

IntelliSense & Intelligent Code Completion

```
///<reference types ="Cypress" />
```

Cypress Within

```
it.only('Examples of within', () => {  
  cy.visit("https://example.cypress.io/commands/actions")  
  cy.get('.javascript').eq(0).within(() => {  
    cy.get(".hljs-string").should("exist")  
    cy.get(".hljs-comment").should("exist")  
  })  
});
```

Cypress JS constant

const is a keyword in JavaScript that is used to declare a constant variable. A constant variable cannot be reassigned once it is defined.

```
const myURL = "https://www.saucedemo.com/v1/inventory.html"
```

```
it.only('Constant', () => {  
    cy.visit(myURL)  
});
```

Cypress DropDown

Select dropdown

```
it.only('DropDown Concept', () => {  
  cy.visit(myURL).get('select').select('Price (low to high)')  
});
```

Custom Dropdown

```
it.only('DropDown Concept', () => {  
  cy.visit("https://www.ebay.com/")  
  cy.get('#gh-shop-a')  
    .should('be.visible')  
    .and('have.attr', 'aria-controls', 'gh-sbc-o').click()  
    .get('.scnd').should('be.visible').contains("Collectibles").click();  
});
```

Cypress Photo Upload

```
it.only('Photo Upload', () => {  
  cy.visit("https://the-internet.herokuapp.com/upload")  
  cy.get("#file-upload").selectFile("cypress/fixtures/test.png")  
  
  cy.get(".button").click()  
});
```

Cypress File Drag and Drop

1. Install the File upload Plugin

```
npm install --save-dev cypress-file-upload
```

2. Add import in command.js

```
import 'cypress-file-upload';
```

3. Executive this command

```
it.only('File Upload', () => {  
  cy.visit("https://the-internet.herokuapp.com/upload")  
  cy.get("#drag-drop-upload").attachFile("/test.pdf", { subjectType: 'drag-n-drop' })  
});
```

Cypress Photo Upload by attachFile

```
it.only('File Upload', () => {  
  cy.visit("https://the-internet.herokuapp.com/upload")  
  cy.get("#file-upload").attachFile("/test.pdf")  
});
```

Cypress Environment Variable

- **Benefits:** Great for values that need to be checked into source control and remain the same on all machines.
- **Downsides:** Only works for values that should be the same on across all machines.

Cypress.env

In config file

```
const { defineConfig } = require("cypress");

module.exports = defineConfig({
  e2e: {

    setupNodeEvents(on, config) {
      // implement node event listeners here
    },
    env: {
      username: '',
      password: '',
    },
  },
});
```

```
cy.get('[placeholder="Username or Email"]').type(Cypress.env('username'))
```

Cypress.env

```
const username = Cypress.env('username')  
const password = Cypress.env('password')
```

```
cy.get('[placeholder="Username or Email"]').type(username)  
cy.get('[placeholder="Password"]').type(password)
```

Using cypress.env.json

Create a new file name cypress.env.json

```
{  
  "stag_url": "https://example.com",  
  "live_url": "https://docs.cypress.io/api/commands/within"  
}
```

Then in test mention this by using Cypress.env('stag_url')

```
it('json.evn', () => {  
  cy.visit(Cypress.env('live_url'));  
});
```

Implicit Assertion and Explicit Assertion

Implicit Wait and Explicit Wait

Implicit Wait and Explicit Wait

Explicit waits can be useful in certain scenarios, such as waiting for animations to complete or giving time for elements to load. While explicit waits can be effective in some cases, it's generally recommended to use Cypress commands that automatically wait for elements to be available or for conditions to be met.

While Cypress doesn't have a dedicated "implicit wait" command like Selenium, it has built-in mechanisms that achieve similar outcomes:

Cypress's design emphasizes explicit coordination and assertions over implicit waits.

cy.wait(2000) this is explicit wait

Instead of giving time to wait, we can use assertion method to wait for the element

```
cy.get('.some-element').should('be.visible'); // Waits for the  
element to become visible
```

Cypress exception Waiting TimeOut

```
it.only('DropDown Concept', () => {  
  cy.visit("https://www.saucedemo.com/v1/inventory.html")  
  cy.get('.product_label')  
});
```

```
it.only('DropDown Concept', () => {  
  cy.visit("https://www.saucedemo.com/v1/inventory.html")  
  cy.get('.product_labelssssssss')  
});
```

```
it.only('DropDown Concept', () => {  
  cy.visit("https://www.saucedemo.com/v1/inventory.html")  
  cy.get('.product_labelssssss',{ timeout: 1000000})  
});
```

```
defaultCommandTimeout: 10000,
```

Cypress Wait

- Avoid using `cy.wait(300)` like this while waiting for request.
- It makes your system run time more and occupied.
- Instead of Wait will use network intercept.

Cypress Network Intercept and Request

Cypress Network Intercept

```
it.only('Network Intercept Wait', () => {  
  cy.visit('https://hris.infodev.com.np/')  
  cy.get('[placeholder="Username or Email"]').type(username)  
  cy.get('[placeholder="Password"]').type(password)  
  cy.get('[type="submit"]').click()  
  
  cy.intercept("GET", '/Core/Home/DashboardChartResult').as("waitPage")  
  cy.wait("@waitPage")  
  
});
```

```
cy.intercept( 'https://hris.infodev.com.np/').as("waitingPage")  
cy.wait("@waitingPage")
```

Cypress Network Intercept

```
it.only('Network Intercept Wait', () => {  
  cy.visit('https://hris.infodev.com.np/')  
  cy.get('[placeholder="Username or Email"]').type(username)  
  cy.get('[placeholder="Password"]').type(password)  
  cy.get('[type="submit"]').click()  
  
  cy.intercept("GET", '**/NoticeInfo/**').as("waitPages")  
  cy.wait("@waitPages")  
  
});
```

Cypress Network Intercept

```
it('should intercept and read a JSON file', () => {  
    cy.intercept('GET', '**/2024-01-04.json').as('getJSONFile');  
  
    cy.visit('https://www.nytimes.com/games/wordle/index.html')  
  
    cy.get('[data-testid="Play"]').should('have.text', 'Play').click()  
  
    cy.get('.Modal-module_closeIcon__TcEKb').should('be.visible').click()  
  
    cy.wait('@getJSONFile')  
});
```

Any Questions??

Thank You!!