Effective Programming Practices for Economists

Software engineering

Pure functions

Janoś Gabler and Hans-Martin von Gaudecker

Definition of pure functions

In computer programming, a pure function is a function that has the following properties:

1. the function return values are identical for identical arguments (no variation with local static variables, non-local variables, mutable reference arguments or input streams)
2. the function has no side effects (no mutation of local static variables, non-local variables, mutable reference arguments or input/output streams).

– Wikipedia

Benefits of pure functions

- **Explicit interfaces**: What is used in the function is passed in as argument
- **Testability**: While testing, control everything that is relevant
- **Parallelization**: No worries when calling pure functions in parallel
- **Reduced mental load**: No worries about side effects when calling pure functions
- **Powerful tools**: Pure functions are compatible with powerful concepts from functional programming and some libraries expect them

Push impurities to the boundaries!

```
def task_clean_data(
    data = SRC / "original_data" / "data.csv",
    produces = BLD / "data.pkl",
):
    df = pd.read_csv(data)
    clean = clean_data(df)
    clean.to_pickle(produces)
```

- Functions that read or write files are impure, but unavoidable
- Solution: Push impurities to the boundaries
- We have covered examples in pytask