

# Learning digital fabrication by yourself

Ko Tsuruta, Zurich, 1st, November, 2021

---

**Preface:** This is a list of learning materials which I used during MAS2021. First of all, do not try to follow or read all of the materials, and do not feel stressed even if you cannot understand all of them because they were customized for my own learning process and I haven't fully understood yet either. Also, you will get amazing lectures from experts through the MAS programme depending on the research topics. So there is no need to read everything beforehand, and I hope you can find your own learning materials. Second, in general, when you learn code, keep in mind that programming is not magic. Someone invented, designed and is maintaining it in its long history. It will take some time to learn how to read, write and communicate with programming languages. Stay calm and Write codes until you can encode what you want. Finally, you can create something nice even if you cannot code. The quality of code does not guarantee the quality of design.

---

**Note:** Before Starting your journey, it's nice to have an overview of the context. It's possible to start without this knowledge. However, I prepared this note for those who feel awkward when you cannot understand behind the scenes. Again, no need to be perfectionist at the beginning.

[How do our code work in computers?](#)

And then, what and why [python](#), [rhino](#) and [grasshopper](#)?

---

## For Beginners

1. **Introduction to RH&GH (Rhino and Grasshopper):** For those who are new to RH/GH
  - 1.1. [Digital in Architecture](#) : overview of basic knowledge for digital fabrication
  - 1.2. [Plethora Project for RH](#) : Nurbs modeling in rhino
  - 1.3. [Plethora Project for RH/GH](#) : Basic Grasshopper tutorials
  - 1.4. [Plethora Project for python in RH/GH](#)
2. **Introduction to mathematical theories:** For those who want to learn theories behind programming.
  - 2.1. [Essence of Linear Algebra](#) : What is the concept of vector...?
  - 2.2. [Essence of calculus](#) : Derivatives / Integration
3. **Introduction to python:** For those who want to learn what python is from a programmer perspective.
  - 3.1. [Introduction to python](#) : no need to understand everything, but helpful to have an overview.
  - 3.2. [PEP8](#) : For those who want to write clean code in python.

---

Before diving into elementary...

[How to read an error message](#) : Go to stack overflow to find an answer

[How to use RhinoCommon API](#) : This is C# though helpful to understand the concept

---

## For Elementaries

1. **Essential mathematics for computational design:** programming in RH/GH
  - 1.1. [Essential mathematics](#)
  - 1.2. [Essential guide to C# scripting](#) : For those who have pre-knowledge of C#
2. **Detailed RH/GH documentations**
  - 2.1. [The why and how of data trees](#) : detailed explanation of data tree in GH
  - 2.2. [Rhino Python Primer](#) : detailed explanation of nurbs and python in rhino
3. **The Nature of Code:** how to simulate natural phenomena in digital using "class".
  - 3.1. [NOC for python in GH](#)
  - 3.2. [The Nature of Code](#) : This is an original version for processing.

---

*Before diving into intermediate...*

[Understanding Directory Structure in Windows.](#)

[What is the difference between Module, Package, Library and Framework?](#)

[What](#) and [Why](#) Anaconda?

---

## *For Intermediate*

### 1. **COMPAS tutorials**

1.1. [COMPAS tutorials](#): A python framework for digital fabrication

1.2. [Anaconda environment control](#): a package manager for python

### 2. **Data transfer from one to the other through json file**

2.1. [What is a json file?](#)

2.2. [How to read and write json using python](#)

---

*Before diving into advanced...*

Materials are topic specific from this level. I just list up what I watched or read in 2020-2021 based on my interests and during MAS-thesis with Simon Griffioen. Try some if you have interest or search for your own learning materials!

---

## *For Advanced*

### 1. **COMPAS extensions (could be intermediate)**

1.1. [COMPAS extensions](#)

### 2. **Documents around computation**

2.1. [Two Minutes Papers](#)

2.2. [How to Think Like a Computer Scientist](#)

### 3. **Introduction to Machine Learning**

3.1. [An introduction to Machine Intelligence](#)

3.2. [Newral Networks by 3Blue1Brown](#)

### 4. **Data processing with OpenCV and Open3d**

4.1. [OpenCV Basic tutorial](#)

4.2. [OpenCV website](#)

4.3. [Open3d website](#)

### 5. **How to create python package**

5.1. [Packaging python](#)

### 6. **Introduction to Robotics**

6.1. [ROS](#)

6.2. [Moveit!](#)

---

*For more contribution in a long coding journey...*

[Why opensource?](#)

[How to control versions with semantic versioning](#)

[How to choose a lisenace for your work](#)

---