

안녕하세요.
골드메탈입니다.

모두 추석 잘 보내셨는지요?
필자는 추석 음식을 많이많이 먹어서
체중이 늘어난 느낌이에요.



(30대가 가까워지는 마당에 뱃살관리도 해야하는데..)

여튼 추석도 끝났으니
다시 유니티 공부를 계속해야겠지요?



11. 캐릭터 회전

이번 편은 회전에 대해서 짧막하게 다루어보도록 하죠.
프로젝트를 열고 바로 스크립트로 넘어갑시다.

```
//-----[ Action Function ]
void Run ()
{
    movement.Set (horizontalMove, 0, verticalMove);
    movement = movement.normalized * speed * Time.deltaTime;

    rigidbody.MovePosition (transform.position + movement);
}

void Turn ()
{
}

void Jump ()
{
    if(!isJumping)
        return;

    rigidbody.AddForce(Vector3.up * jumpPower, ForceMode.Impulse);

    isJumping = false;
}
```

회전에 대한 함수를 FixedUpdate에 넣어주었습니다.
물론 리지드바디를 컨트롤 하기 때문이죠.

이제 새로운 클래스 **Quaternion** 을 소개합니다.
 Quaternion는 회전만을 다루는 클래스로
 회전 값을 정할 수 있습니다.

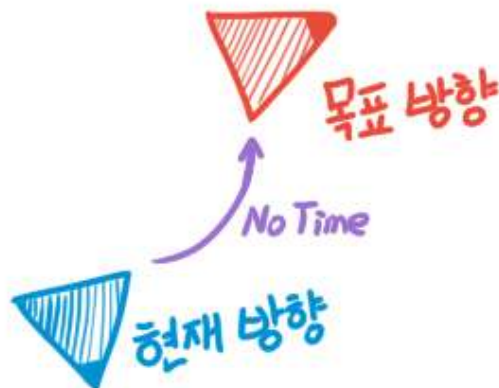
다만 Transform 같이 X, Y, Z 축을
 직접 접근하여 변경할 수는 없고,
 제공해주는 함수를 사용해야 하죠.

```
void Turn ()
{
    Quaternion newRotation = Quaternion.LookRotation(movement);
}
```

가장 기본적인 함수는 **LookRotation (Vector)** 입니다.
 벡터가 가지고 있는 방향으로, **앞을 보는 회전 값**을 만들어줍니다.

```
void Turn ()
{
    Quaternion newRotation = Quaternion.LookRotation(movement);
    rigidbody.MoveRotation(newRotation);
}
```

이 회전 값을 리지드바디의
MoveRotation (Quaternion) 함수를 통하여
 적용해보도록 하죠.



MoveRotation 함수는 목표 회전 값으로
 바로 회전시켜줍니다.



이제 방향을 바꾸면
 캐릭터 그 방향을 앞으로 바라보도록 회전하네요.

하지만 순식간에 핵 바뀌어
회전한다는 느낌이 들지 않습니다.

```
void Turn ()
{
    Quaternion newRotation = Quaternion.LookRotation(movement);

    rigidbody.rotation = Quaternion.Slerp (rigidbody.rotation, newRotation, rotateSpeed * Time.deltaTime);
}
```

좀 더 부드럽게 회전시키기 위해서
Quaternion 의 **Slerp** 를 사용해봅시다.



Slerp 함수는 목표 회전 값까지
회전 속도에 따라 천천히 회전시키는 함수입니다.



이제 확실히 회전한다는 느낌을 주는 모습입니다.

```
void Turn ()
{
    if(horizontalMove == 0 && verticalMove == 0)
        return;

    Quaternion newRotation = Quaternion.LookRotation(movement);

    rigidbody.rotation = Quaternion.Slerp (rigidbody.rotation, newRc
}
```

하나 더.
회전은 방향 키로 컨트롤할 때만 적용되어야하니까
위 스크립트처럼 방향제어가 없을 땐,

회전 로직을 타지 않도록 막아두어야합니다.



이 로직이 없다면,
이렇게 항상 벡터가 0 값인 방향을 바라보게 되버립니다.

열한번째 강좌는 이렇게 마무리되었습니다.
다음 시간에는 카메라 이동에 대해 다루어보겠습니다.
그럼.