- Biggest benefit of Ivy, the rendering pipeline removes actually reads the code vs. static analysis. This removes imports that are never used, even if they are referenced in the code.

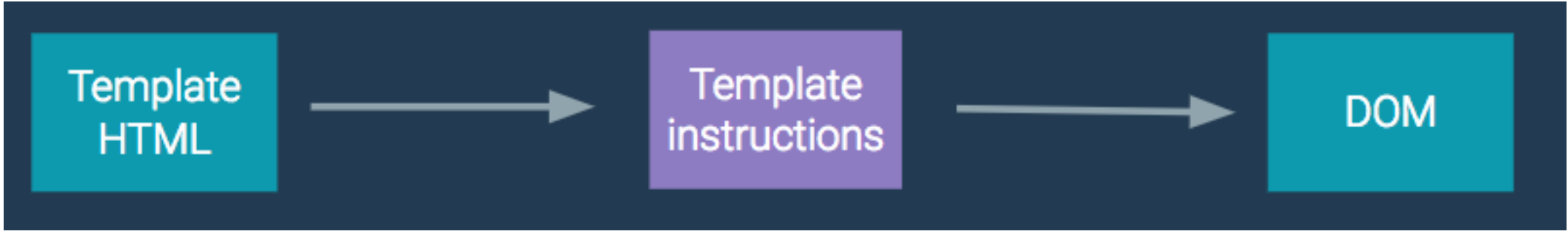
- Backwards Compatible - 6 months out (status)

- Locality - Allows AOT dependencies - & no meta data. Local meaning only information which is directly tied to the annotation of the component

- Stack trace is much more accurate, and identifies the issue.

- Build size is insanely small!
  - Today Compressed → 36K
  - Minified → 7.3Kb
  - Ivy Compressed → 2.7Kb


Template HTML
Template Data
Angular Interpreter
DOM


Template
HTML
Template
instructions
DOM

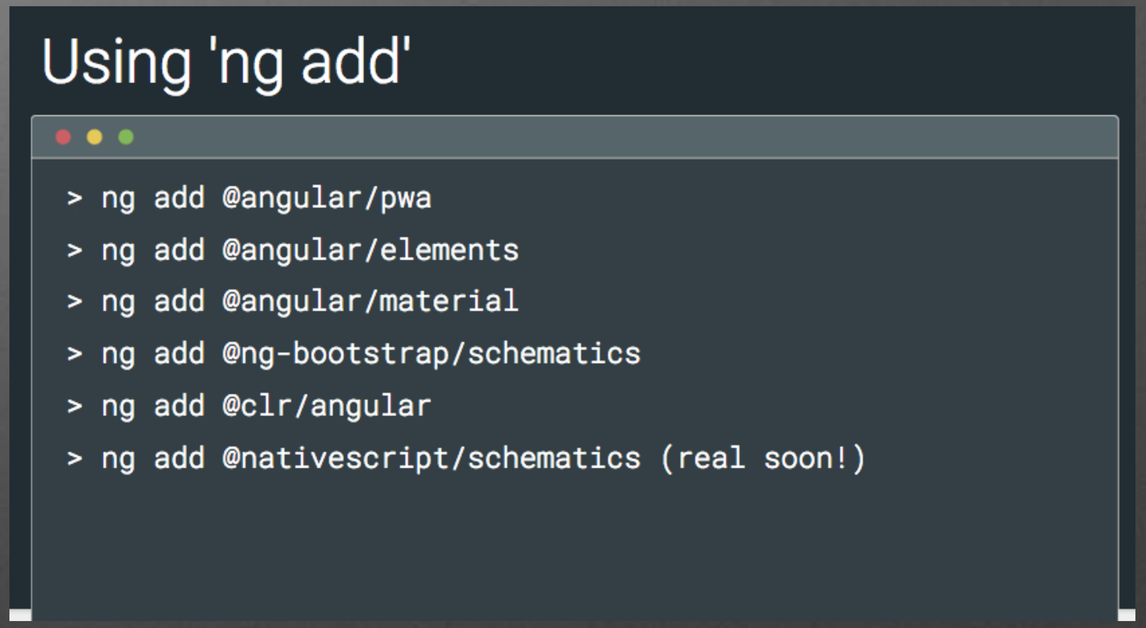
```
import {someFn, unusedFn} from './third-party';

function main(x) {
    someFn();
    if (x) unusedFn();
    return x;
}

main(false);
```

Schematics

Schematics allow you to “do things” to your project.

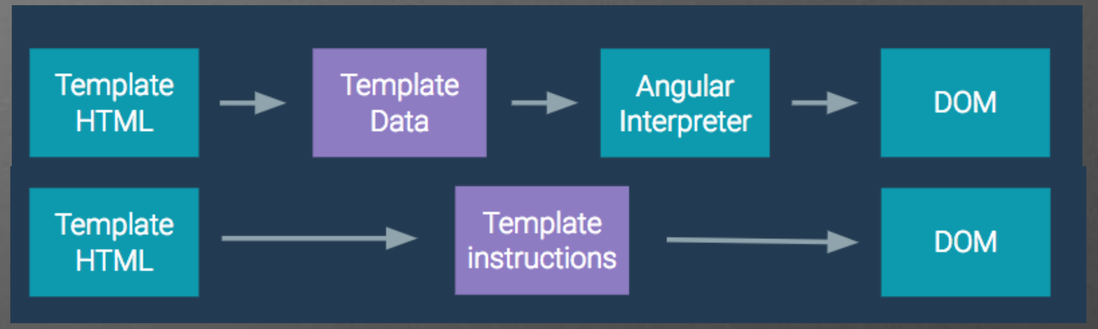
- update = single command to update dependencies
  - *ng update = npm, code, RxJS, Material*
- add = instant code and library scaffolding
  - Material, Elements, PWA, bootstrap, clarity, nativescript
  - *ng add @angular/pwa = does all the setup.*
- generate = Per-library 'generate' schematics
  - *ng generate @angular/material:material-nav --name=main-nav*

Vendors and Us can publish our own schematics. Goal is to make universal libraries easier to build: project scaffolding, test, build system. N



Ivy

- Biggest benefit of Ivy, the rendering pipeline removes actually reads the code vs. static analysis. This removes imports that are never used, even if they are referenced in the code.
- Backwards Compatible - 6 months out (status)
- Locality - Allows AOT dependencies - & no meta data. Local meaning only information which is directly tied to the annotation of the component
- Stack trace is much more accurate, and identifies the issue.
- Build size is insanely small!
  - Today Compressed → 36K
  - Minified → 7.3Kb
  - Ivy Compressed → 2.7Kb




```
import {someFn, unusedFn} from './third-party';

function main(x) {
    someFn();
    if (x) unusedFn();
    return x;
}

main(false);
```