Fluid

- Create a cube and set it to physics → fluid → domain
- Create an object inside cube and set it to physics → fluid → fluid
- On domain click to physics → bake button

/lab/blender/fluid

bake.py

```python
import bpy
import sys
from bpy.app.handlers import persistent



#@persistent
def do_bake():
    for scene in bpy.data.scenes:
        for object in scene.objects:
            for modifier in object.modifiers:
                if modifier.type == 'FLUID_SIMULATION':
                    if modifier.settings.type == 'DOMAIN':
                        bpy.ops.fluid.bake({'scene': scene, 'active_object':
object})

                        break



#@persistent
def do_render_opengl():
    bpy.ops.render.opengl(animation=True, view_context=False)
    bpy.ops.wm.quit_blender()


print("BAKE")
do_bake()
print("RENDER")
do_render_opengl()
```

```python
import bpy
import sys
from bpy.app.handlers import persistent



#@persistent
def do_bake():
    for scene in bpy.data.scenes:
```

```python
        for object in scene.objects:
            for modifier in object.modifiers:
                if modifier.type == 'FLUID_SIMULATION':
                    if modifier.settings.type == 'DOMAIN':
                        bpy.ops.fluid.bake({'scene': scene, 'active_object':
object})
                        break



#@persistent
def do_render_opengl():
    bpy.ops.render.opengl(animation=True, view_context=False)
    bpy.ops.wm.quit_blender()


print("BAKE")
do_bake()
print("RENDER")
do_render_opengl()

#bpy.app.handlers.load_post.append(do_bake)
#bpy.app.handlers.load_post.append(do_render_opengl)
```