

# Scripting

## What Is Scripting?

Most of the time when we talk about "scripting" in Ignition we are talking about Python scripting, or writing code in the Python language. Python is a general purpose programming language that was developed in the early 90s and has gained significant popularity in the 2000s. We like it because it is extremely readable, elegant, powerful, and easy to learn. As an added bonus, it gracefully interacts with Java, giving programmers an extremely powerful tool when paired with Ignition, which is written in Java.

Since Python is such a popular and well-regarded language, there are many high-quality tutorials available on the web. The official Python tutorial, written by the inventor of Python himself, Guido van Rossum, is a great reference: <https://docs.python.org/2/tutorial/index.html>. In Ignition, you will be mixing the core Python language with references to other components and a variety of our built-in **system** functions.

### On this page

...

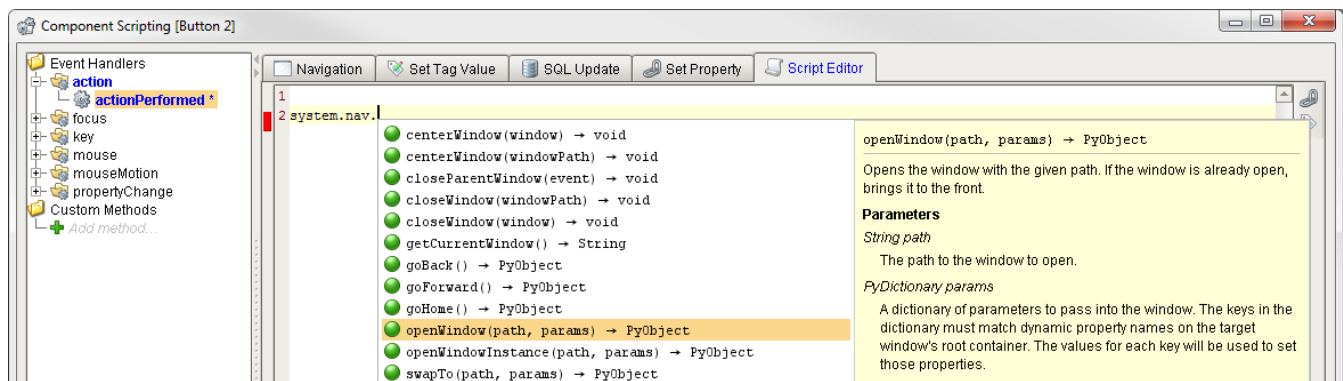
- What Is Scripting?
- Is It Easy to Use?
- Where Is Scripting Used?

### Pseudocode - Interacting With a Component

```
text = event.source.parent.getComponent('My Component').text
if text == "":
    print "You must enter a name"
```

## Is It Easy to Use?

Luckily, Python is a simple language to get started with, and using it in an event-driven system takes away a lot of the extra code that normally makes programming time consuming. For those that are already familiar with scripting (and those of you that are learning), we also have a huge list of functions inside Ignition to do some of the common tasks in a single line of code. These **System Functions** are available while typing. Just start with "**system**" and press "ctrl-space" to see a list of available functions. The list will filter itself as you continue typing.



## Where Is Scripting Used?

Python is used in many places in Ignition. Each location has its own events that trigger your scripts to run, and add functionality to your projects in different ways.

- **Components** - Add actions to components like buttons, customize the look and feel of charts and tables, and even setup a custom navigation schema.
- **Tags** - Create a script that runs on a Tag change, when an alarm goes active, and more!
- **Reports** - Use scripting to create a customized datasource, or create your own unique action to use with the scheduling system.
- **Alarm Notification** - Create custom rosters using scripting to dynamically change who gets notified with each new alarm event.
- **Client and Gateway** - Add a script that will run when certain events happen, such as when the Client or Gateway starts up, or on certain keystrokes.

Users that are new to Ignition focus mainly on the **Component Binding system**, and for good reason. It's simple, flexible, and generally easy to understand without much of a computer background. However, Ignition has a complete scripting system built into every place you can think of. Using it is not a requirement, but it can add a significant degree of flexibility and customization to your projects. It allows you to create exactly what you need, giving you total control where pre-canned options fall short.

The majority of your scripting will be done in Event Handlers inside of components. This system makes it very easy to get started scripting with little to no experience. With the four [script builders](#), basic scripting like Navigation and setting Tag values takes just a few clicks!

#### Related Topics ...

- [Getting Started with Scripting in Ignition](#)

#### In This Section ...