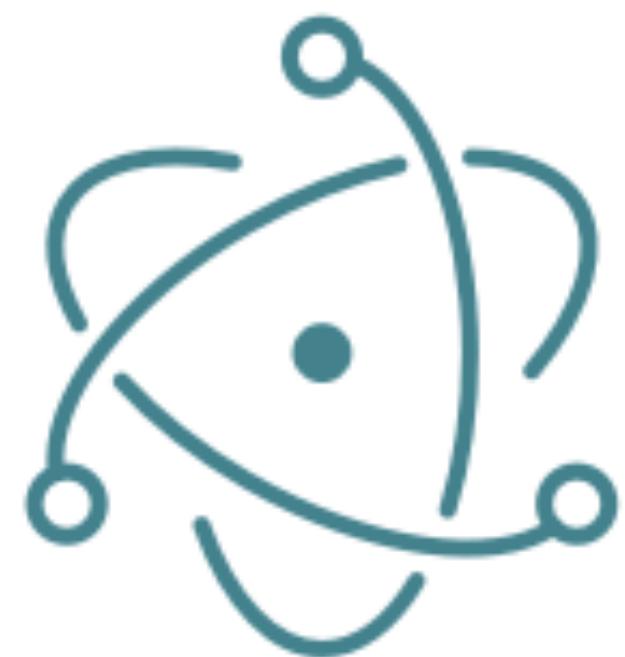
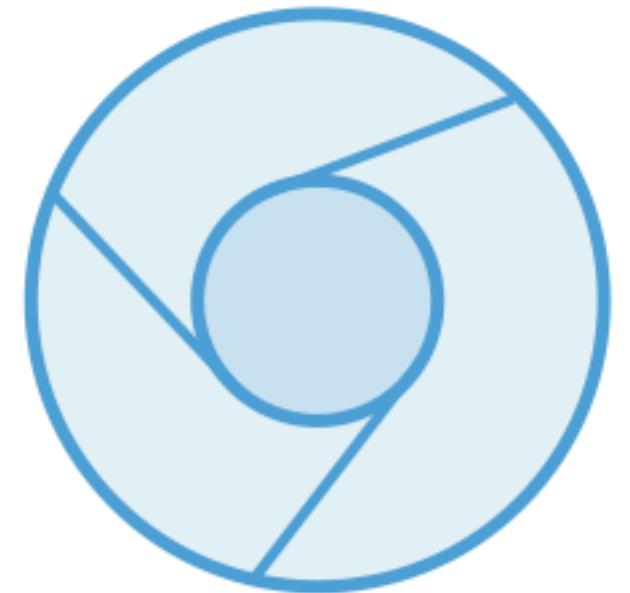


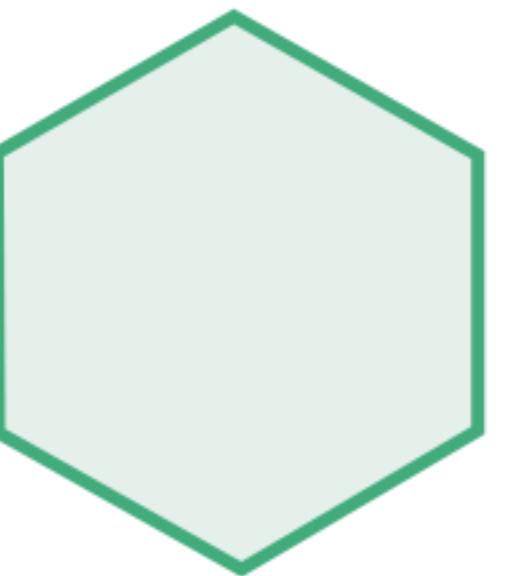
Make your Electron app feel at home everywhere



ELECTRON



+



+



=



Chromium
for making
web pages

Node.js
for filesystems
and networks

Native APIs
for three
systems

ELECTRON

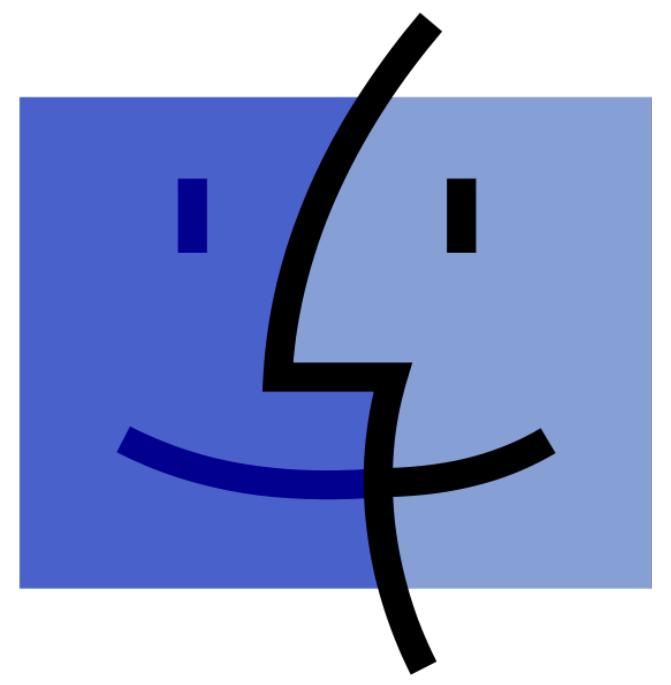
```
const { app, BrowserWindow } = require('electron');

app.whenReady().then(() => {
  let mainWindow = new BrowserWindow({
    width: 800,
    height: 600
  });

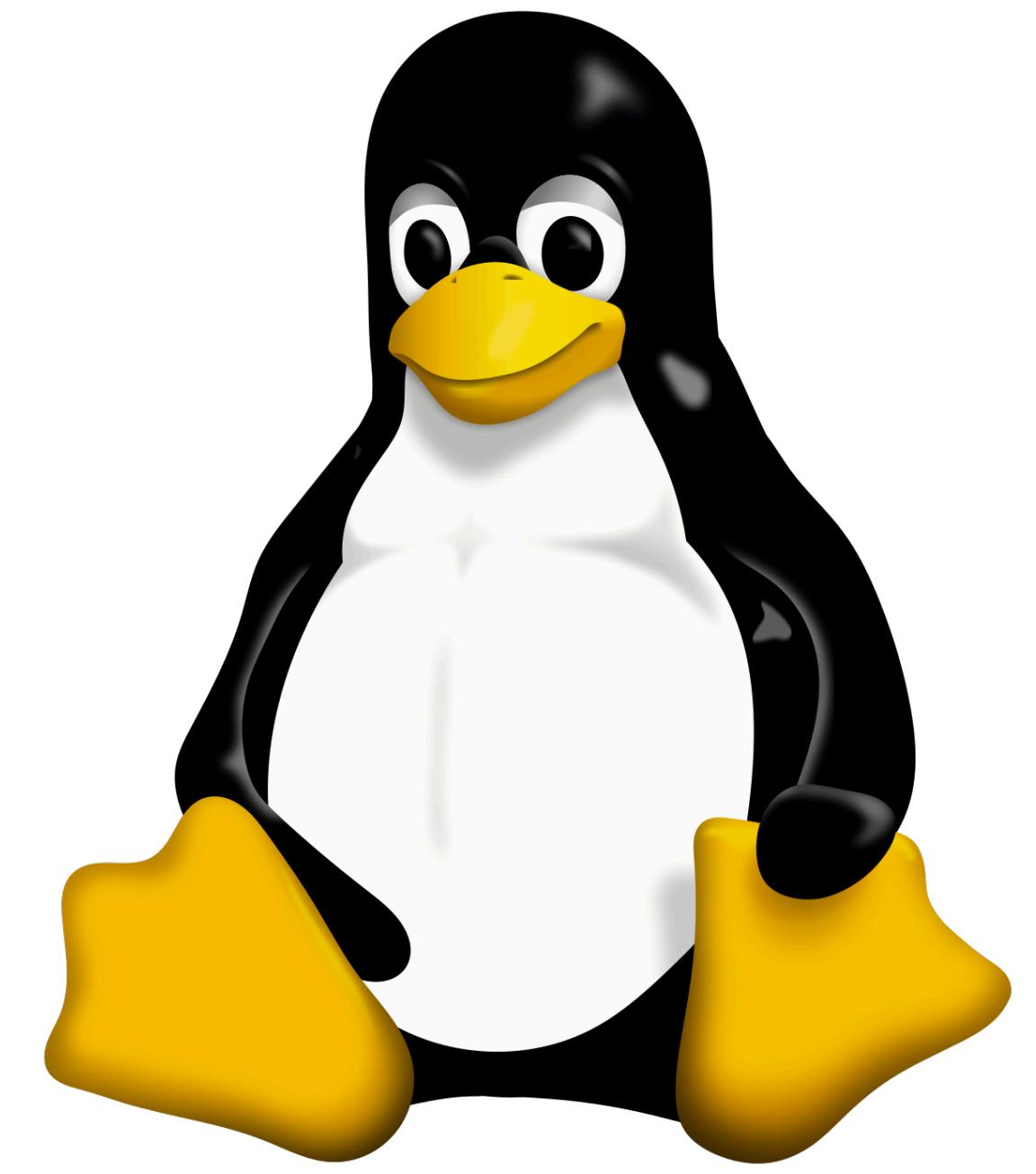
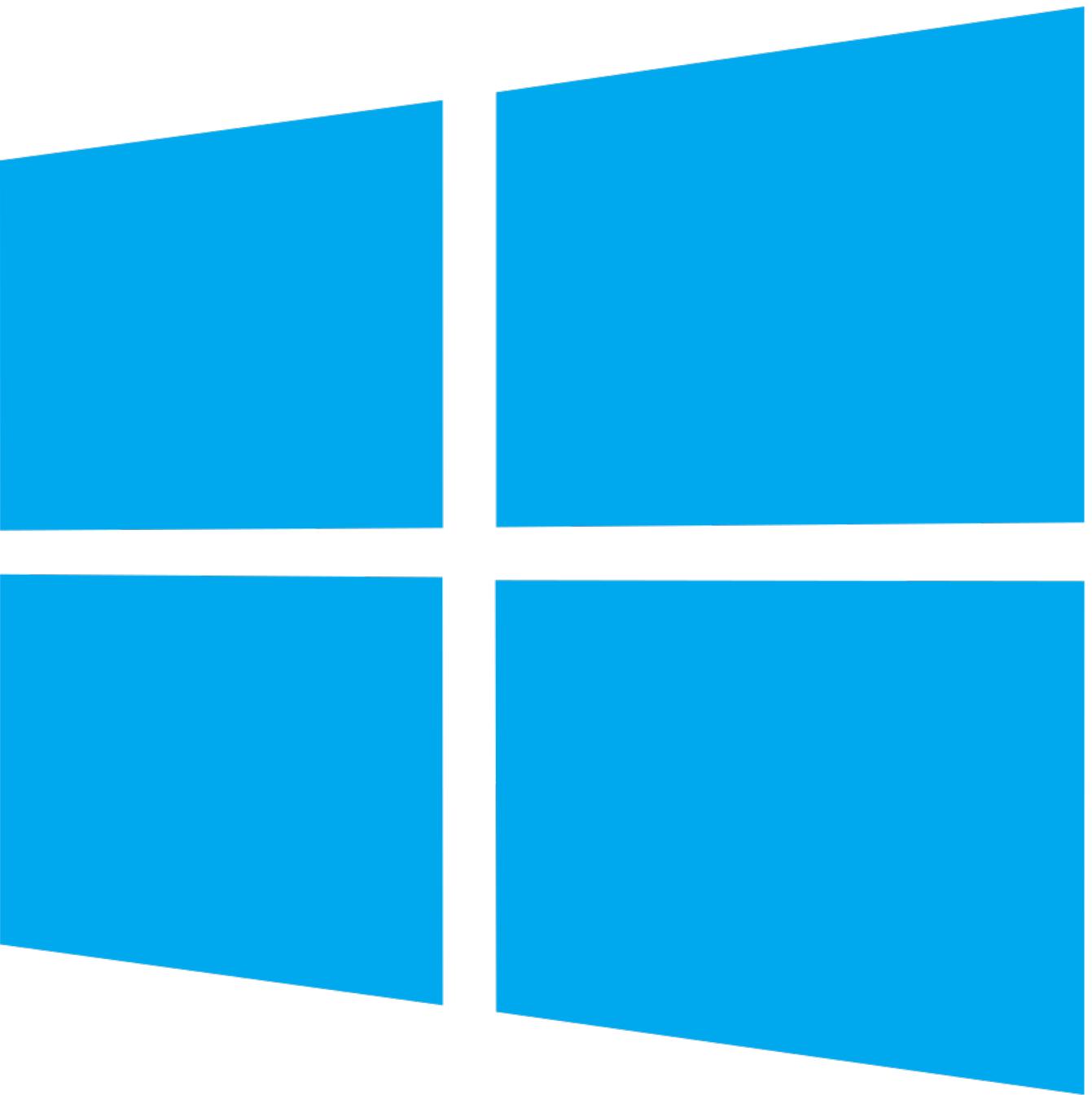
  mainWindow.loadURL('https://www.conf42.com/js2021');
}

app.on('window-all-closed', () => {
  app.quit();
});
```

\$ electron index.js



Mac OS



[@gmail.com](#) | [Report bug](#) | [Contacts](#) | [Settings](#) | [Help](#) | [Sign out](#)

[Compose Mail](#) [Archive](#) [More actions...](#)

Inbox

[Starred](#) [Sent Mail](#) [All Mail](#) [Spam](#) [Trash](#)

Labels

- [quirk](#)
- [Orkut](#)
- Prius (18)**
- [Edit labels](#)

Screenshot [Inbox](#) [Apply label...](#)

Jason Shellen Kevin, Please send me screenshots... 3:31pm (7 minutes ago)

Kevin Fox to Jason [More options](#) 3:37pm (1 minute ago)

Don't worry, I'm sure nobody'll find out that you're using work resources to serve these images from [shellen.com](#). Just in case they cut you off though, I'll post the screenshots on my blog at [fury.com](#) as well.

Seeya later,
Kevin
[fury.com](#)

- Show quoted text -

[Reply](#) [Forward](#)

[« Back to Inbox](#) [Archive](#) [More actions...](#)

You are currently using 0 MB (0%) of your 1000 MB.

Shortcuts: **o**-open **y**-archive **c**-compose **j**-older **k**-newer [more »](#)

[Terms of Use](#) - [Privacy Policy](#) - [Program Policies](#) - [Google Home](#)

©2004 Google

1 of 25 Older

[Open in new window](#)
[Print conversation](#)
[Expand all](#)

Related Pages

[Mac OS X Hosting - Apache Hosting - Macintosh Web Site Hosting ...](#)
Offers affordable Mac OS X Apache web server unix hosting - BSD unix ... [www.inno-tech.com](#)

[Rivercom - Web Hosting - Web Development - Multimedia Development ...](#)
We combine original graphic design, intuitive navigational controls, ... [www.rivercom.com](#)

[About these links](#)

Closing our deal

Laura Per our conversation, I am pleased to present you with a term sheet... FEB 20

Me Thanks Laura, I appreciate the speed — will review. FEB 20

Laura

Sounds good. Do let me know if you have any questions.

Draft to Laura Shea

Hi Laura — I think we're ready! Can we have a quick call in the next 2 days?

SUPERHUMAN

Laura Shea
[laura@uxventures.com](#)
San Francisco

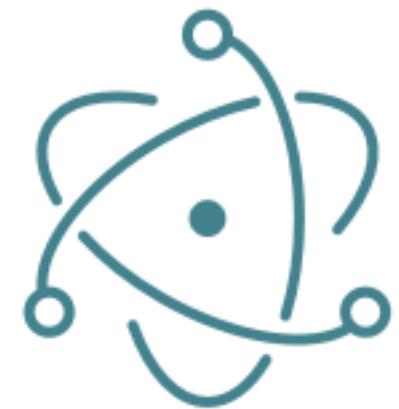
Early stage VC focused on dormant but massive markets

Mail
Diligence list
Your thoughts on this article
Re: messaging & bots

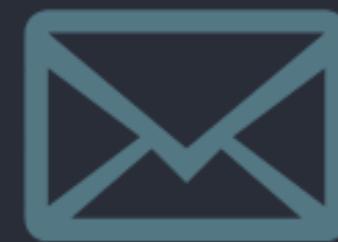
LinkedIn
Facebook
AngelList
@laurashea

Our portfolio @uxventures is ramping hard – dm if you know a great senior product manager

Very excited to announce our self-driving investment @driverless! get on the list at [driverless.com](#)



ELECTRON

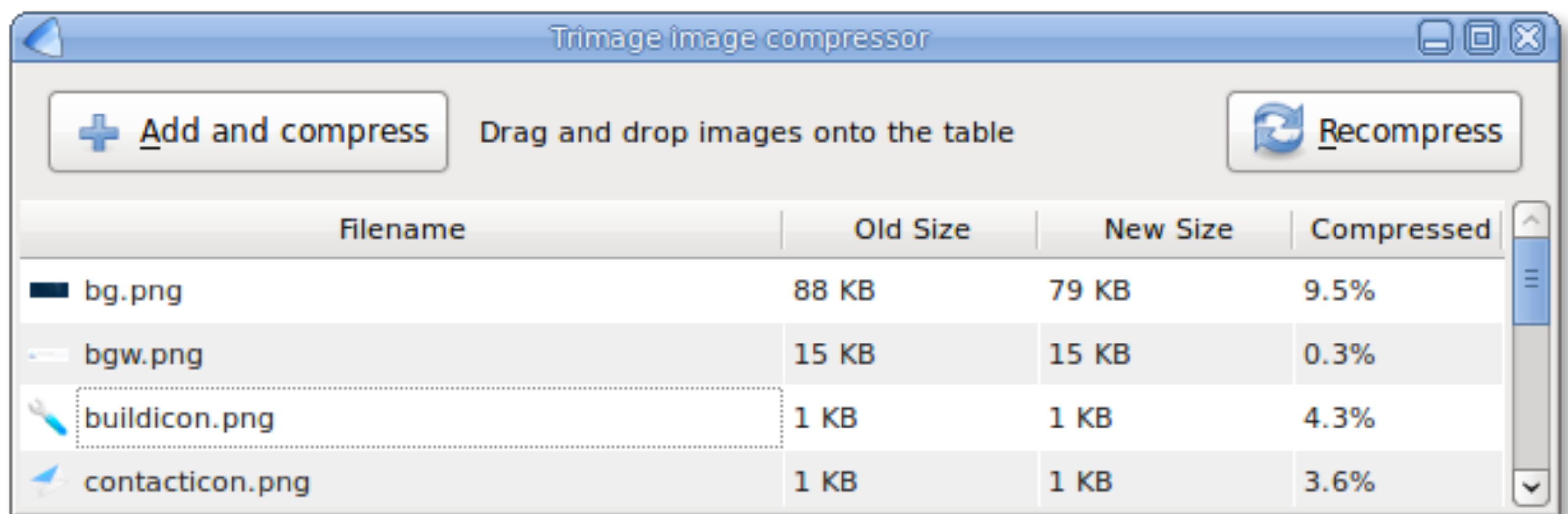


Kilian Valkhof

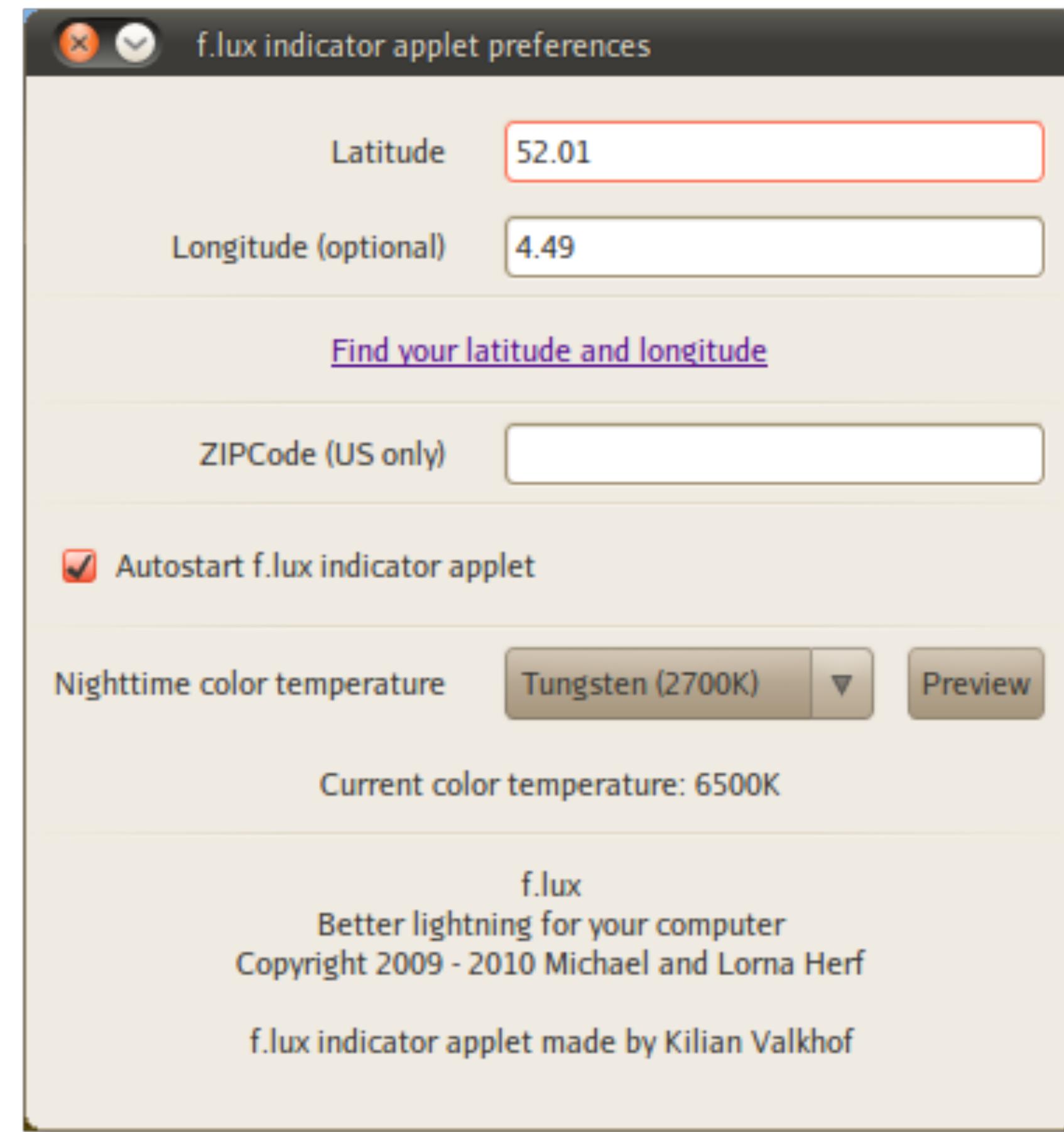


kilianvalkhof.com | @kilianvalkhof





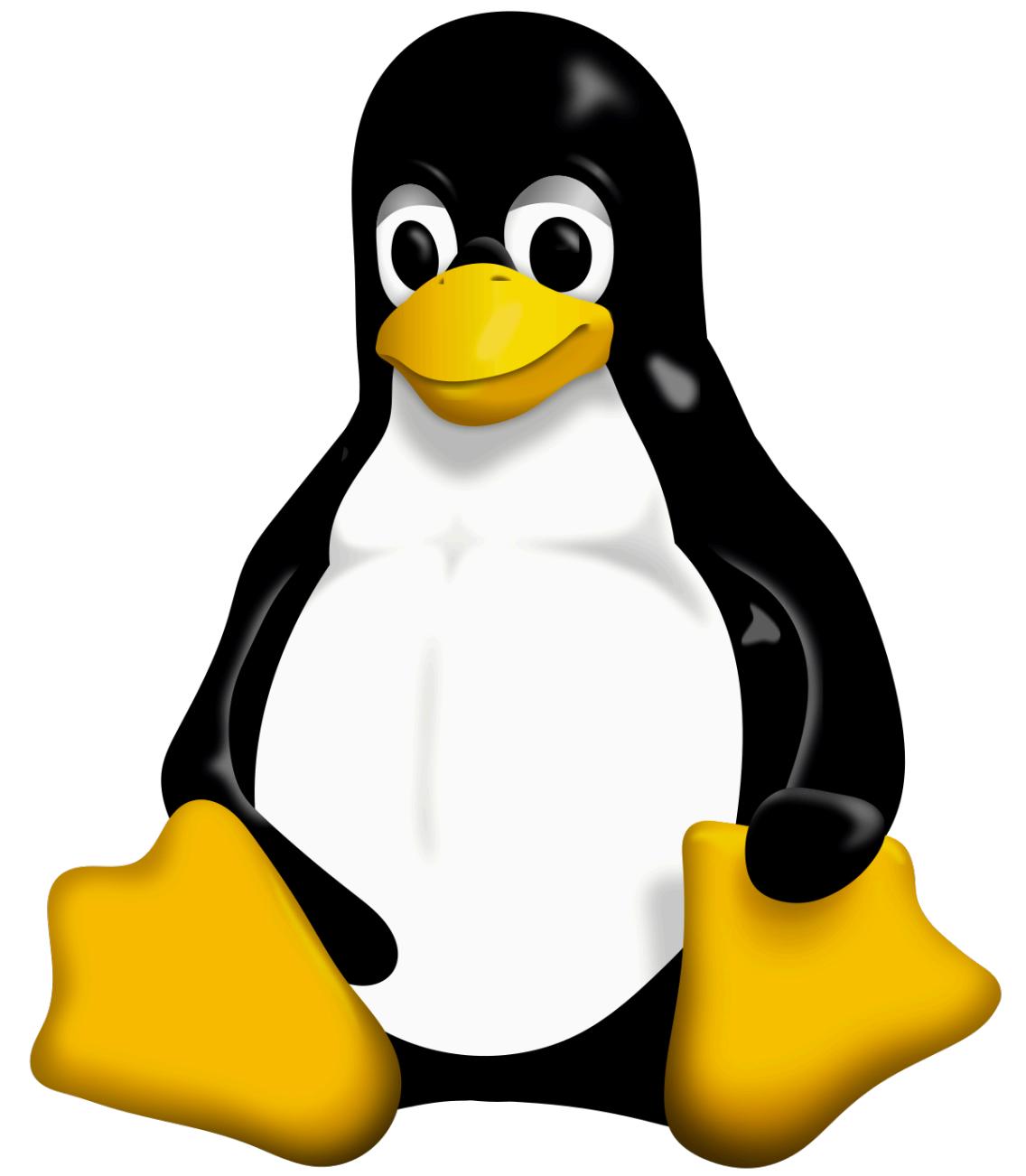
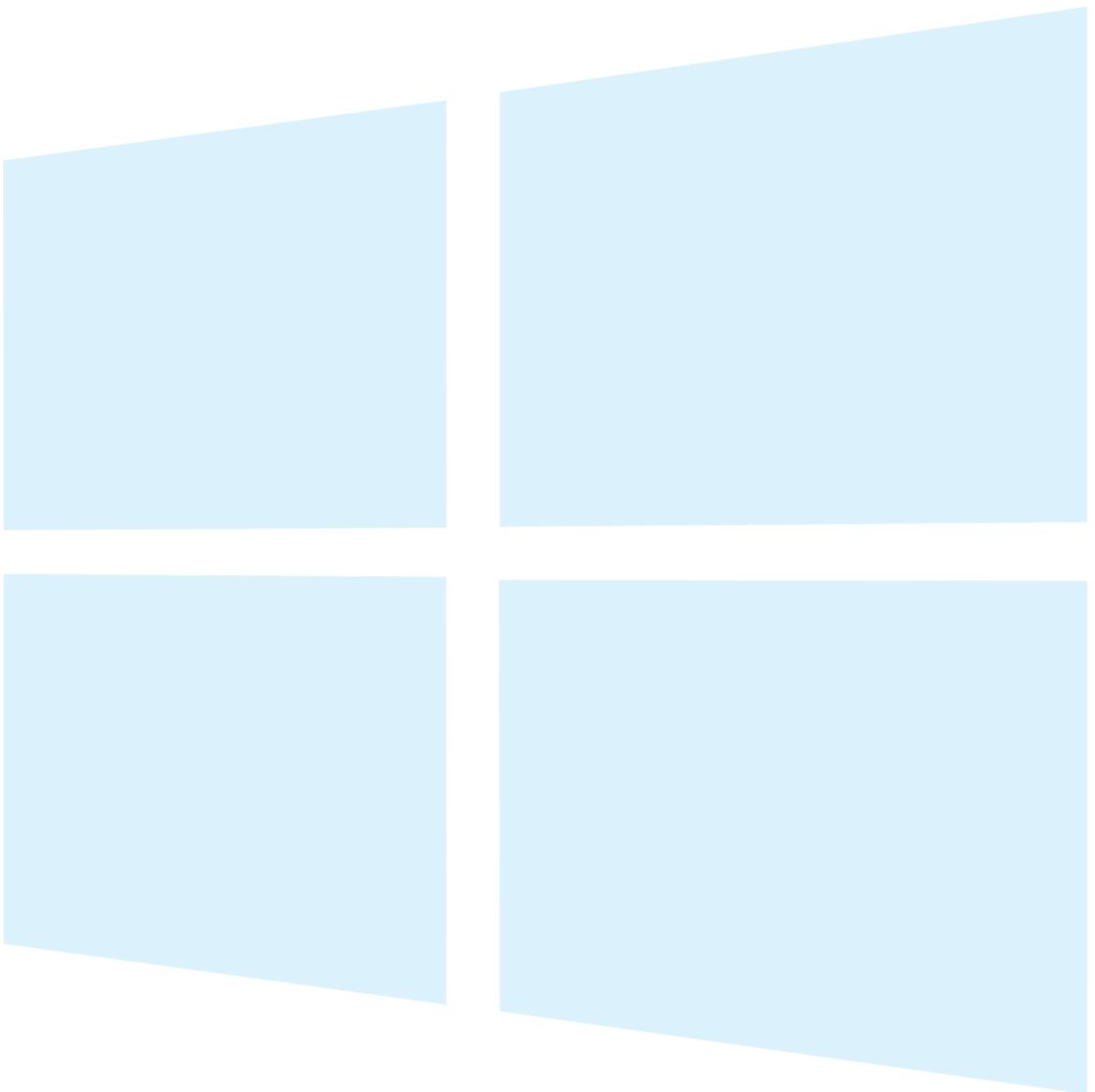
Qt



GTK



Mac OS



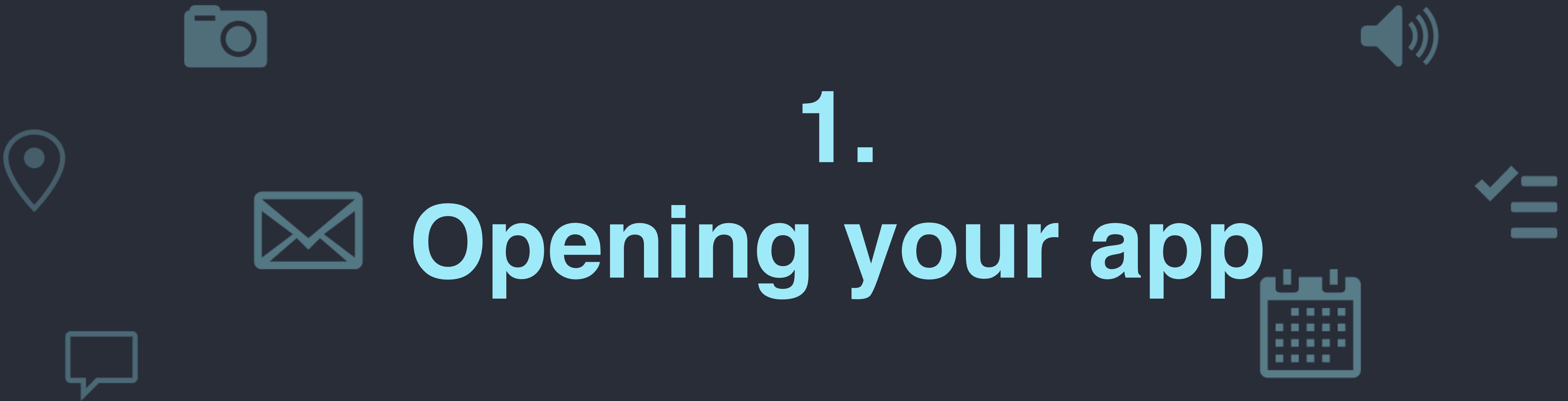
The image displays four distinct application windows arranged horizontally:

- Messenger Demo Viewer:** A window titled "Messenger Demo Viewer" showing a "Messenger Adventure". It contains several text messages from a character named "A" and a blue button labeled "Descend staircase".
- Terminal Window:** A window titled "FromScratch" showing a terminal session. The user has typed "ls" and is viewing the contents of a "tmp" directory, which is currently empty.
- Color Palette Tool:** A window titled "Color" from the website smashingmagazine.com. It displays a grid of color swatches with names and RGB values. The categories include Color, Typography, Spacings, Border Radii, Shadows, and Motion. An "Export" menu is visible at the top right.
- Responsive Web Design Browser:** A window titled "Polypane" showing a responsive web design interface. It displays three mobile device emulators (iPhone SE, iPhone 7/8, iPad) and a desktop view of the same website. The interface includes a navigation bar, a main content area with promotional text and calls-to-action, and a footer with device icons.



8 ways to make your Electron app feel great on all three platforms

1. Opening your app



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

app.whenReady().then(() => {
  let mainWindow = new BrowserWindow({
    width: 800,
    height: 600
  });

  mainWindow.loadURL('https://www.conf42.com/js2021');
}

app.on('window-all-closed', () => {
  app.quit();
});
```

\$ electron index.js

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

app.whenReady().then(() => {
  let mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    show: false
  });

  mainWindow.once('ready-to-show', () => {
    mainWindow.show();
    mainWindow.focus();
  })

  mainWindow.loadURL('https://www.conf42.com/js2021');
})
```

Hide the app until the page has loaded

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

app.whenReady().then(() => {
  let mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    show: false,

    backgroundColor: "#fafafa",
  });

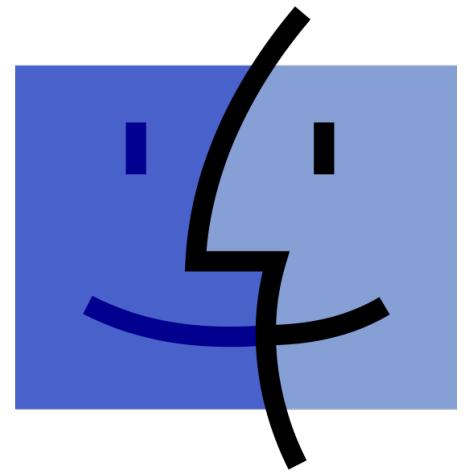
  mainWindow.once('ready-to-show', () => {
    mainWindow.show();
    mainWindow.focus();
  })

  mainWindow.loadURL('https://www.conf42.com/js2021');
})
```

Use a background color that matches your app

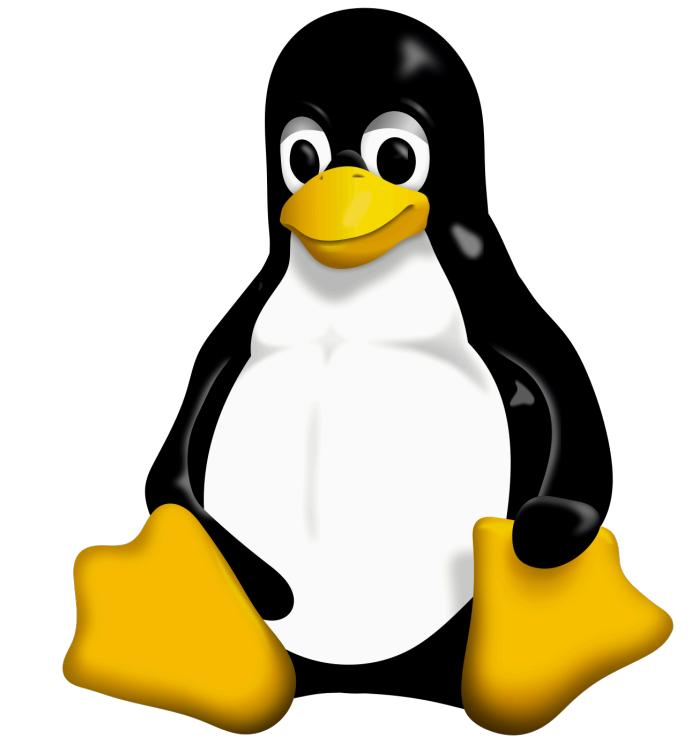
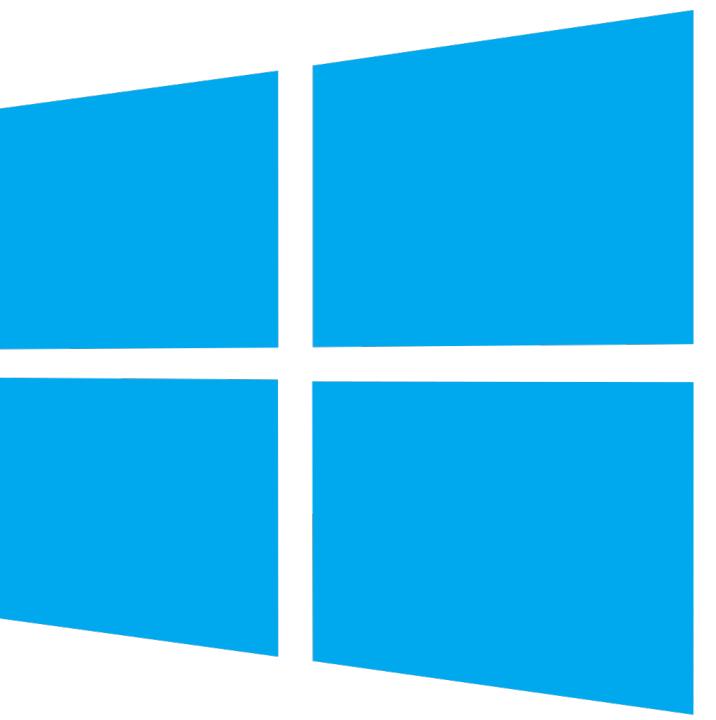
2. Closing your app





Mac OS

The app *has* windows



The window *is* the app

```
app.on('window-all-closed', () => {  
  app.quit()  
})
```

Windows and Linux:

All app windows closed = close the app

```
app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') {
    app.quit()
  }
})
```

Don't close the app on MacOS.

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

let mainWindow;

function createWindow() {
  mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    show: false,
  });

  mainWindow.once('ready-to-show', () => { ... })
  mainWindow.loadURL('https://www.conf42.com/js2021');

  mainWindow.on('closed', () => {
    mainWindow = null;
  })
}

app.whenReady().then(createWindow);

app.on('window-all-closed', () => {
  if( process.platform !== 'darwin' ) {
    app.quit();
  }
})
```

**And make sure to
recreate the window
when the dock icon
is clicked.**

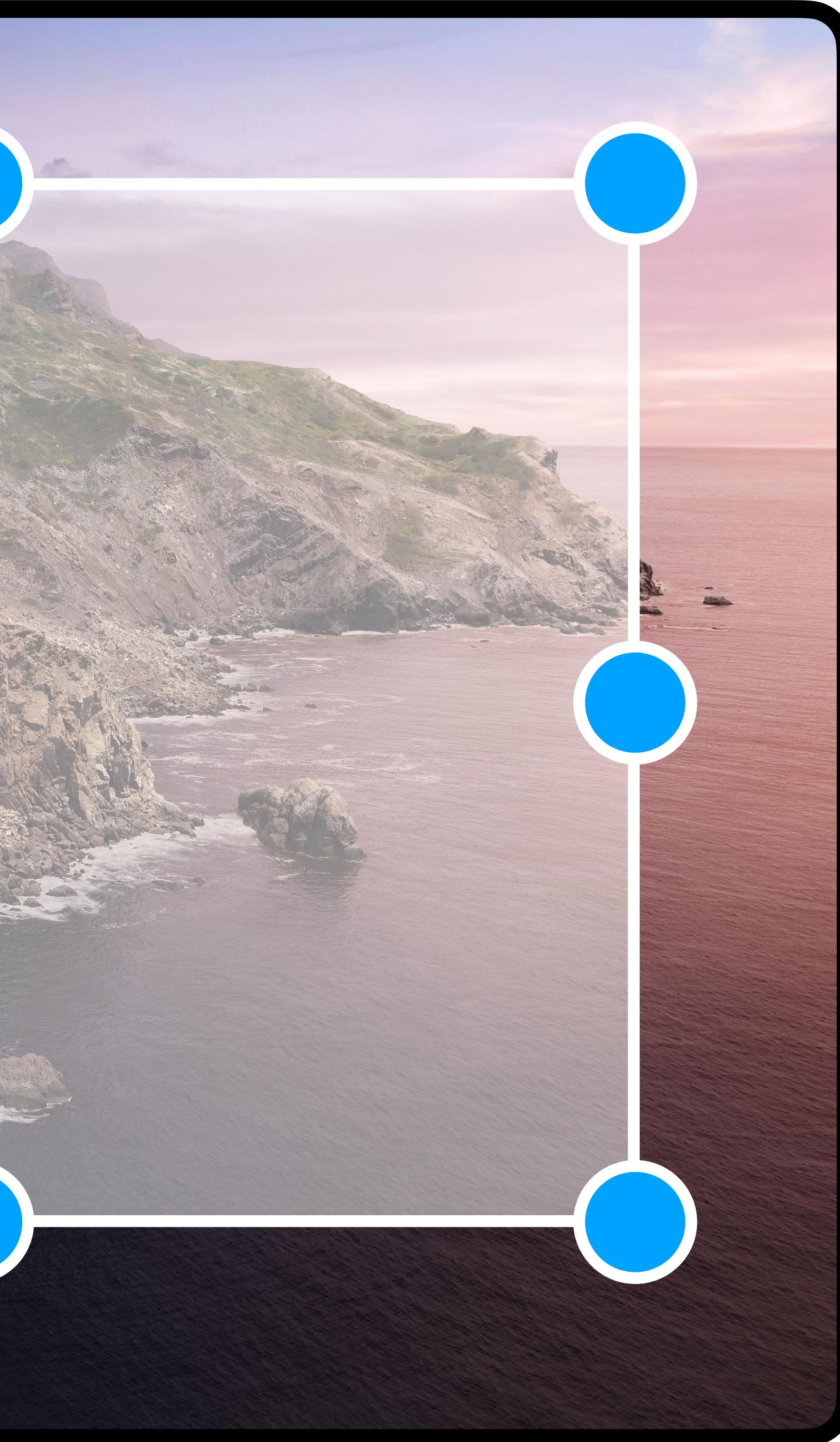
```
app.on('activate', function () {
  if (mainWindow === null) {
    createWindow()
  }
})
```

Remember user preferences

3.







- **Window dimensions**
- **Window position**
- **Which monitor**
- **If the app is maximised**

- **Window dimensions**
- **Window position**
- **Which monitor**
- **If the app is maximised**

```
let windowState = settings.get('windowState')

['resize', 'move', 'close'].forEach(event => {
  mainWindow.on(event, () => {
    windowState.isMaximized = mainWindow.isMaximized()

    if (!windowState.isMaximized) {
      windowState.bounds = mainWindow.getBounds()
    }

    settings.set('windowState', windowState)
  });
});
```

Restore the window position on load

```
let windowState = settings.get('windowState')

mainWindow = new BrowserWindow({
  x: (windowState.bounds && windowState.bounds.x) || undefined,
  y: (windowState.bounds && windowState.bounds.y) || undefined,
  width: (windowState.bounds && windowState.bounds.width) || 800,
  height: (windowState.bounds && windowState.bounds.height) || 600,
  ...
})

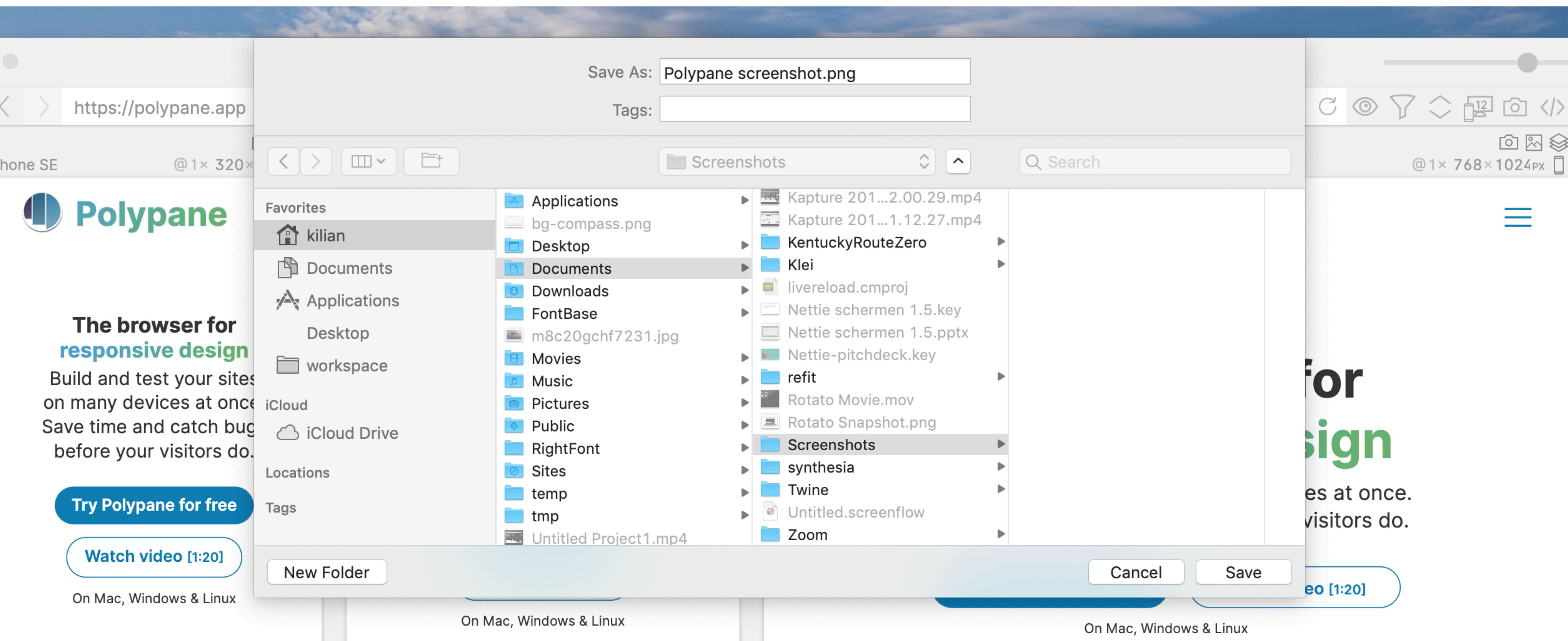
mainWindow.once('ready-to-show', () => {
  mainWindow.show()

  if (windowState.isMaximized) {
    mainWindow.maximize()
  }

  mainWindow.focus()
});
```

Or use electron-window-state

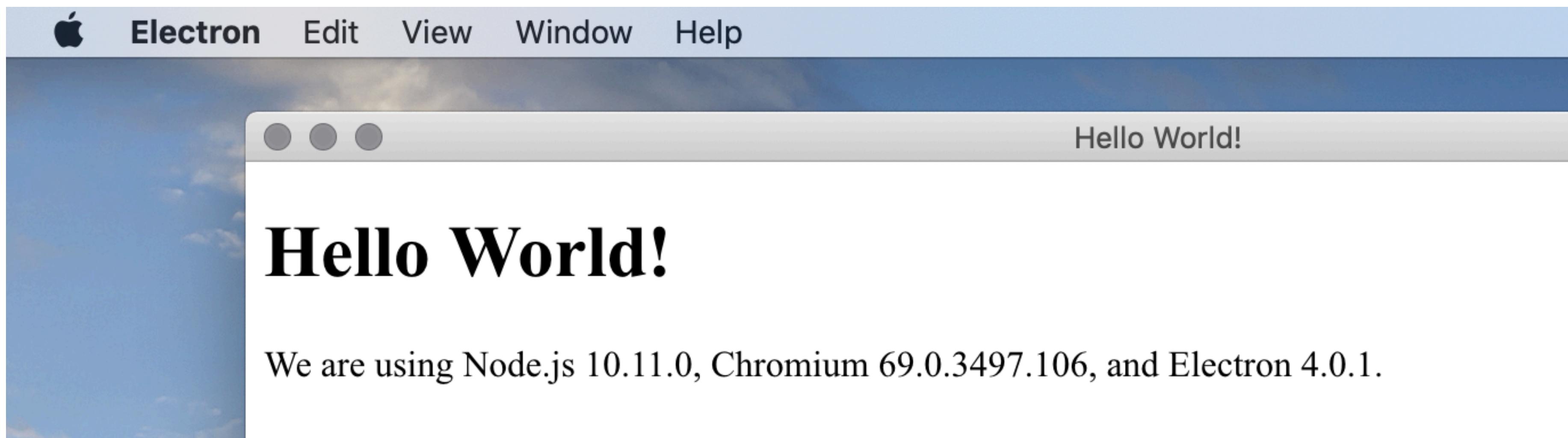
Keep track of the last used folder



```
dialog.showSaveDialog(  
  mainWindow,  
  {  
    title: 'Save Polypane screenshot',  
    defaultPath: `${settings.get('screenshot-path')}/screenshot.png`,  
    filters: [{ name: 'Images', extensions: ['png'] }],  
  },  
  filename => {  
    if (filename) {  
      setting.set('screenshot-path', path.dirname(filename))  
    }  
  }  
)
```

4. OS-specific menu





npm install electron-create-menu

```
const menuTemplate = [
{
  label: app.getName(),
  showOn: ["darwin"],
  submenu: [
    { role: "about" },
    SEPARATOR(),
    { role: "services" },
    SEPARATOR(),
    { role: "hide" },
    { role: "hideothers" },
    { role: "unhide" },
    SEPARATOR(),
    { role: "quit" }
  ]
},
{
  label: i18nFunc("File"),
  hideOn: ["darwin"],
  submenu: [{ role: "quit" }]
},
]
```

5. Text highlighting

[Events](#)[Podcast](#)[Blog](#)[Discord](#)[Sponsor](#)[Subscribe for FREE](#)

Conf42: JavaScript 2021

October 28 2021 - premiere 5PM GMT

Thu Oct 28 2021 19:00:00 GMT+0200 (Central European Summer Time) in Europe/Amsterdam

[Subscribe](#)[Sponsors](#)

13d

3h

10m

16s

View Zoom Add Page

Insert Table Chart Text Shape Media Comment

Collaborate

Format Document

Some text in Writer

Default*

Update

Style Layout More

Font

SF UI Text

Bold

14 pt

B *I* U ~~S~~ ⚙️

Character Styles None

Text Colour █ █ █ █ █ █

Alignment

Left Center Right Justify

Left Center Right

Top Middle Bottom

► Spacing

1,1

► Bullets & Lists

None

```
body {  
    user-select: none;  
}
```

6. Context menus

```
const { Menu } = require('electron')

mainWindow.webContents.on('context-menu', (e, props) => {
  const menuTemplate = []

  Menu.buildFromTemplate(menuTemplate).popup(mainWindow)
})

})
```

```
const { Menu } = require('electron')

mainWindow.webContents.on('context-menu', (e, props) => {
  const menuTemplate = []

  if (props.setEditable) {
    menuTemplate.push(
      { label: 'Undo', role: 'undo', enabled: props.editFlags.canUndo },
      { label: 'Redo', role: 'redo', enabled: props.editFlags.canRedo },
      { type: 'separator' },
      { label: 'Cut', role: 'cut', enabled: props.editFlags.canCut },
      { label: 'Copy', role: 'copy', enabled: props.editFlags.canCopy },
      { label: 'Paste', role: 'paste', enabled: props.editFlags.canPaste },
      { type: 'separator' },
      { label: 'Delete', role: 'delete', enabled: props.editFlags.canDelete },
      { label: 'Select all', role: 'selectall', enabled: props.editFlags.canSelectAll }
    )
  }

  Menu.buildFromTemplate(menuTemplate).popup(mainWindow)
})
```

```
const { Menu, shell, clipboard } = require('electron')

mainWindow.webContents.on('context-menu', (e, props) => {
  const menuTemplate = []

  if (props.isEditable) { ... }

  if (props.linkURL) {
    menuTemplate.push(
      { label: 'Open link', click() { shell.openExternal(props.linkURL) } },
      { label: 'Copy link location', click() { clipboard.writeText(props.linkURL) },
    );
  }

  Menu.buildFromTemplate(menuTemplate).popup(mainWindow)
}

}))
```

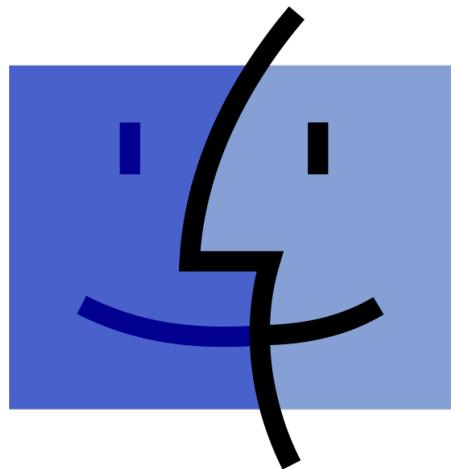
Or use electron-context-menu

7. Keyboard shortcuts



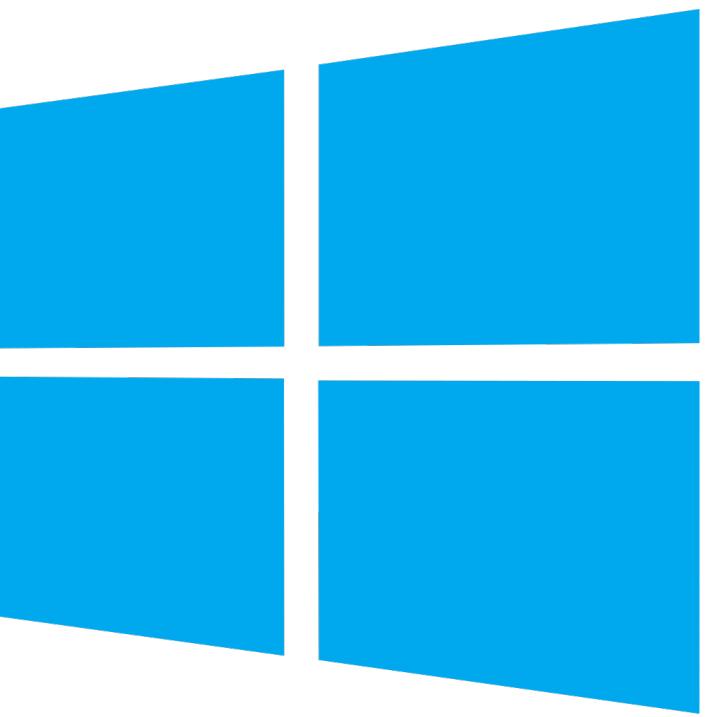
7.



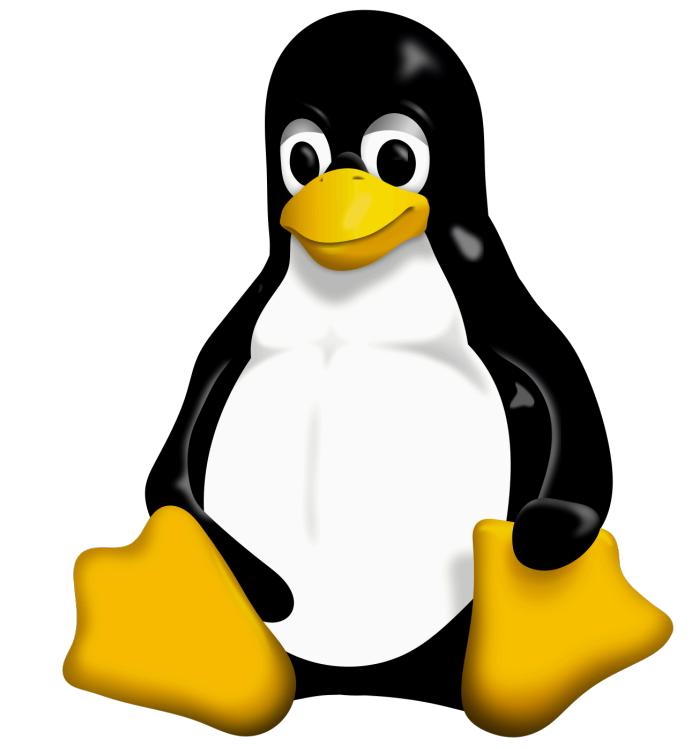


Mac OS

**Keyboard
shortcuts use
cmd (⌘)**



**Keyboard
shortcuts use
ctrl**



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

globalShortcut.register('backspace', () => { /* ... */ })
```

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

if (process.platform === 'darwin') {
  globalShortcut.register('cmd+s', () => { /* ... */ })
} else {
  globalShortcut.register('ctrl+s', () => { /* ... */ })
}
```

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

globalShortcut.register('CmdOrCtrl+s', () => { /* ... */ })
```

CmdOrCtrl

8. Using system fonts



8.



```
body {  
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica,  
    Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";  
}
```

```
body {  
  font-family: system-ui;  
}
```

```
@font-face {  
    font-family: 'Inter';  
    src: url(../font/Inter-Regular.ttf) format('truetype');  
    font-weight: 300;  
    font-style: normal;  
}
```

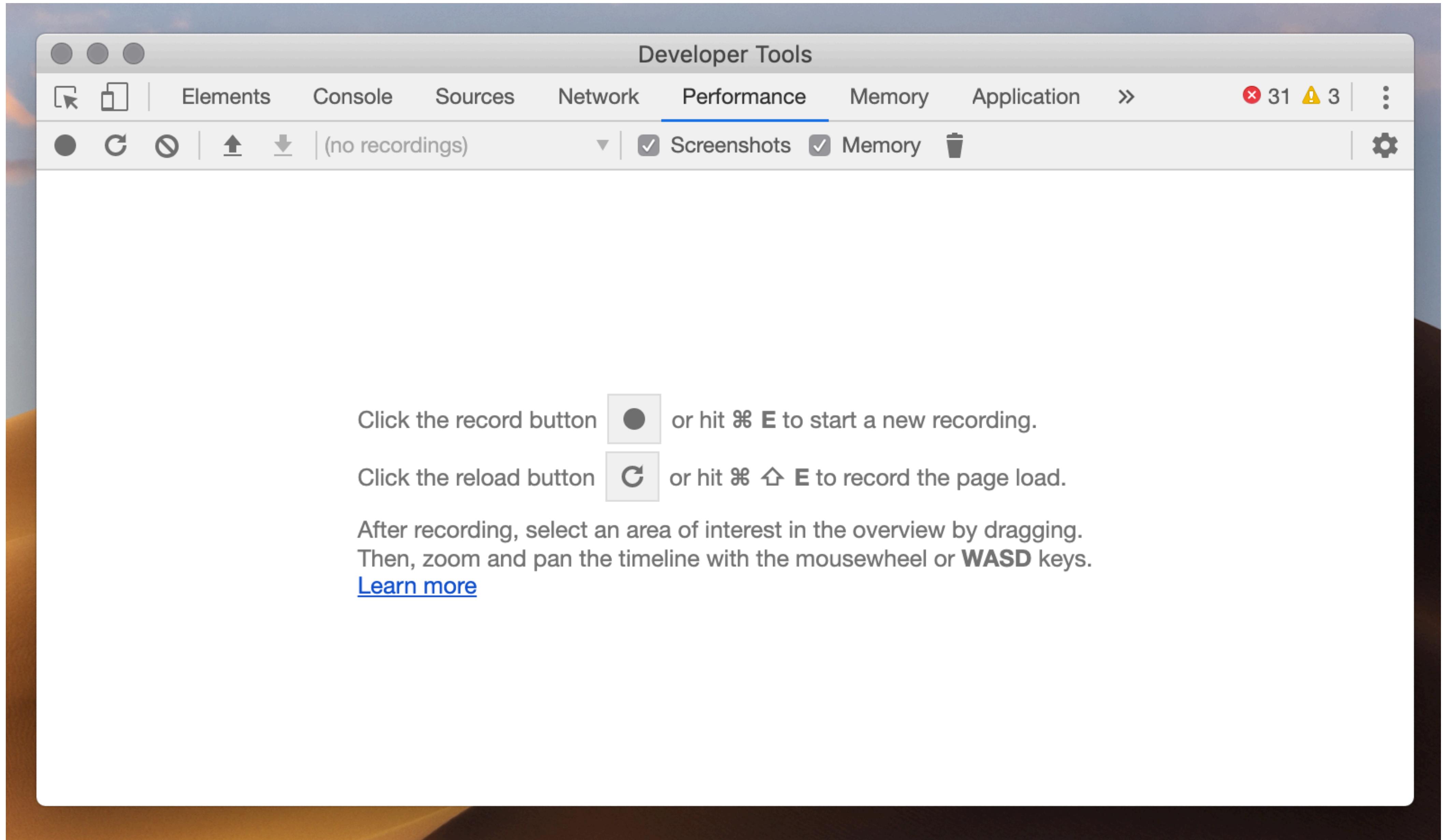


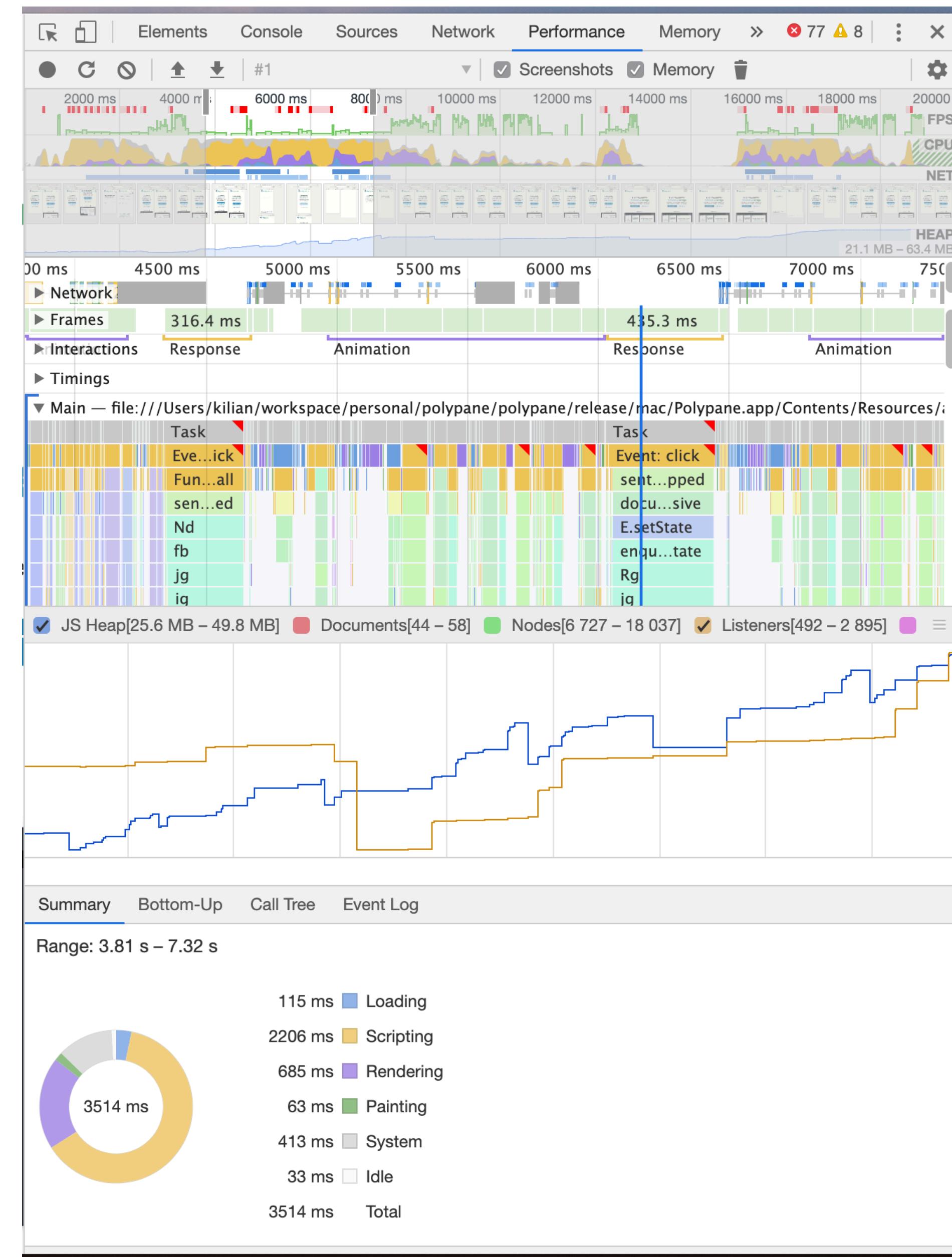

Memory



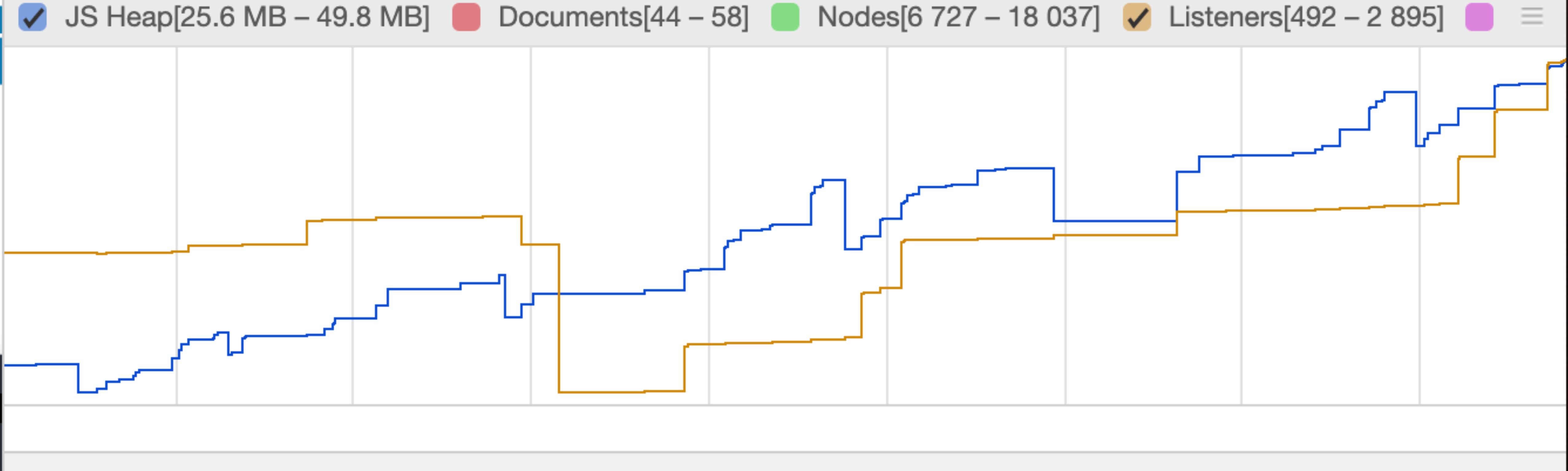
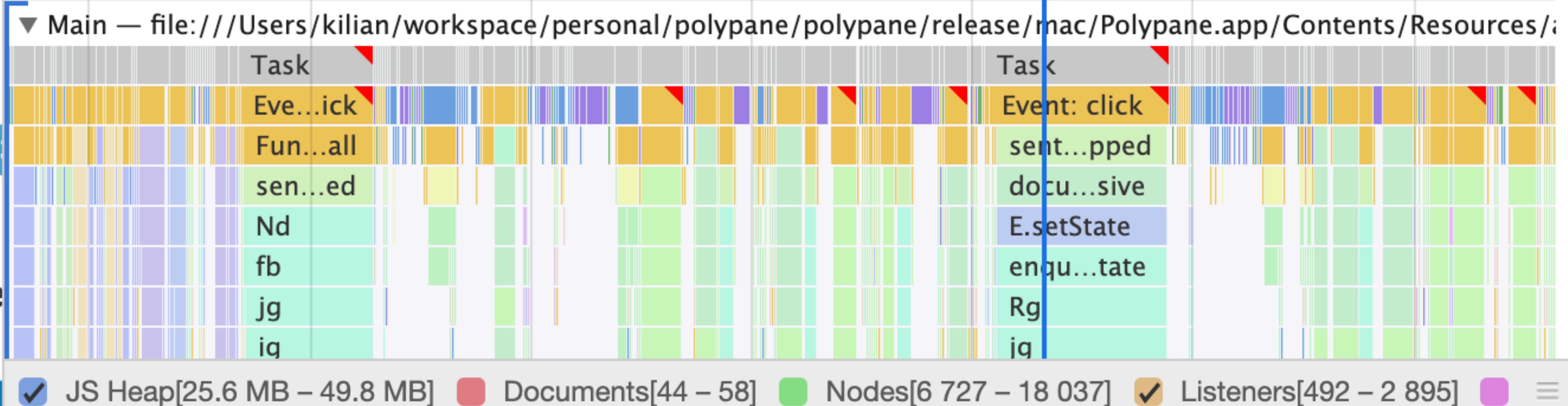
Memory

<https://electronjs.org/docs/tutorial/performance>





► Timings





```
$ electron --inspect=5858 index.js
```

chrome://inspect

In conclusion

1.

Don't launch your app like a site

Hide until
“ready-to-show”

Use a fitting
backgroundColor

2.

Handle window closing like the OS

Keep app running
on MacOS, re-
open on click.

Closing the (last) window
should close the app on
Windows and Linux

Remember user preferences

3.

**Restore windows
to their last
position and size**

**Remember the last used
folder**

4. OS specific menu

Use electron-create-menu



5.

Prevent text highlighting



```
body {  
  user-select: none;  
}
```

6.

Context menus

**Right clicking on
text should have
cut, copy etc.**

**Use contextual info to
give extra options for
images, links etc.**

7.

OS-specific keyboard shortcuts

Use CmdOrCtrl



8.

Use the system font



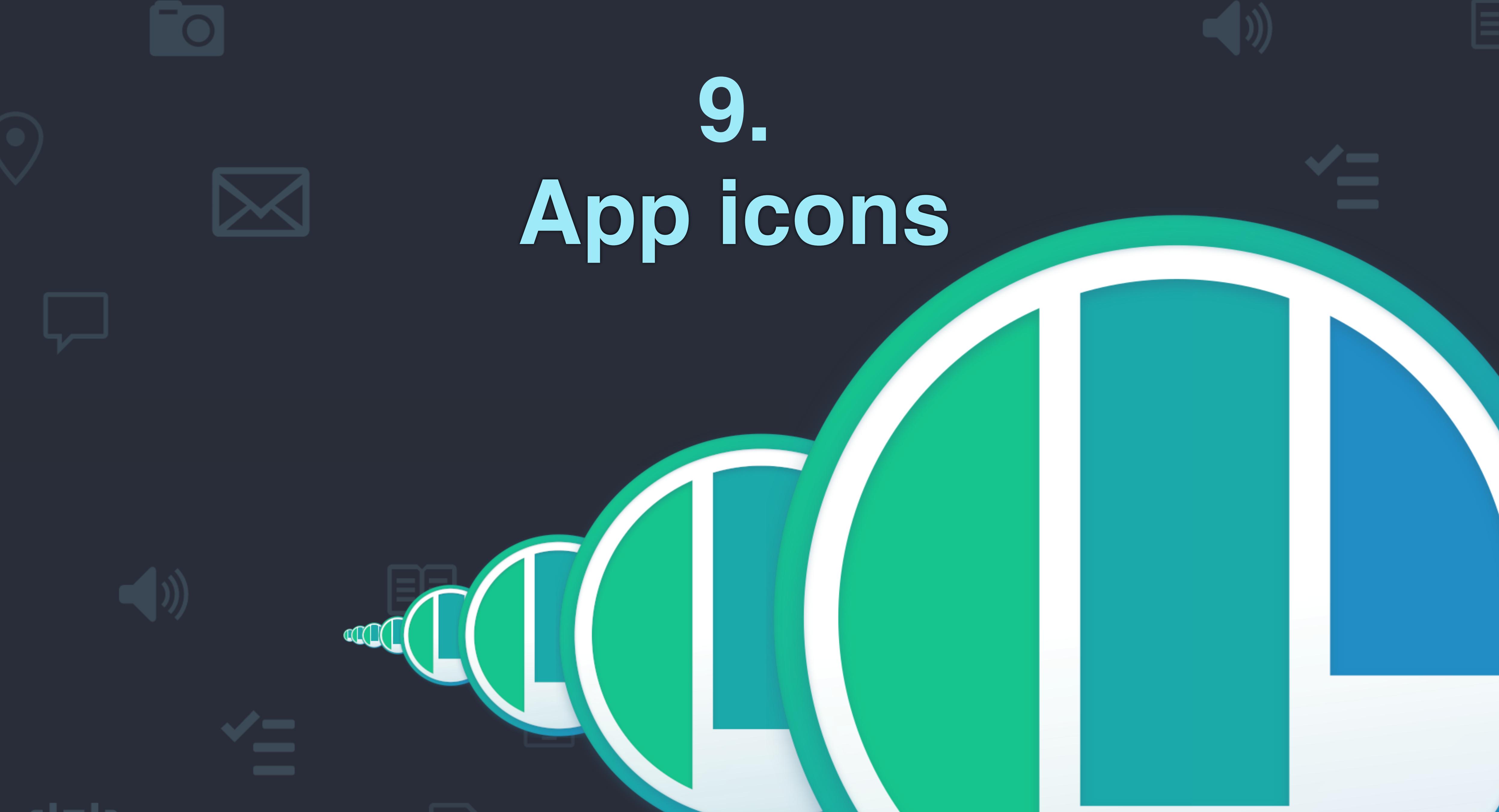
```
body {  
    font-family: system-ui;  
}
```

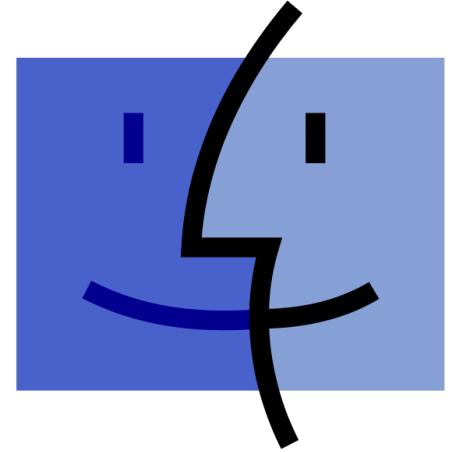


electronjs.org

kilianvalkhof.com | @kilianvalkhof | polypane.app

9. App icons





Mac OS

.icns

16x16

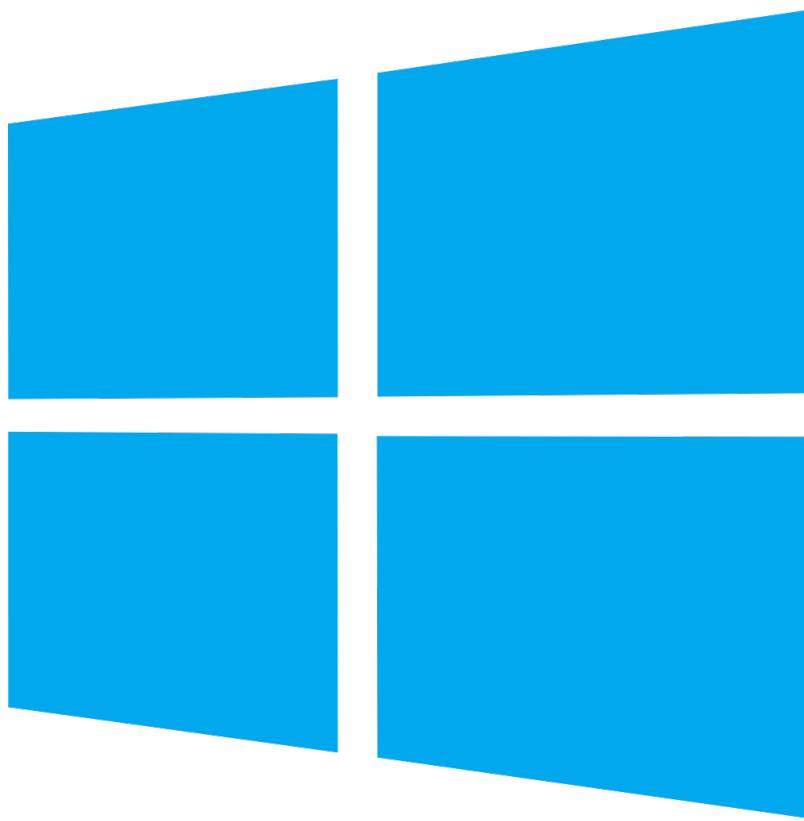
32x32

128x128

256x256

512x512

1024x1024



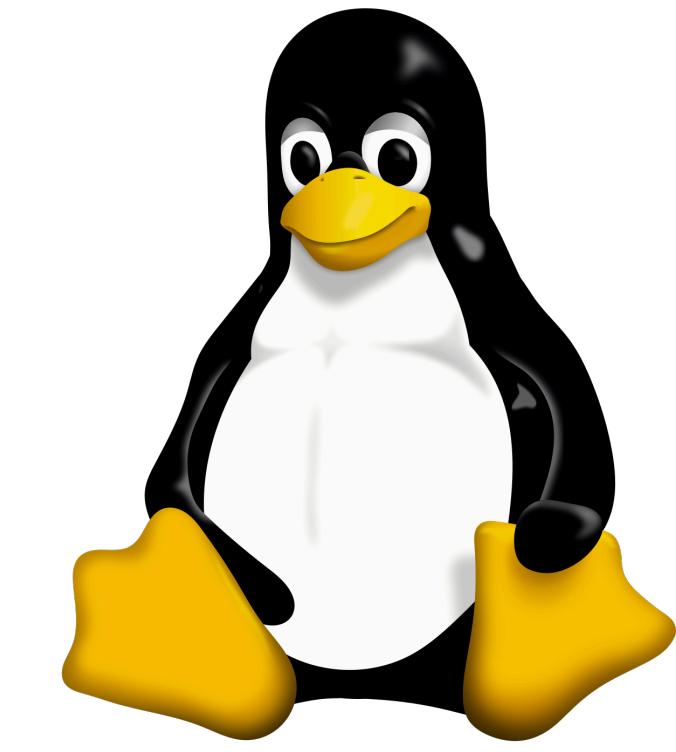
.ico

16x16

24x24

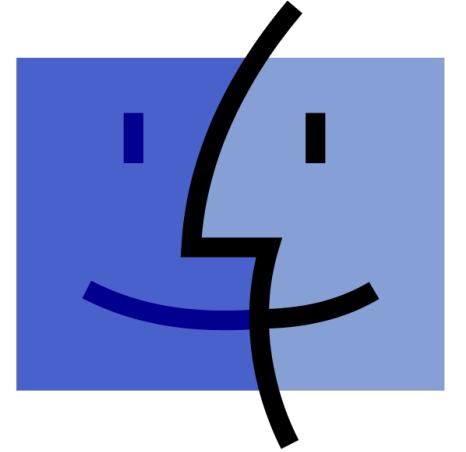
48x48

256x256



.png

1024x1024



Mac OS

png2icns

16x16

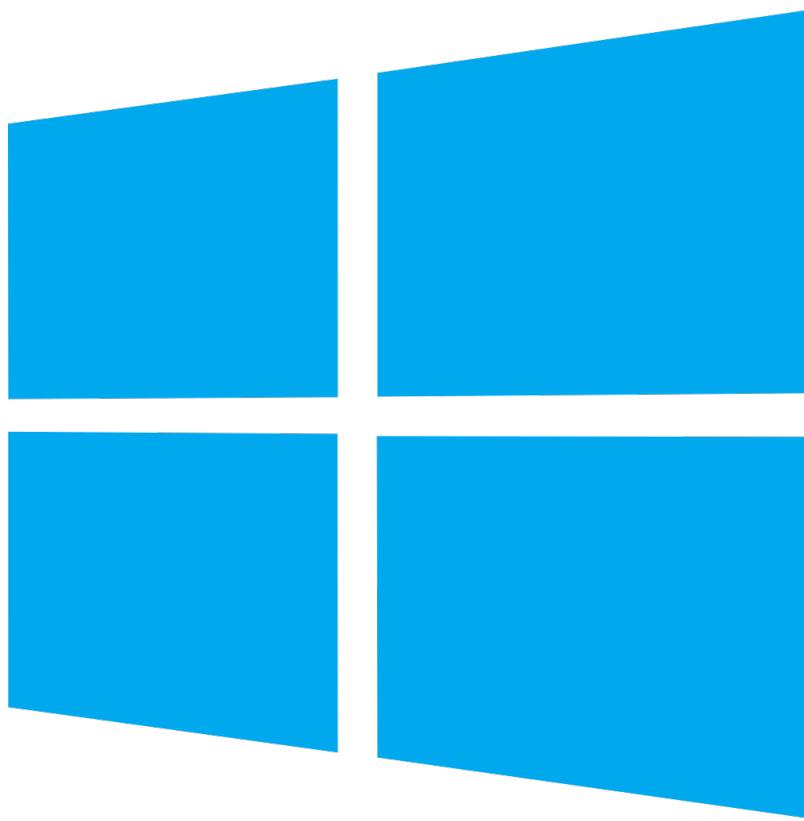
32x32

128x128

256x256

512x512

1024x1024



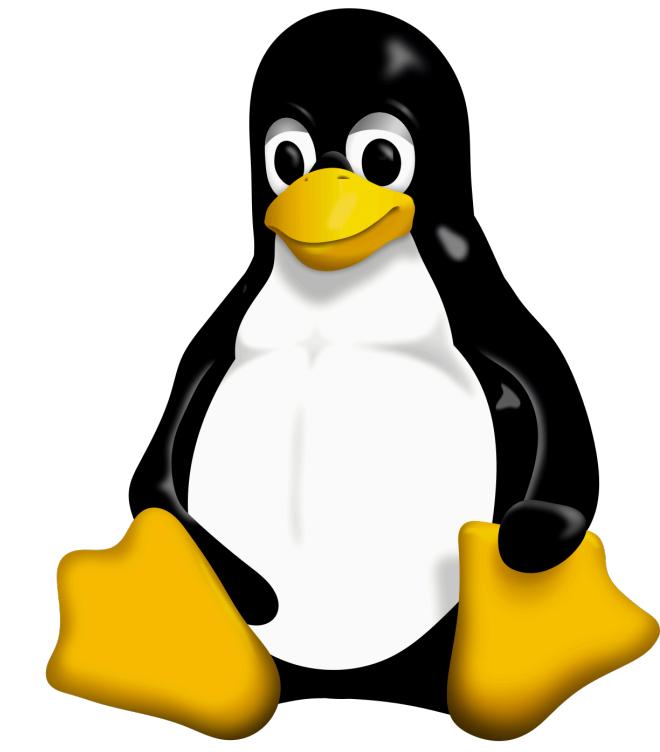
icotool

16x16

24x24

48x48

256x256



whatever

1024x1024

10. App layout

Flexbox
CSS Grid

Grid stylesheets:
constraint based css

gss.github.io