

Execution Order Attributes

A collection of attributes to control the execution order of your scripts in Unity from your source code.

Use

Add one of these attribute to your script's class definition to change its script execution order :

- `[ExecutionOrder(<order>)]` : The script execution order is set to `order`
- `[ExecuteAfter(<type>[, orderIncrease = <increase>])]` : The script execution order is set to a value greater than the one of the script `type`, ensuring that your script will be executed after this script. By default, the script execution order is increased by 10 but this can be changed by adding the `orderIncrease` argument.
- `[ExecuteBefore(<type>[, orderDecrease = <decrease>])]` : Same as `ExecuteAfter` except that the execution order will be lowered, ensuring that your script will be executed before this script.

A script can have multiple `ExecuteAfter` attributes and will be executed after all the scripts given in parameters. It is the same for the `ExecuteBefore` attributes except it will be executed before all the scripts given in parameters.

A script can only have one type of these attributes and cannot mix the `ExecutionOrder`, `ExecuteAfter` and `ExecuteBefore` attributes.

Changing the execution order of a script with one of these attributes in the *Script Execution Order* inspector from Unity will have no effect. The order will be reset to the one defined by the attributes.

Changing a script execution order, either from the inspector or with an attribute, will cause a recompilation of the scripts.

Example

```
// Let's say that this script has an execution order of 50
// set from the Script Execution Order inspector in Unity
public class Script1 : MonoBehaviour {
    void Start () {
        Debug.Log("Second");
    }
}
```

```
// this script will have an execution order of 40 (50-10)
[ExecuteBefore(typeof(Script1))]
public class Script2 : MonoBehaviour {
    void Start () {
        Debug.Log("First");
    }
}
```

```
}
```

```
// this script will have an execution order of 100
[ExecutionOrder(100)]
public class Script3 : MonoBehaviour {
    void Start () {
        Debug.Log("Third");
    }
}
```

```
// this script will have an execution order of 120 (max(40+10, 100+20))
[ExecuteAfter(typeof(Script2)), ExecuteAfter(typeof(Script3), orderIncrease
= 20)]
public class Script4 : MonoBehaviour {
    void Start () {
        Debug.Log("Fourth");
    }
}
```

The resulting script execution orders in the inspector:

