



Introduction à Deno

Présenté par Logan TANN
Pour OnePanthéon





Audience

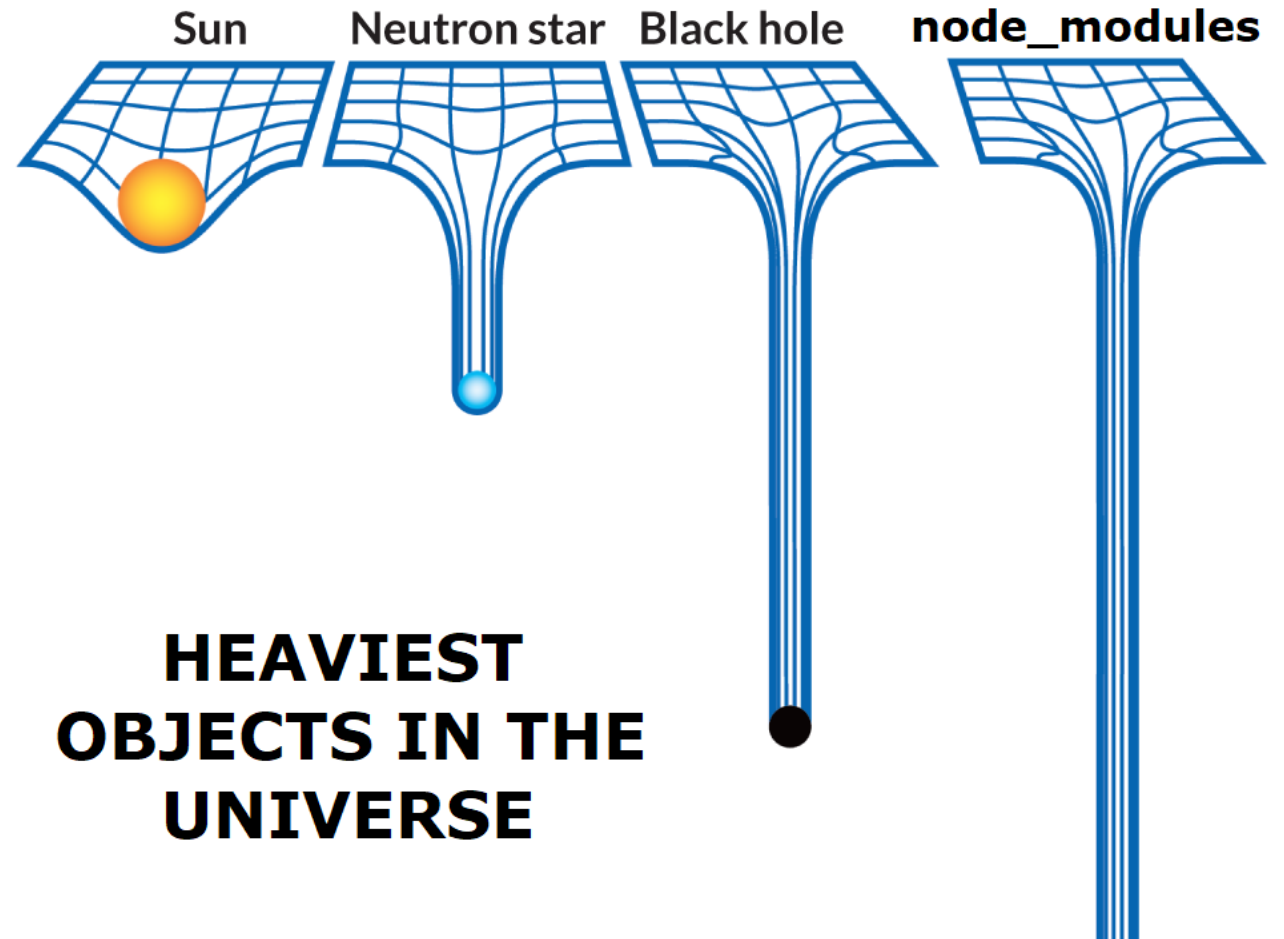
- Pour les curieux du web en général 🧐 -

👉 En particulier : les Typescript Enthousiasts, ou pour ceux qui ont réalisé un projet avec NodeJs.

🕶 Note : Pour comprendre les exemples, une expérience en Javascript moderne est recommandée.

Un peu d'histoire

> De Node.JS à Deno





Les débuts de Node.JS

- ☹️ Ryan Dahl est frustré des limitations d'Apache
- Difficultés d'implémenter des applis en temps réel
 - Mécanismes de prog évènementielle et concurrente pas assez efficace.



JSConf 2009 : Il présente un projet perso, qui deviendra Node.JS

👉 Bénéficie aujourd'hui d'une grande popularité



Un projet devenu trop populaire

10 ans plus tard, il présente à la JSConf 2019 :
« 10 Choses que je regrette à propos de Node.js »



Il évoque par exemple :

- Node n'est pas assez proche des standards JS des navigateur
- NPM souffre de plein de défauts

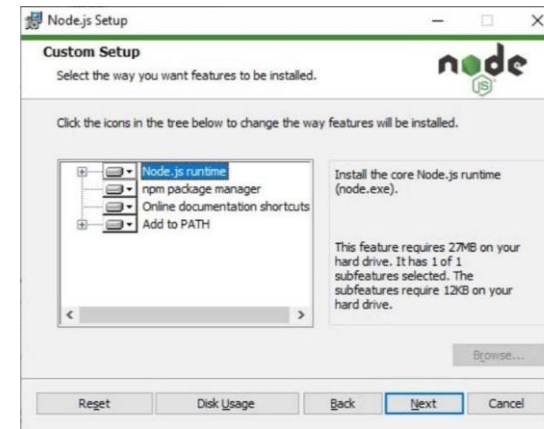
Prise en main de Deno

Installation et découverte en exemples





Installation



1 - Juste le binaire à installer depuis github



Installation



1 - Juste le binaire à installer depuis github

2 – Installeur automatique en ligne de commande

Shell (Linux & Mac) :

```
$ curl -fsSL https://deno.land/x/install/install.sh | sh
```

Windows :

```
$ irm https://deno.land/install.ps1 | iex
```

Docker :

```
$ docker run -it denoland/deno:1.28.1
```




Exemple 1 : Hello World

Consigne :

1. Faire un programme qui affiche "Hello World".
2. Faire un programme qui affiche le contenu du fichier "canard.txt"

```
~/p/A/0/préparations main 4 $ deno run --allow-read .\01_HelloWorld.ts  
Hello World !
```

```
>(. )__ <(. )__ =(. )__  
(___/ (___/ (___/
```



Exemple 1 : Hello World

Solution :

```
7  console.log("Hello World !");
8
9  // support du top level await
10 // ... et aussi un système de permissions vraiment bien foutu.
11 // avant ça faisait un throw, mais depuis la version 1.28, ça a changé
12 const texte = await Deno.readFile("canard.txt");
13
14 console.log(texte);
15
```



Exemple 2 : Appel d'API

1. Voici une API très sympa : <https://ifuckinghatejira.deno.dev/>
2. Faire appel à cette API
3. Ne rendre l'affichage du message que si l'utilisateur n'aime pas Jira.

```
~/p/A/0/préparations main ?4 $ deno run --allow-net .\02_ApiCall.ts  
Aimez-vous jira ? [y/N] N
```

Voici une opinion honnête sur jira :

```
#22 : Every time I open Confluence or Jira I want to claw my fucking eyes out.  
It's slow, hard to navigate, and terrible.
```



Exemple 2 : Appel d'API

Solution :

```
7  
8 // Les APIs web sont supportées :D  
9 const request = await fetch("https://ifuckinghatejira.deno.dev/");  
10 const message = await request.text();  
11  
12 const result: boolean = confirm("Aimez-vous jira ? ");  
13 if (! result) {  
14     console.log("\nVoici une opinion honnête sur jira : ");  
15     console.log(message);  
16 } else {  
17     console.log("\nVous aimez jira ? Mieux vaut ne pas dire la vérité.");  
18 }
```



Exemple 3 : Serveur web et NPM

1. Importer la librairie Aqua (<https://deno.land/x/aqua@v1.3.5/mod.ts>) et faire un hello world
2. Importer une librairie NPM (npm:qrcode@1.5.0) et afficher un QR Code
3. Utiliser un import map pour rendre l'importation des dépendances plus maintenables

```
~/p/A/0/préparations main ?4 $ deno run --allow-net --import-map=../../import_map.json .\03_WebServer.ts
Démarrage du serveur...
Serveur web démarré.
```



The screenshot shows a web browser window with the address bar set to `localhost:3001`. The page content includes the text "Hello world" in a large, bold, serif font, a square QR code centered below it, and the text "Un super site :" at the bottom left of the page area.

```

7  ∨ import Aqua from "aqua";
8  import QRCode from "qrcode";
9
10 console.log("Démarrage du serveur... ");
11
12 const aqua = new Aqua(3001);
13 ∨ aqua.get("/", async () => {
14     const url = await QRCode.toDataURL('https://soup.kagescan.fr');
15     ∨ return `
16         <H1>Hello world</H1>
17         <p>Un super site :
18         |   <img src=${url} />
19         </p>
20     `;
21 });
22
23 console.log("Serveur web démarré.");

```

Exemple 3 : Appel d'API (Solution)

```

{} import_map.json > ...
1  {
2      "imports": {
3          "aqua": "https://deno.land/x/aqua@v1.3.5/mod.ts",
4          "qrcode": "npm:qrcode@1.5.1"
5      }
6  }
7

```



Récap



Avantages :

1. Expérience développeur très satisfaisante.
2. Un binaire standalone avec plein d'outils built-in
3. Veut faire changer les habitudes.



Inconvénients :

1. Difficile de faire changer les habitudes
2. Librairies souvent peu documentées ou supportées
3. VsCode est le seul éditeur fiable

=> Génial pour des scripts d'automatisation ou des petits projets



Pour aller plus loin...

- **Documentation** : <https://deno.land/manual@v1.28.1/introduction>
- **DenoDeploy** : plateforme de déploiement (gratuit)
- **Fresh** : framework web fullstack concurrent à React
- **Deno News** : Veille numérique officielle sur Deno

- **Discordeno** : Lib pour faire des bots discord
- **Caviar** : Game Engine basé sur le WebGPU
- **Awesome deno** sur github



Prochaine séance

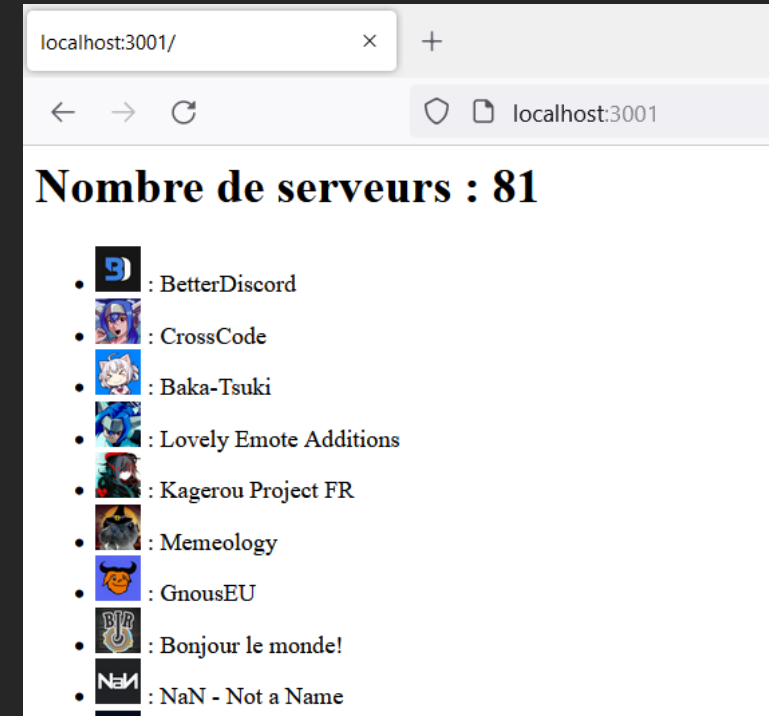
- Pratique avec le protocole Oauth 2 -

Sujet : Faire une application listant les différents serveurs discord d'une personne, et la déployer en edge functions.

Principe :

- Un mélange entre cours et TP (prepared kata)
- Une correction distribuée en PDF

Date : Fin décembre / Début janvier





Des questions ?

Merci pour votre écoute !

Lien github vers les slides et le code :
<https://github.com/CPU-Paris/Ateliers-deno/tree/main/01 - Introduction>

The screenshot shows a GitHub repository interface. At the top, there is a file list with the following items:

File/Folder	Time
documents	32 minutes ago
pratique	32 minutes ago
solutions	32 minutes ago
.gitkeep	32 minutes ago
readme.md	1 minute ago

Below the file list, the content of the `readme.md` file is displayed. The title is "Séance 1 - Introduction à Deno". The text reads:

Découvrez Deno, un runtime moderne pour Javascript, Typescript et WebAssembly.

Prérequis :

Connaissances en javascript fortement recommandé