

Master Cypress in 15 minutes a day

Maurice de Beijer - @mauricedb

Course Goals

Course Goals

- Learn what Cypress is
 - And what it can do for you
- Understand what the best practices are
 - And what not to do
- See how Cypress differs from other End to End testing tools
 - Like Selenium, Nightwatch.js etc.

Why this course?

- Cypress is not hard to use most of the time
 - Not knowing the best practices can trip you up
- Going beyond the basics can be tough

15 minutes

- Can I really do this in 15 minutes a day?

See you in the next video

What is Cypress

And why use it?

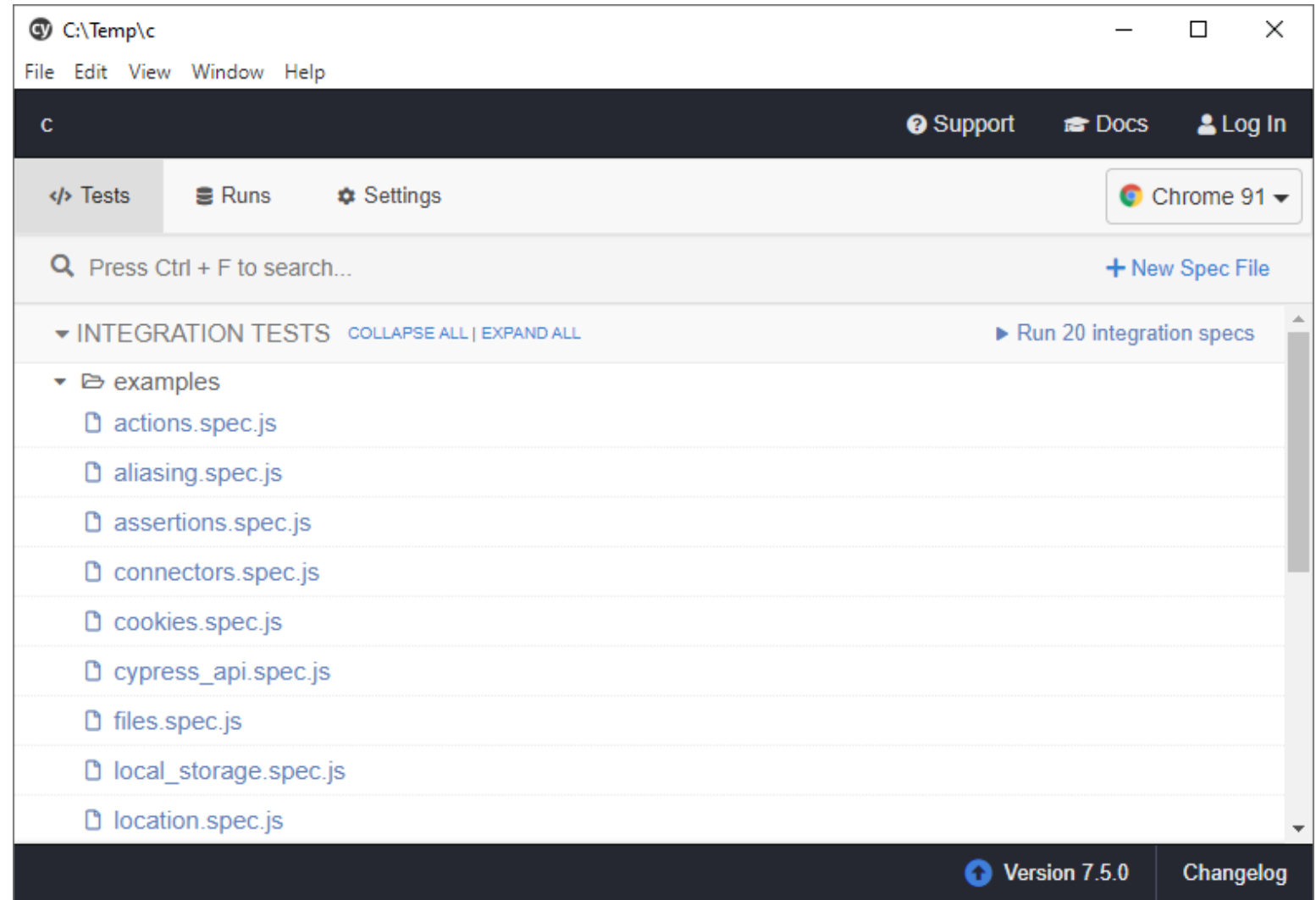
What is Cypress

- Cypress is a front end testing tool
 - Built for the modern web
- Tests are easy to write
 - Using simple JavaScript
- Time Travel runner lets you inspect test at each step
 - Click on a step to see the state of the browser
- Automatic waiting for elements
 - Makes tests more resilient
- Video recording of running test
 - Helps debugging failing tests on the CI server

What is Cypress

- Lets you fake the network
 - Or use the real backend as appropriate
- Automatically retry
 - Run flaky tests multiple times
- Many standard and 3rd party extensions
 - Or write your own in JavaScript
- Can run in Docker containers
 - Consistent cross platform behavior
- Runs inside the browser
 - Not based on Selenium

Interactive Test Runner



Interactive Test Runner

The screenshot displays the Cypress test runner interface. On the left, the 'Tests' panel shows a test suite 'Block Buster Film Reviews' with two passing tests: 'Has the correct title' and 'Can open The Shawshank Redemption'. Below these, the 'TEST BODY' section lists the test steps: 1. visit /top-rated-movies, 2. get #movie-278 a:first, 3. -click, 4. (page Load) --page Loaded--, 5. (new url) https://block-buster-film-reviews.azureedge.net/movie/278, 6. get h1, 7. -assert expected <h1.bd-hd> to contain text The Shawshank Redemption (1994). The right panel shows the application under test, which is a movie review page for 'The Shawshank Redemption (1994)'. The page features a movie poster, a rating of 8.7/10, and a 'WATCH TRAILER' button. The test runner indicates that the test passed successfully.

Chrome is being controlled by automated test software.

Tests: 4 ✓, 1 ✗, 10.52s

https://block-buster-film-reviews.azureedge.net/movie/278

1000 x 660 (89%)

cypress/integration/block-buster.spec.js

Open Selector Playground

Block Buster Film Reviews

- ✓ Has the correct title
- ✓ Can open The Shawshank Redemption

TEST BODY

- 1 visit /top-rated-movies
- 2 get #movie-278 a:first
- 3 -click
- 4 (page Load) --page Loaded--
- 5 (new url) https://block-buster-film-reviews.azureedge.net/movie/278
- 6 get h1
- 7 -assert expected <h1.bd-hd> to contain text The Shawshank Redemption (1994)

✓ The Shawshank Redemption has the expected cast

✓ Can navigate to Morgan Freeman

The Shawshank Redemption (1994)

8.7/10
19112 Reviews

Rate This Movie: ★★★★★★☆☆

OVERVIEW REVIEWS CAST & CREW MEDIA SIMILAR MOVIES

Framed in the 1940s for the double murder of his wife and her lover, upstanding banker Andy Dufresne begins a new life at the Shawshank prison, where he puts his accounting skills to work for an amoral warden. During his long stretch in prison, Dufresne comes to be admired by the other inmates -- including an older prisoner named Red -- for his integrity and unquenchable sense of hope.

Director:
Frank Darabont

Writer:
Stephen King (Novel)
Frank Darabont (Writer)

Pete von Sholly (Storyboard)

Genres:
Drama Crime

VIDEOS & PHOTOS

DOM Snapshot (pinned)

Command Line Interface

```
Windows PowerShell

Running: block-buster.spec.js (1 of 1)

Block Buster Film Reviews
  ✓ Has the correct title (810ms)
  ✓ Can open The Shawshank Redemption (344ms)
  ✓ The Shawshank Redemption has the expected cast (175ms)
  ✓ Can navigate to Morgan Freeman (377ms)

4 passing (2s)

(Results)

Tests:      4
Passing:    4
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      true
Duration:   1 second
Spec Ran:   block-buster.spec.js

(Video)

- Started processing: Compressing to 32 CRF
- Finished processing: C:\Temp\c\cypress\videos\block-buster.spec.js.mp4 (1 second)




=====

(Run Finished)

Spec                                Tests  Passing  Failing  Pending  Skipped
✓ block-buster.spec.js              00:01    4        4        -        -        -
✓ All specs passed!                 00:01    4        4        -        -        -
```

Pros and Cons of Cypress

Pros

- Easy to get started
-  Fast execution
- Automatic waiting on elements
- Access to elements outside the DOM
- Support for FireFox and Chromium based browsers
- No driver dependencies
-  Time travel and debuggability
- Uses the Mocha test framework
- Open source
-  Mostly free

Cons

- Only JavaScript or TypeScript
- No support for Safari
- Only a single browser
- Only a single tab

See you in the next video



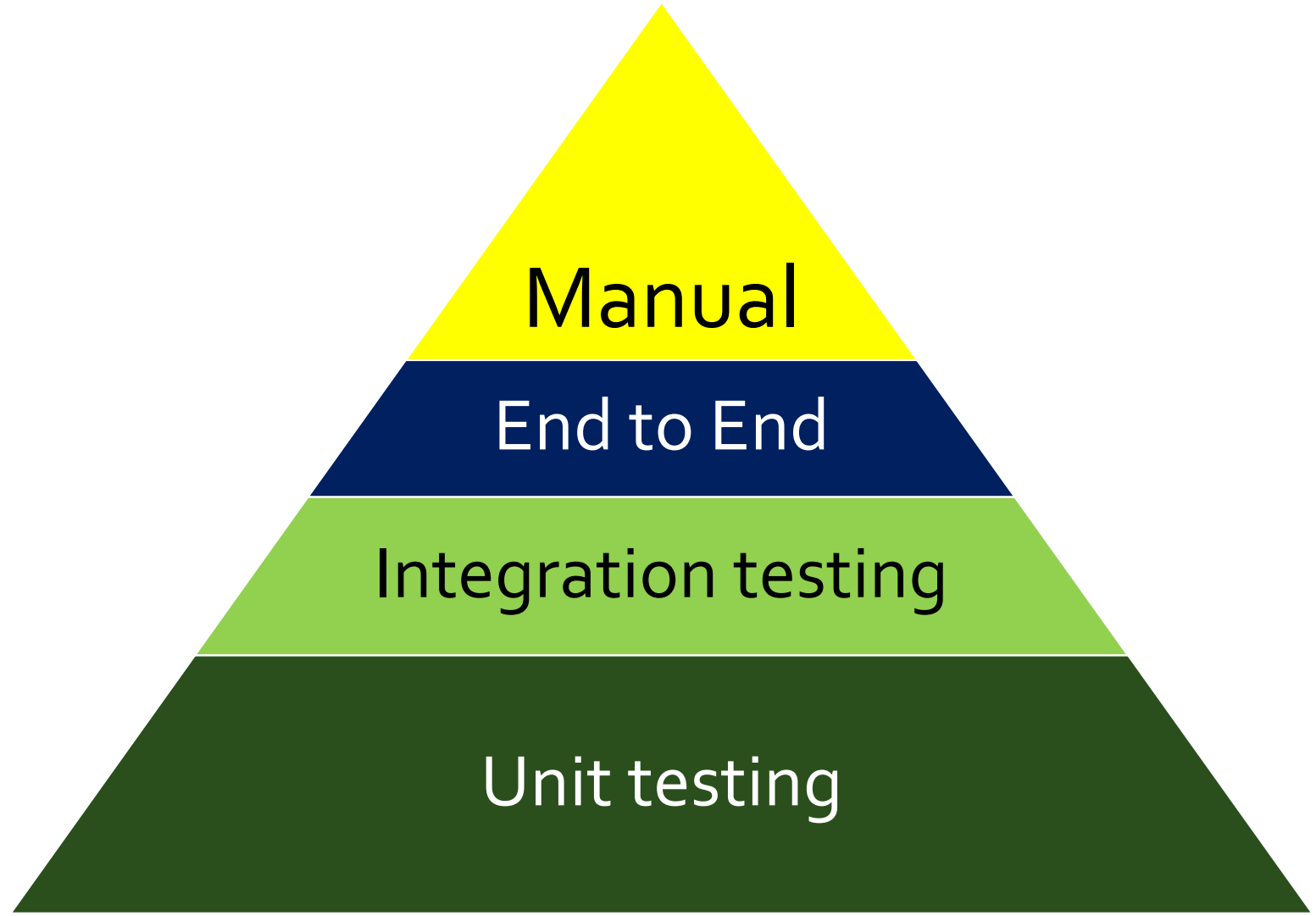
What is end to end testing

What is testing

“Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.”

-- Wikipedia --

The traditional testing pyramid



Unit Testing & Code Coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
All files	98.92	94.36	99.49	100	
yargs	99.17	93.95	100	100	
index.js	100	100	100	100	
yargs.js	99.15	93.86	100	100	
yargs/lib	98.7	94.72	99.07	100	
command.js	99.1	98.51	100	100	
completion.js	100	95.83	100	100	
obj-filter.js	87.5	83.33	66.67	100	
usage.js	97.89	92.59	100	100	
validation.js	100	95.56	100	100	

Both windows
are fine



via reddit.com/r/programmerhumor

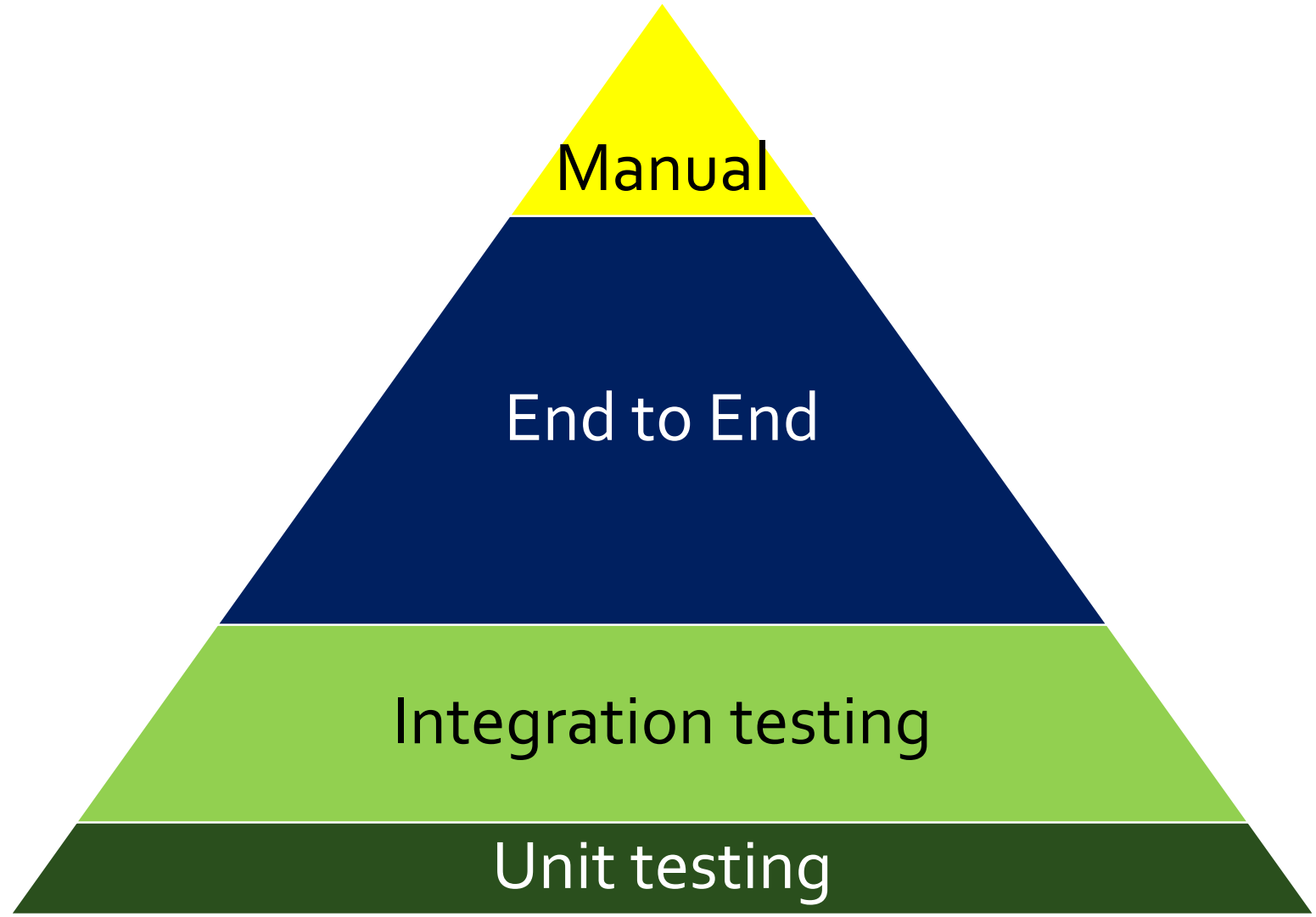
[Source](#)

A sturdy latch



[Source](#)

A better
testing
pyramid for
the web



End-to-end Testing

“End-to-end testing is a methodology used to test whether the flow of an application is performing as designed from start to finish. The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various system components and systems.”

-- Techopedia --

End-to-end Testing

- Test an application's workflow from beginning to end
 - Including AJAX requests to the backend
- Based on workflows the end user actually executes
 - Using the same user interface, browser, network database etc.

See you in the next video

Personal introduction



Maurice de Beijer

Independent software developer and trainer

The Netherlands



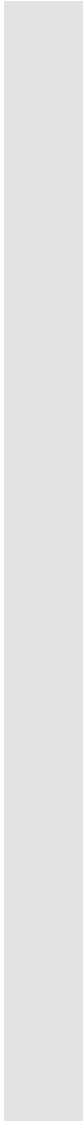







Happily married





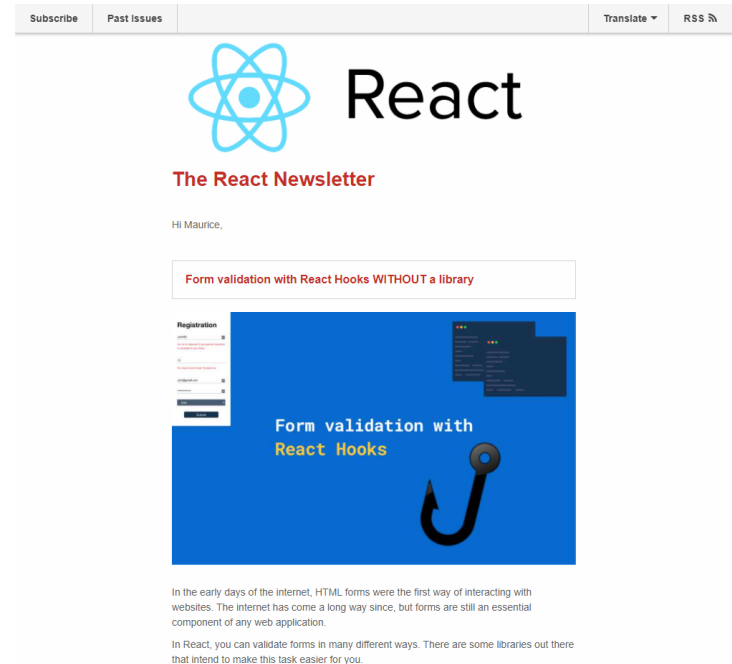
Independent software developer & instructor

Since 1995



**Microsoft[®]
Most Valuable
Professional**

The React Newsletter



See you in the next video

Prerequisites

Install Node & NPM

Install the GitHub starter repository

Following Along



```
1  /// <reference types="cypress" />
2
3  context('Block Buster Film Reviews', () => {
4
5    it('Has the correct title', () => {
6      cy.visit('/')
7      cy.get('h1').should('contain.text', 'Block Buster Film Reviews')
8    })
9
10   it('Can open The Shawshank Redemption', () => {
11     cy.visit('/top-rated-movies')
12     cy.get('#movie-278 a:first').click()
13     cy.get('h1').should('contain.text', 'The Shawshank Redemption (1994)')
14   })
15
16   it('The Shawshank Redemption has the expected cast', () => {
17     cy.visit('/movie/278')
18     cy.get('.col-md-8 > .mvcast-item > #cast-504 > p').should('have.text', '... Andy Dufresne')
19     cy.get('.col-md-8 > .mvcast-item > #cast-192 > p').should('have.text', '... Ellis Boyd "Red" Redding')
20     cy.get('.col-md-8 > .mvcast-item > #cast-192 > .cast-left').should('contain.text', 'Morgan Freeman')
21   })
22
23   it('Can navigate to Morgan Freeman', () => {
24     cy.visit('/movie/278')
25     cy.get('.col-md-8 > .mvcast-item > #cast-192 a').click()
26     cy.get('h1').should('contain.text', 'Morgan Freeman')
27   })
28 })
```

- Slides:
 - <http://theproblemsolver.nl/master-cypress-in-15-minutes-a-day.pdf>

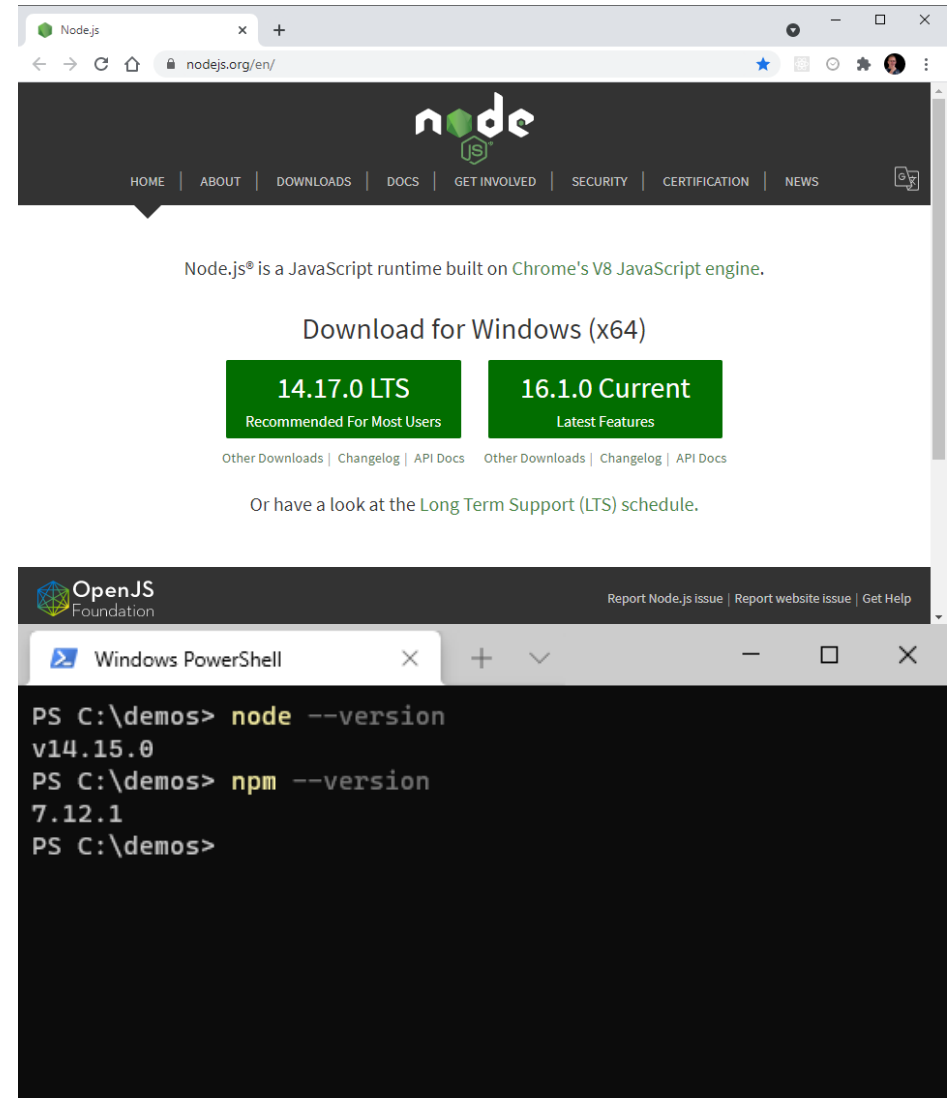
Type it out
by hand?

"Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now!"

Install Node.js



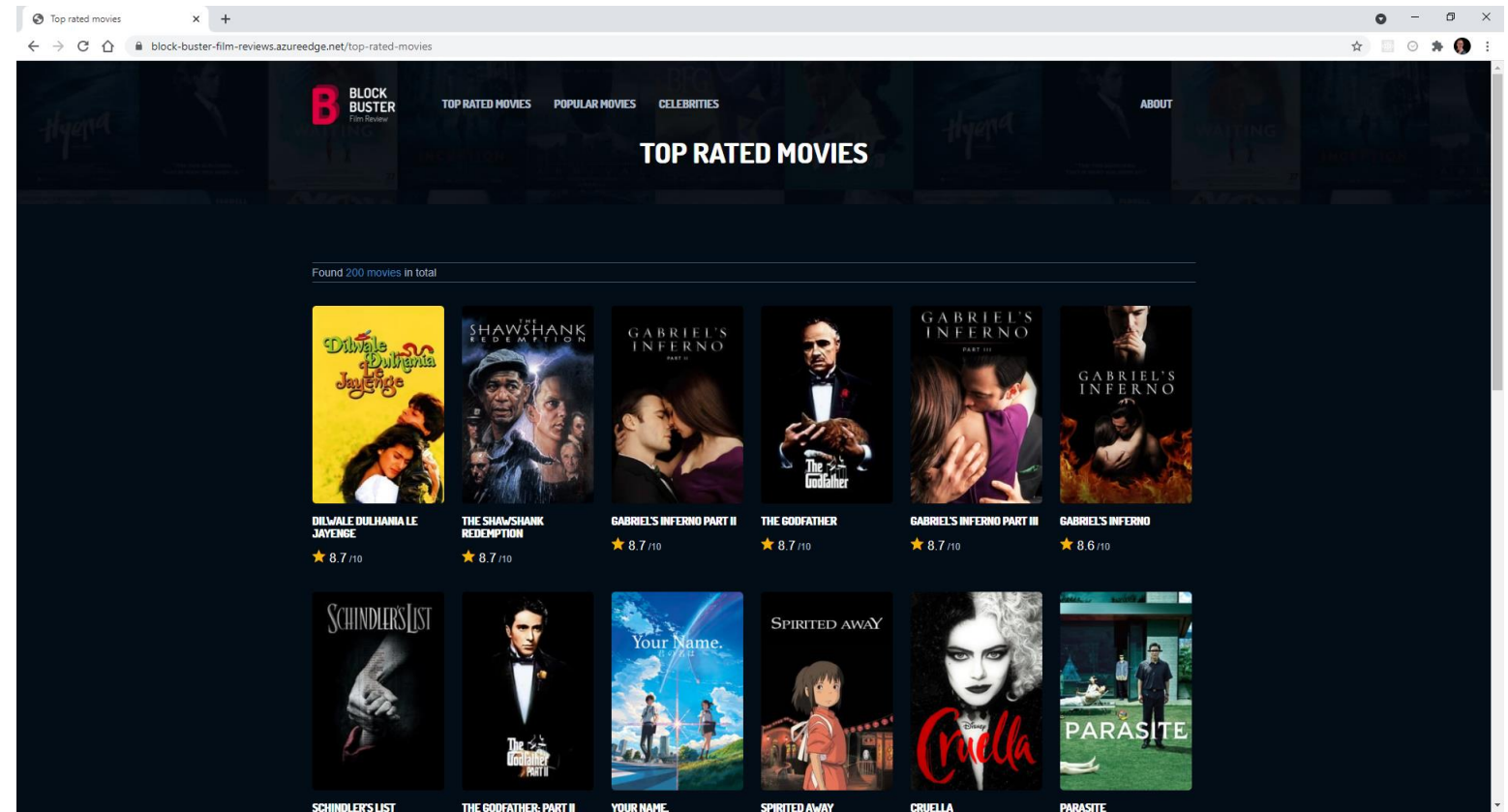
- Minimal:
 - Node version 12



See you in the next video

Creating a first Cypress test

The Application



Opening Cypress

- Start Cypress using “npx cypress open”
 - This will automatically download and run Cypress
- The first time it will create the Cypress a folder structure
 - Including a number of example tests

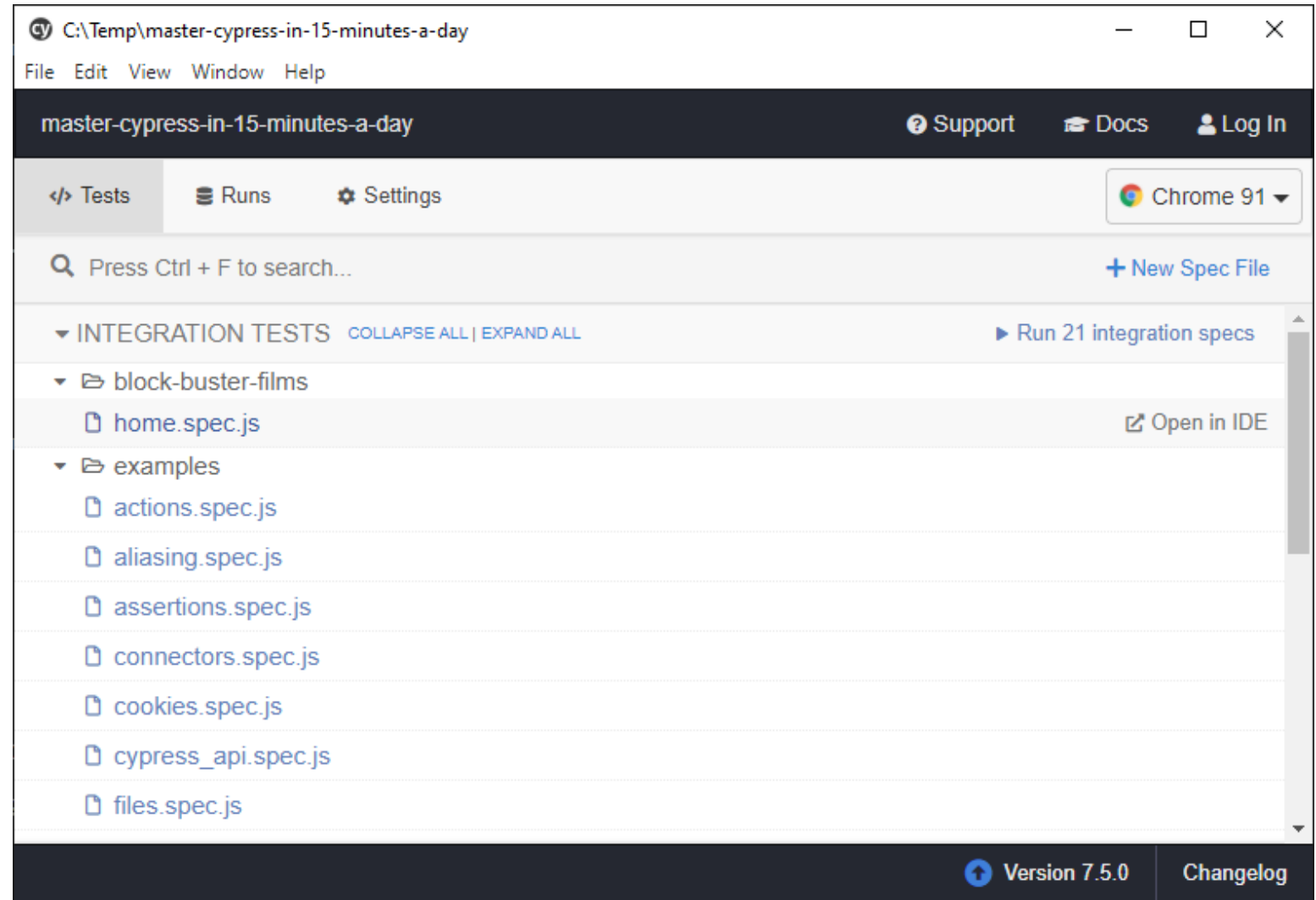
Creating a first Cypress test

- Create a new spec file in the **/cypress/integration** folder
 - The spec file name must end with **spec.js**
- Call the **context()** function to create a group of specification
 - The **describe()** function is an alias for context()
- Call the **it()** function to create a End to End specification
 - Both **context()** and **it()** come from the Mocha test framework

Creating a first Cypress test

```
JS home.spec.js X
cypress > integration > block-buster-films > JS home.spec.js > ...
1 context('Block Buster Film Reviews - Home Page', () => {
2   it('Has the correct title', () => {
3     cy.visit('https://block-buster-film-reviews.azureedge.net');
4
5     cy.title().should('equal', 'Block Buster Film Reviews');
6     cy.get('h1').should('have.text', 'Block Buster Film Reviews');
7   });
8 });
```

Cypress test runner



Running the test



master-cypress-in-15-minutes-a x +

Not secure | block-buster-film-reviews.azureedge.net/_/#/tests/integration/block-buster-films/home.spec.js

Chrome is being controlled by automated test software.

< Tests 1 1 01.34 1000 x 660 (53%)

cypress/integration/block-buster-films/home.spec.js

Block Buster Film Reviews - Home Page

Has the correct title

TEST BODY

- 1 visit https://block-buster-film-reviews.azure...
- 2 title
- 3 - assert expected Block Buster Film Reviews to equal Block Buster Film Reviews
- 4 get h1
- 5 - assert expected <h1> to have text Block Buster Film Reviews

Block Buster Film Reviews



See you in the next video

Adding a package.json

Specifying the Cypress version

- Right now we are always using the **latest version of Cypress**
 - Instead of being explicit about what version to use
- Each developer/tester **has to know** to run “npx cypress open”
 - Maybe look for it in the documentation

Adding a package.json

- When using Node.js the normal way is to **add a package.json**
 - And specify the dependencies with versions in there
 - As well as add important scripts to execute
- **Create a package.json** with the “npm init” command

Adding dependencies

- **Add dependencies** with “npm install <<package>>”
 - Use the --save-dev option to register a development only package
 - Use “npm install cypress --save-dev”

Adding scripts

- Add relevant scripts to the “scripts” section
 - “cypress open”
 - “cypress run”
- Execute with “npm run [script name]”

The package.json



```
{ package.json U x
{ package.json > ...
1  {
2    "name": "master-cypress-in-15-minutes-a-day",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "cypress run",
8      "open": "cypress open"
9    },
10   "repository": {
11     "type": "git",
12     "url": "git+https://github.com/mauricedb/master-cypress-in-15-minutes-a-day.git"
13   },
14   "keywords": [
15     "cypress"
16   ],
17   "author": "Maurice de Beijer",
18   "license": "MIT",
19   "bugs": {
20     "url": "https://github.com/mauricedb/master-cypress-in-15-minutes-a-day/issues"
21   },
22   "homepage": "https://github.com/mauricedb/master-cypress-in-15-minutes-a-day#readme",
23   "devDependencies": {
24     "cypress": "^7.6.0"
25   }
26 }
```

See you in the next video

Fixing a broken Cypress install

A broken Cypress Installation

```
Windows PowerShell
PS C:\Repos\master-cypress-in-15-minutes-a-day> npm run open

> master-cypress-in-15-minutes-a-day@1.0.0 open
> cypress open

No version of Cypress is installed in: C:\Users\mauri\AppData\Local\Cypress\Cache\7.6.0\Cypress

Please reinstall Cypress by running: cypress install

-----

Cypress executable not found at: C:\Users\mauri\AppData\Local\Cypress\Cache\7.6.0\Cypress\Cypress.exe

-----

Platform: win32 (10.0.19042)
Cypress Version: 7.6.0
PS C:\Repos\master-cypress-in-15-minutes-a-day> |
```

Cypress NPM Package

- The Cypress NPM package is not complete
 - It only contains part of the required bits
- The rest is installed into an application cache
 - Once for ever version used

Cypress cache

Binary cache

As of version `3.0`, Cypress downloads the matching Cypress binary to the global system cache, so that the binary can be shared between projects. By default, global cache folders are:

- **MacOS:** `~/Library/Caches/Cypress`
- **Linux:** `~/.cache/Cypress`
- **Windows:** `/AppData/Local/Cypress/Cache`

Cypress cache

- Every version of Cypress is downloaded the first time it's required
 - And reused whenever needed
- Use the “cypress cache” command to manage the Cypress cache
 - `npx cypress cache path`
 - `npx cypress cache list [--size]`
 - `npx cypress cache prune`
 - `npx cypress cache clear`

Cypress cache



```
Windows PowerShell
PS C:\Repos\master-cypress-in-15-minutes-a-day> npx cypress cache path
C:\Users\mauri\AppData\Local\Cypress\Cache
PS C:\Repos\master-cypress-in-15-minutes-a-day> npx cypress cache list --size
```

version	last used	size
6.0.1	26 minutes ago	516.1MB
6.5.0	23 minutes ago	609.0MB
6.9.1	20 minutes ago	579.1MB
7.0.1	17 minutes ago	587.4MB
7.3.0	12 minutes ago	574.9MB
7.5.0	14 minutes ago	575.4MB
7.6.0	unknown	444.3MB

```
PS C:\Repos\master-cypress-in-15-minutes-a-day> |
```

See you in the next video

The most important commands

cy.visit()

- Visit a URL in the browser
 - Takes the URL to navigate to
 - An optional options object with many settings
- <https://docs.cypress.io/api/commands/visit>

cy.get()

- Queries the DOM for one or more elements
 - Uses the jQuery selector engine
- <https://docs.cypress.io/api/commands/get>

.should()

- Check if a given condition is as expected
 - Based on the Chai NPM package
 - Often chain after a cy.get()
 - .and() is an alias
- Check for example
 - If an elements is visible and enabled
 - A value is equal to another
 - Many more...
- <https://docs.cypress.io/api/commands/should>

Asynchronous

- Commands are asynchronous
 - They not executed immediately but placed in a queue

Retry behavior

- Commands like `get()` and `should()` will retry upon failure
 - By default 4 seconds
- The retry timeout can be configured
 - Either globally or per command

Chainable

- Most commands return a Chainable object

Top rated movies Specs



```
JS top-rated-movies.spec.js U x
cypress > integration > block-buster-films > JS top-rated-movies.spec.js > ...
1 // top-rated-movies.spec.js created with Cypress
2 //
3 // Start writing your Cypress tests below!
4 // If you're unfamiliar with how Cypress works,
5 // check out the link below and learn how to write your first test:
6 // https://on.cypress.io/writing-first-test
7
8 context('Top rated movies', () => {
9   it('Has 20 movies per page', () => {
10     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
11
12     cy.get('.movie-item-style-1').should('have.length', 24)
13   })
14
15   it('Has the correct title', () => {
16     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
17
18     cy.title().should('equal', 'Top rated movies')
19     cy.get('h1')
20       .should('be.visible')
21       .should('have.text', 'Top rated movies')
22   })
23
24   it('The movie should be Dilwale Dulhania Le Jayenge', () => {
25     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
26
27     cy.get('#movie-19404 > .mv-item-infor > h6 > a')
28       .should('have.text', 'Dilwale Dulhania Le Jayenge')
29   })
}
```

See you in the next video

Adding IntelliSense

IntelliSense

- **IntelliSense** in VS Code can be added with:
 - `/// <reference types="Cypress" />`
- Uses the **TypeScript type declarations**
 - From the Cypress NPM package

IntelliSense



```
JS home.spec.js M x
cypress > integration > block-buster-films > JS home.spec.js > context('Block Buster Film Reviews') callback > it('Has the correct title - Home Page') callback

You, 2 minutes ago | 1 author (You)
1  /// <reference types="Cypress" />
2
3  context('Block Buster Film Reviews', () => {
4    it('Has the correct title - Home Page', () => {
5      cy.visit('https://block-buster-film-reviews.azureedge.net')
6
7      cy.title().should('equal', 'Block Buster Film Reviews')
8      cy.get('h1').should('have.text', 'Block Buster Film Reviews')
9    })
10 }
11
```

- have.text
- have.nested.property
- have.returned
- have.deep.nested.property
- have.length.gte
- have.length.lte
- have.lengthOf.gte
- have.lengthOf.lte
- have.always.returned
- have.length.greaterThan
- have.length
- have.length.above

See you in the next video

Mocha hooks

Mocha hooks

- Mocha provides **hooks** to run code **before and after tests**
 - The **before()** and **after()** run once per block
 - The **beforeEach()** and **afterEach()** run for every test
- Using the **beforeEach()** is very convenient
 - Use `cy.visit()` to load the right page

Using beforeEach()



```
JS top-rated-movies.spec.js M x
cypress > integration > block-buster-films > JS top-rated-movies.spec.js > ...

10 context('Top rated movies', () => {
11
12   beforeEach(() => {
13     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
14   })
15
16   it('Has 24 movies per page', () => {
17     cy.get('.movie-item-style-1').should('have.length', 24)
18   })
19
20   it('Has the correct title', () => {
21     cy.title().should('equal', 'Top rated movies')
22     cy.get('h1')
23       .should('be.visible')
24       .and('have.text', 'Top rated movies')
25   })
26
27   it('The movie should be Dilwale Dulhania Le Jayenge', () => {
28     cy.get('#movie-19404 > .mv-item-infor > h6 > a')
29       .should('have.text', 'Dilwale Dulhania Le Jayenge')
30   })
31 })
```

See you in the next video

Cypress configuration

Cypress configuration

- The `cypress.json` is used to store **global configuration values**
 - Base URL
 - Environment variables
 - Retries
 - Timeouts
 - Etc.

IntelliSense

- Add a **\$schema** key to get **IntelliSense**
 - <https://on.cypress.io/cypress.schema.json>

Cypress.config()

- The **Cypress.config()** function
 - Retrieve a configuration value
 - Override a configuration value for a single test

Command Line

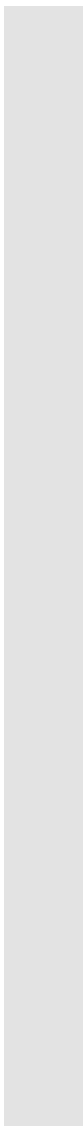
- Use the `--config` option to **override config using the CLI**
 - Both with `cypress run` and `cypress open`

cypress.json



```
cypress.json M x
cypress.json > ...
You, seconds ago | 1 author (You)
1 {
2   "$schema": "https://on.cypress.io/cypress.schema.json",
3   "baseUrl": "https://block-buster-film-reviews.azureedge.net",
4   "defaultCommandTimeout": 10000,
5   "ignoreTestFiles": [
6     "**/1-getting-started/*",
7     "**/2-advanced-examples/*"
8   ]
9 }
```

See you in the next video





See you in the next video





See you in the next video

Maurice de Beijer

@mauricedb

maurice.de.beijer
@gmail.com

