

Programmation Web Avancée et Mobile

CM 1 : Stack JavaScript

Aurélien Tabard - Lionel Médini

But de ce cours

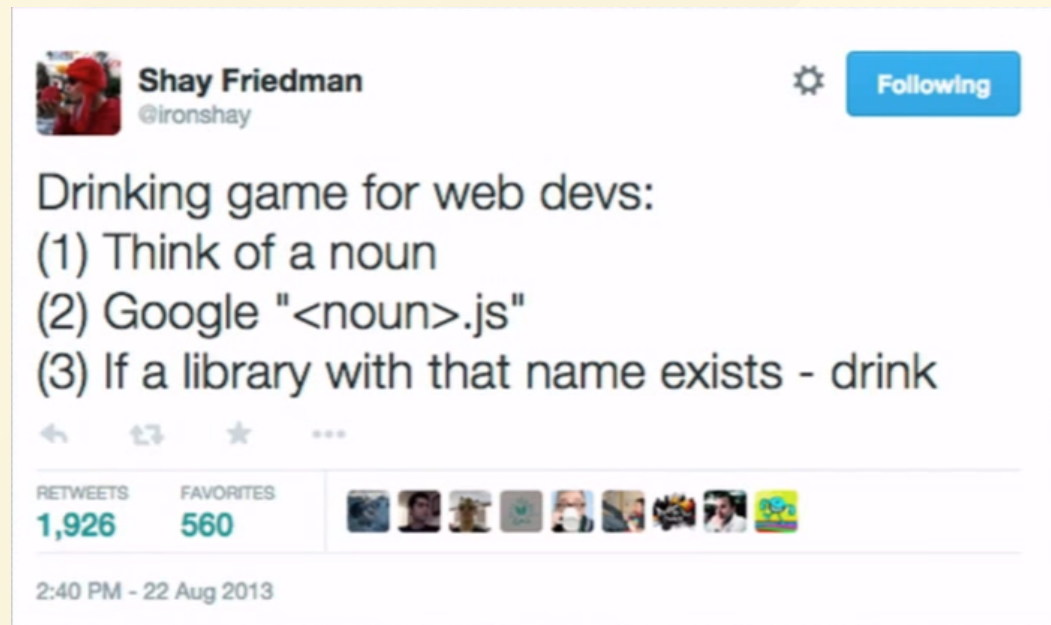
- Introduire les outils d'aide au développement JS
Quelques [rappels sur JS au besoin](#).
- Vous rendre plus efficaces pour :
 - développer
 - packager une application Web
 - débbugger

Contenu de ce cours

- Grands principes d'un outil
- Lien vers le site / la doc
- Remarques sur l'utilisation pratique

JavaScript c'est cool

Parfois un peu trop...



- [Wat par Gary Bernhardt \(vidéo\)](#)
- [How it feels to learn JavaScript in 2016](#)



Plateforme d'exécution basée sur le moteur Chrome

- Permet de :
 - mettre en place de “grosses” applications
 - en JS
 - modulaires
 - programmer dans le même langage côté serveur et côté client

Quelques différences

avec du JS côté navigateur

- Pas de Browser Object Model (BOM)
- Paquets spécifiques
- Mode de gestion des modules : CommonJS



- Gestionnaire de paquets pour node
- S'utilise "à la" apt-get
 - localement : `npm install <package>`
 - globalement : `npm install -g <package>`
- Description de l'application dans un fichier package.json

Gestion des dépendances

Require

- Chargement des dépendances dans les modules
- S'utilise
 - côté navigateur (utilise [require.js](#) ou d'autres méthodes)
 - côté Node (vient avec sa propre commande `require`)
- Configuration en JSON



Bower

- Gestionnaire de paquets applicatif côté client
- Description de l'application dans un fichier `bower.json`
- Exécution : `bower install`
- Différence avec NPM :
 - Optimisé pour le front-end
 - Meilleure gestion des dépendances croisées
Graphe de dépendance plat : on télécharge pas jquery 2x côté client



Grunt

- Task manager / runner
- S'appuie sur un ensemble de plugins (cf. Maven)
- Configuration du projet : Gruntfile
- Utilisation : lancement des tâches via la Command Line Interface (CLI)

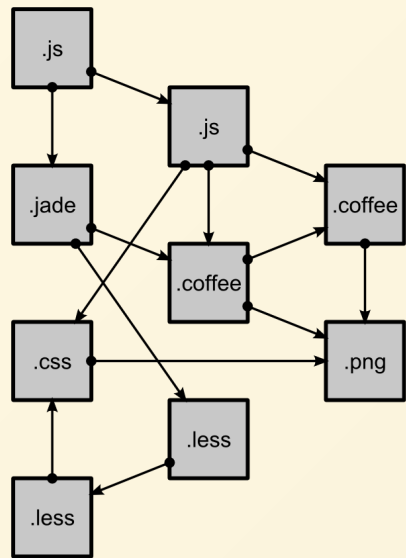
Voir aussi [Gulp](#)



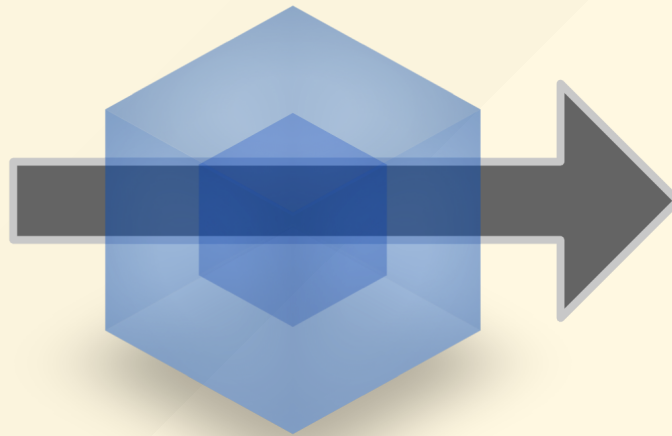
Yeoman

- Outil de “Scaffolding” d'applications Web
- Permet de créer rapidement un projet avec de nombreux composants
- 2 possibilités
 - CLI
 - Bibliothèque de générateurs prêts à l'emploi

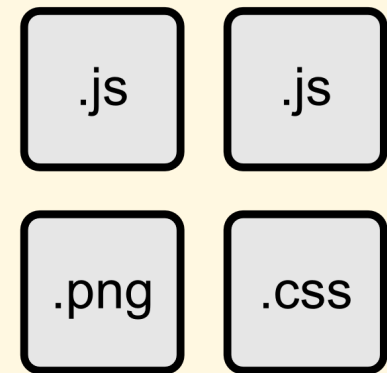
Webpack



modules
with dependencies



webpack
MODULE BUNDLER



static
assets

Webpack

1. Permet la gestion de module (require / import)
2. Agnostique au type de modules : AMD, UMD, CommonJS etc.
3. Traite les assets statiques comme des modules (CSS, images)
4. Découpe le code pour optimiser le chargement
5. Injection de dépendance et multi-compilation
6. S'intègre aux autres outils de build (Grunt/Gulp/etc)

Webpack

Write your code

app.js

```
import bar from './bar';  
  
bar();
```

bar.js

```
export default function bar() {  
  //  
}
```

Bundle with webpack

webpack.config.js

```
module.exports = {  
  entry: './app.js',  
  output: {  
    filename: 'bundle.js'  
  }  
};
```

page.html

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
    <script src="bundle.js"></script>  
  </body>  
</html>
```

Then run `webpack` on the command-line to create `bundle.js`.

via [Getting started with Webpack](#)

Webpack debugging

Problème : tout est optimisé / minifié

Solution : création d'une [source-map](#) liant code bundlé au code "réel".

[Intro au debbuging Vuejs](#)

Autres outils JS

- Framework node : [Express](#)
- Tests : [Mocha](#) ou [Jasmine](#)
- Vérification de code : [ESLint](#)
- Compilation ES6 → ES5 : [Babel](#)

Modularité

Modularité Principes

- isolation
- réutilisabilité
- gestion des dépendances
- téléchargement

Différentes solutions

- Asynchronous Module Definition (AMD)
- CommonJS (CJS)
- Universal Module Definition (UMD)
- ES6 modules

Référence : <http://addyosmani.com/writing-modular-js/>

Deux approches

- Approche serveur
 - un répertoire `node_modules` avec une arborescence de dépendances
 - chaque module peut avoir ses propres dépendances, avec des versions différentes
- Approche client
 - un répertoire avec des dépendances “à plat”
 - les dépendances partagées doivent avoir les mêmes versions
 - il faut gérer les conflits

Modules et sécurité

Lire:

I'm harvesting credit card numbers and passwords from your site. Here's how.