

# *Working with* JavaScript

*The Other Stuff*

# You Know JavaScript

What else is there?

# JavaScript, HTML and CSS

The Web Presentation Engine

# JavaScript, HTML and CSS

JavaScript is the programming language for the web.

HTML and CSS is the presentation language for the web.

A solid knowledge of HTML and CSS is crucial to create complex JavaScript web applications.

# Frameworks

jQuery, Angular, React, Vue, ...

# Frameworks

Manually updating HTML with JavaScript is easy, until it's not, even with tools like jQuery and MooTools.

Frameworks provide abstraction in increase development speed.

Provide software design patterns such as MVC and MVVM.

Introduce the *Shadow DOM* and React's *JSX* syntax.

# Package Management

In an open-source world

# Package Management

Speed up development by including pre-written packages.

E.g. `npm i react`

Automate the installation of open-source and private code packages.

E.g. `npm ci`

Share your code with the community.

E.g. `npm publish`



# Debugging

The art of making it work

# Debugging

Browser developer tools are crucial for development.

```
console.log("My Variables", a, b, c);
```

Break execution and open the console with `debugger()` ;  
or by setting breakpoints.

# Testing

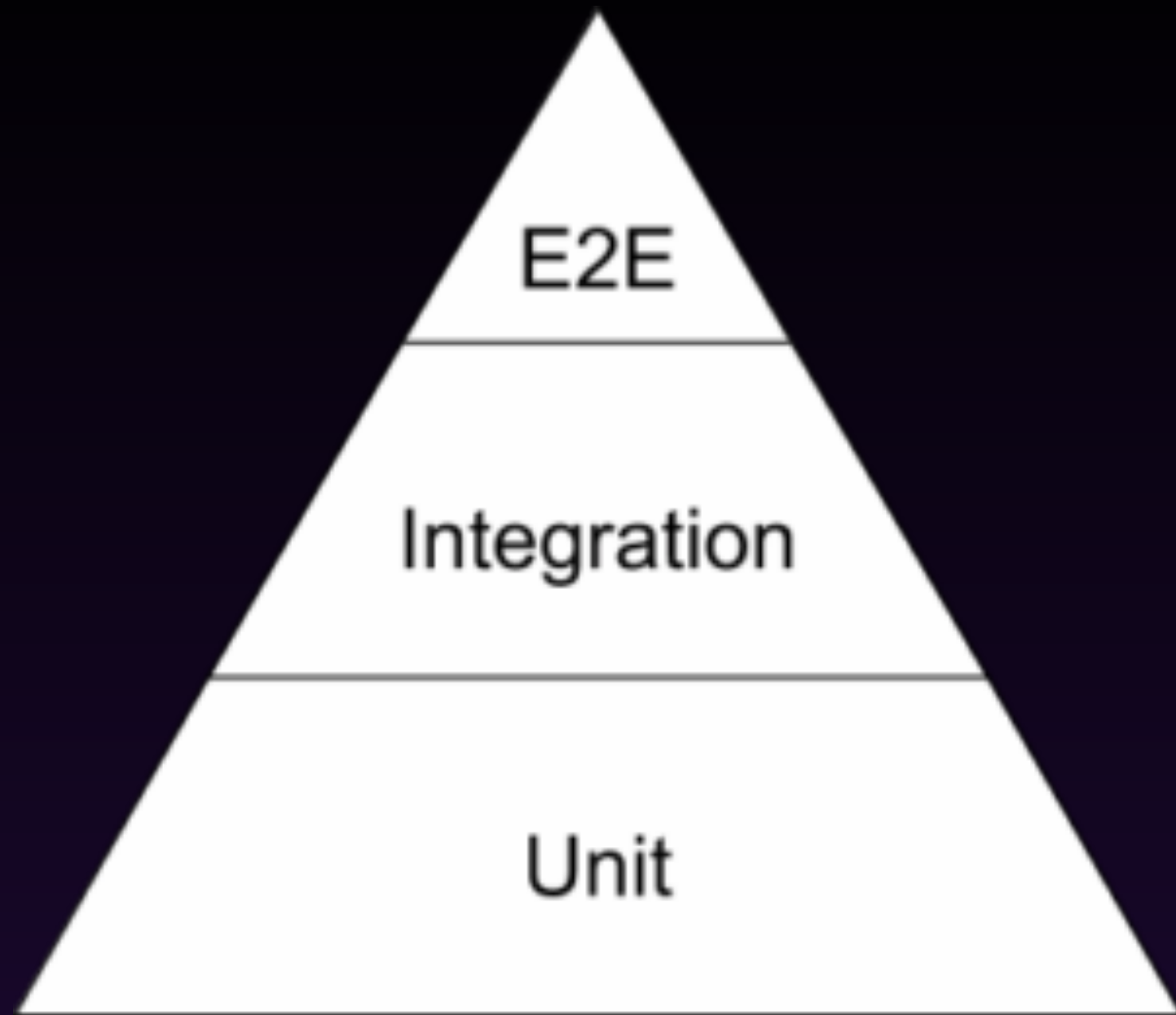
Making sure you know if it doesn't work

# Testing

Unit tests cover the external API of small functions.

Integration tests cover deep and complex interactions.

End to End tests cover the full-stack of the application.



# Testing Pyramid

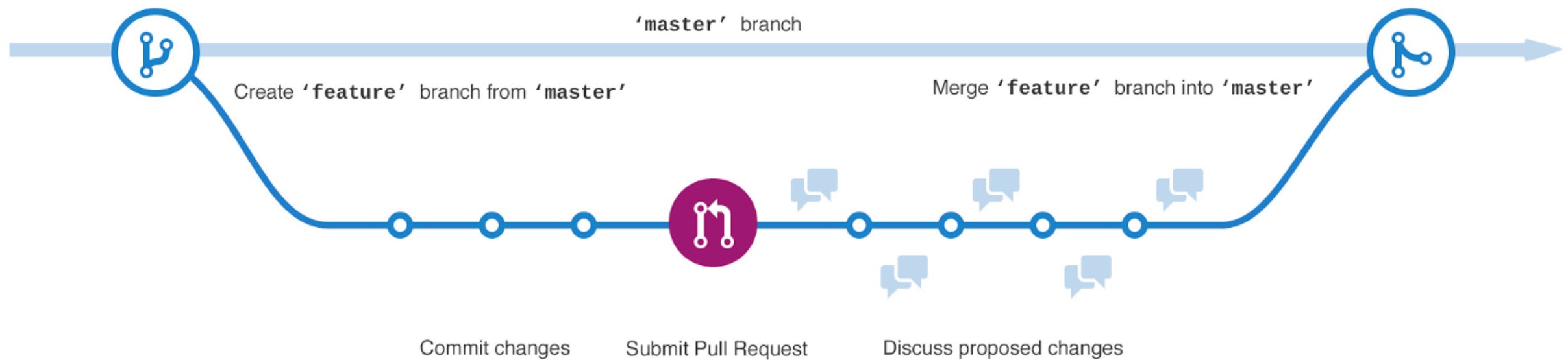
# Version Control

git good

# Version Control

Track changes and feature development in a project.

Distributed code management to share & review with others.



# Branches, PRs and Merging



# Deployment

Push to production on a Friday afternoon?

# Deployment

Continuous Integration and Continuous Deployment ensures code is always tested and ready to release before it can be merged.

Integrated into a version control system, CI/CD will build and test a release candidate for the product.

CI/CD will usually deploy to a staging environment to allow for a QA check.



# CI/CD Pipeline

# Documentation

Define your assumptions

# Documentation

Documenting the purpose of code can help you write it.  
E.g. BDD

Documenting the APIs/interfaces can help with testing.  
E.g. TDD

Document the assumptions and boundary conditions.

Documentation for different audiences.

# Product

What to build

# Product

Tickets describe the work

Projects describe the scope

Roadmap describes the business value

Done