

Static Analysis and Continuous Integration

Fall 2022

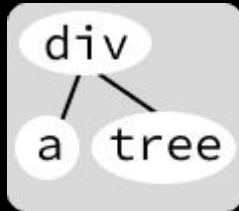
Static Analysis

Looking at the program

Purely syntactic

```
<p>↵  
__text↵  
</p>↵
```

```
<p>tokens</p>
```



Dynamic Analysis

Running the program

Dynamic properties

e.g.

A variable is never false

No uncaught exceptions

No null pointers

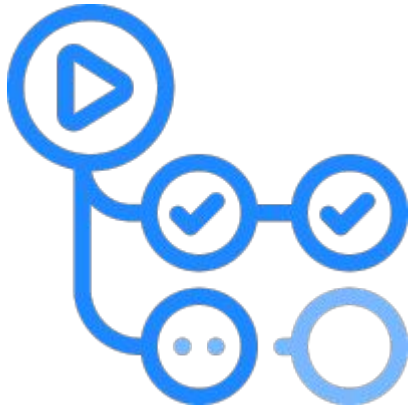
Static example: Linter

Static language-specific analysis tool for code styling and formatting

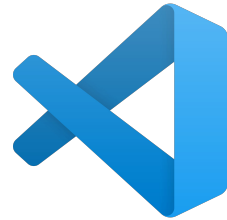
Checks for

- Syntax errors
- Code standards adherence
- Code smells
- Security

You are already running analysis tools



GitHub
Actions



This recitation: add more to your GH Actions workflow (static *and* dynamic tools)

How does GH Actions work

- Runs on containers: it's not on your laptop!
 - Platform might be different
- Like all boring code, it's written in YAML (JSON as bullets)
 - Triggers: push, pull_requests (useful for forks)
 - Steps, working directories, timeout etc etc
- Testing actions:
 - Can be frustrating to debug. Try commands locally as much as possible

YAML is basically fancy JSON

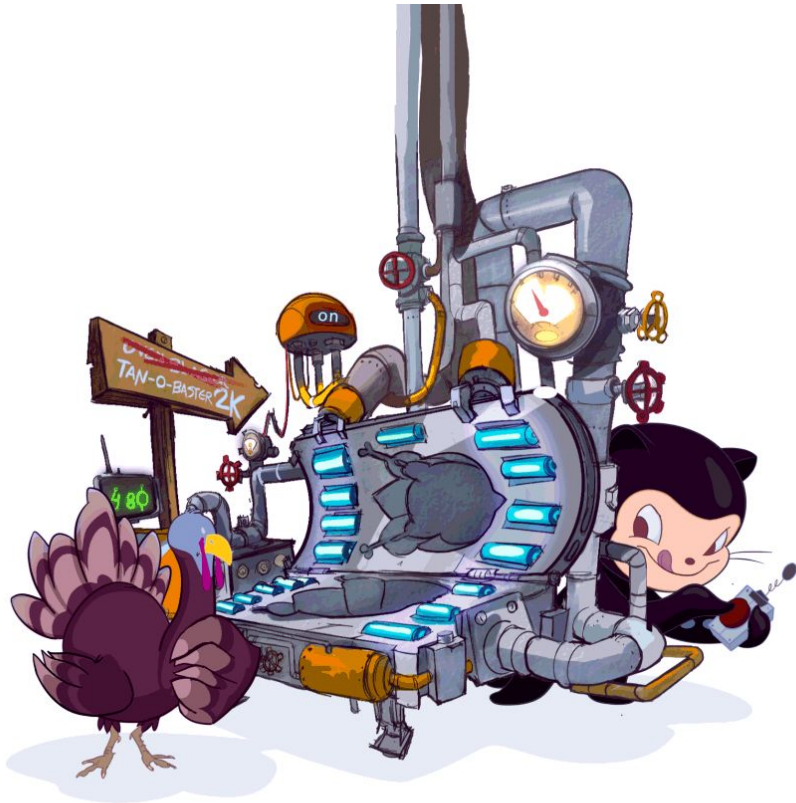
```
doe: "a deer, a female deer"  
ray: "a drop of golden sun"  
pi: 3.14159  
xmas: true  
french-hens: 3  
calling-birds:  
  - huey  
  - dewey  
  - louie  
  - fred  
xmas-fifth-day:  
  calling-birds: four  
  french-hens: 3  
  golden-rings: 5  
  partridges:  
    count: 1  
    location: "a pear tree"  
  turtle-doves: two
```

```
{  
  "doe": "a deer, a female deer",  
  "ray": "a drop of golden sun",  
  "pi": 3.14159,  
  "xmas": true,  
  "french-hens": 3,  
  "calling-birds": [  
    "huey",  
    "dewey",  
    "louie",  
    "fred"  
  ],  
  "xmas-fifth-day": {  
    "calling-birds": "four",  
    "french-hens": 3,  
    "golden-rings": 5,  
    "partridges": {  
      "count": 1,  
      "location": "a pear tree"  
    },  
    "turtle-doves": "two"  
  },  
}
```

Try it out

Navigate to the recitation handout and begin

- Fix the CI
- Add a lint job to CI
- Add a code coverage reporter to CI
- (Bonus) PR bot comments about coverage



**Some example
tools**

Semantic

- Language support system that powers code navigation on GH
- Processing source code into appropriate representation
- <https://github.com/github/semantic>

Code Climate

- Syntactic analysis for determining code quality
- Provides a maintainability score on your application
- Reports various code smells in application code

Refer to <https://docs.codeclimate.com/> for more information

CodeQL

- Discover vulnerabilities across a codebase
- Query code as though it were data
- Discover a bad pattern
- Find similar occurrences across the entire codebase

Refer to <https://codeql.github.com/> for more information