

Simulateur “Grenelle de l’Éducation”

Objectifs

L’objectif principal du simulateur est de simuler le montant de la *prime d’attractivité* en négociation lors du “Grenelle de l’Éducation” de la fin de l’année 2020.

Ce que le simulateur fait

- Demander le statut, et éventuellement l’échelon
- Retourner selon les cas une simulation de prime
- Donner le rapport à l’inflation des salaires dans la fonction publique.

Aspects techniques

Pré-requis

Le site est généré par [Hugo](#). Il faut donc l’installer sur la machine locale pour reconstruire le site.

Hugo est un générateur de site statique. Il suffit, à la racine du site, d’invoquer la commande `hugo` pour générer le site entier. Il se trouve alors dans le dossier `public` et peut être copié sur n’importe quel hébergement en ligne.

Si on veut faire tourner le site localement, il suffit d’invoquer Hugo ainsi : `hugo server -D`. Les changements conduiront à un rechargement immédiat des pages. Le site de test sera disponible par défaut à l’URL locale suivante : <http://localhost:1313>.

Prérequis utilisateur

Les utilisateur-ice-s du simulateur devraient pouvoir utiliser la page dans n’importe quel navigateur sur n’importe quel terminal, à la condition de pouvoir exécuter du javascript.

Les navigateurs ie9+, Edge, Safari, Firefox, Chrome, Opera, Vivaldi sont supportés sur Windows / Mac / Linux. Les navigateurs Safari, Chrome, Firefox, Vivaldi sont supportés sur mobile.

Configuration du site

Le thème de base est une modification du thème [Copper](#) pour Hugo. Il se trouve dans le répertoire `themes/copper-sud`.

La configuration de base du site se trouve dans le fichier `config.toml` à la racine du site. Le fichier est assez simple à comprendre : on peut définir des rubriques et des pages.

Le Design se trouve en particulier dans le fichier `themes/cupper-sud/assets/css/template-styles.css`. C'est à cet endroit que la majorité des modifications doivent être effectuées pour modifier le design du site. De nombreux ajouts et règles ont été produites pour les besoins du simulateur.

L'image en une doit porter le nom `logo.svg` et se trouver dans le dossier `static/images` du site. Les favicons doivent être placées dans le dossier `static`.

Composition du site

Dans le dossier `content` se trouvent les pages en racine du site. La page spéciale `_index.md` est la page d'accueil. C'est dans cette page que le code permettant de gérer le simulateur est mis en place.

Les fichiers sont écrits en Markdown, un langage à balisage léger bien plus facile à écrire que le HTML. L'extension d'un fichier Markdown est `.md`.

Pour avoir un aperçu efficace de la syntaxe Markdown, il suffit de se rendre ici : <https://wprock.fr/blog/markdown-syntaxe/>.

Chaque fichier commence par 5 lignes :

```
---
title: "Comparateur de retraites entre le système actuel et le projet du
gouvernement"
date: 2019-10-12T08:08:47+02:00
draft: false
---
```

- title: Ce sera le titre de la page. Il ne faut pas l'omettre.
- date: la date de création. Inutile sauf si on veut classer par ordre chronologique les pages.
- draft: il faut mettre à **false** pour être certain-e que le site sera généré avec la page.

Pour éviter d'avoir à copier/coller ces lignes, on peut créer un post en tapant simplement depuis la racine du site :

```
hugo new content/nomdelarticle.md
```

Simulateur

Le simulateur est composé de code javascript (sans appel à une bibliothèque externe) dans une page html.

Il est intégré grâce à un *shortcode* du thème défini dans le répertoire `themes/cupper-sud/layouts/shortcodes` : `rawhtml.html`.

La syntaxe est la suivante : dans un article rédigé en Markdown, il suffit d'insérer :

```
{{< rawhtml >}}
Le code html
{{< /rawhtml >}}
```

La logique de masquage et affichage des différents menus est implémentée dans le fichier `static/js/esthetique.js`.

L'algorithme

L'algorithme compare essentiellement la différence de revenu projeté entre le système actuel et le système par points annoncé par Delevoye.

L'implémentation se trouve dans `static/js/salaires.js`. Le code est documenté au maximum pour permettre sa réutilisation.

Les variables utiles sont déclarées en début de script pour permettre au maximum une appropriation de la logique du code.