# Unity Doc 1: Giriş

Methods:

A method is a block of code which only runs when it is called. You can pass data, known as parameters, into a method. Methods are used to perform certain actions.

*Write something onto console using Debug.Log("");

Script:

In computer programming, a script is a program sequence of instructions that is interpreted or carried out by another program rather than by the computer processor (as a compiled program is).

Start and Update Functions:

Start function is called at the beginning of the program but update is called every frame.

```
//Rotates the object around given axes relative to world or self
transform.Rotate(x,y,z,RelativeSpace); //0.1f space.self or space.world
```

*Red axis is X and green axis is Y in unity.

---

```
//Moves to object at given axes relative to world or self
transform.Translate(x,y,z,RelativeSpace); //space.self or space.world
```

---

```
//Declate a variable and make it visible in the inspector
[SerializeField] float speed = 0.1f;
```

---

Axis Input:

Edit > Project Settings > Input Manager
"Horizontal" for x axis and "Vertical" for y axis.

```
Input.GetAxis("Horizontal"); // or "Vertical"
//Returns a value for horizontal axis from -1 to 1 according to input. If the input key is "a" it returns -1 which is in the direction of -x, if it is "d" it returns 1 which is in the direction of x.
```

*For the case of transform.Rotate, because of positive rotation is counter clockwise we put a minus sign in front of the value. This way the rotation becomes clockwise.

---

## Time.deltaTime:

```
Time.deltaTime; //Returns how many seconds have passed between one frame to another.
```

*For example if our speed is equal to 2:

Slow computer : FPS: 10 Duration Of A Frame: 0.1s

Distance per second: 2 x 10 x 0.1 = 2

Fast computer : FPS: 100 Duration Of A Frame: 0.01s

Distance per second: 2 x 100 x 0.01 = 2

---

## Colliders & Rigidbodies:

We can use colliders and rigid bodies to detect collisions and do stuff. Rigidbodies are for physics interactions.

*Don't forget to set gravity to 0 for 2d project.

```
//On collision enter works one time when the collision happens and executes the code inside it. In this case printing a message onto the console.
private void OnCollisionEnter2D(Collision2D other)
{
      Debug.Log("Collided!");
}
//Private keyword is for saying this can only be used in this class.
```

*collision.gameObject.CompareTag

---

## Triggers:

Triggers are used for detecting something and executing instructions accordingly.

```
private void OnTriggerEnter2D(Collider2D other)
{
     Debug.Log("Triggered!");
}
```

*remember to tick the trigger on collision component.

---

## Hierarchy and Game World

---

## Follow Camera:

```
[SerializeField] GameObject thingToFollow; //Declaring a new object thingToFollow which is a GameObject
```

```
    //Inside the late update assign the position of the object to follow to the object that
    the script is attached.
    transform.position = thingToFollow.transform.position + new Vector3 (0,0,-1);
```

https://docs.unity3d.com/Manual/ExecutionOrder.html

## Tags:

```
    private void OnTriggerEnter2D(Collider2D other)
    {
    //Execute the following code if the trigger object has the tag something
        if(other.tag == "something")
        {
            //Do something
        }
    }
```

*using bools

## Destroying Objects:

```
    Destroy(other.gameObject,destroy_Delay); //Destroys the game object that is
    attached to other
```

## Getting Components:

```
    //In the start function
    spriteRenderer = GetComponent<SpriteRenderer>();

    //Gets the sprite renderer component of object.
```

```
(
[SerializeField] Color32 has_Package_Colour = new Color32(1,1,1,1); // define new
color
spriteRenderer.color = has_Package_Colour; // assign color to the object
)
```