



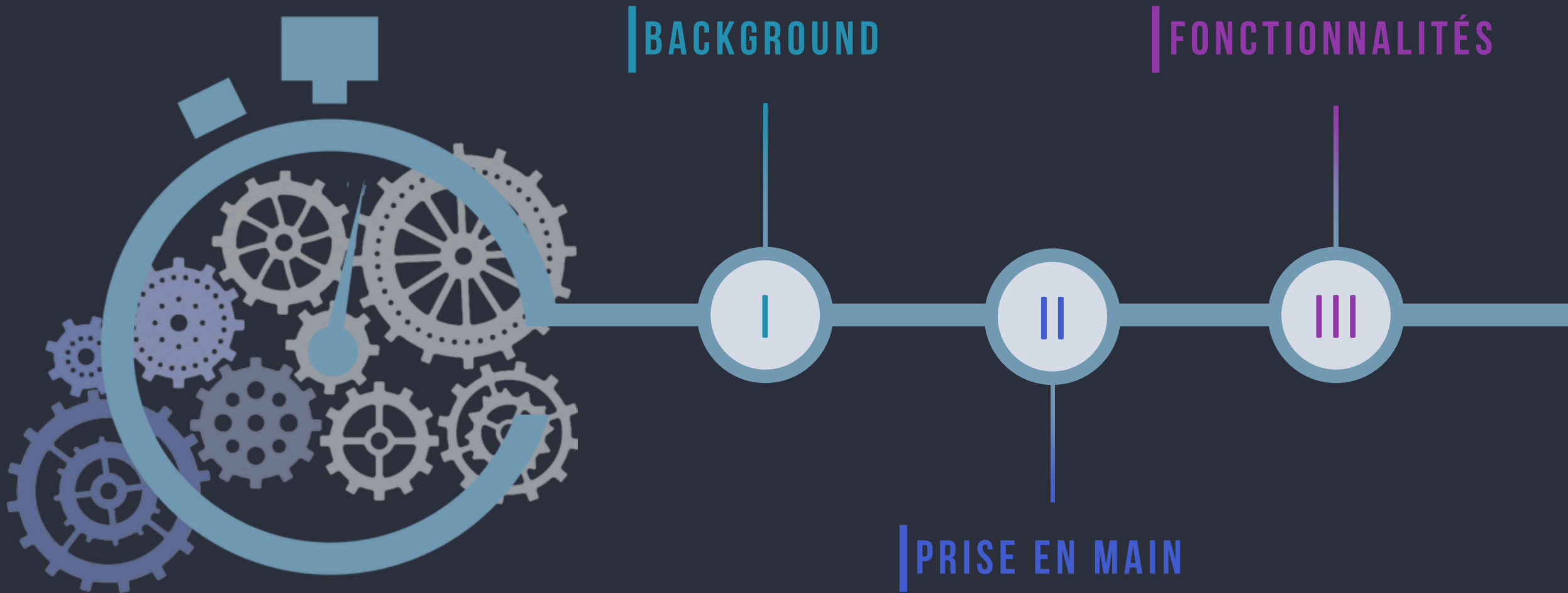
# ELECTRON

ANDRIAMARO RANDY

GRANJON JULIEN

REISSER JULIETTE

# Sommaire



Background

---

# Electron ?

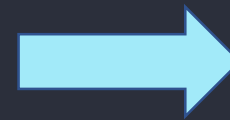


Une application web qui a accès au système d'exploitation  
Bibliothèque open source développée par GitHub

HTML - CSS - Javascript



+



# À propos

Principal concurrent

**NW.JS**

Ancien nom

**ATOM SHELL**

Principal donateur

**FACEBOOK**

Mise à jour d'Electron en rapport avec le dev de la librairie mais aussi des MàJ de Chromium

# Qui utilise Electron ?



Prise en main

---

# Installation

## PRÉREQUIS



```
$ npm install electron --save-dev
```

```
–
```



# Première application

## STRUCTURE SIMPLE

```
votre-app/  
├── package.json  
├── main.js  
└── index.html
```

```
$ npm init
```

## PACKAGE.JSON

```
{  
  "name": "your-app",  
  "version": "0.1.0",  
  "main": "main.js"  
}  
index.js
```

# Première application

## STRUCTURE SIMPLE

```
votre-app/  
├── package.json  
├── main.js  
└── index.html
```

```
$ npm init
```

## PACKAGE.JSON

```
{  
  "name": "your-app",  
  "version": "0.1.0",  
  "main": "main.js",  
  "scripts": {  
    "start": "electron ."  
  }  
}
```

**ELECTRON**

# Le développement en résumé

## MAIN.JS

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

function createWindow () {
  // Cree la fenetre du navigateur.
  let win = new BrowserWindow({ width: 800, height: 600 })

  // and load the index.html of the app.
  win.loadFile('index.html')
}

app.on('ready', createWindow)
```

CRÉATION FENÊTRE

GESTION ÉVÉNEMENTS SYSTÈME

OUVRIR OUTILS DÉVELOPPEURS

GESTION FERMETURE FENÊTRES

...

## INDEX.HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    Nous utilisons Node <script>document.write(process.versions.node)</script>,
    Chrome <script>document.write(process.versions.chrome)</script>,
    et Electron <script>document.write(process.versions.electron)</script>.
  </body>
</html>
```

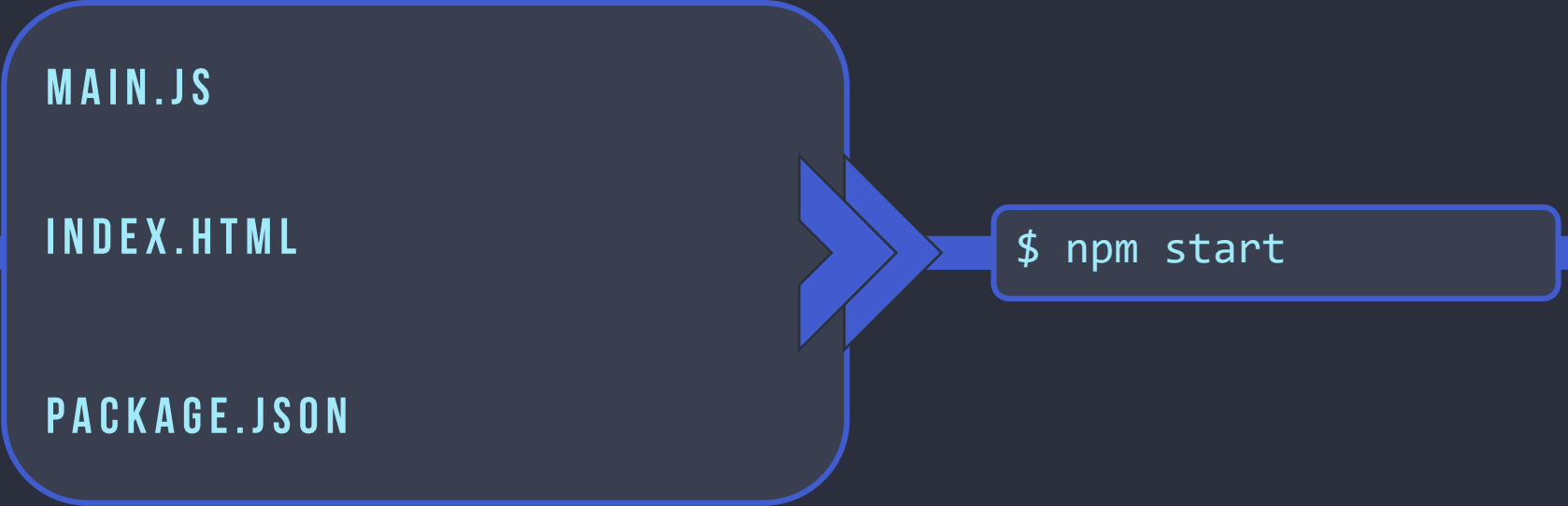
# Lancement de l'application

MAIN.JS

INDEX.HTML

PACKAGE.JSON

\$ npm start

A diagram illustrating the application launch process. On the left, a dark blue rounded rectangle contains the file names 'MAIN.JS', 'INDEX.HTML', and 'PACKAGE.JSON' stacked vertically. A large blue arrow points from this box to a smaller dark blue rounded rectangle on the right containing the terminal command '\$ npm start'. A thick blue horizontal line passes through the center of the diagram, behind the arrow and the terminal box.

# Architecture

## PROCESSUS PRINCIPAL

- ➔ Script main dans le package.json
- ➔ Affiche interface graphique en créant des pages web
- ➔ 1 seul, jamais plus

## PROCESSUS DE RENDU

- ➔ Architecture multiprocessus de Chromium utilisée
- ➔ 1 page web = 1 processus de rendu

## INTÉRÊT

Navigateur normal : pages web non autorisées à accéder aux ressources natives.

Utilisateurs d'Electron : API Node.js dans les pages web → permet interactions bas niveau avec l'OS

## PROCESSUS PRINCIPAL

1 instance `BrowserWindow` = 1 page web

Gère toutes les pages web  
+ leurs processus de rendu associés

## PROCESSUS DE RENDU

~~= 1 processus de rendu~~ arrêté

isolé et se soucie uniquement de la  
page web où il est exécuté

Opérations GUI 

doit communiquer





# Notifications

## API HTML5

```
let myNotification = new Notification('Title', {  
  body: "Le dîner est prêt, vous pouvez manger :)"  
})
```

```
myNotification.onclick = () => {  
  console.log('Notification clicked')  
}
```

# Notifications

## ELECTRON

```
ipcRenderer.send("dinner_ready")
```

```
ipcMain.on("dinner_ready", (event) => {  
  new Notification({  
    title: "A table !",  
    body: "Le dîner est prêt, vous pouvez manger :)«  
  }).show()  
})
```

# Drag & Drop

## PROCESSUS DE RENDU

```
<a href="#" id="drag">item</a>
<script type="text/javascript" charset="utf-8">
  document.getElementById('drag').ondragstart = (event) => {
    event.preventDefault()
    ipcRenderer.send('ondragstart', '/path/to/item')
  }
</script>
```

## PROCESSUS PRINCIPAL

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

ipcMain.on('ondragstart', (event, filePath) => {
  event.sender.startDrag({
    file: filePath,
    icon: '/path/to/icon.png'
  })
})
```

# Progress Bar

## PROCESSUS DE RENDU

```
const { BrowserWindow } = require('electron')  
const win = new BrowserWindow()  
  
win.setProgressBar(0.5)
```

WINDOWS

MAC OS

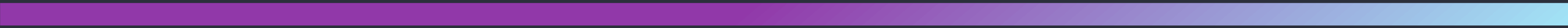
# Dialog

## PROCESSUS DE RENDU

```
dialog.showMessageDialog(mainWindow, {  
    type: "info",  
    title: "Temps écoulé",  
    message: "Le timer " + timer.id + " est arrivé au bout du décompte"  
})
```

# Raccourcis claviers

```
// Raccourcis locaux
const { Menu, MenuItem } = require('electron')
const menu = new Menu()
menu.append(new MenuItem({
  label: 'Print',
  accelerator: 'CmdOrCtrl+P',
  click: () => { console.log('time to print stuff') }
}))
// Raccourcis globaux
const { app, globalShortcut } = require('electron')
app.on('ready', () => {
  globalShortcut.register('CommandOrControl+X', () => {
    console.log('CommandOrControl+X is pressed')
  })
})
// BrowserWindow
window.addEventListener('keyup', doSomething, true)
```







Démonstration