

# Accelerator

*Define keyboard shortcuts.*

Accelerators are strings that can contain multiple modifiers and a single key code, combined by the `+` character, and are used to define keyboard shortcuts throughout your application.

Examples:

- `CommandOrControl+A`
- `CommandOrControl+Shift+Z`

Shortcuts are registered with the [globalShortcut](#) module using the [register](#) method, i.e.

```
const { app, globalShortcut } = require('electron')

app.whenReady().then(() => {
  // Register a 'CommandOrControl+Y' shortcut listener.
  globalShortcut.register('CommandOrControl+Y', () => {
    // Do stuff when Y and either Command/Control is pressed.
  })
})
```

## Platform notice

On Linux and Windows, the `Command` key does not have any effect so use `CommandOrControl` which represents `Command` on macOS and `Control` on Linux and Windows to define some accelerators.

Use `Alt` instead of `Option`. The `Option` key only exists on macOS, whereas the `Alt` key is available on all platforms.

The `Super` (or `Meta`) key is mapped to the `Windows` key on Windows and Linux and `Cmd` on macOS.

## Available modifiers

- `Command` (or `Cmd` for short)
- `Control` (or `Ctrl` for short)
- `CommandOrControl` (or `CmdOrCtrl` for short)
- `Alt`
- `Option`
- `AltGr`
- `Shift`
- `Super`
- `Meta`

## Available key codes

- `0` to `9`
- `A` to `Z`
- `F1` to `F24`

- Punctuation like `~`, `!`, `@`, `#`, `$`, etc.
- `Plus`
- `Space`
- `Tab`
- `Capslock`
- `Numlock`
- `Scrolllock`
- `Backspace`
- `Delete`
- `Insert`
- `Return` (or `Enter` as alias)
- `Up`, `Down`, `Left` and `Right`
- `Home` and `End`
- `PageUp` and `PageDown`
- `Escape` (or `Esc` for short)
- `VolumeUp`, `VolumeDown` and `VolumeMute`
- `MediaNextTrack`, `MediaPreviousTrack`, `MediaStop` and `MediaPlayPause`
- `PrintScreen`
- NumPad Keys
  - `num0` - `num9`
  - `numdec` - decimal key
  - `numadd` - numpad `+` key
  - `numsub` - numpad `-` key
  - `nummult` - numpad `*` key
  - `numdiv` - numpad `÷` key