

## Runner

**OneShot** Wraps an anonymous method in an IEnumerator.

Runs method and then continues after the defined number of seconds.

```
public class RunnerTest : MonoBehaviour
{
    private CandyCoded.Runner runner;

    private void Start()
    {
        StartCoroutine(Sequence());
    }

    private IEnumerator Sequence()
    {
        yield return runner.OneShot(() => { Debug.Log("Hello!"); }, 2.0f);
    }
}
```

Runs method and then continues on the next frame.

```
public class RunnerTest : MonoBehaviour
{
    private CandyCoded.Runner runner;

    private void Start()
    {
        StartCoroutine(Sequence());
    }

    private IEnumerator Sequence()
    {
        yield return runner.OneShot(() => { Debug.Log("Hello!"); });
    }
}
```

Used in a sequence.

```
public class RunnerTest : MonoBehaviour
{
    private CandyCoded.Runner runner;
```

```
private void Start()
{
    StartCoroutine(Sequence());
}

private IEnumerator Sequence()
{
    yield return runner.OneShot(() => { Debug.Log("Hello,"); });

    yield return new WaitForSeconds(2.0f);

    yield return runner.OneShot(() => { Debug.Log("World!"); });
}
}
```