

C#  
(C Sharp)

Instantiate??



# Object.Instantiate

```
public static Object Instantiate(Object original);  
public static Object Instantiate(Object original, Transform parent);  
public static Object Instantiate(Object original, Transform parent, bool worldPositionStays);  
public static Object Instantiate(Object original, Vector3 position, Quaternion rotation);  
public static Object Instantiate(Object original, Vector3 position, Quaternion rotation, Transform parent);
```

## Parameters

|                           |  |
|---------------------------|--|
| <b>original</b>           | An existing object that you want to make a copy of.                            |
| <b>position</b>           | Position for the new object (default <a href="#">Vector3.zero</a> ).           |
| <b>rotation</b>           | Orientation of the new object (default <a href="#">Quaternion.identity</a> ).  |
| <b>parent</b>             | The transform the object will be parented to.                                  |
| <b>worldPositionStays</b> | If when assigning the parent the original world position should be maintained. |

## Returns

**Object** A clone of the original object.

# Casting & Manipulating

```
// Spawning and casting
Vector3 spawnPoint = new Vector3(1f, 0f, 0f);
Quaternion spawnRotation = Quaternion.identity;
GameObject clone = (GameObject) Instantiate(Prefab, spawnPoint, spawnRotation, transform);

// Now we have a GameObject, rather than an Object. We can use any of the methods
// available on a GameObject:

// Apply a random scale
Vector3 randomScale = new Vector3(1f, Random.Range(1f, 3f), 1f);
clone.transform.localScale = randomScale;
```

# Arrays

```
int[] HighScores;
```



ARRAY TYPE



# Ways to Create an Array

```
// Empty integer array  
int[] HighScores;
```

```
// Empty integer array with four element  
int[] HighScores = new int[4];
```

```
// Integer array with specific values  
int[] HighScores = { 10, 12, 15, 20 };
```

# Resources

- Ray Wenderlich – [Video](#) on arrays
- Unity [tutorial](#) on arrays
- Blog [post](#): data structures in Unity and when to use them
- Unity [tutorial](#) on Lists and Dictionaries