

# Boxes.py

## More than a Box generator

Florian Festi  
2014/03/15

Mainframe  
Hackspace Oldenburg

# Outline

- Brief overview
- Showing existing objects
- How to use and adjust
- Software architecture and how to use it
- Outlook

# Motivation

- Boxes for the Lasercutter
- Drawing (finger) joints in CAD is a drag
- Online box generators only allow fixed styles
- No flexible opensource generators found

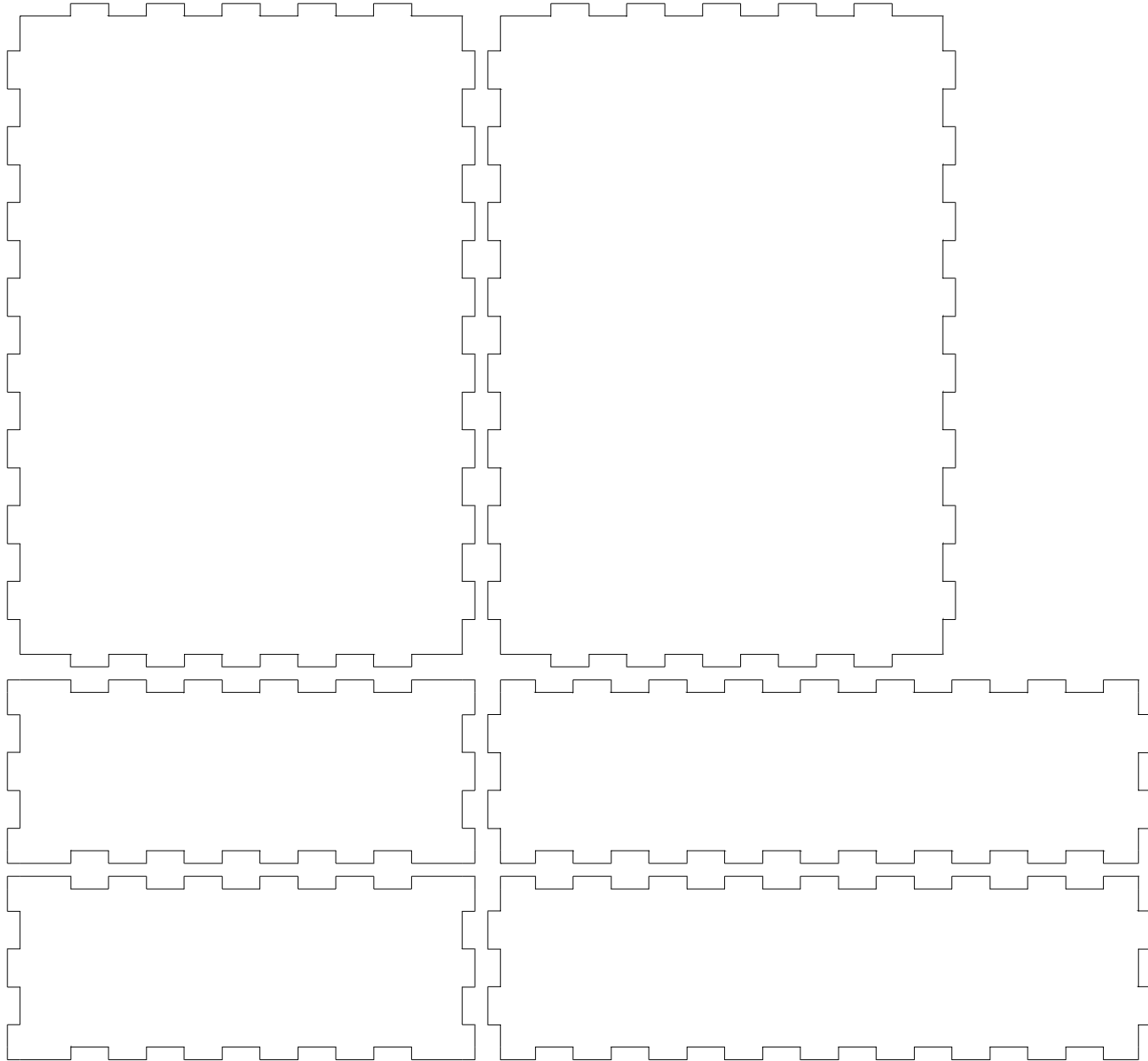
# What?

- Box generator
- Written in Python
- Creates SVG
- Several supported models
- Library for own creations
- Supports flex, finger and flat dovetail joints
- Can be use to create more complicated things than a simple box

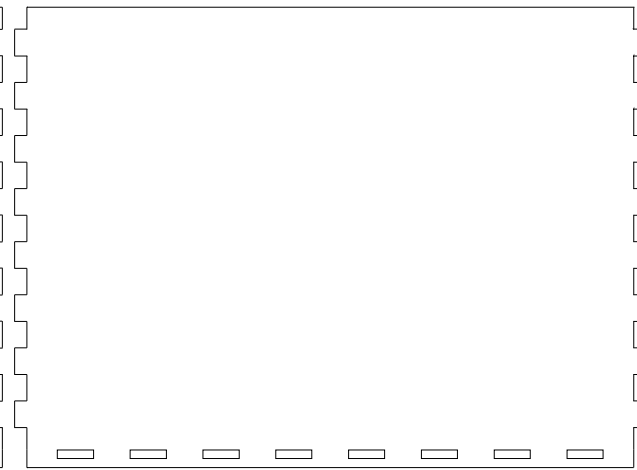
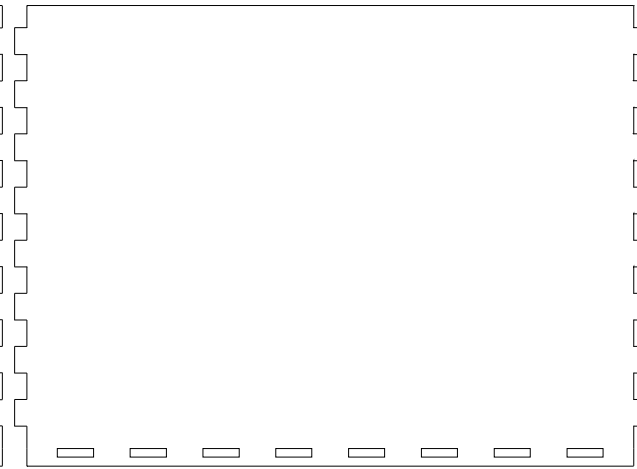
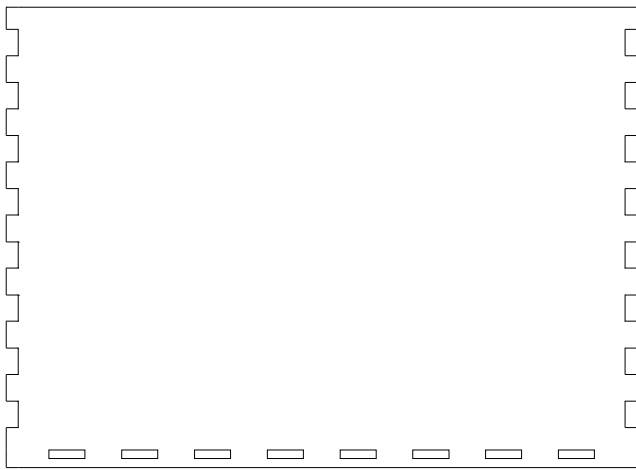
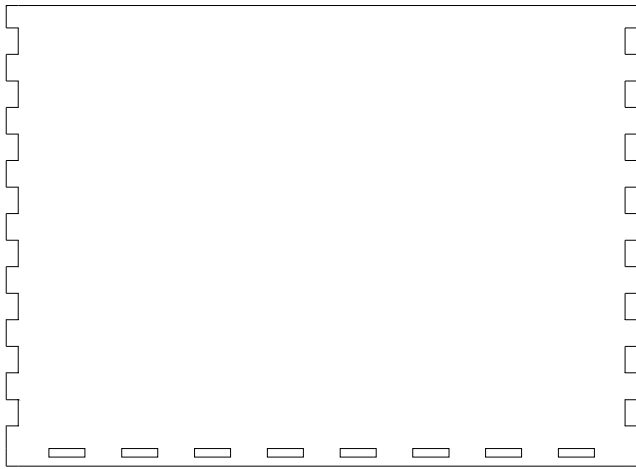
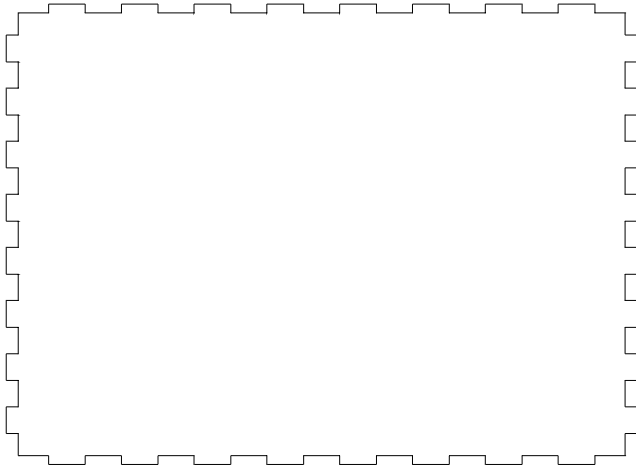
# Features

- Flex
- Finger joints
- Dovetail joints (flat only)
- Holes in honey comb pattern
- Several finished Models

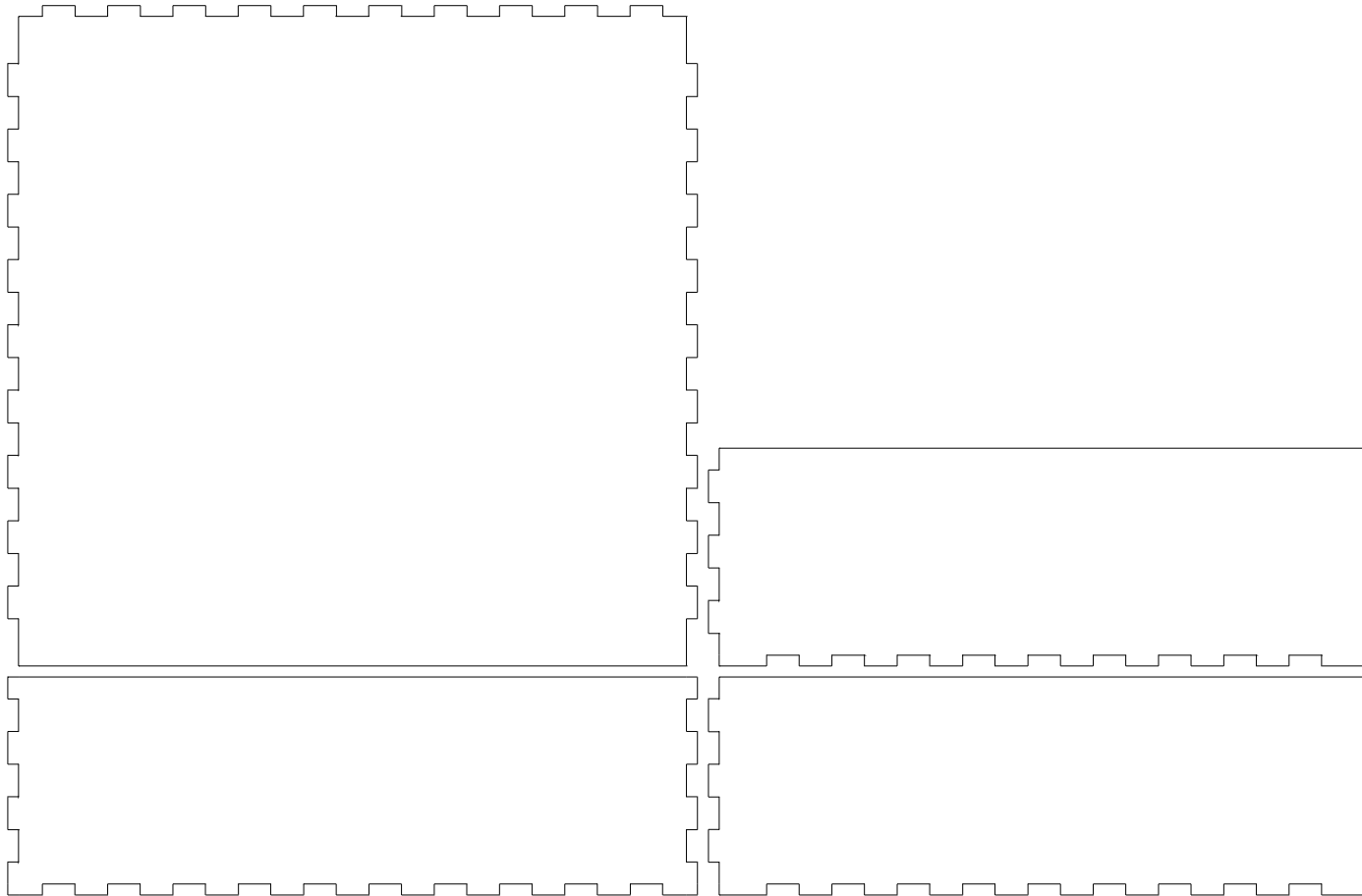
# Box.py



# Box2.py

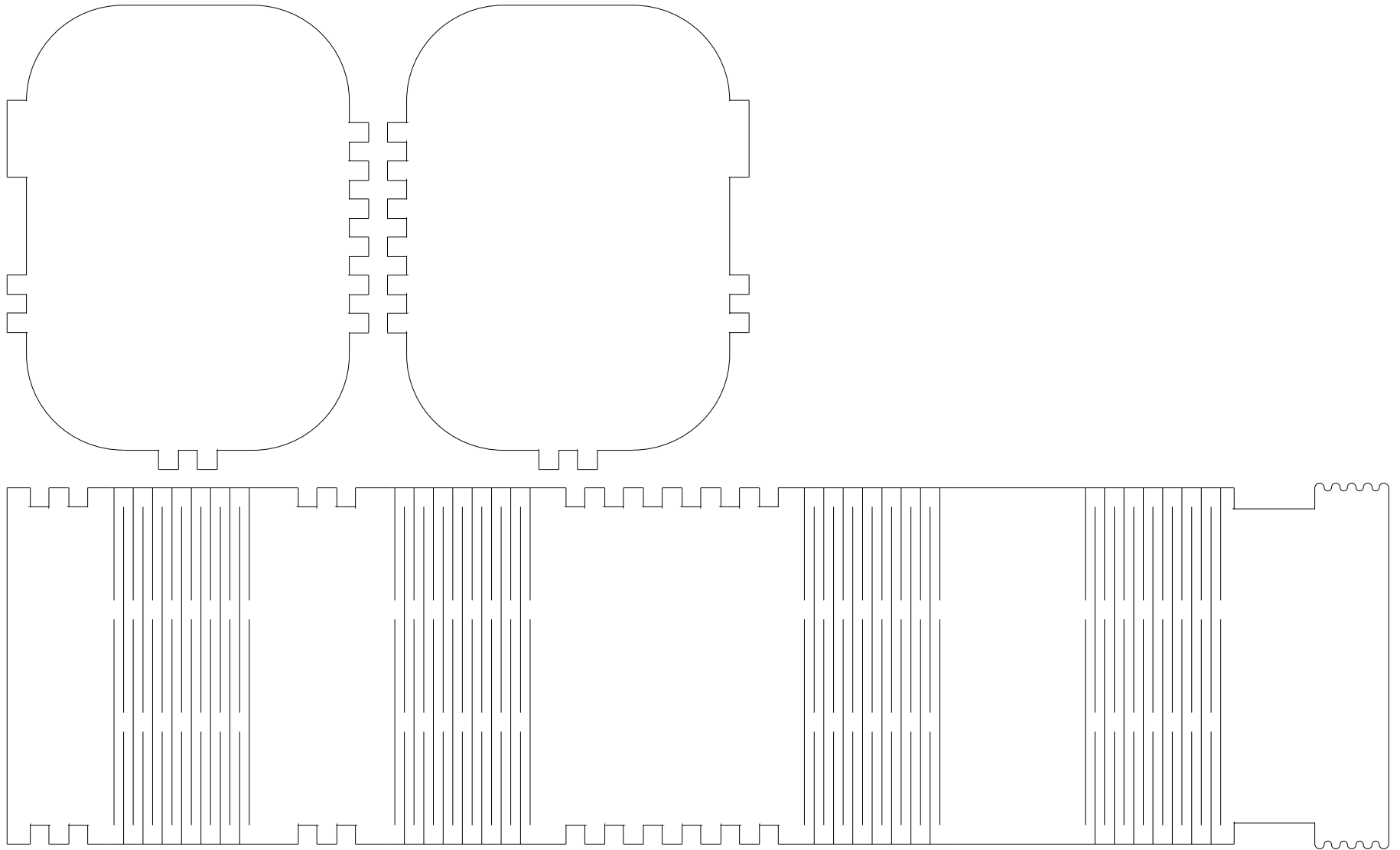


# Box3.py

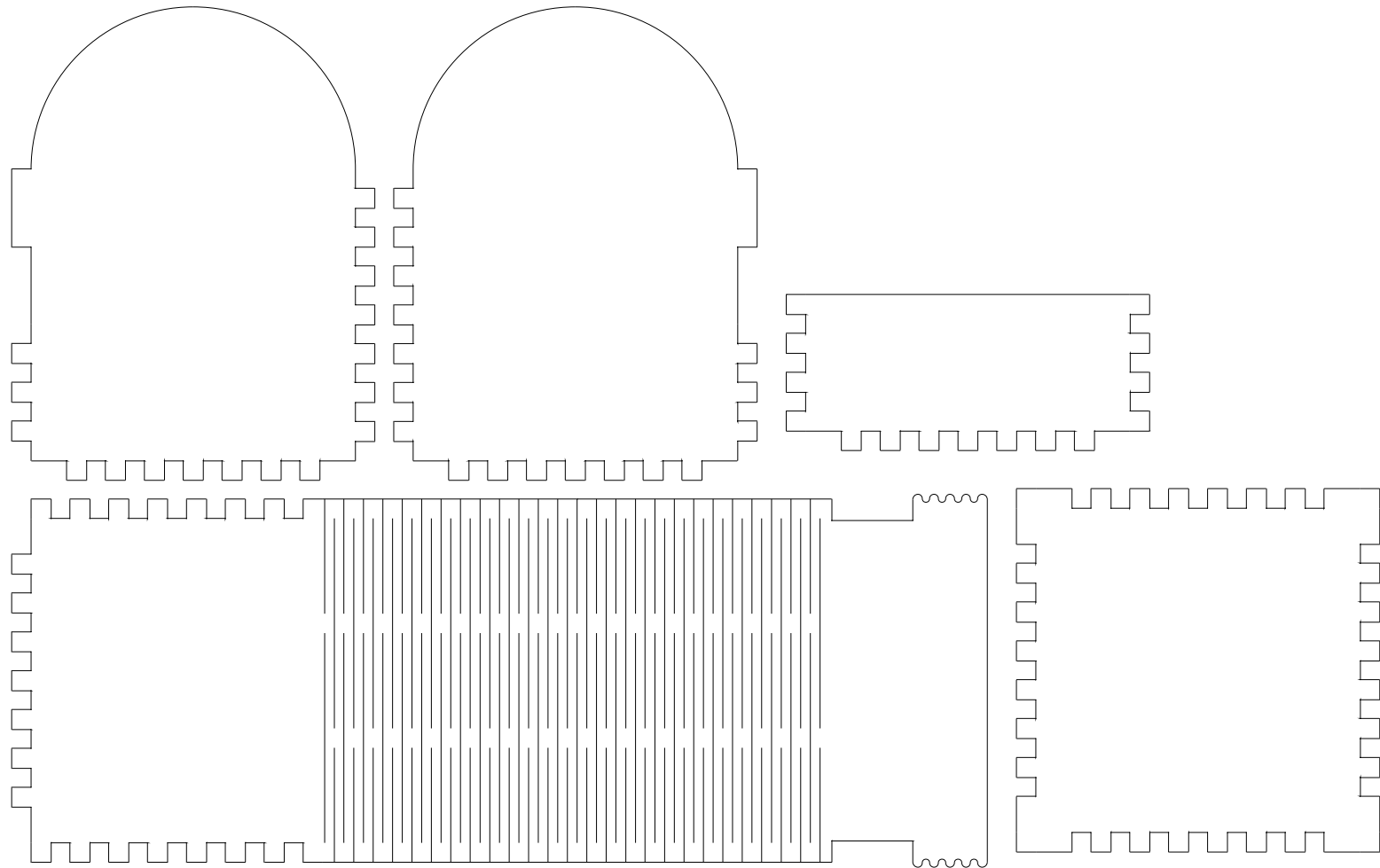




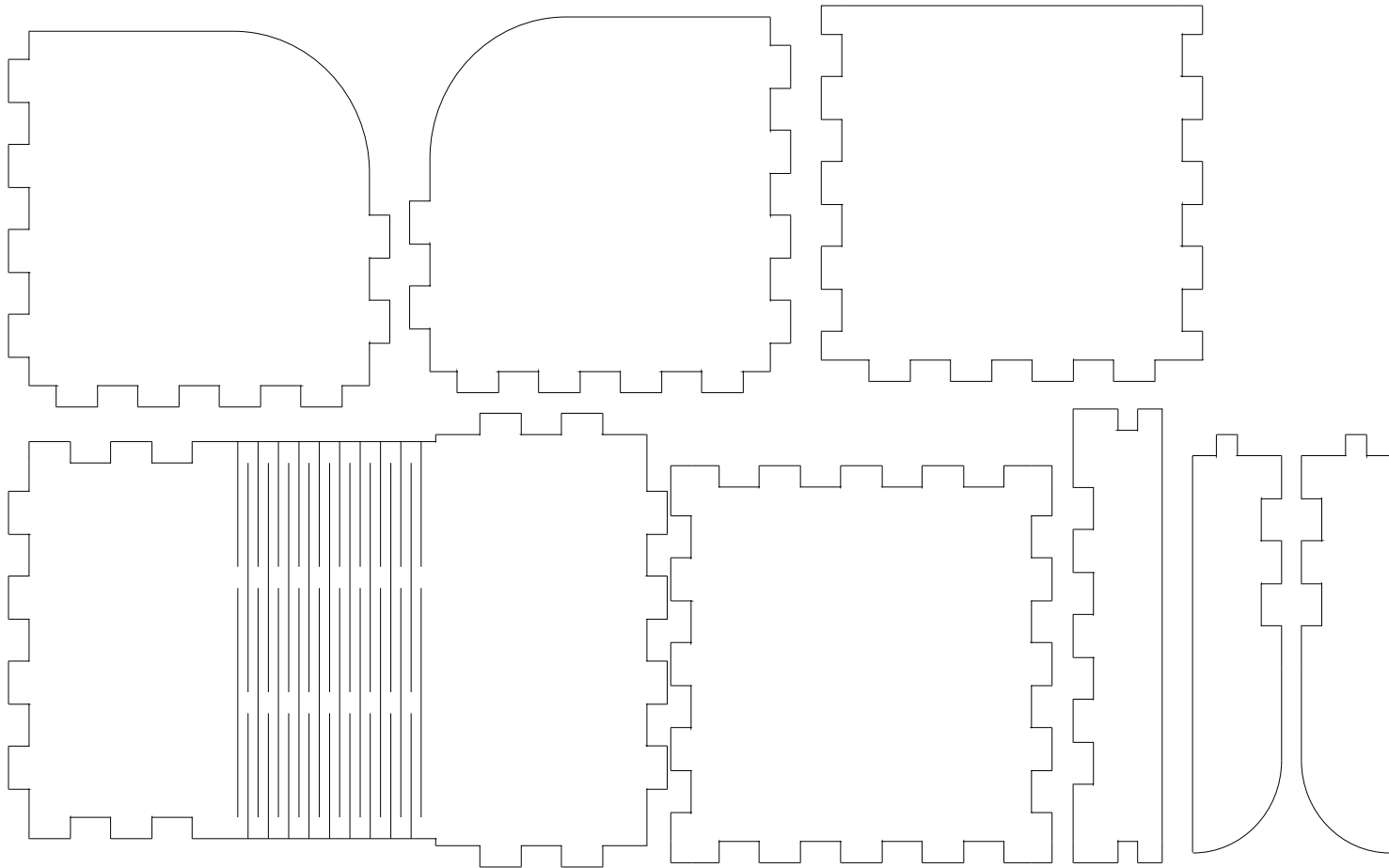
# Flexbox.py



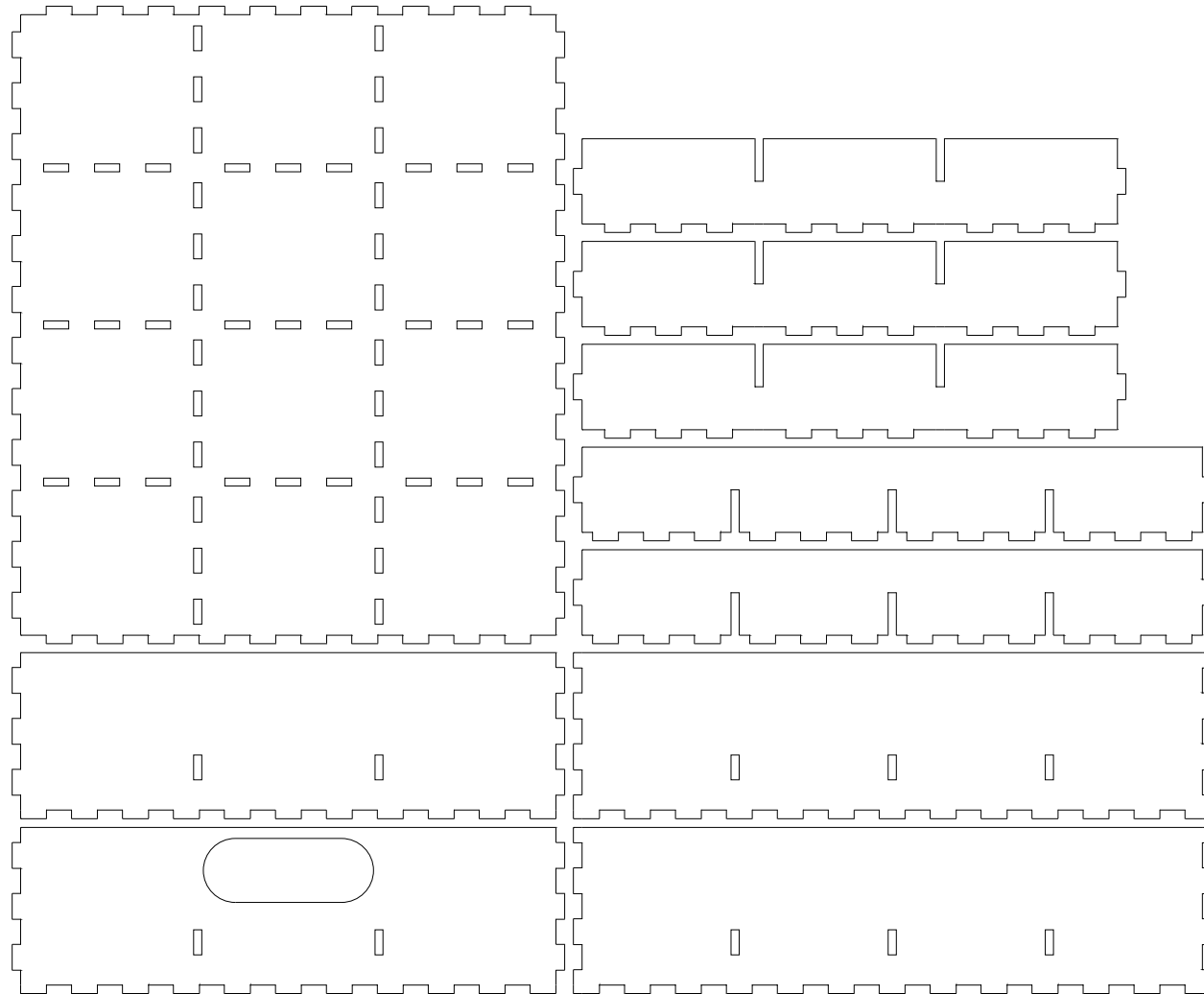
# Flexbox2.py



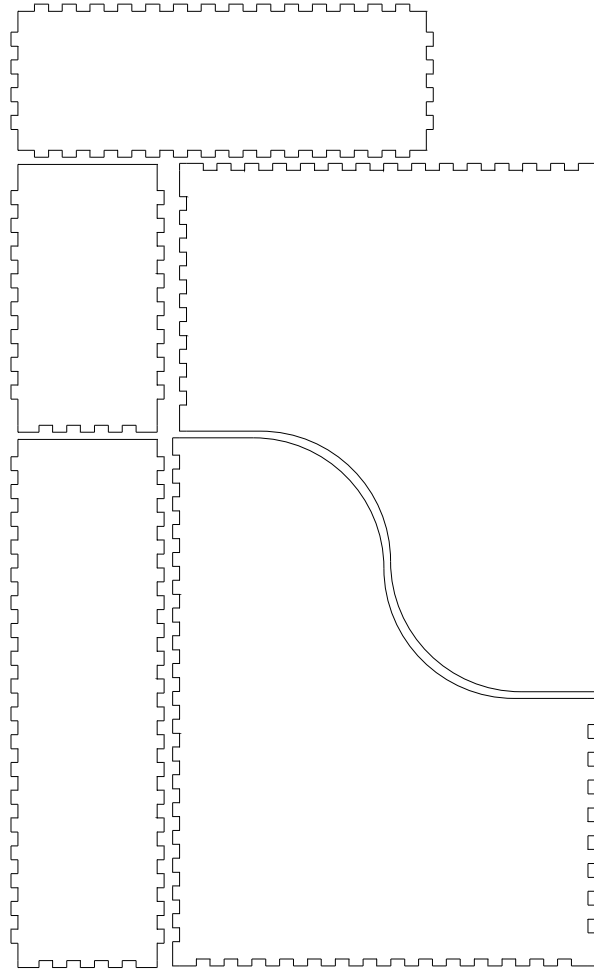
# Flexbox3.py



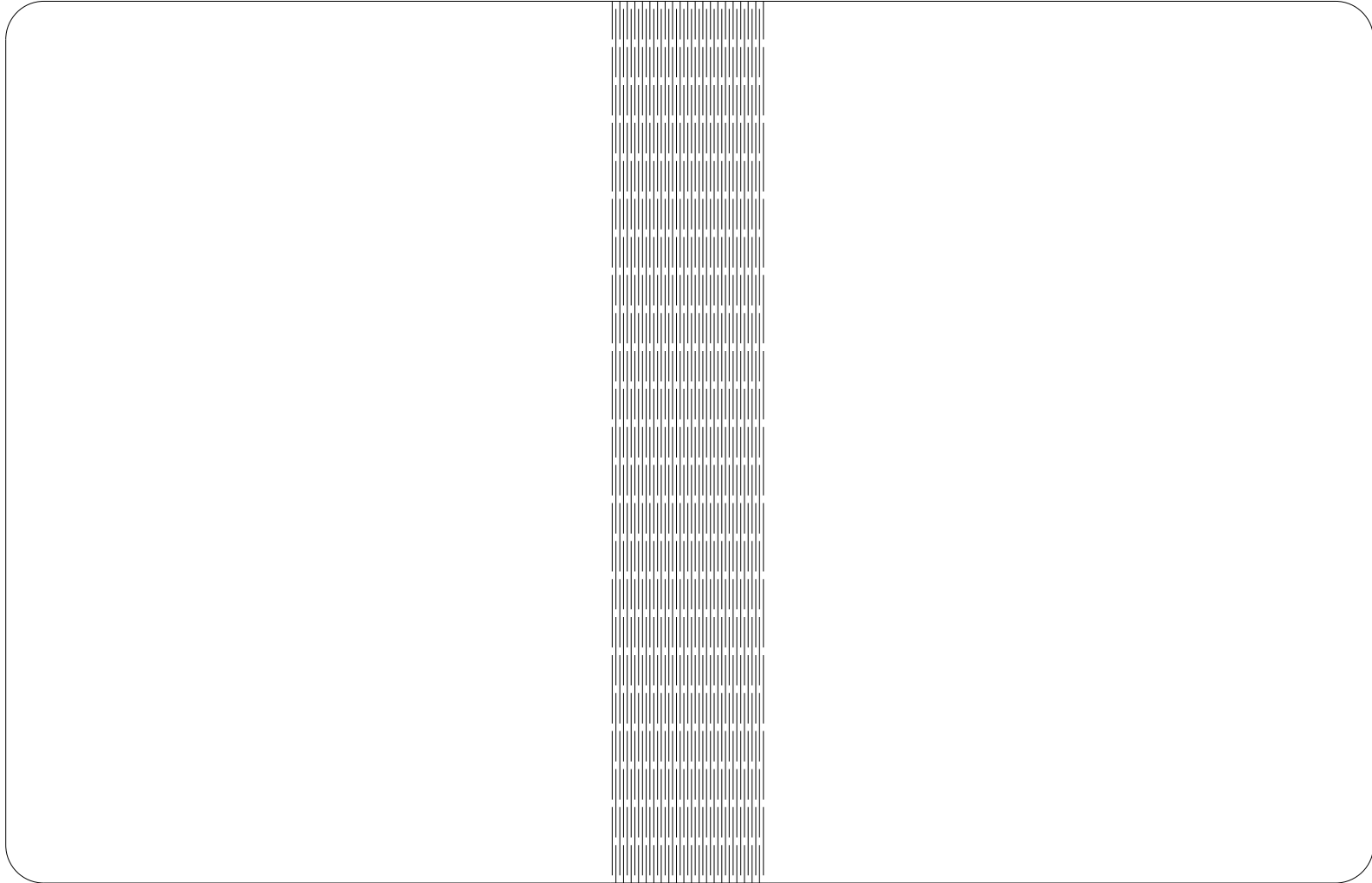
# Typetray



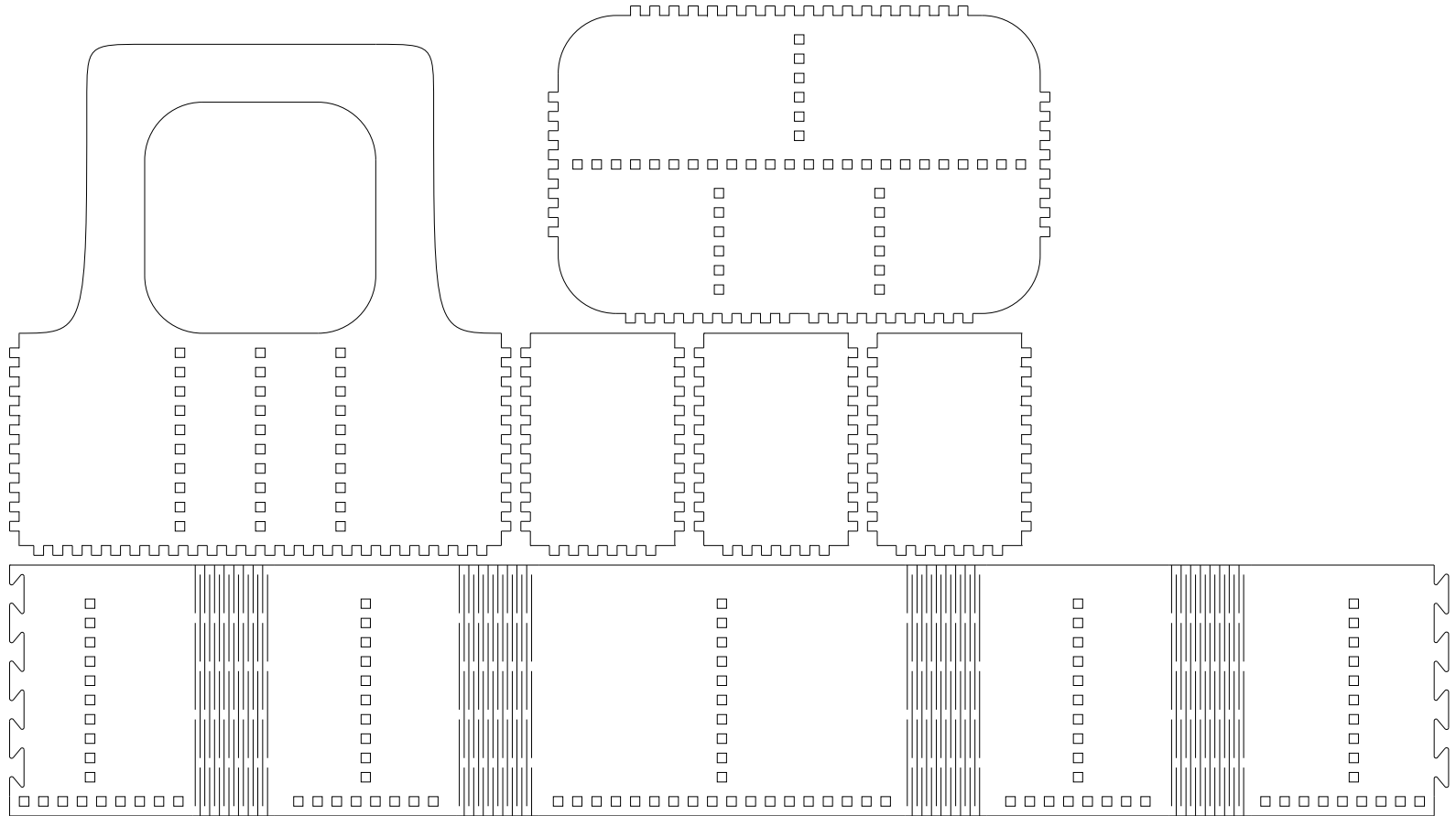
# Magazinefile.py



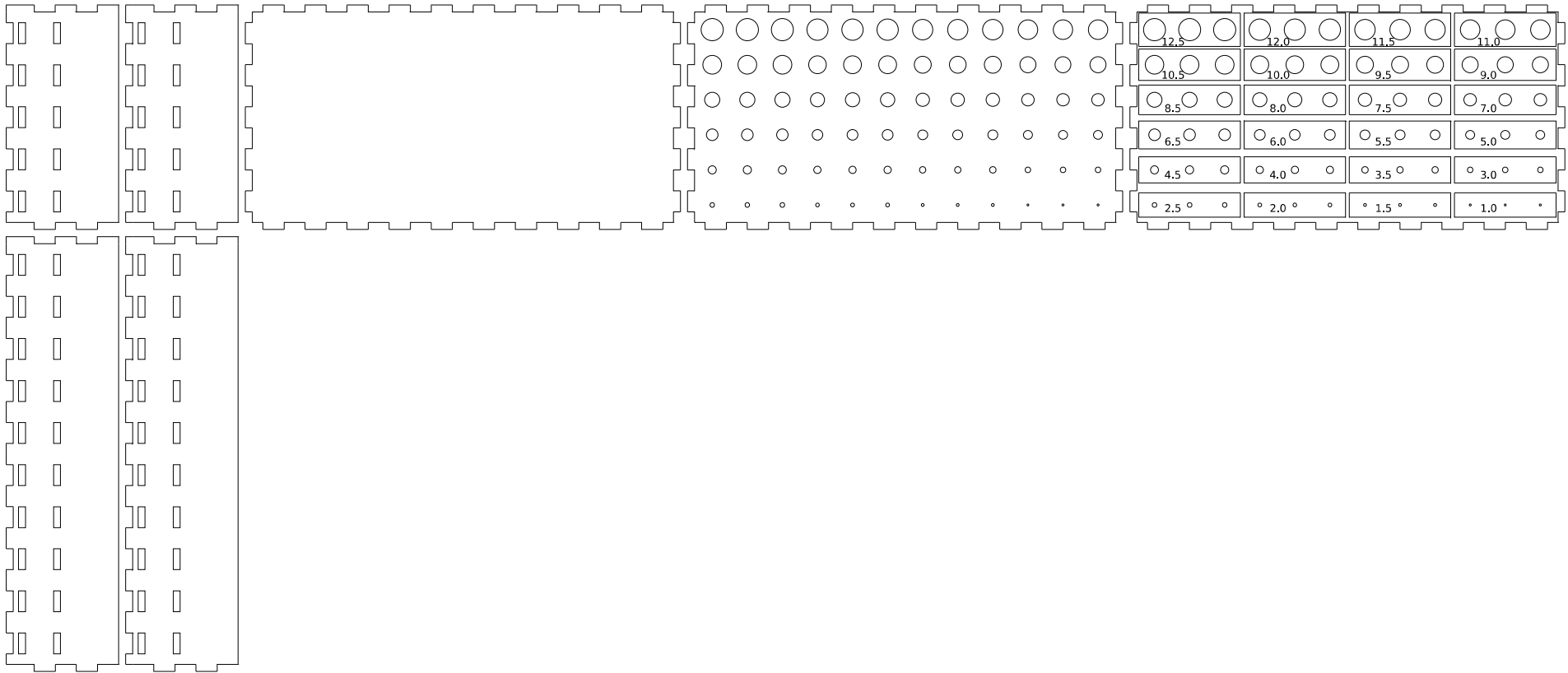
# Folder.py



# Silverwarebox

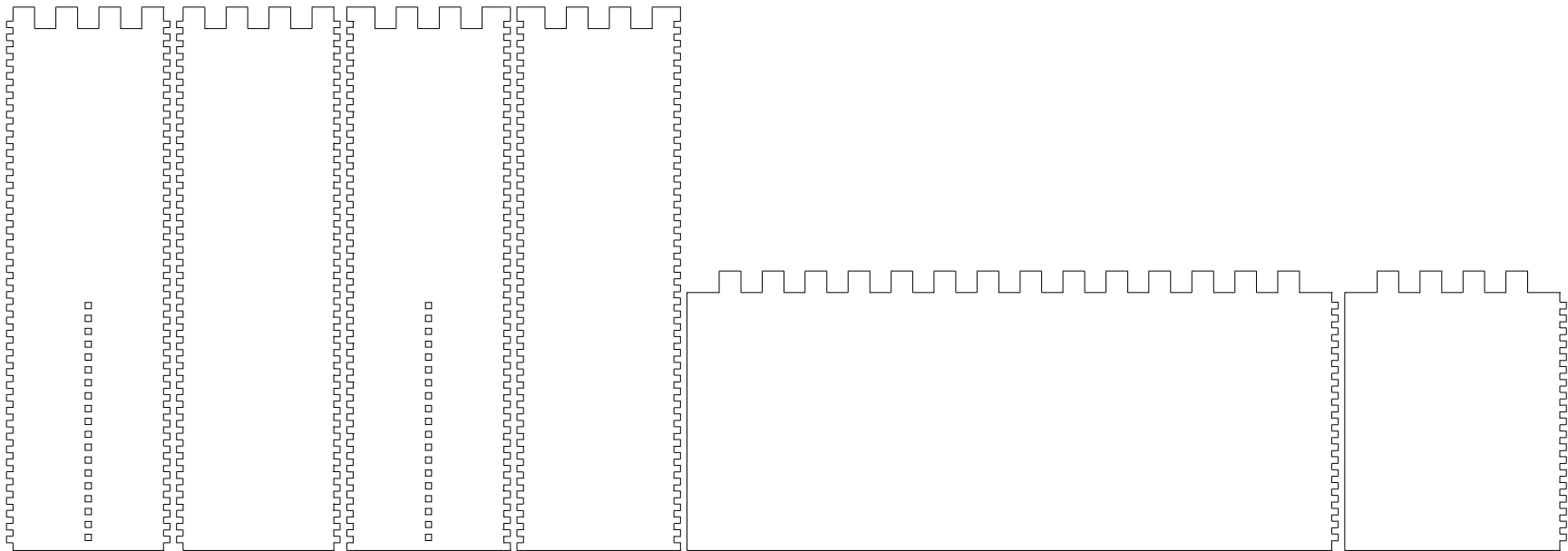


# Drillbox.py

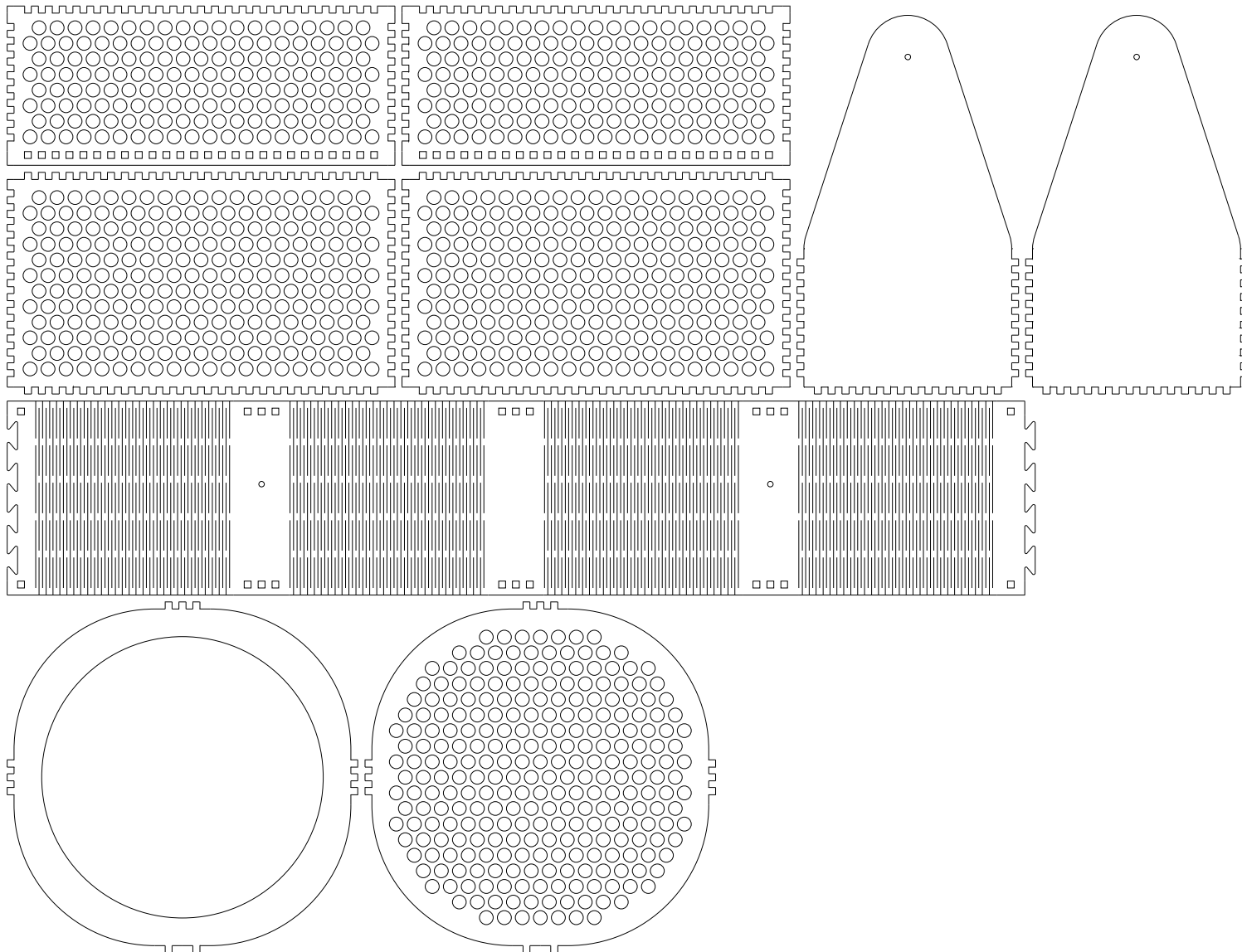




# Castle.py



# Lamp.py



# Getting started

- Get from <https://github.com/florianfesti/boxes>
  - Git clone
  - Or download
- Requires Python 2 or 3
  - Defaults to `/usr/bin/python`
- And Pycairo
  - Package `pycairo`, `python-cairo` or `python3-cairo`
  - Windows install instructions still wanted

# Using boxes.py

- Select script you want to use
  - You don't want boxes.py
- Adjust sizes and parameters
- Execute script
- Find result in box.svg

# Adjusting Sizes

- Skip to the end of the file
- Edit params given
- E.g. in Flexbox2.py:

```
if __name__=="__main__":  
    b = FlexBox(50, 70, 50, r=25,  
                thickness=3.0)  
    b.render()
```

# Typical Parameters

- $x, y, z, h$ : inner sizes
  - Often  $x, y$  is the side view and  $z$  width
- $r$ : radius of corners
- thickness: strength of the wood
  - Influences how joints are made
- burn: correction for the beam width
  - This is the radius of the beam
  - the amount of the offset at one side

# Fingerjoint Settings

```
b.edges["f"].settings.setValues(  
b.thickness, space=3, finger=3,  
surroundingspaces=1)
```

- Space: width in between fingers
- Finger: width of fingers
- Surroundingspaces: width at the end of the edge
- All in multiples of thickness
  - as passed as first param

# Using the boxes Framework

- Several layers of abstraction
- Base class Boxes in boxes.py
- Scripts sub class Boxes



# Primitives and Building blocks

- Cairo primitives (on `self.ctx`)
- Building blocks
  - Get coordinates passed to them
  - Text and all kind of holes
- `self.moveTo(x, y, dir)`
  - Change coordinate system to given point
- `self.ctx.save()`
- `self.ctx.restore()`
  - To return to previous positions

# Turtle Graphics Commands

- Edges and corners
- Start at 0, 0 heading right
- Move coordinate system to their endpoint
- And their end orientation
- Part is above the X Axis
  - Go to right and turn mathematically positive
  - to get a closed part
- Automatically do burn correction

# Edges

- Classes on their own
- Separate Settings classes
- Allow sharing of settings among both sides
- `Boxes.edges[char]` and `Boxes.name`
- `width`: outset needed to begin this edge
- `margin`: additional space needed (e.g. fingers)

# Parts

- Walls and pieces
  - `rectangularWall()`
  - `roundedPlate()`
  - `surroundingWall()`
- move parameter
  - right, left up, down, only
- Callback for all sides
  - To put holes and other building blocks
  - Pass either single function or list of callables

# Edges parameter

- Iterable with characters or Edge objects
- e, E: straight edge, normal and outset
- f, F: Finger joints
- h: Straight edge with holes for finger joint
- d, D: Dovetails joints
- X: Flex edge
- You can register your own edges
- Or pass them as Objects

# Still missing

- Command line interface
- Smarter output file name (scriptname + params)
- Support for Edges other than  $90^\circ$ 
  - Switch lamp.py over to hexagonal head
- Cleaning stuff up
  
- Patches welcome!

Questions?