Python - Report Lambdas

Report Lambda Cheatset

Voxel Farm Python - Report Lambda Cheatset

VOXEL FARM

Typical sections in a Report Lambda

```
import voxelfarm as vf
```

Get Inputs

Load Voxels

Combine Voxels

Compute & Output

Getting inputs that were passed to the lambda

Inputs allow the calling program to provide parameters to a Lambda execution

```
entity = vf.input_entity(id, label, type)

double = vf.input(id, label, default)
str = vf.input_string(id, label, default)
bool = vf.input_bool(id, label, default)
```

Input Entity Types

```
vf.type.voxel_terrain
vf.type.block_model
vf.type.point_cloud
vf.type.voxel_generator
vf.type.mesh
vf.type.voxel_mesh
vf.type.drill_holes
vf.type.indexed_ortho
vf.type.voxel_ortho
vf.type.geo_chem
```

Loading voxel data and its attributes

```
voxels = vf.load_voxels(entity, attr, custom_attr)
```

Native Attributes

```
vf.type.none
vf.type.volume
```

Custom Attributes

A string containing comma-separated attribute identifiers

* Combine multiple types using binary OR operator

Vanilla Python (no Pandas)

```
volume_total = 0.0
tonnes_total = 0.0
for v in vf.voxels:
  volume = vf.get_volume(voxels, v)
  grade = vf.get_attribute(voxels, v, 'fe')
  volume_total += volume
  tonnes_total += volume * grade * D

sum_volume = vf.start_sum('Object', 'Volume')
sum_tonnes = vf.start_sum('Object', 'Tonnes')
vf.sum(sum_volume, volume_total)
vf.sum(sum_tonnes, tonnes_total)
```

	Volume	*Tonnes*
Object		

Voxel Operations - Volumetric Booleans and Voxel Stacks

```
voxels = vf.voxels_union(voxelsA, voxelsB, attr, custom_attr)
voxels = vf.voxels_complement(voxelsA, voxelsB, attr, custom_attr)
voxels = vf.voxels_intersection(voxelsA, voxelsB, attr, custom_attr)

voxels = vf.start_composite()
vf.add_layer(voxels, voxelsA, mode)
vf.add_layer(voxels, voxelsB, mode)
vf.add_layer(voxels, voxelsC, mode)
```

Composition Modes

```
vf.layer_merge_mode.none
vf.layer_merge_mode.oldest
vf.layer_merge_mode.newest
vf.layer_merge_mode.min
vf.layer_merge_mode.max
```

Using Pandas

```
df_model = vf.load_pandas_dataframe(voxels)
df_model['tonnes'] = df_model['fe'] * df_model['volume'] * D
df_model = df_model[['volume','tonnes']]
result = df_model.sum(axis=0)
vf.sum_pandas_result('report', result)
```

	Volume
Tonnes	

report.csv

Voxel Joins

```
df_A.merge(df_B,
  left_on='voxel_id',
  right_on='voxel_id')
```

* Join data frames using *voxel_id*

Reading Results Back

```
vf.download_file(
  projectId, entityId,
  'report.csv', local_file)
```

Define Report Program

This example creates a new report program.

```
result = vf.create_lambda_python(
    project=project,
    type=vf.lambda_type.Report,
    name="My Volume Report Lambda",
    fields={},
    code=open('./lambdas/ComputeVolumeLambda.py').read())
if not result.success:
    print(result.error_info)
    exit()
report_lambda_id = result.id
```

Run Report Program

The following example runs an existing Report Program with a given set of inputs.

```
# We will use a region that defines an 160 meter cube centered in the origin
region = '1,-80,80,-80,-80,80,-80,80,80,-80,80'
result = vf.create_report(
    project=project,
    program=report_lambda_id,
    region=region,
    lod=0,
    name='Volume of Cube',
    fields={},
    inputs={
        'entity' : voxmesh_id
    })
if not result.success:
    print(result.error_info)
    exit()
report_id = result.id
```

For more information on regions and how to trigger report programs, please check out the Triggering Reports topic.

Read Report Results

This example shows how to read the results of a Report Lambda execution.

```
# The "project" variable contains the Project ID
# The "report_id" variable contains the ID for the report entity
csvdata = vf.get_file(project, report_id, 'report.csv')
```

Initialize Report Lambda

This example shows how to initialize the Voxel Farm library in a report Lambda program.

```
import voxelfarm as vf
```

Read inputs of Report Lambda

The following example shows how to read inputs from a report lambda program.

```
mesh_entity = vf.input_entity("mesh_entity_id", "Entity", vf.type.voxel_mesh)
string_input = vf.input_string("string_input", "String Input", "")
numerical_input = vf.input("numerical_input", "Numerical Input", 0.0)
```

Volumetric Booleans in Report

This example shows how to create volumetric boolean operations between volumetric datasets.

```
# Get inputs, in this case the entities that will have they volumes computed
entityA = vf.input_entity("entityA", "Select Entity A", vf.type.voxel_generator | vf.type.voxel_terrain |
vf.type.voxel_mesh)
entityB = vf.input_entity("entityB", "Select Entity B", vf.type.voxel_generator | vf.type.voxel_terrain |
vf.type.voxel_mesh)

# Next, we will request to load voxel data for the entity
# We are only interested in volume, we will request no other attributes for voxels
voxelsA = vf.load_voxels(entityA, vf.attribute.volume, None)
voxelsB = vf.load_voxels(entityB, vf.attribute.volume, None)

voxels_union = vf.voxels_union(voxelsA, voxelsB, vf.attribute.volume, None)
voxels_intersection = vf.voxels_intersection(voxelsA, voxelsB, vf.attribute.volume, None)
voxels_complementAB = vf.voxels_complement(voxelsA, voxelsB, vf.attribute.volume, None)
voxels_complementBA = vf.voxels_complement(voxelsB, voxelsA, vf.attribute.volume, None)
```

Load and Write Pandas

This example shows how to create a Pandas dataframe from voxels inside a Report Lambda program and output a different dataframe as a result.

```
import pandas
import voxelfarm as vf

entity = vf.input_entity("entity", "Select Entity", vf.type.block_model)
voxels = vf.load_voxels(entity, vf.attribute.volume, 'fe, density')

df_model = vf.load_pandas_dataframe(voxels)
df_model['tonnes'] = df_model['fe'] * df_model['volume'] * df_model['density']
df_model = df_model['volume', 'tonnes']
df_output = df_model.sum(axis=0)
vf.sum_pandas_result('report', df_output)
```