



Angular 19

Développement d'applications Web

Intervenant : Jean-Frédéric VINCENT



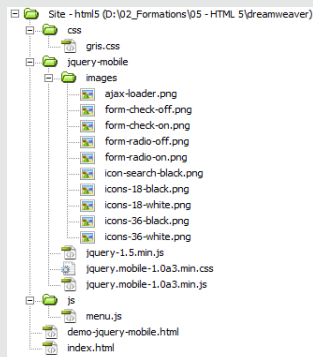
Contenu du module

- Javascript ou Vanilla Javascript
- Les Frameworks JavaScript
- Historique
- Caractéristiques du langage
- Une **S**ingle **P**age **A**pplication ?
- Présentation du serveur NodeJS
- Le principe de component avec Angular

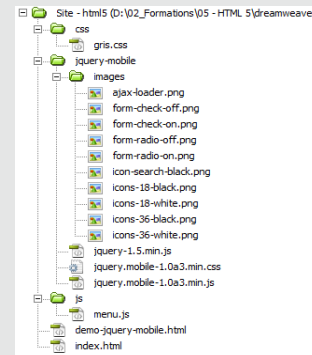
Le javascript

- Le fichiers JavaScript est téléchargé sur l'ordinateur client
- C'est le navigateur web qui exécute le JavaScript (et non le serveur)
- Problème le code source est accessible !

JS

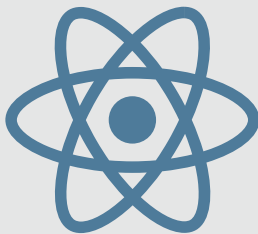


HTTP



Les Framework JavaScript ?

- Vanilla Javascript **VERSUS** Framework JavaScript
- Moins de codes
- Plus de fonctionnalités
- Code mieux "rangé"
- Maintenance



React



Angular



Vue

Framework ou bibliothèque

- un **Framework** est :
 - un ensemble d'outils ultra complets permettant de créer une application de A à Z
 - et fournissant tous les outils nécessaires au développement d'une application.
- Alors qu'une **Bibliothèque (Library en anglais)**
 - s'ajoute à une partie de votre application

Les Frameworks JavaScript

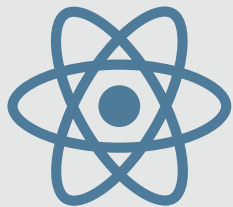
- La fin d'un règne 2009 - **2019** !
 - *Vive Vanilla JavaScript* ou juste JavaScript



- Mais encore bien présent
 - Plugin **Select2** et autres ...

Les Frameworks JavaScript

- Les 3 Frameworks JavaScript les plus utilisés



React

 Meta



Angular 2+



Vue JS

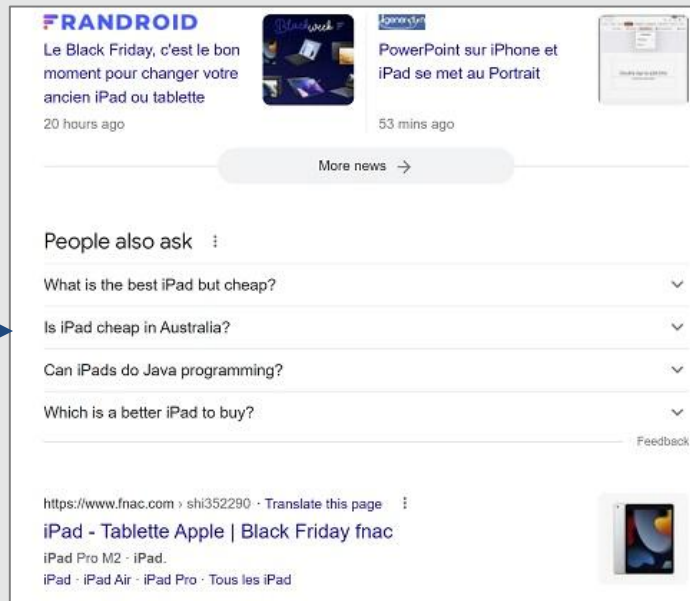


Evan You

Historique

- Création en **2009** dans les locaux de Google
- Projet **Google Feedback**
- 17 000 lignes sans Framework puis 1 500 lignes avec **Angular 1**
- Date de sortie des différentes versions :
 - juin **2012** **Angular 1** ou **AngularJS**
 - mai **2016** sortie de **Angular 2**
 - novembre **2022** sortie de **Angular 15**
- Google sort 2 versions par an

Google Feedback



Caractéristiques du langage

- Pourquoi parle-t-on de Angular **2+** ?
 - Angular **2+** ?
 - Angular **2** ?
 - Angular **14** ?
- Angular **1** ou **AngularJS**
 - Fin du support janvier **2022**

Caractéristiques du langage

- Le CLI **C**ommand **L**ine **I**nterface
 - Utilise le serveur NodeJS
 - Permet de générer des projets Angular ou des fichiers spécifiques
- Le principe de component avec 3 fichiers
 - TypeScript
 - CSS
 - HTML
- Modèle MVVM : **M**odel **V**ue **V**ue **M**odel

Une Single Page Application : qu'est ce que c'est ?

- Ou **SPA**
- Toutes les pages sont chargées en une seule fois!
 - dès le chargement de la première page
- Donc plus de latence
- Attente uniquement si requête AJAX
- Bande passante réduite
 - Plus rapide pour l'utilisateur
 - Serveur moins sollicité

Présentation du serveur NodeJS

- NodeJS est un serveur JavaScript
 - Il exécute donc du code Javascript
- Il est basé sur le moteur V8 de Chrome
- Le serveur NodeJS est installé en local sur la machine de développement
- NodeJS permet de transcompiler les fichiers TypeScript en JavaScript
- Le navigateur web **ne** peut **pas** exécuter de fichiers TypeScript
- Le CLI à partir du serveur NodeJS
 - permet de générer des projets Angular
 - permet de générer des fichiers Angular (du type component)



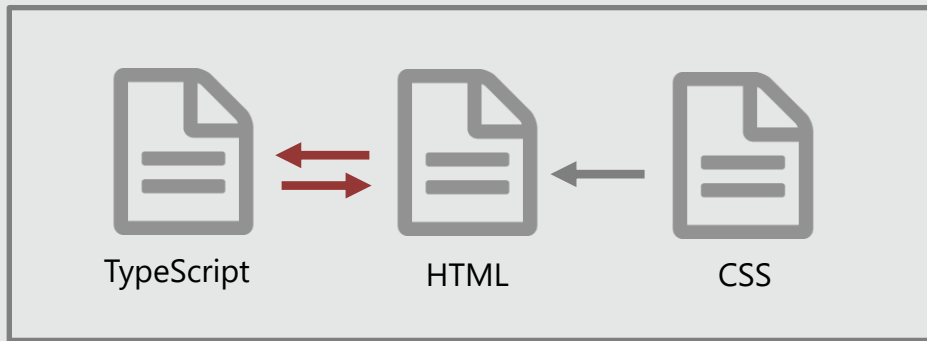
Le principe de component

- Un component est constitué de 3 fichiers
 - TypeScript
 - CSS
 - HTML
- Une page peut contenir plusieurs components imbriqués
- Une page est un component
- Interaction entre la vue et le model : le fichier HTML et TypeScript
 - **Liaison** **binding**
 - **Directive** **directive**

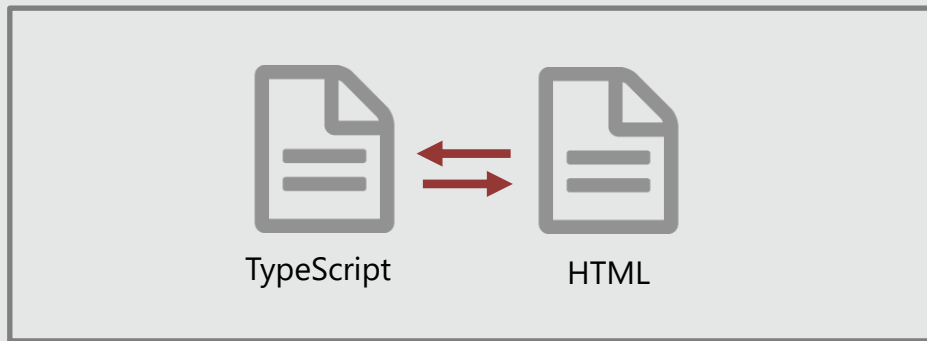


Le principe de component

- La richesse du framework Angular se trouve dans l'interaction entre le fichier TypeScript et HTML
 - Interpolation →
 - Liaison *ou binding en anglais* →
 - Directive →
- Grande économie de code



un component

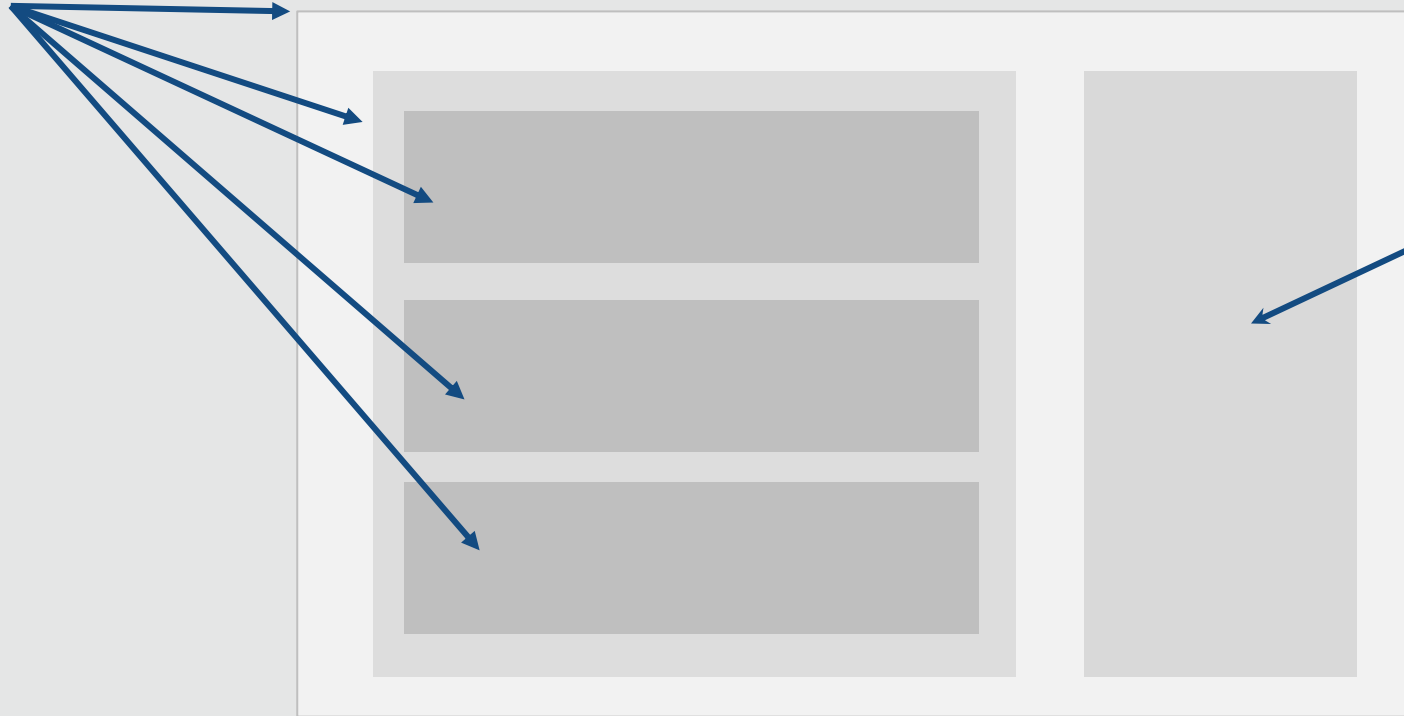


un component

Le principe de component

- Imbrication de components sur une page

component



component