



# Media Computing Project

Python and Fusion 360 API

Prof. Dr. Jan Borchers  
M.Sc. René Schäfer



**RWTH**AACHEN  
UNIVERSITY

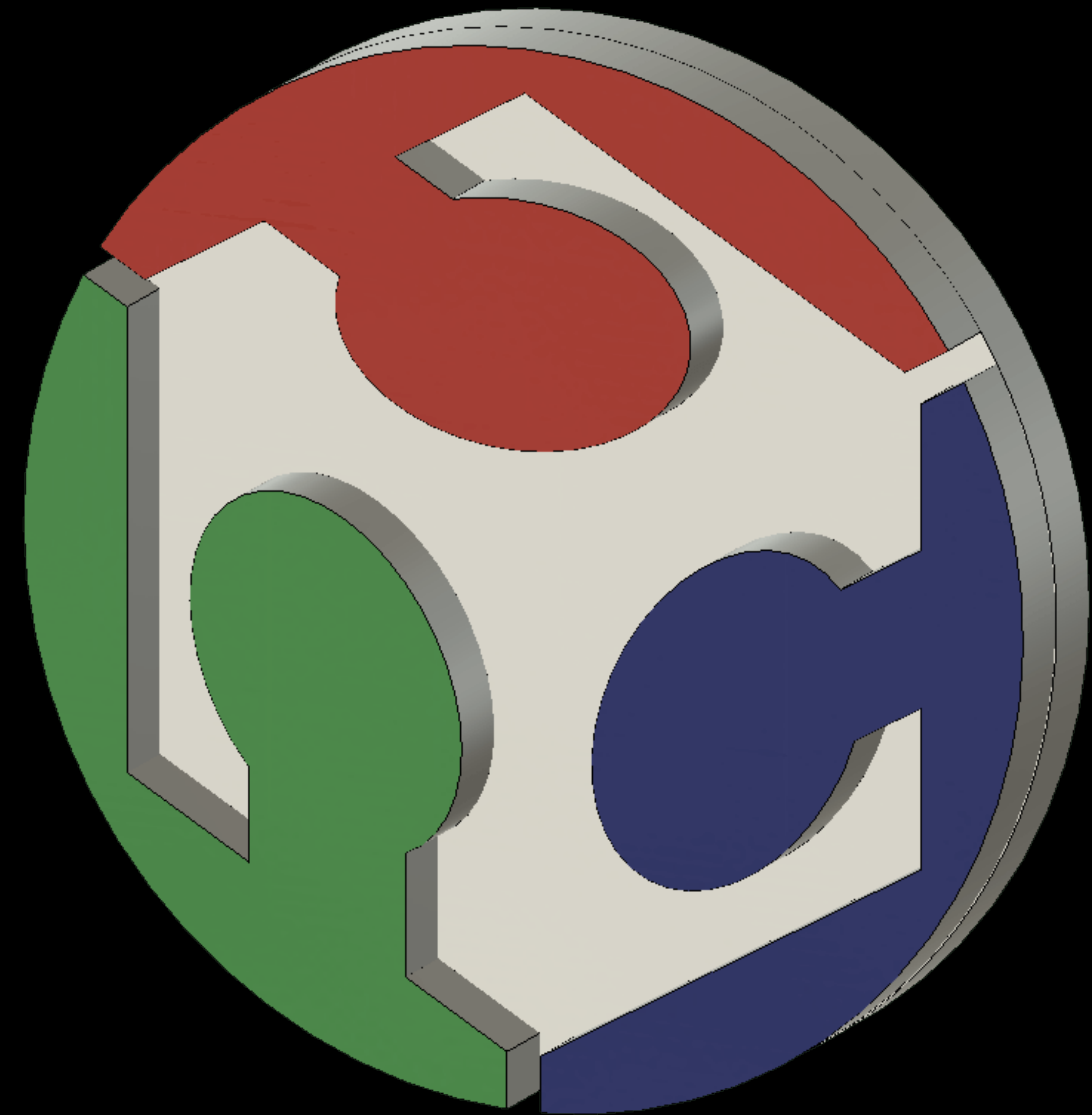


**ASSIGNMENT**

# Heroshots







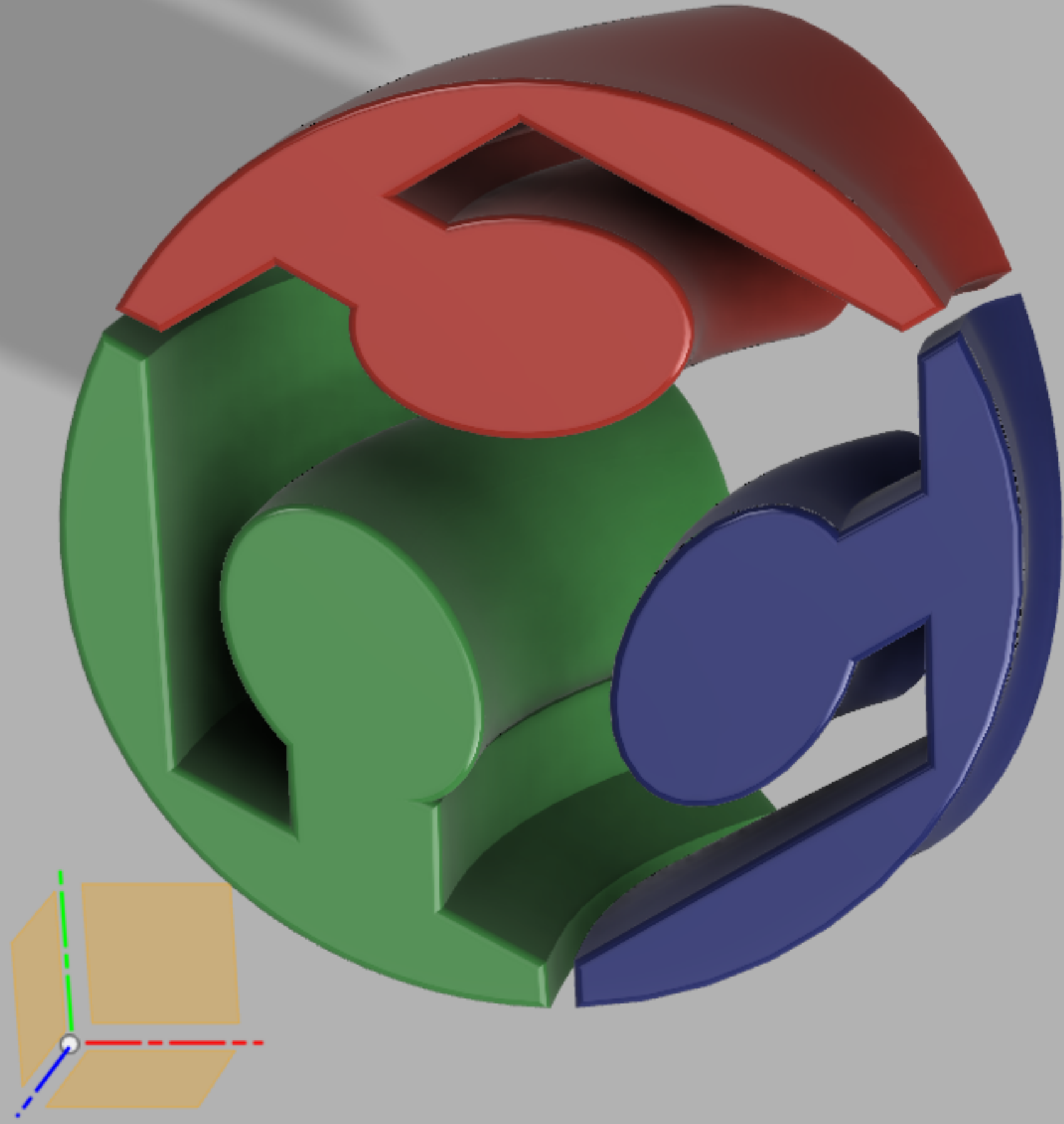
FAB LAB  
AACHEN

GERMANY'S FIRST FAB LAB



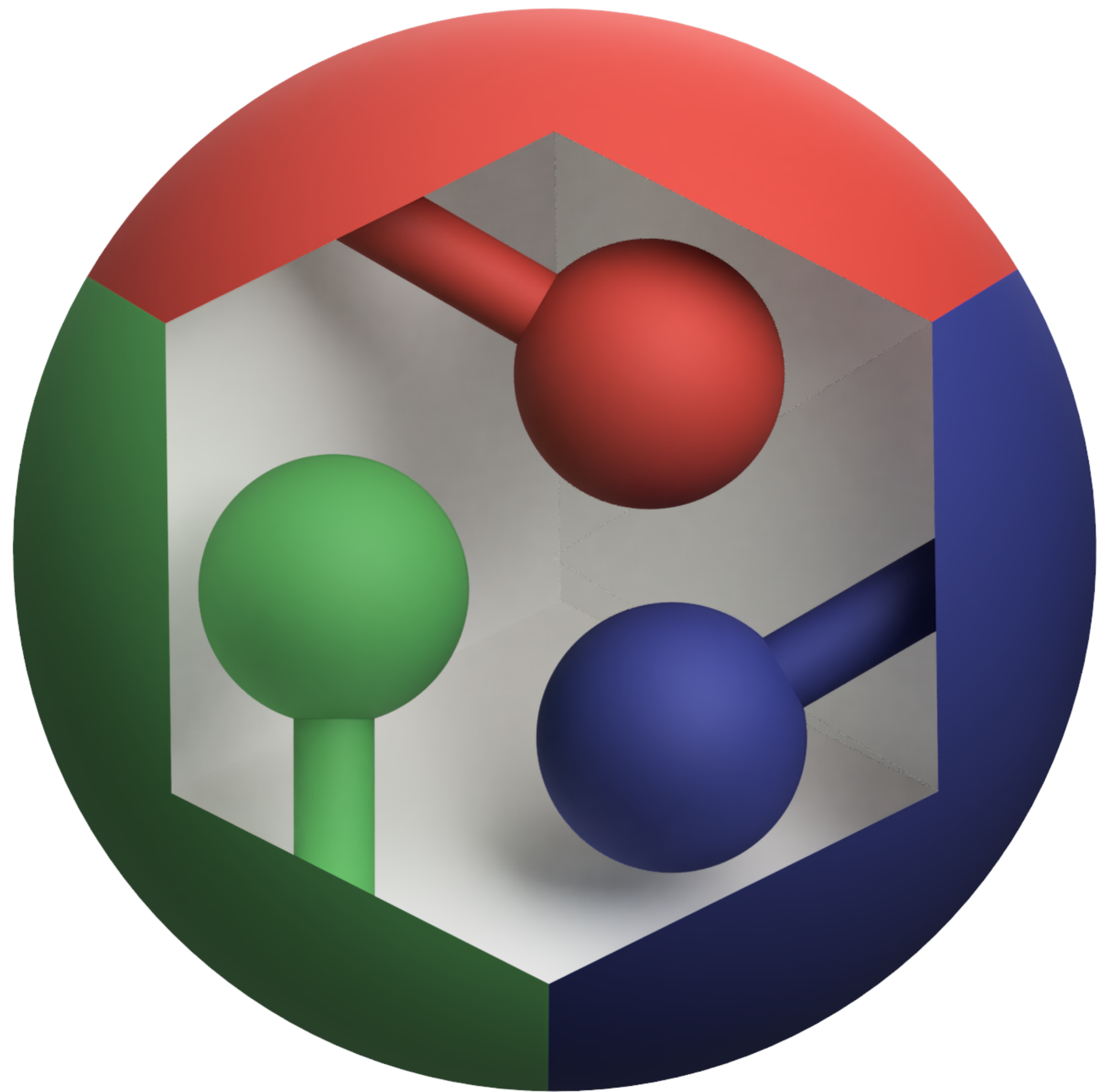
**FAB LAB**  
**AACHEN**  
GERMANY'S FIRST FAB LAB





**FAB LAB**  
**AACHEN**  
GERMANY'S FIRST FAB LAB

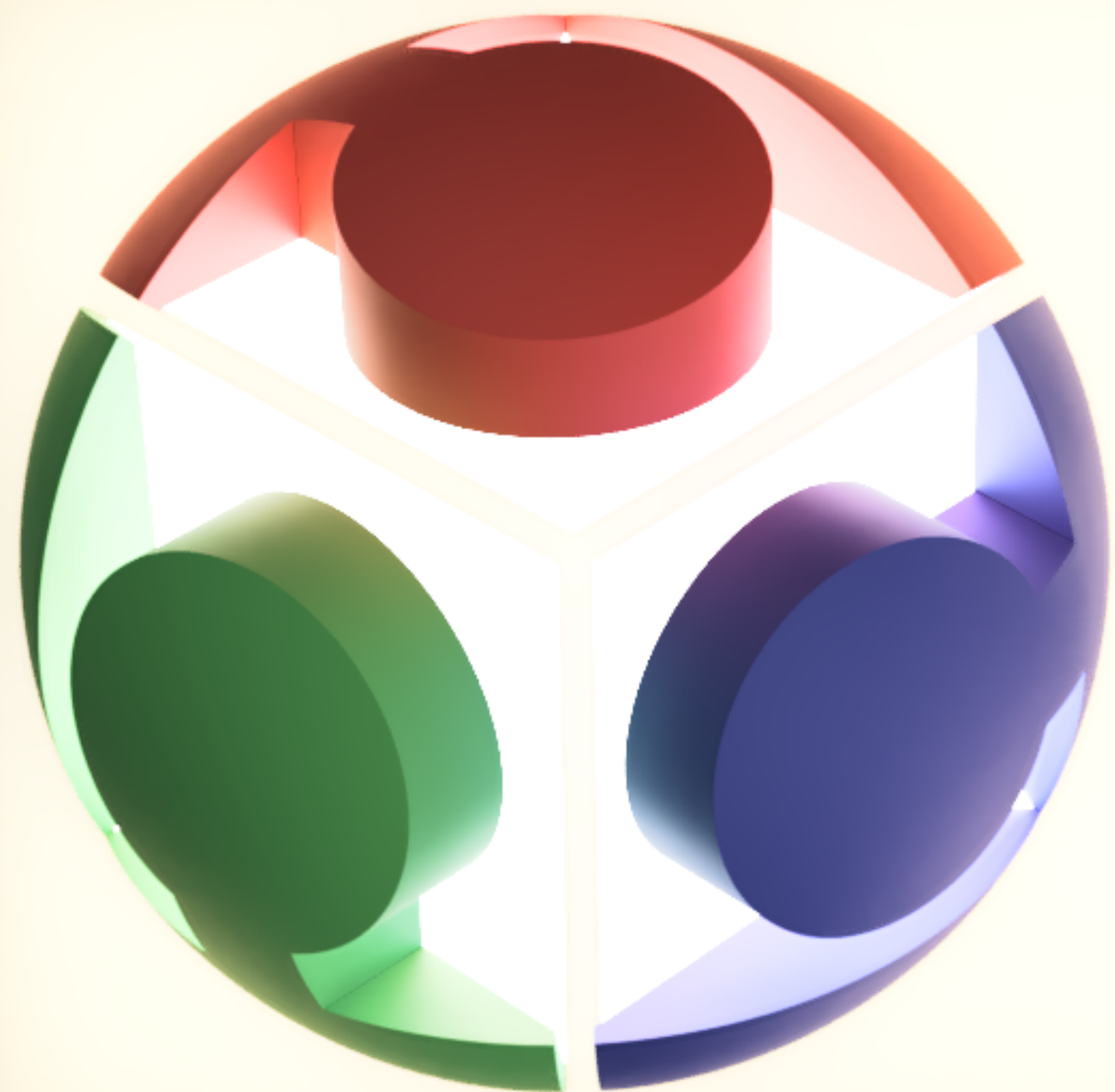




# FAB LAB AACHEN

GERMANY'S FIRST FAB LAB



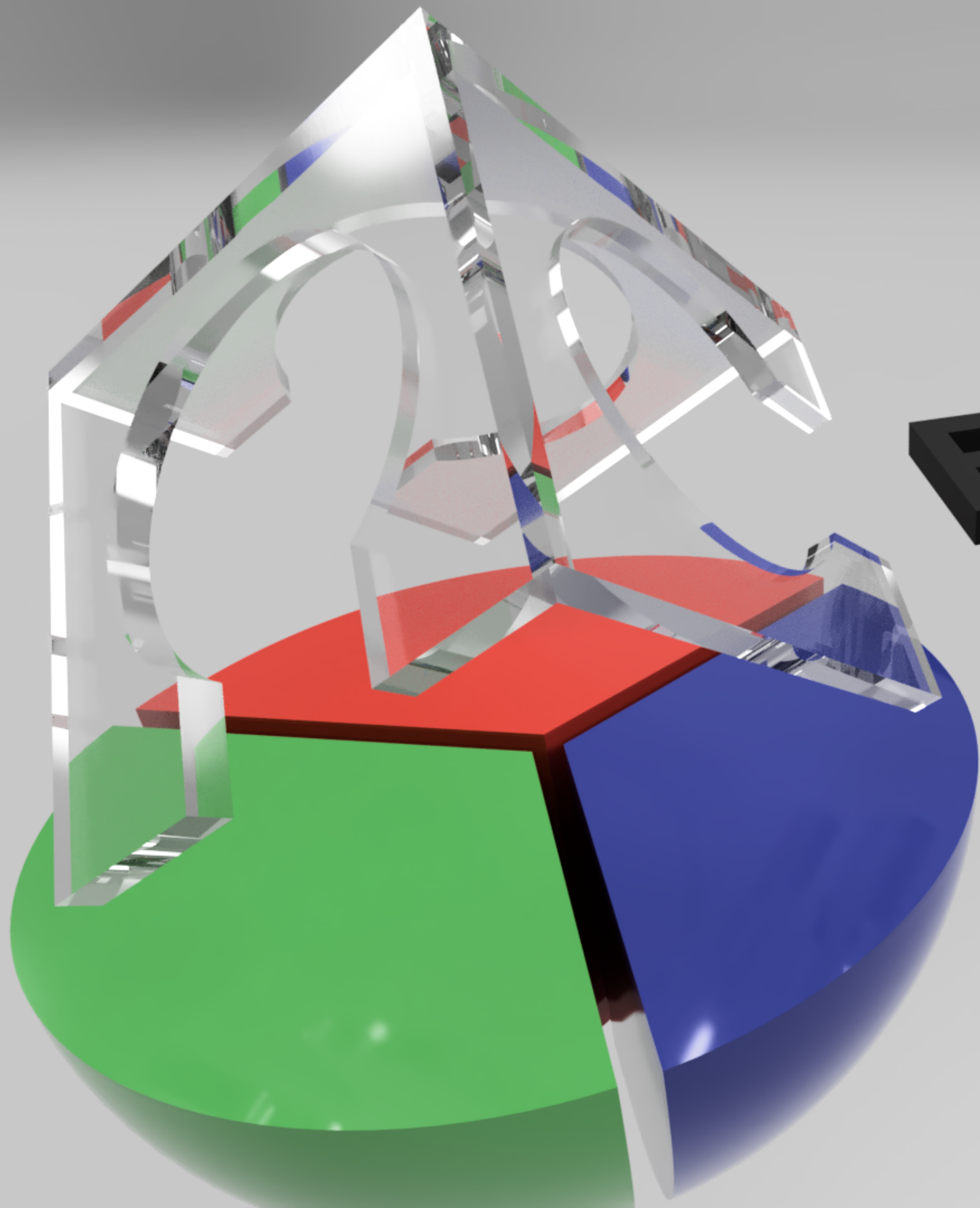


**FAB LAB**

**AACHEN**

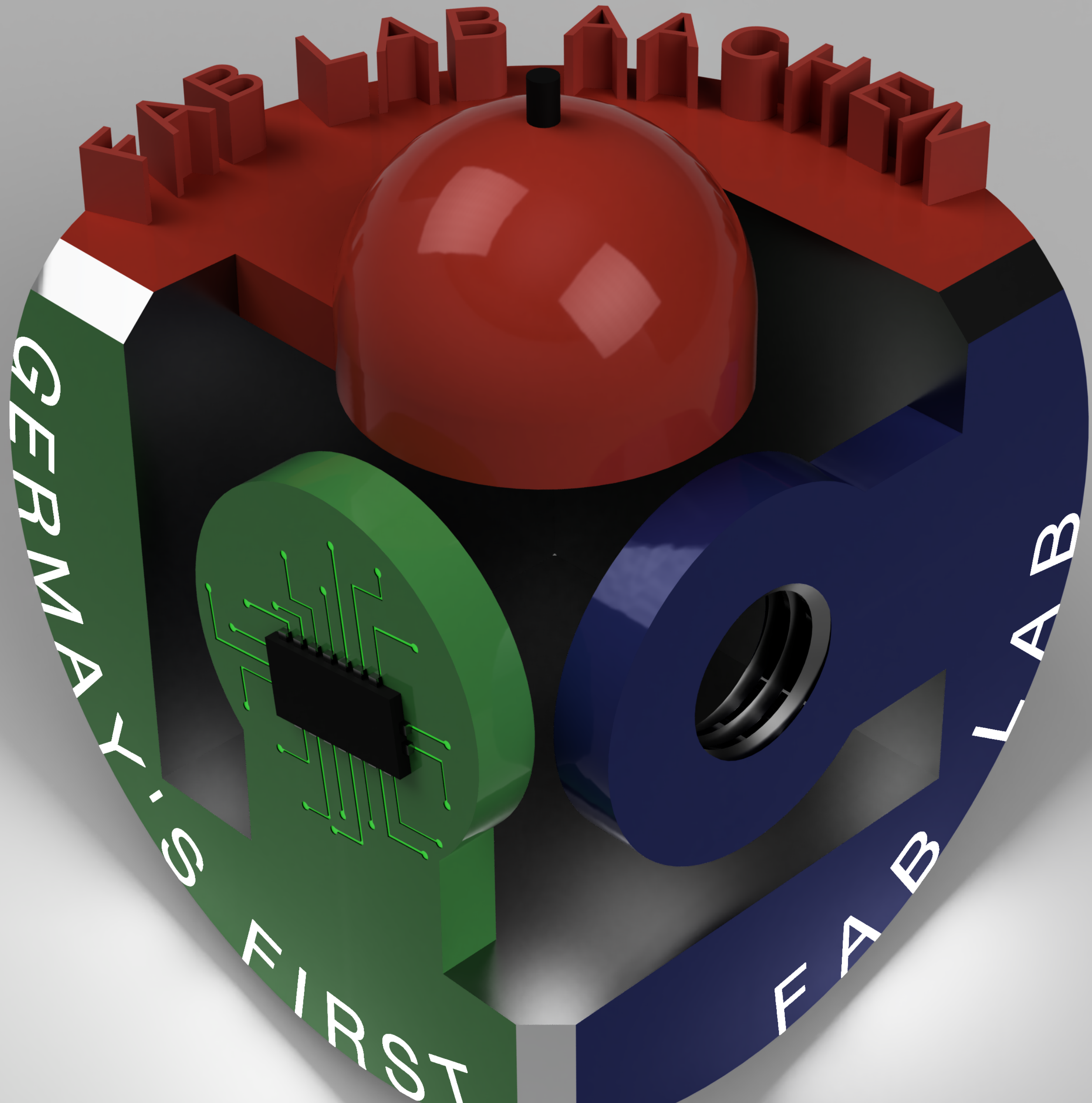
**GERMANY'S FIRST FAB LAB**





**FAB LAB  
ZACHEN**  
GERMANY'S FIRST FAB LAB









FAB LAB

AACHEN

GERMANY'S FIRST FAB LAB









# ASSIGNMENT Script







**BASICS**

# Python

```

1  #Author-Autodesk Inc.
2  #Description-Etract BOM information from active design.
3
4  import adsk.core, adsk.fusion, traceback
5
6  def spacePadRight(value, length):
7      pad = ''
8      if type(value) is str:
9          paddingLength = length - len(value) + 1
10     else:
11         paddingLength = length - value + 1
12     while paddingLength > 0:
13         pad += ' '
14         paddingLength -= 1
15
16     return str(value) + pad
17
18 def walkThrough(bom):
19     mStr = ''
20     for item in bom:
21         mStr += spacePadRight(item['name'], 25) + str(spacePadRight(item['instances'], 15)) + str(item['volume']) + '\n'
22     return mStr
23
24 def run(context):
25     = None
26     = None
27     = None
28     = None
29
30     product = app.activeProduct
31     design = adsk.fusion.Design.cast(product)
32     title = 'Extract BOM'
33     if not design:
34         ui.messageBox('No active design', title)
35         return
36
37     # Get all occurrences in the root component of the active design
38     root = design.rootComponent
39     occs = root.allOccurrences
40
41     # Gather information about each unique component
42     bom = []
43     for occ in occs:
44         comp = occ.component
45         jj = 0
46         for bomI in bom:
47             if bomI['component'] == comp:
48                 # Increment the instance count of the existing row.
49                 bomI['instances'] += 1
50                 break
51             else:
52                 bomI += 1

```





# Decorators

- Allow to wrap a function with additional instructions
- Functions can be a parameter for another function
  - Logging
  - Runtime
  - ...

```
def my_awesome_function()  
    ...  
my_awesome_function = some_other_function(my_awesome_function)
```

```
@some_other_function  
def my_awesome_function()  
    ...
```





# List Comprehension

- Increase readability

```
numbers = []  
for i in range(100):  
    if i%2 == 0:  
        numbers.append(i)
```

```
numbers = [i for i in range(100) if i%2 == 0]
```





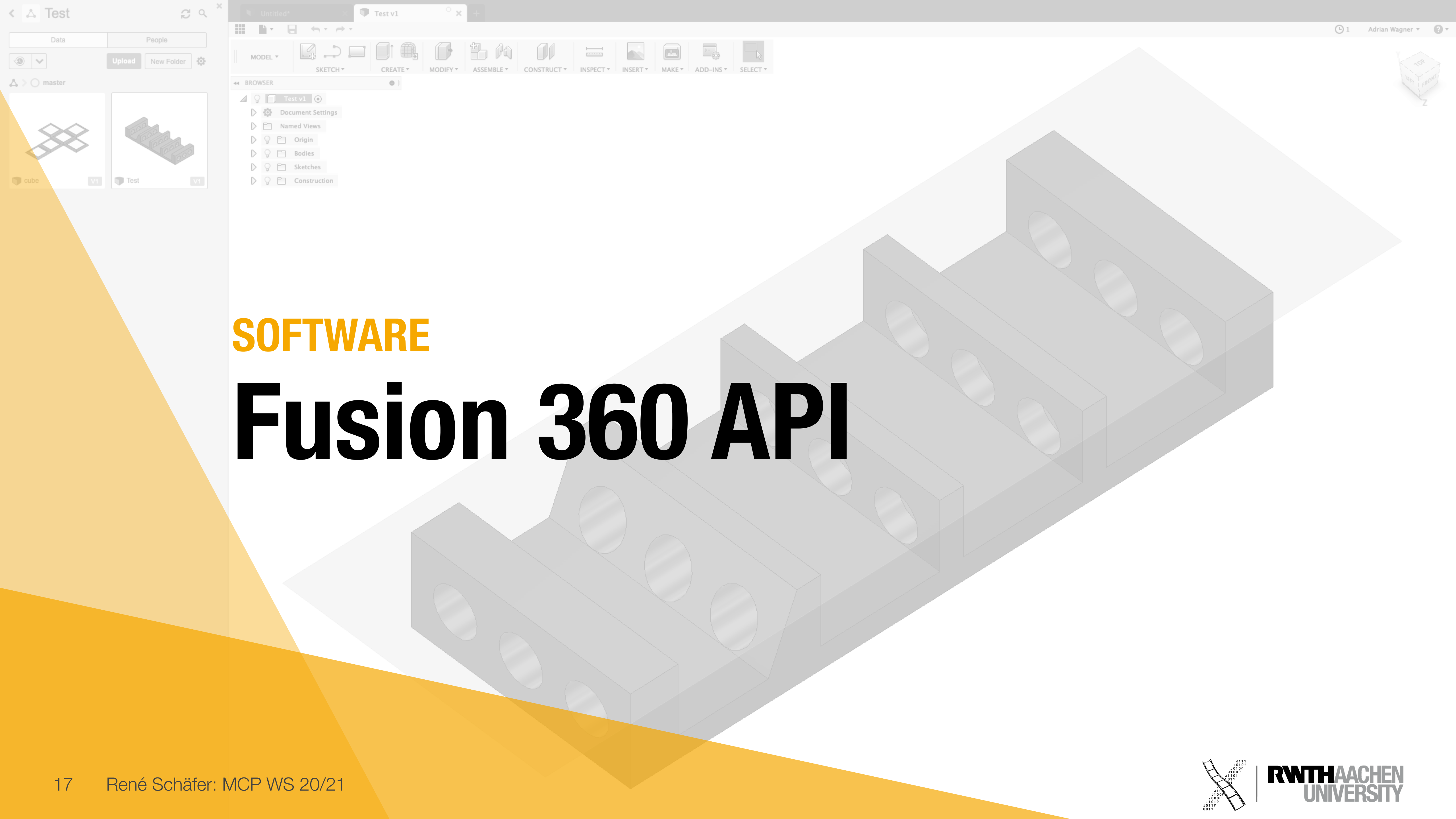
# Generators

- Do not store all values in memory
- Used e.g. for context managers
- Use **yield** instead of **return** to pause the function
- Use **list()** to convert it to a list

```
def mcp_generator():  
    yield("M")  
    yield("C")  
    yield("P")
```

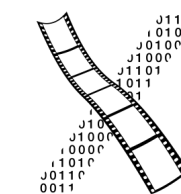
```
gen = mcp_generator()  
print(gen.next()) # M  
print(gen.next()) # C  
print(gen.next()) # P
```





**SOFTWARE**

# Fusion 360 API





# Imports

- If possible only use packages which are already available in Fusion
- Otherwise there are multiple options:
  - Use pip from Fusion to install a package
  - Install the package on your computer and adjust the system path within your script (not recommended)
  - Use python from your system to install the package into the fusion folder
    - Use the folder of your script for this
  - If the package is pure python, just place the file into the script folder

# Script vs Add-In

- Access to the same functions
- Add-Ins usually also change the UI
  - Buttons
  - Dialogs
  - ...
- Add-Ins can be run on startup
- Add-Ins need to be cleaned up when unloaded





# Resources

- Fusion documentation:
  - “Programming Interface” —> “User Interface”





**ASSIGNMENT**

# Tasks for next week

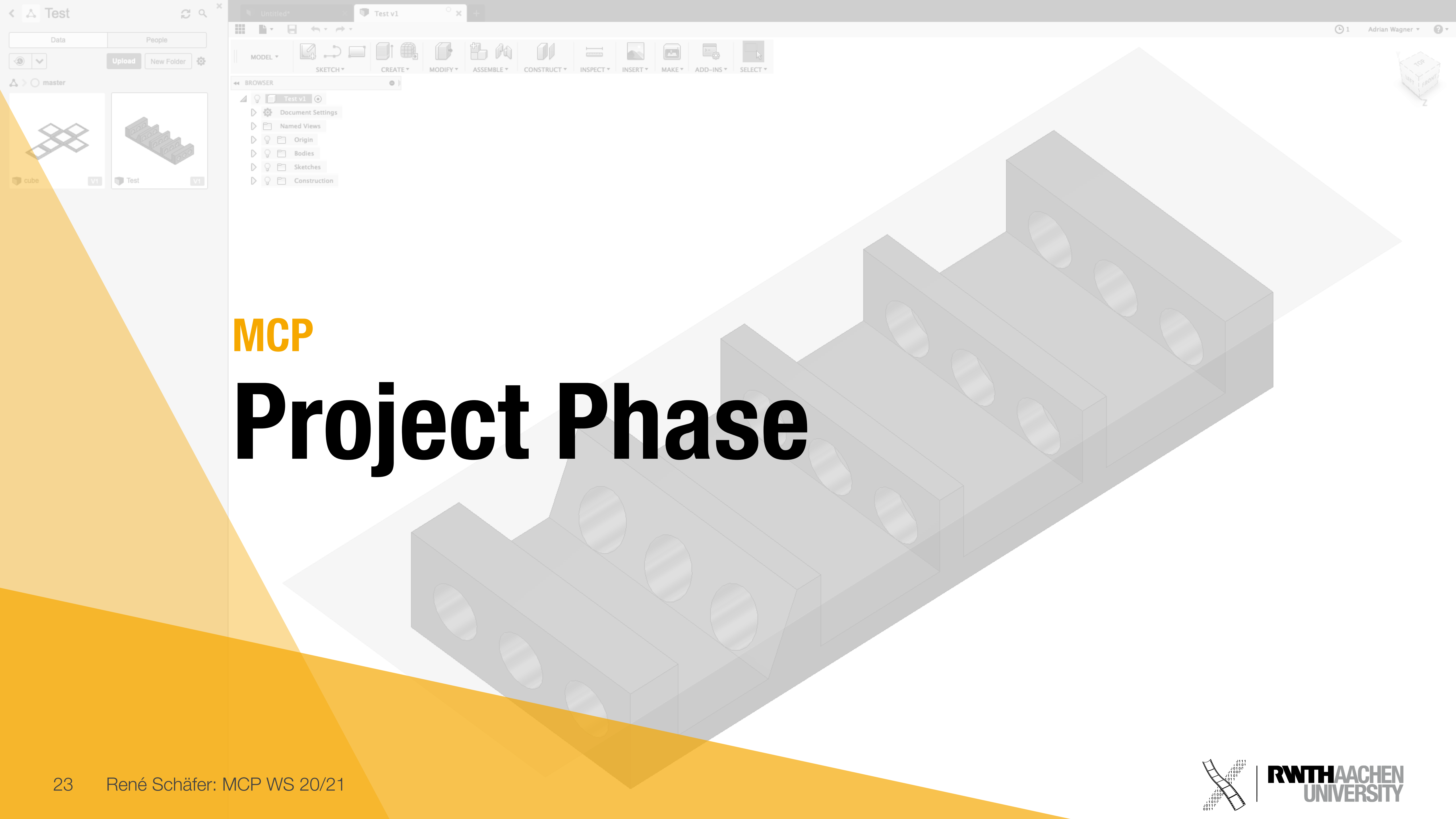




# Tasks for next week

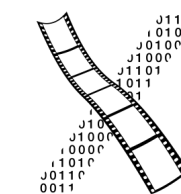
- **Convert the script from the last assignment to an Add-In**
  - **Allow user input with a dialog**
    - **Random number**
    - **Chosen number**
    - **At least one more functionality (be creative)**
  - **Clean up changes in the UI when unloading the Add-In**
- **Create a schedule for your project including all deadlines**





**MCP**

# Project Phase





# Project Phase

- Weekly individual group meetings
  - Schedule a regular meeting with your supervisor
- Deadlines
  - **December 16, 2020:** alpha version
  - **January 27, 2020:** beta version (final documentation)
  - **February 10, 2020:** final presentation (including video trailers)
- Next meeting with everyone: December 16, 2020

