

Houdini: Tool Development



Brief description :

Houdini Technical Director specialization course is aimed at training Houdini specialists responsible for the technical side of the Pipeline in the studio. Using knowledge in various fields, solving all sorts of technical problems, programming tools, Technical Director is the link between technologies and artists.

The course analyzes a variety of issues: from the beginning of the development of an instrument, selection of the approach and the method to the features and subtleties in solving of various tasks. A lot of code, practical advice and a large amount of Python.

The course will be useful for:

- Those who want to work as technical specialists, engaged in Pipeline construction, instruments creation, technology development, etc.
- Houdini VFX artists, especially FX TD.
- All Houdini artists, who want to explore Python and improve the quality and productivity of their work.

Introduction

- TD'S Tools and utils for daily work
- Version control system (Git/Mercurial)
- Cygwin/Terminal
- Text editor
- IDE

- - Quick look at Houdini configuration and pipeline setup
- - Folder structure
- - Config files and environment variables
- - Initialisation and startup scripts

- - Extending houdini:
- - Hscript and Python scripting
- - VEX operators
- - C++ HDK plugins

- - Code VS Nodes. Speed VS flexibility and ease of use

Python

- - Why learn Python.
- - Benefits of dynamic language
- - The Zen of Python. Batteries included

- - Which Python version to choose and why?
- - Working interactively in Python interpreter.
- - Useful tools and alternative python shells

- - Build-in command and functions
- - User defined function:
- - Passing arguments
- - Return values
- - Local and Global variables

- - Python data types:
- - Integer, float and boolean types.
- - Strings:
- - Strings methods
- - String formatting syntax
- - Unicode strings

- - Containers:
- - Lists, Tuples, Sets
- - Methods, sorting, use case examples

- Dictionary:
 - Methods, iteration, copying
 - Use cases
-
- Loops, conditional statements
 - Iterators and Generators:
 - Generators explained
 - Creating custom Iterator/generator
-
- List comprehensions syntax
 - Working with files:
 - Reading and writing data to file
-
- Modules and Packages:
 - What is module?
 - Working with modules. Importing, reloading, customizing
 - How to create and organize modules?
 - Where python can find your modules
 - Namespaces in python
 - Organizing modules into a package
-
- Object Oriented Programming in Python:
 - Basic OOP concepts, class members and attributes
 - Examples
-
- Starting other system process from Python:
 - Communicating with process
 - Stopping, killing process, waiting for complete.

Python In Houdini

General use

- Python integration overview. What can we do with python in houdini.
- Custom Python operators.
- Introduction to HOM:

- - hou module and submodules
- - Working with nodes, parameters and geometry
- - Learn python scripting in practice. Couple of small and useful scripts
- - Working with shelves. toolutils, soptoolutils modules
- - Python in parameter expressions

- - Python in HDA - ultimate power:
- - Event handlers
- - Parameter callbacks
- - HDA Sections

- - Python and Mantra rendering:
- - SOHO Basics. Custom rendering properties
- - Python filters and callbacks

- - Exception handling. Houdini specific python errors.
- - Hython.

Python SOP - full control over geometry:

- - hou.Geometry class overview.
- - Point, Vertex and Prim classes.
- - Geometry attributes and groups - reading, creating and modifying.
- - Iterating over geometry entries (points, prims, vertices)
- - Using interruptible operation context.
- - How fast your code is? Code profiling with timeit and cProfile modules.
- - Code optimisation tips.
- - Calling VEX functions within Python.
- - InlineCPP - C++ speed in Python SOP.
- - Various Python operators examples.
- - Creating powerful geometry cache asset using Python.
- - Creating Python DOP Solver.
- - Creating Python OBJ Operator.

HDA - Advanced uses and preferences:

- - Advanced HDA uses and preferences.
- - OTL search paths.
- - Operators versioning - different approaches.
- - Namespaces. Operator loading priorities.
- - Packing and unpacking otl file.
- - Help card for HDA.

- - Storing arbitrary data in HDA sections.
- - Storing user data in operator for cache or exchange.

User Interfaces:

- - Using Python to create and modify operator parameters with ParmTemplates.
- - Access houdini interface from python.
- - Interacting with user with hou.ui module.
- - Running PyQt application inside houdini.
- - PyQt basics. Writing simple dialog.
- - Houdini event loop. Use cases.

Vex

- - General information. Language overview.Contexts.
- - VOP Networks vs code.
- - Using vex in inlineVOP.
- - Wranglers.
- - Compiling vex code with VCC compiler.
- - Writing simple surface shader in vex.

HDK

- - Introduction to HDK.
- - When and why use C++ plugins.
- - Visual Studio setup for compiling houdini files.
- - Setup Cygwin for easy compilation and testing.
- - Learning HDK concepts - general classes, inheritance.
- - H12 geometry concepts.
- - Writing simple HScript command for counting number of prims in scene.
- - In-Depth explanation of SOP operators concepts:
 - - General classes
 - - Cooking inputs
 - - Geometry operations
 - - Primitive and point groups
 - - Working with attributes
 - - Geometry iteration
 - - Error handling
- - Writing expression function - to find max attribute value.
- - Writing simple VEX function to manipulate strings.
- - Writing SOP operators:
 - - OP for computing primitive centroids
 - - OP for creating offscreen attributes and groups

- - Writing SOP operator which imports data from DOP network.
- - Writing VRAY Mantra Procedurals - simple examples.

Information about the course



Lecturer: Alexey Rusev (Sadrutdinov)

Course result: Certificate and knowledge, scripts and plugins that we will write during the training.

Course duration: 4 months

Price: \$500 per month

Course: Houdini: Tool Development

Автор: Administrator
21.06.2011 00:00

Type of course: online

The class will consist of weekly lecture videos containing 1-2 integrated questions per video. You'll be emailed with a download link each Sunday or Monday. You can ask any question in Skype chat any time during this course.

Requirements for listeners entering the course:

Good knowledge of Houdini at 3d Artist level. Basic knowledge of programming. Desire to study.

Start date: January 24, 2016

Registration for the course: opened

[SEND REQUEST](#)

[Tool Development \(Russin group\)](#)

{youtube}poUMmApjB7k{/youtube}

{youtube}JHjm-zfP6sM{/youtube}