

```
Q:\>QFSW
Q:\>Vx.y.z
```

Q>Quantum Console_

Full documentation here <https://qfsw.co.uk/docs/QC/>

- [Email](#)
- [Discord](#)
- [Issue Tracker](#)
- [Twitter](#)
- [Demo](#)

All code is kept under the `QFSW.QC` namespace. This document serves as a brief starting guide and the full documentation should be consulted when possible.

Getting Started

In order to get started as quickly as possible, simply follow the following steps:

1. Ensure the Text Mesh Pro package is installed; this is done by default for new projects
2. Add an Event System to your scene if you do not already have one `GameObject > UI > EventSystem`
3. Navigate to `Assets/Plugins/QFSW/Quantum Console/Source/Prefabs` and add the `Quantum console` prefab to your scene.

After this, you are ready to try out Quantum Console!

This prefab includes a [QuantumConsole](#) component that provides I/O to the core, autocomplete, formatting and much more. You are free however to use your own prefab with the [QuantumConsole](#) script or to make your own script entirely, as all the core functionality resides in the [QuantumConsoleProcessor](#).

1. To get started, use the `help` command. This will give you a brief intro into using the console
2. For more help about any specific command, use `man commandName` to see its user manual
3. For a list of all commands loaded by the processor, use `all-commands`

Adding Commands

When integrating Quantum Console into your project you will likely want to add your own commands to the console.

To do this, you will need to use the `[Command]` attribute.

By adding this to your function, property or field you can turn it into a command that will be loaded by the Quantum Console Processor.

Example

```
using QFSW.QC;
...

[Command]
public static int Add(int a, int b)
{
    return a + b;
}
```

Which can be invoked by entering to the console

```
Add 10 20
```

For more, see the documentation on [Commands](#)