**Adding a new tax function in PIT Microsimulation**

1. New tax functions are written in functions.py. Say the function is “calc\_income\_house\_property”

|  |
| --- |
| @iterate\_jit(nopython=True)  def calc\_income\_house\_property(HP\_deduction, INCOME\_HP, Income\_House\_Property):  Income\_House\_Property = INCOME\_HP - HP\_deduction  return Income\_House\_Property |

The variables “INCOME\_HP” is obtained from the Tax Return (or Survey) and hence is a “records” variable

“HP\_deduction” is a policy variable

“Income\_House\_Property” is a calculated variable which reflects the income from House Property after Deduction and is included as a “records” variable

Note: Even though “Income\_House\_Property” is returned by the function, it is declared as an input at the function definition

1. Declare the variables in the respective json files.

“INCOME\_HP” and “Income\_House\_Property” are declared in records\_variable.json. “INCOME\_HP” is declared as a “read” variable as it is directly read from the tax return/survey while “Income\_House\_Property” is declared as a “calc” (calculated) variable.

|  |  |
| --- | --- |
|  |  |

“HP\_deduction” is a policy variable and is declared in current\_law\_policy.json.

|  |
| --- |
|  |

1. Next call the function in the calculator by updating calculator.py with the new function. This is done in two places, at the beginning when we import it and in the calc\_all function where we call the function.

|  |  |
| --- | --- |
|  |  |

**Note:** The declaration looks for “self.\_\_policy” variables and “self.\_\_records” variables hence the order of the variables in the functions.py has to be in the same order – policy and then record variables.