esri

Esri International Developer Summit

Palm Springs, CA

Creating Geoprocessing Tools in a Python Toolbox

David Wynne

Abstract - Creating Geoprocessing Tools in a Python Toolbox

- Python toolboxes offer a way to efficiently build custom geoprocessing tools entirely in Python.
- A Python toolbox and the tools within look and act just like toolboxes and tools created in any other way.
- Join me as we step through examples of creating Python toolbox tools from scratch.

http://www.esri.com/events/devsummit/session-rater

Why we create tools

- *Tools solve problems*

- Tools are:
  - Easy to share
  - Generic
  - Becomes part of the geoprocessing framework
    - Python, Dialogs, ModelBuilder, Services
  - Basic validation saves coding checks

Python toolbox tool structure

A tool does 3 types of work:

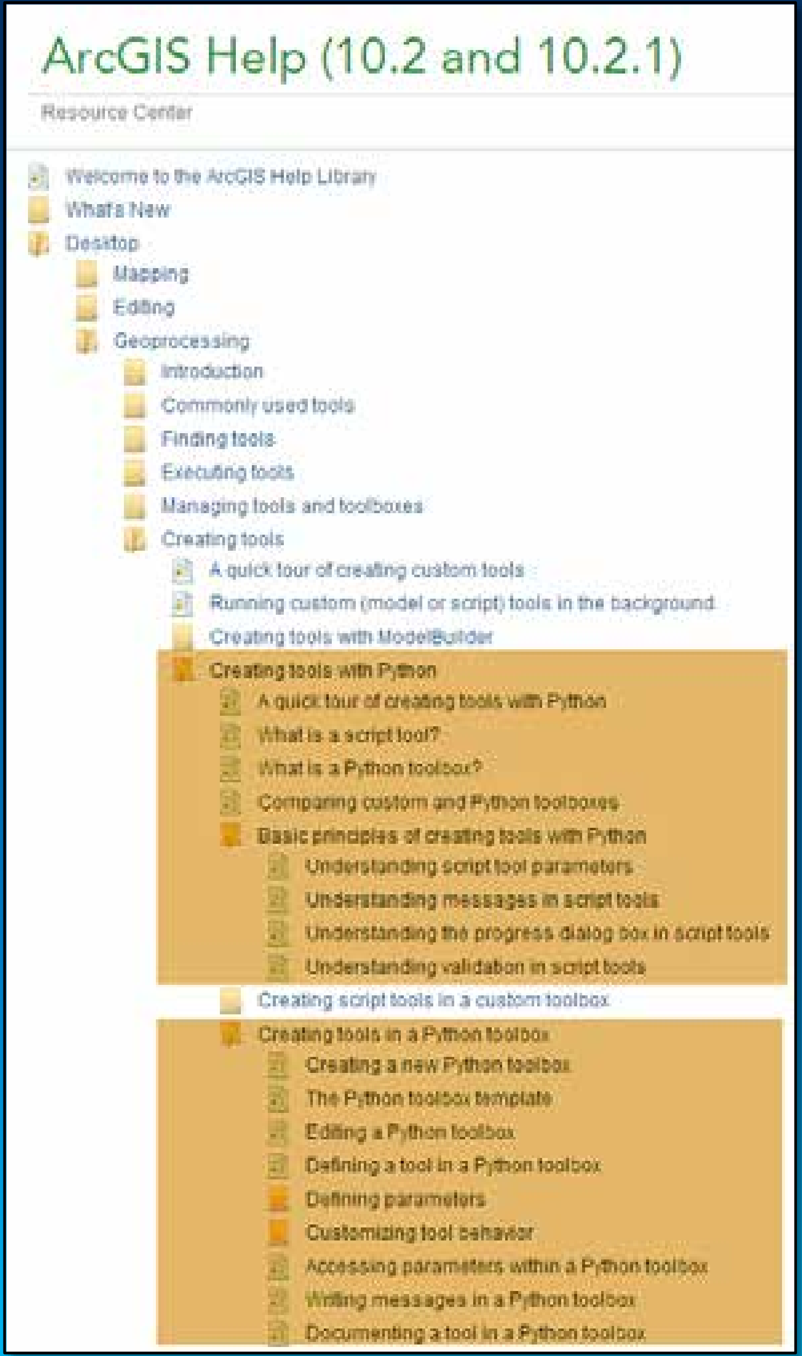
- Parameters
- Validation
- Source

```
class Tool(object):
    def __init__(self):
        """Define the tool (tool name is the name of the class)."""
        self.label = "Tool"
        self.description = ""
        self.canRunInBackground = False

    def getParameterInfo(self):
        """Define parameter definitions"""
        params = None
        return params

    def isLicensed(self):
        """Set whether tool is licensed to execute."""
        return True

    def updateParameters(self, parameters):
        """Modify the values and properties of parameters before internal
        validation is performed.  This method is called whenever a parameter
        has been changed."""
        return

    def updateMessages(self, parameters):
        """Modify the messages created by internal validation for each tool
        parameter.  This method is called after internal validation."""
        return

    def execute(self, parameters, messages):
        """The source code of the tool."""
        return
```

Creating a Python toolbox

Getting more information

- esriurl.com/pythontoolbox



Understanding our world.