

Keyboard class

Keyboard provides an api for managing a virtual keyboard. The high level api is [Keyboard.type\(\)](#), which takes raw characters and generates proper keydown, keypress/input, and keyup events on your page.

Signature:

```
export declare class Keyboard
```

Remarks

For finer control, you can use [Keyboard.down\(\)](#), [Keyboard.up\(\)](#), and [Keyboard.sendCharacter\(\)](#) to manually fire events as if they were generated from a real keyboard.

On MacOS, keyboard shortcuts like `⌘ A` -> Select All do not work. See [#1313](#).

The constructor for this class is marked as internal. Third-party code should not call the constructor directly or create subclasses that extend the `Keyboard` class.

Example 1

An example of holding down `Shift` in order to select and delete some text:

```
await page.keyboard.type('Hello World!');
await page.keyboard.press('ArrowLeft');

await page.keyboard.down('Shift');
for (let i = 0; i < ' World'.length; i++)
  await page.keyboard.press('ArrowLeft');
await page.keyboard.up('Shift');

await page.keyboard.press('Backspace');
// Result text will end up saying 'Hello!'
```

Example 2

An example of pressing `A`

```
await page.keyboard.down('Shift');
await page.keyboard.press('KeyA');
await page.keyboard.up('Shift');
```

Methods

Method	Modifiers	Description
down(key, options)		Dispatches a <code>keydown</code> event.

press(key_options)		Shortcut for Keyboard.down() and Keyboard.up() .
sendCharacter(char)		Dispatches a <code>keypress</code> and <code>input</code> event. This does not send a <code>keydown</code> or <code>keyup</code> event.
type(text_options)		Sends a <code>keydown</code> , <code>keypress/input</code> , and <code>keyup</code> event for each character in the text.
up(key)		Dispatches a <code>keyup</code> event.