



# Dropdowns

---

# Topics:

- What is a dropdown?
- How many types of dropdowns are there?
- How do we handle dropdowns using Selenium WebDriver?
- Learn some methods to handle dropdowns

# What is a dropdown?

- Dropdowns are one of the important web elements in the HTML
- They are used to create an organized way to display different **options** to user

State selection

Select a State ▼

State selection

▼ Select a State  
Alabama  
Alaska  
Arizona  
Arkansas  
California  
Colorado  
Connecticut  
Delaware  
District Of Columbia  
Florida  
Georgia  
Hawaii  
Idaho

# There are 2 types of dropdowns.

1. HTML DROPDOWNS (non-select)
2. SELECT DROPDOWNS

# 1- HTML DROPDOWNS (non-select)

- If a dropdown is created using anything other than `<select>` tag, it is called HTML dropdown.
- We handle just as any other web element.
- We locate. We click.

## 2- SELECT DROPDOWNS

- Select dropdowns are created using `<select>` tags, and the options are created using `<option>` tag

# How to handle Select dropdowns?

- We can NOT just directly locate and click or operate.
- We must use the “Select” class and methods that comes with Selenium Library.

# Syntax and order

1. We locate the select dropdown

```
WebElement dropdown = driver.findElement(By.id("dropdown"));
```

2. We create an object of "Select" (from Selenium library)
3. We pass the already located web element into the constructor of Select object.

```
Select select = new Select(dropdown);
```

4. Using the object, we will be able to reach the methods coming from this.

```
select.selectByIndex(1);
```



## Preferred way:

- Give the name of the dropdown to Select object
- Pass the WebElement into the Select constructor
- Instead of creating everything in 2 lines, you create in line.
- More organized and easier to keep track of things when it comes to dropdowns.

```
Select dateDropdown = new Select(driver.findElement(By.id("dropdown")));
```

# What would you want to verify on a dropdown?

- What is selected by default?
- How many options are there?
- When you select something, is it actually selected?
- Select multiple options at once, and verify? (only some dropdowns)

# How do you get default selected value?

- `.getFirstSelectedOption();`
- This method allows us to get default selected option
- We can also verify currently selected option after selecting something else.
- Return type: `WebElement`
- Syntax:

```
dateDropdown.getFirstSelectedOption();
```

# How do you get all the options from a dropdown?

- .getOptions(); method
- Returns all of the options in the dropdown.
- Return type : List<WebElement>
- Syntax:

```
List<WebElement> allOptions = dateDropdown.getOptions();
```

# How do you select an option using select object?

- There are 3 methods provided from Select class.
- Therefore we have 3 different way of selecting options.
  1. By text
  2. By value
  3. By Index

# #1- By text

- We use `.selectByVisibleText()` method
- Whatever text the `<option>` has can be used to select the option
- This method accepts a String argument as the text of the goal option
- Syntax:

```
dateDropdown.selectByVisibleText("2020");
```



2022  
2021  
✓ 2020  
2019  
2018  
2017  
2016  
2015  
2014  
2013  
2012

```
<option value="2020">2020</option>
```

## #2- By value

- We use `.selectByValue()` method
- Whatever value of attribute “value” the `<option>` has can be used to select the option
- This method accepts a String argument as the text of the goal option
- Syntax:

```
dateDropdown.selectByValue("2020");
```



2022  
2021  
✓ 2020  
2019  
2018  
2017  
2016  
2015  
2014  
2013  
2012

```
<option value="2020">2020</option>
```

## #3- By index

- We use `.selectByIndex()` method
- Whatever index number the `<option>` has can be used to select the option
- This method accepts a `int` argument as the text of the goal option
- Syntax:

```
dateDropdown.selectByIndex(2);
```

2022	0
2021	1
✓ 2020	2
2019	3
2018	4
2017	
2016	
2015	
2014	
2013	
2012	

```
<option value="2020">2020</option>
```