

# End-to-end Vue Testing with Cypress

---

## Getting Started with Cypress



**Marko Vajs**

Software Development Engineer in  
Test



Cypress.io

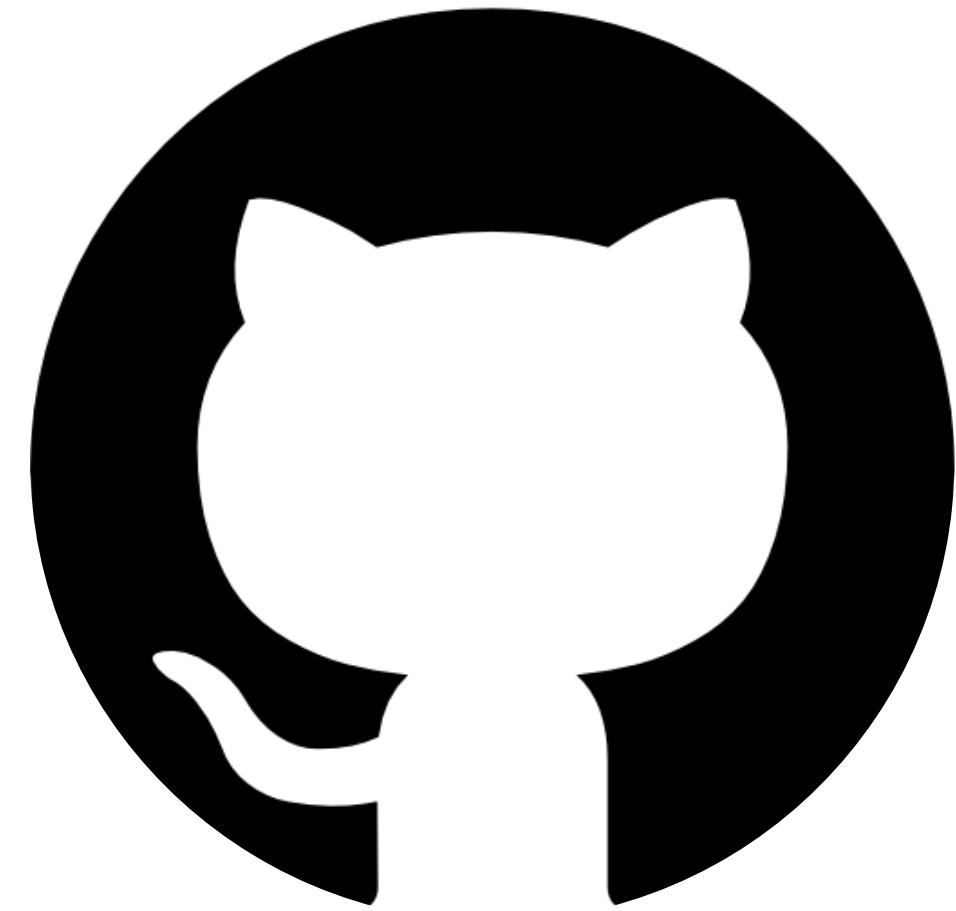
# What Is Cypress?



**Simplifies end-to-end  
testing**



**Based on a  
completely new  
architecture**



**Has a great open-  
source community**

## Course Overview



**Learn in detail about Cypress and the way it works**

**Explore temporary and permanent trade-offs**

**Set up environment and create a Cypress project**

**Learn how to interact with different UI elements and organize testing project**

**Explore topics like conditional testing, retry-ability, sync and async tests, commands and plugins, etc.**

**Set up continuous integration for tests**

**Explore features specific to Vue**

A man and a woman in professional attire stand in the foreground, looking towards the right. They are positioned in front of a dark, slightly blurred background that suggests a group of people. The man is on the left, wearing a dark suit, white shirt, and tie. The woman is on the right, wearing a dark blazer over a light-colored collared shirt and dark pants. Both have their hands in their pockets.

Cypress will help you  
increase your confidence  
in delivering high-quality  
products.

# What Is Cypress and How It Works?

---



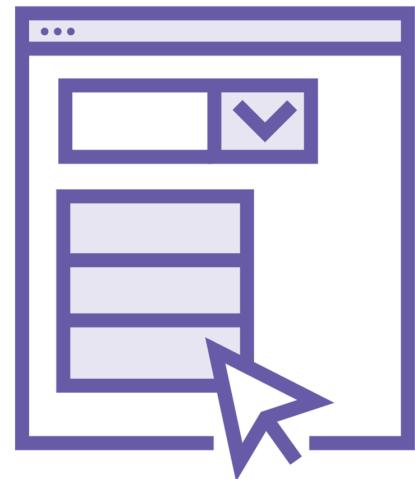
## What is Cypress?

**Cypress is a very popular end-to-end testing framework.**

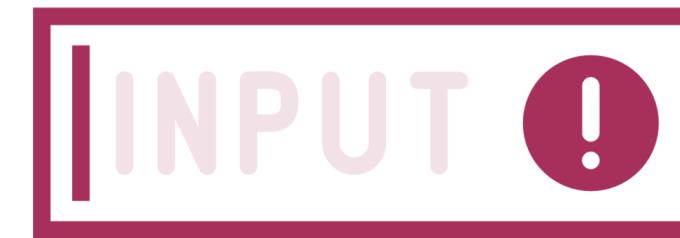
# End-to-end Testing

**A type of testing in which business processes are tested from start to finish under production-like circumstances.**

# Example of Login



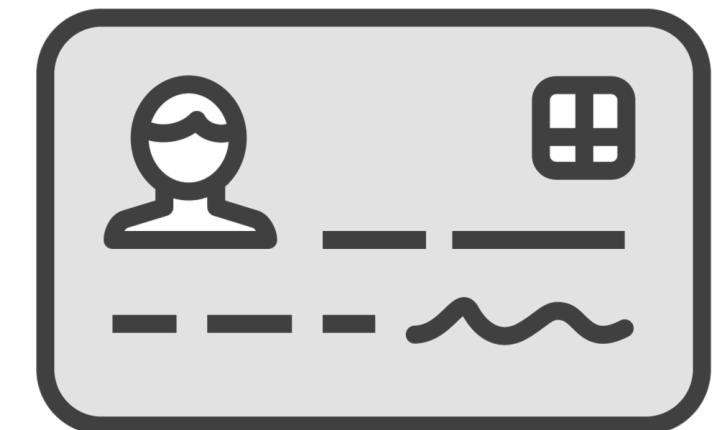
User interface



Validations and  
other protection  
mechanisms



3<sup>rd</sup> party  
identity provider



Authentication  
middleware



UI testing is not the same as  
end-to-end testing.

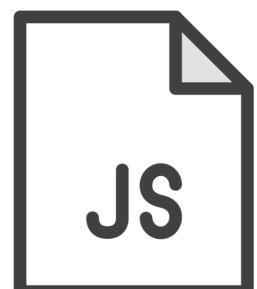
# Cypress



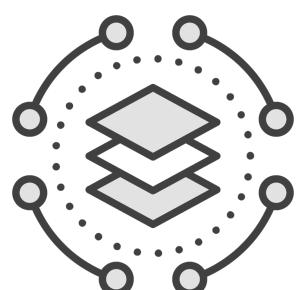
**Offers different features that are combined in a single framework**



**Framework-agnostic tool and can be used to test anything that runs in a web browser**



**Cypress tests are written in JavaScript**



**It has a completely different architecture and runs in the same run-loop as the application that's being tested**

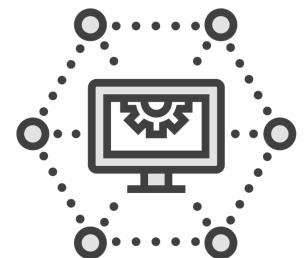
# Exploring Trade-offs

---

Trade-offs can be seen as good boundaries that are there to prevent us from writing bad tests.



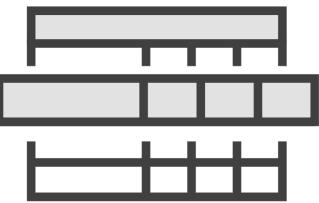
# Permanent Trade-offs



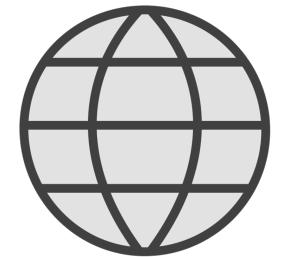
**Cypress is not used to index web pages, spider links, perform performance testing or to script 3<sup>rd</sup> party sites**



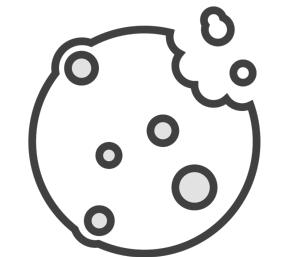
**Communication with back-end like server or database is limited**



**There is no multi-tabs support**

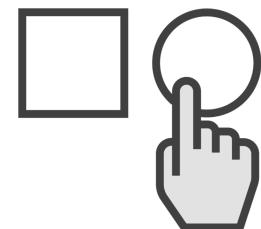


**Cypress does not support multiple browsers open at the same time**

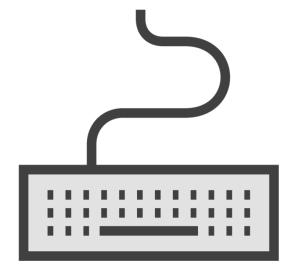


**Cypress tests are limited to the same origin**

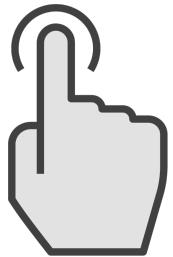
# Temporary Trade-offs



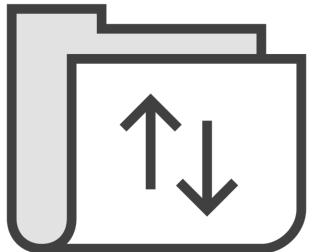
**Workaround is required for a cy.hover() command**



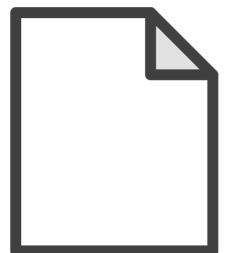
**There is no cy.tab() command**



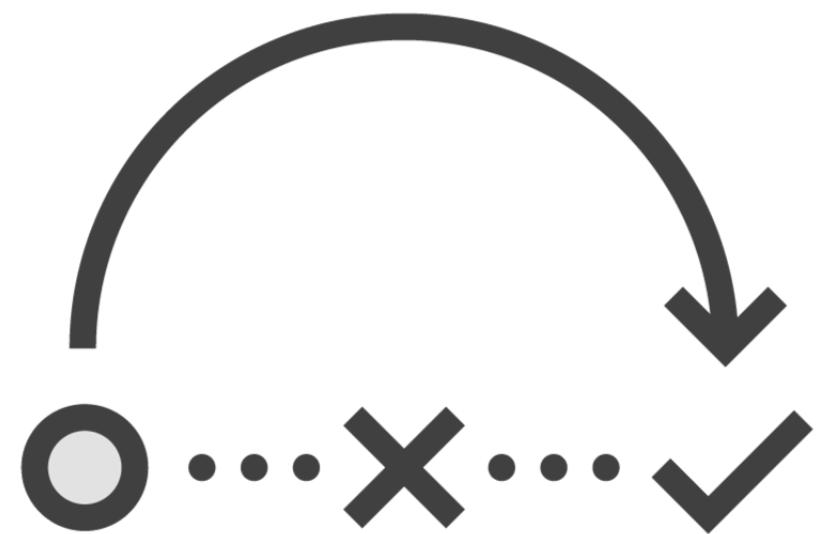
**There is not any native or mobile events support**



**Testing file uploads and downloads are application-specific**



**iframe support is somewhat limited**



## Workarounds

**There is a workaround for every limitation,  
but it's better to re-think the test case design  
before searching for it.**

# Installing Cypress and Configuring Environment

---

# Demo



- 1. Install Node.js**
- 2. Initialize an empty project**
- 3. Add Cypress as a dependency and open it**

# Module Summary

---

# Module Summary



**Cypress is the new standard in front-end testing that has many benefits**

**It has permanent and temporary trade-offs**

**Cypress is designed for end-to-end testing**

**It's easy to get started with Cypress**

Up Next:  
Testing Your Application with Cypress

---