



# **Cypress .should() Command: A Detailed Guide**



# What are Assertions?



An assertion is a way to validate that the application or system under test is functioning as expected. It is a crucial part of the testing process, as they determine whether a test has passed or failed based on its successful functionality.

In Cypress, assertions are used to examine the state and behavior of the application under test

# Should Assertions in Cypress



One of the key features of Cypress is its assertion library which provides functions for writing assertions and `.should()` is one of the widely used chai assertion.

```
.should(chainers)
.should(chainers, value)
.should(chainers, method, value)
.should(callbackFn)
```

# Most common ways to handle assertion in Cypress.

- **Assert based on class value**
- **Assert based on length**
- **Assert based on text**
- **Assert based on value tag**
- **Assert based on HTML attribute**

# Most common ways to handle assertion in Cypress.

- **Assert for checkboxes**
- **Chaining Multiple Assertions**
- **Assert if the text is visible/hidden**
- **Assert if the value is empty**
- **Assert if the URL is correct**
- **Assert if the element is not present**

# Assert based on class value



To perform assertions where the class of an element can change dynamically based on user interactions or other events, you can use Cypress `.should('have.class', 'className')` command.



```
.should('have.class', className)
```

# Assert based on length



To perform assertions where you want to verify the expected number of elements for a specific set of elements, you can use Cypress `.should('have.length',totalLength)` command.



```
.should('have.length', totalCount)
```

# Assert based on attribute



To assert based on attribute, you can use the `should` function with the `have.attr` property. The `have.attr` property allows you to check the value of any attribute on an element.

```
.should('have.attr', attributeName, attributeValue)
```



# Assert for checkboxes



To assert if checkbox is selected or not, you can use cypress should assertion  
`.should('be.checked')` or `.should('be.unchecked')`



```
.should('be.checked')  
.should('not.be.checked')
```



**For a more detailed understanding of `.should()` commands, refer to the below blog**

Blog : <https://www.lambdatest.com/blog/cypress-should-command/>

