Step 1: Walk through ReadMe (install stuff)

- ReadMe: https://github.com/OpenMined/OpenMined
- **If on Mac OSX - Turn on Metal Support**

Step 2: Turn on Metal and Background Process

- Run Bash Commands:
  - `git clone https://github.com/OpenMined/PySyft.git`
  - `cd PySyft`
  - `python setup.py clean install` (might need sudo)
    - You can run `python setup.py develop` after the first install to prevent you from needing to restart the notebook in OpenMined to uptake your local changes.
    - if any file dependency error regarding mpc and mpfr appears on Mac, you might want to check: https://stackoverflow.com/questions/23187801/installing-gmpy-on-osx-mpc-h-not-found/27663446).
  - `cd ../`
  - `git clone https://github.com/OpenMined/OpenMined.git`
  - `cd OpenMined`
  - `jupyter notebook` (might need to install jupyter)
- Start Unity App
- location (download and install free ver if you don't have it: Windows/Mac: https://unity3d.com/get-unity/download and Linux: http://beta.unity3d.com/download/ee86734cf592/public_download.html)
- Within Unity editor select: Edit -> Project Settings -> Player
- In the Inspector, scroll down until you find "Metal Editor Support" under "Other Settings" and click the checkbox to turn it on. (skip this if not on Mac OSX)
- In the Inspector, expand "Resolution" and check the box for "Run in Background"

Step 3: Get Current Float Tensor Working

- Open OpenMined/OpenMined in Unity Application er
- In the Project Pane, Double Click Assets/_Scenes/DefaultScene
- Click "Play" button at the top
- Open the Notebook Syft Tensor Example Notebook
- Run the Notebook Cells

Step 4: Add new function to Float Tensor (this example "abs()" function)

- Open PySyft/syft/tensor.py

- (the simple version is in this picture - although we also have helper functions if you look closely... many people use those instead ... look for “params_func”)

```
def abs(self):
    command = {}
    command["functionCall"] = "abs"
    command["objectType"] = "tensor"
    command["objectIndex"] = self.id
    command["tensorIndexParams"] = []

    self.controller.socket.send_json(command) # sends the command
    return self.controller.socket.recv_string() # receives output from command
```

tx

- Open https://github.com/OpenMined/OpenMined/blob/master/UnityProject/Assets/OpenMined/Syft/Tensor/FloatTensor.cs
- Add an CASE statement to check for the “functionCall” string you initialized in your JSON command object like so.

```
switch (msgObj.functionCall)
{
case "abs":
        {
                // calls the function on our tensor object
                this.Abs ();
                // returns the function call name with the OK status
                return msgObj.functionCall + ": OK";
        }
case "add_":
        {
```

- Open https://github.com/OpenMined/OpenMined/blob/master/UnityProject/Assets/OpenMined/Syft/Tensor/FloatTensor.MutatingOps.cs to add the “Abs_()” function to our FloatTensor class.

```
public FloatTensor Abs()
{
        if (dataOnGpu) {
                // GPU Absolute Value Code Here
        } else {
                for (int i = 0; i < size; i++) {
                        if (data [i] < 0) {
                                data [i] = -data [i];
                        } else {
                                // do nothing
                        }
                }
        }
        return this;
}
```

- At this point - it should work in your Python Notebook for Tensors that are on the CPU. You should test it out in the Notebook (don't forget to python setup.py install pysyft)!!!

```
In [1]: from syft.syft import FloatTensor, SyftController
        import numpy as np

In [2]: # connect
        # identity = 'worker-%d' % (choice([0,1,2,3,4,5,6,7,8,9]))
        identity = 'worker-0'
        sc = SyftController(identity)

In [3]: # create tensors

        l0_tensor = sc.FloatTensor([0,0,-1,0,1,1,1,0,1,1,1,1], [3,4])

        FloatTensor.__init__: 0

In [4]: l0_tensor.abs()

Out[4]: 'abs: OK'

In [5]: l0_tensor.print()

        0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1

In [ ]:
```

Step 5: Add GPU Support for your Function

- Open OpenMined/Assets/OpenMined/Syft/Math/Shaders/FloatTensorShaders.compute to add the function

```
#pragma kernel AbsMain

RWStructuredBuffer<float> abs_data;

[numthreads(1,1,1)]
void AbsMain (uint3 id : SV_DispatchThreadID) {
    if(abs_data[id.x] < 0) {
        abs_data[id.x] = -abs_data[id.x];
    }
}
```

- Open OpenMined/Assets/OpenMined/Syft/Tensor/FloatTensor.ShaderOps.cs to add the field for the new function and save shaders and kernels

```
        }

        public void AbsGPU_() {
            if (dataOnGpu) {
                shader.SetBuffer (Abs_Kernel, "abs_data", dataBuffer);
                shader.Dispatch (Abs_Kernel, this.size, 1, 1);
            }
        }

        public void MulScalarGPU(float value)
```

- Open OpenMined/Assets/OpenMined/Syft/Tensor/FloatTensor.Ops.cs to call the GPU function if data is on GPU

```
public FloatTensor Abs_()
{
    if (dataOnGpu)
    {
        // GPU Absolute Value Code Here
        AbsGPU_();
    }
    else
    {
        for (int i = 0; i < size; i++)
        {
            if (Data[i] < 0)
            {
                Data[i] = -Data[i];
            }
        }
    }
    return this;
}
```

Step 6: Write & Run unit tests

- Open OpenMined/OpenMined/Assets/OpenMined.Tests/Editor/FloatTensorTest.cs
- Add unit tests
- Open Unity

- Open TestRunner (Window -> TestRunner)
- Select unit test and “run selected”
- Fdsfdsffsd <- what is this part for…?