

Effective Programming Practices for Economists

Debugging

Using the Pdb+ debugger

Janoś Gabler and Hans-Martin von Gaudecker

Setting a breakpoint

Simple

```
def cobb_douglas(x1, x2, gamma1, gamma2, a):  
    import pdb; breakpoint()  
    return (a * x1**gamma1 * x2**gamma2,)
```

Conditional

```
def cobb_douglas(x1, x2, gamma1, gamma2, a):  
    if gamma1 <= 0.5:  
        import pdb; breakpoint()  
    return (a * x1**gamma1 * x2**gamma2,)
```

- Set a breakpoint with `import pdb; breakpoint()`
- You can do that anywhere!
 - Inside function definitions
 - In loops
 - In if conditions!
- Execution will stop at the breakpoint and show you the interactive debug prompt

Important commands

Command Action

<code>`n`</code>	Execute the n ext line
<code>`s`</code>	Execute the next s tep
<code>`c`</code>	c ontinue until the next breakpoint
<code>`u`</code>	Go one frame u p (go backwards through code)
<code>`d`</code>	Go d own one frame (go forward through code)
<code>`exit`</code>	Stop the debugging (also <code>ctrl + d`</code>)

- More commands here
- Do not use any of those as variable names!

Graphical alternatives

- VScode and other IDEs have graphical debuggers
 - Set breakpoints via clicking
 - Variable explorers
- We prefer the terminal for several reasons
 - Integrates perfectly with pytask and pytest
 - Extremely fast once you get a bit of practice
 - More robust (in our experience)