# BIPSCRIPT BASIC WORKSHOP

*John Hammen*

bipscript.org
Portland, Oregon, USA
`jhammen@j2page.com`

## ABSTRACT

Bipscript is a scripting language for music creation. The Bipscript Basic Workshop is a two hour guided session that gives a quick introduction to the language and its main uses. This workshop assumes basic scripting experience on the part of attendees but does not assume any prior knowledge of bipscript. Core topics include working with instrument and effects plugins, MIDI and OSC messages. More advanced topics can be addressed if time permits. Software prerequisites are the latest bipscript and the CALF plugin pack, participants should bring a portable computer with these installed.

## 1. INTRODUCTION

Bipscript is a scripting language designed for creating interactive sequenced music. It is based on the Squirrel scripting language, augmented with a class library of objects to manipulate audio-related events such as MIDI and OSC messages or the extracted features of an audio stream.

## 2. PREREQUISITES

### 2.1. Software

Participants should install the following software packages before the start of the workshop:

- bipscript
- Calf Studio Gear plugins[1]
- jack-keyboard[2]

The latest version of the bipscript interpreter application is available at bipscript.org, source code and binary install packages are available for several versions of Ubuntu Linux as well as Arch Linux and its derivatives.

The Calf plugins and the jack-keyboard virtual MIDI keyboard are both free and open source software and available as packages in many common GNU/Linux distributions.

### 2.2. Hardware

Any hardware that can run the above is suitable, this would typically be a laptop but other form factors such as netbook, raspberry pi etc. are fine as long as they support the software prerequisites.

[1] https://calf-studio-gear.org/

[2] http://jack-keyboard.sourceforge.net/

## 3. CORE TOPICS

### 3.1. Language and API Overview

This section of the workshop will give a quick introduction to the language and syntax as well as the API and its documentation.

### 3.2. Working with Instrument Plugins

An example script will be created by each participant, showing how to create an instance of a synthesizer or sampler plugin and connect it to system MIDI inputs and audio outputs. This section of the workshop also serves as a test of attendees' software and hardware configuration.

### 3.3. Working with Effects Plugins

Effects plugins will be added to the signal chain of the example script, participants will learn how to control plugin parameters.

### 3.4. Handling OSC Messages

OSC ("Open Sound Control") functionality will be added to the script with demonstration of how to receive and handle incoming OSC messages.

### 3.5. MIDI Sequencing

Various ways of creating and MIDI patterns will shown, these patterns will be scheduled to play on an instrument plugin to demonstrate MIDI sequencing.

## 4. ADVANCED TOPICS

More advanced topics can be addressed based on the interests of the participants, as time permits:

### 4.1. Advanced Timing

This optional section is a demonstration of using timing other than the default clocks. For example the script can become the timebase master in order to control the system transport and allow synchronization with other applications.

### 4.2. JIT scheduling

This section gives an introduction to just-in-time scheduling functionality, showing how incoming events of various types can affect outgoing MIDI or OSC sequences in real time.