

Mouse

The Mouse class operates in main-frame CSS pixels relative to the top-left corner of the viewport.

Every `page` object has its own Mouse, accessible with `page.mouse`.

Sync

Async

```
# using 'page.mouse' to trace a 100x100 square.
page.mouse.move(0, 0)
page.mouse.down()
page.mouse.move(0, 100)
page.mouse.move(100, 100)
page.mouse.move(100, 0)
page.mouse.move(0, 0)
page.mouse.up()
```

Methods

click

Added in: v1.8

Shortcut for `mouse.move()`, `mouse.down()`, `mouse.up()`.

Usage

```
mouse.click(x, y)
mouse.click(x, y, **kwargs)
```

Arguments

- `x` float
- `y` float

- `button` "left"|"right"|"middle" (*optional*)

Defaults to `left`.

- `click_count` `int` (*optional*)

defaults to 1. See [UIEvent.detail](#).

- `delay` `float` (*optional*)

Time to wait between `mousedown` and `mouseup` in milliseconds. Defaults to 0.

dblclick

Added in: v1.8

Shortcut for [mouse.move\(\)](#), [mouse.down\(\)](#), [mouse.up\(\)](#), [mouse.down\(\)](#) and [mouse.up\(\)](#).

Usage

```
mouse.dblclick(x, y)
mouse.dblclick(x, y, **kwargs)
```

Arguments

- `x` `float`
- `y` `float`
- `button` "left"|"right"|"middle" (*optional*)

Defaults to `left`.

- `delay` `float` (*optional*)

Time to wait between `mousedown` and `mouseup` in milliseconds. Defaults to 0.

down

Added in: v1.8

Dispatches a `mousedown` event.

Usage

```
mouse.down()  
mouse.down(**kwargs)
```

Arguments

- `button` "left"|"right"|"middle" (*optional*)

Defaults to `left`.

- `click_count` `int` (*optional*)

defaults to 1. See [UIEvent.detail](#).

move

Added in: v1.8

Dispatches a `mousemove` event.

Usage

```
mouse.move(x, y)  
mouse.move(x, y, **kwargs)
```

Arguments

- `x` `float`
- `y` `float`
- `steps` `int` (*optional*)

Defaults to 1. Sends intermediate `mousemove` events.

up

Added in: v1.8

Dispatches a `mouseup` event.

Usage

```
mouse.up()  
mouse.up(**kwargs)
```

Arguments

- `button` `"left"|"right"|"middle"` (*optional*)

Defaults to `left`.

- `click_count` `int` (*optional*)

defaults to 1. See [UIEvent.detail](#).

wheel

Added in: v1.15

Dispatches a `wheel` event.

NOTE

Wheel events may cause scrolling if they are not handled, and this method does not wait for the scrolling to finish before returning.

Usage

```
mouse.wheel(delta_x, delta_y)
```

Arguments

- `delta_x` `float`

Pixels to scroll horizontally.

- `delta_y` `float`

Pixels to scroll vertically.