

[Home](#) > [puppeteer](#) > [Keyboard](#) > [down](#)

## Keyboard.down() method

Dispatches a `keydown` event.

### Signature:

```
down(key: KeyInput, options?: {  
  text?: string;  
}): Promise<void>;
```

## Parameters

Parameter	Type	Description
key	<a href="#">KeyInput</a>	Name of key to press, such as <code>ArrowLeft</code> . See <a href="#">KeyInput</a> for a list of all key names.
options	{ text?: string; }	An object of options. Accepts text which, if specified, generates an input event with this text.

### Returns:

Promise<void>

## Remarks

If `key` is a single character and no modifier keys besides `Shift` are being held down, a `keypress` / `input` event will also generated. The `text` option can be specified to force an input event to be generated. If `key` is a modifier key, `Shift`, `Meta`, `Control`, or `Alt`, subsequent key presses will be sent with that modifier active. To release the modifier key, use [Keyboard.up\(\)](#).

After the key is pressed once, subsequent calls to [Keyboard.down\(\)](#) will have [repeat](#) set to true. To release the key, use [Keyboard.up\(\)](#).

Modifier keys DO influence [Keyboard.down\(\)](#). Holding down `Shift` will type the text in upper case.