
```