Construire un site web : quelques informations

Séminaire Infomath

https://infomath.pages.math.cnrs.fr/tutorial/website/

Julien Guillod

4 novembre 2021

Fortement inspiré de la présentation de Bertrand Thierry

Le web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

1989 Projet de World Wide Web au CERN[1]

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

1989 Projet de World Wide Web au CERN[1]

1990 Premier serveur et navigateur : info.cern.ch

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

1989 Projet de World Wide Web au CERN[1]

1990 Premier serveur et navigateur : info.cern.ch

1993 Le CERN fait du World Wide Web un logiciel libre

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

1989 Projet de World Wide Web au CERN[1]

1990 Premier serveur et navigateur : info.cern.ch

1993 Le CERN fait du World Wide Web un logiciel libre

1994 10k serveurs, 10M utilisateurs. Formation du W3C

[1] https://home.cern/fr/science/computing/birth-web/short-history-web
[2] https://websitesetup.org/news/how-many-websites-are-there/

Bref historique

1969 Lancement d'ARPANET (qui deviendra Internet)

1971 Courrier électronique (séparateur @)

1989 Projet de World Wide Web au CERN[1]

1990 Premier serveur et navigateur : info.cern.ch

1993 Le CERN fait du World Wide Web un logiciel libre

1994 10k serveurs, 10M utilisateurs. Formation du W3C

1994-2005 Yahoo! (1994), Amazon (1995), Google (1998), Wikipédia (2001), Facebook (2004), Youtube (2005)

[1] https://home.cern/fr/science/computing/birth-web/short-history-web

[2] https://websitesetup.org/news/how-many-websites-are-there/

Bref historique

1969	Lancement d'ARPANET (qui deviendra Internet)
1971	Courrier électronique (séparateur @)
1989	Projet de World Wide Web au CERN[1]
1990	Premier serveur et navigateur : info.cern.ch
1993	Le CERN fait du World Wide Web un logiciel libre
1994	10k serveurs, 10M utilisateurs. Formation du W3C
1994-2005	Yahoo! (1994), Amazon (1995), Google (1998), Wikipédia (2001), Facebook (2004), Youtube (2005)
2008	Sortie de l'iPhone

[1] https://home.cern/fr/science/computing/birth-web/short-history-web
[2] https://websitesetup.org/news/how-many-websites-are-there/

Bref historique

1969	Lancement d'ARPANET (qui deviendra Internet)
1971	Courrier électronique (séparateur @)
1989	Projet de World Wide Web au CERN[1]
1990	Premier serveur et navigateur : info.cern.ch
1993	Le CERN fait du World Wide Web un logiciel libre
1994	10k serveurs, 10M utilisateurs. Formation du W3C
1994-2005	Yahoo! (1994), Amazon (1995), Google (1998), Wikipédia (2001), Facebook (2004), Youtube (2005)
2008	Sortie de l'iPhone
2020	4G d'internautes (60% de la population), 576000 nouveau site par jour, 100M d'années pour *télécharger internet* à 1Go/s[2].

[1] https://home.cern/fr/science/computing/birth-web/short-history-web
[2] https://websitesetup.org/news/how-many-websites-are-there/

Bref historique : technologies et langages

1990 HTML : HyperText Markup Language

URI : Uniform Resource Identifier

HTTP : Hypertext Transfer Protocol

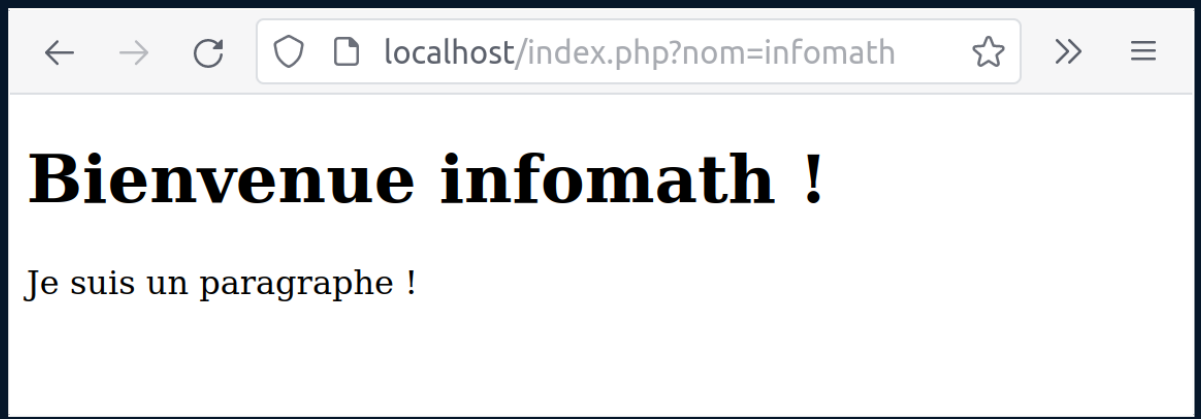
```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  <h1>Bienvenue !</h1>
  <p>Je suis un paragraphe !</p>
  </body>
</html>
```

Bref historique : technologies et langages

1990 HTML : HyperText Markup Language
URI : Uniform Resource Identifier
HTTP : Hypertext Transfer Protocol

1994 PHP: Hypertext Preprocessor

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6   <h1>Bienvenue <?php echo $_GET['nom']; ?> !</h1>
7   <p>Je suis un paragraphe !</p>
8   </body>
9 </html>
```



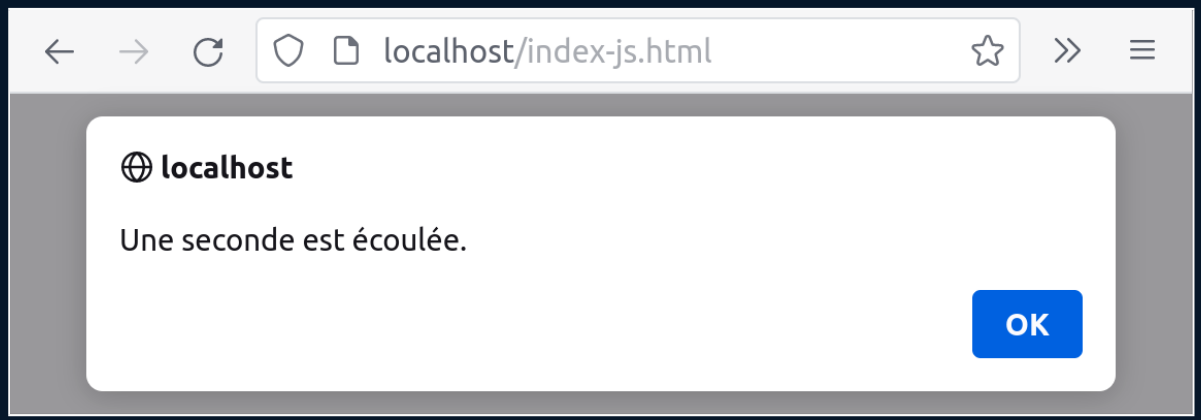
Bref historique : technologies et langages

1990 HTML : HyperText Markup Language
URI : Uniform Resource Identifier
HTTP : Hypertext Transfer Protocol

1994 PHP: Hypertext Preprocessor

1995 JavaScript

```
<!DOCTYPE html>
<html>
  <script>
    setTimeout(function () {
      alert('Une seconde est écoulée.');
    }, 1000);
  </script>
</html>
```



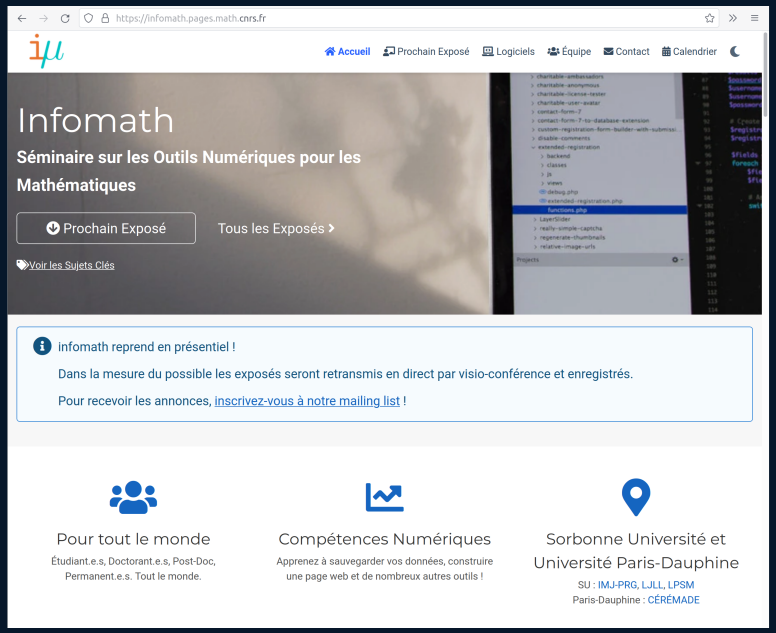
Bref historique : technologies et langages

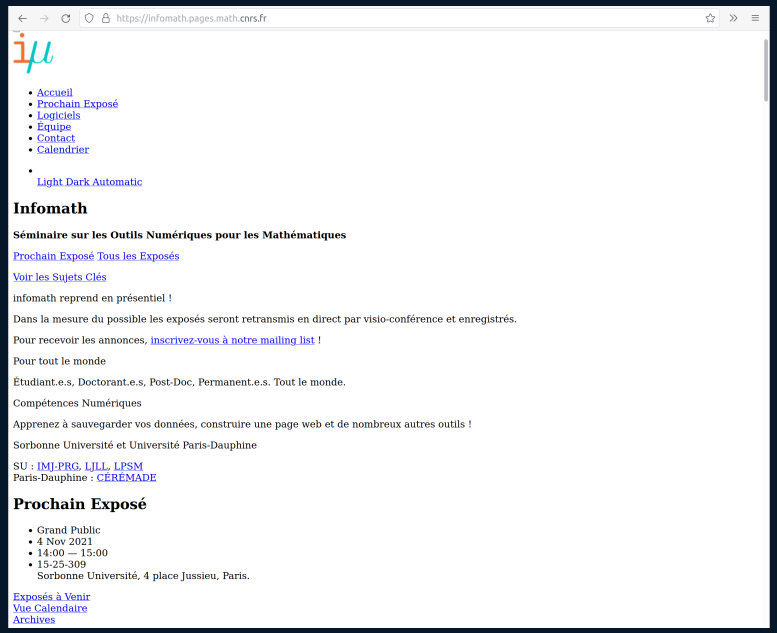
1990 HTML : HyperText Markup Language
URI : Uniform Resource Identifier
HTTP : Hypertext Transfer Protocol

1994 PHP: Hypertext Preprocessor

1995 JavaScript

1996 CSS : Cascading Style Sheets
(séparation contenu et de la forme)

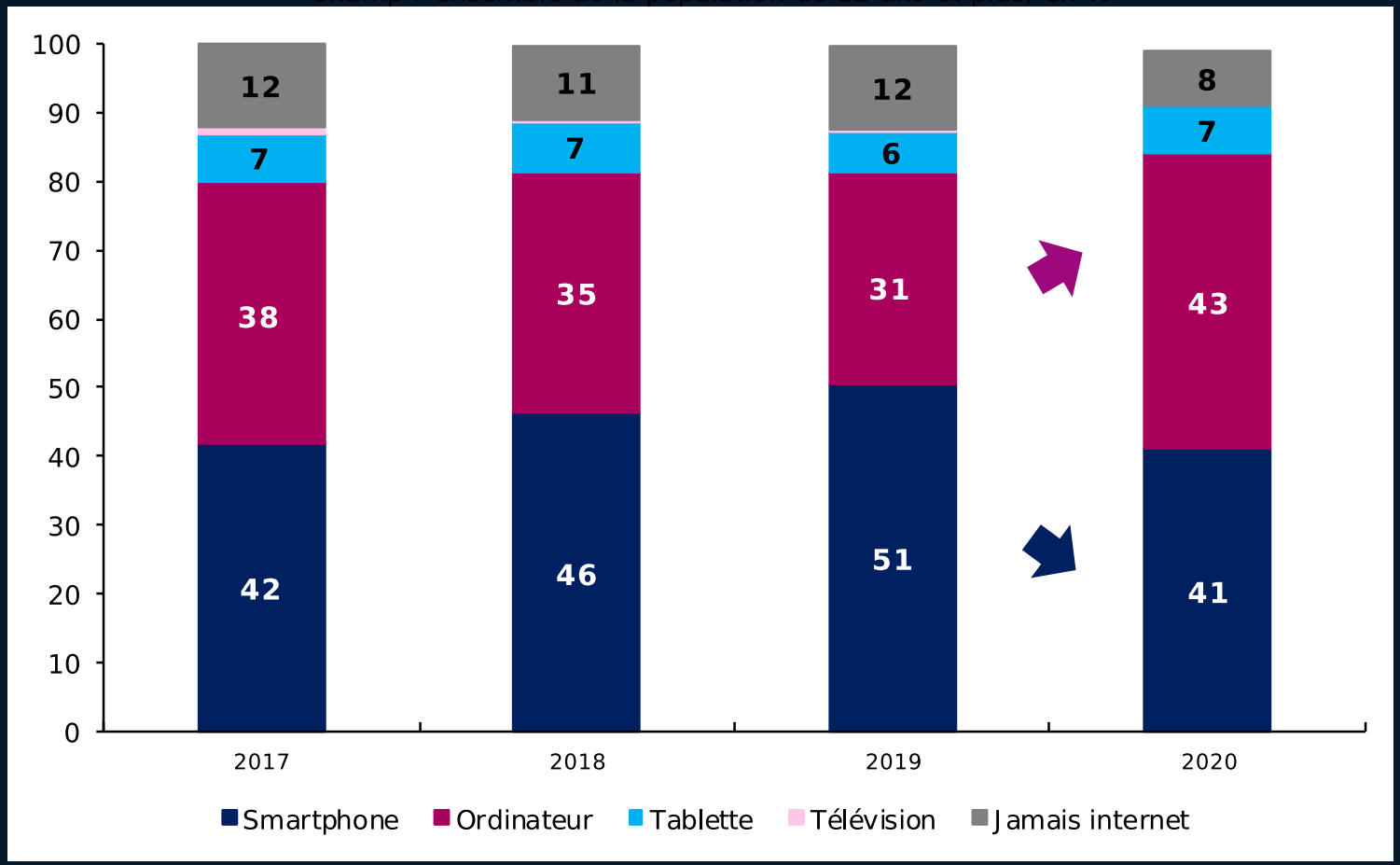




Avec et sans fichier CSS

Bref historique : terminal[1]

Quel équipement utilisez-vous le plus souvent pour vous connecter à internet ?
- Champ : ensemble de la population de 12 ans et plus, en % -

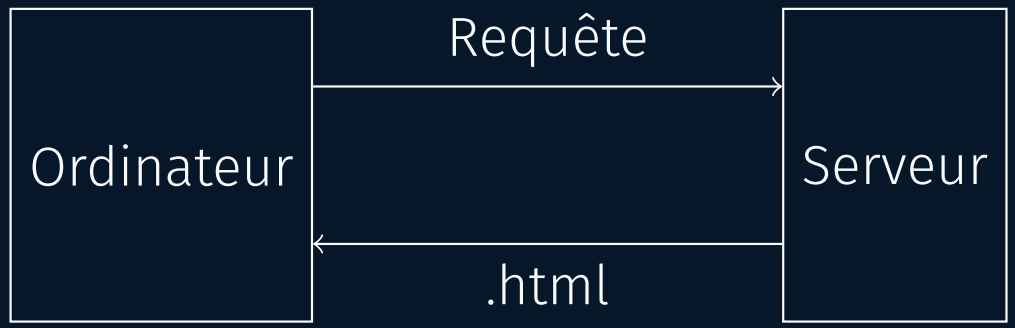


Source : CREDOC, Baromètres du numérique.

$\Longrightarrow$ Un site internet doit s'adapter à l'écran (et non l'inverse) !

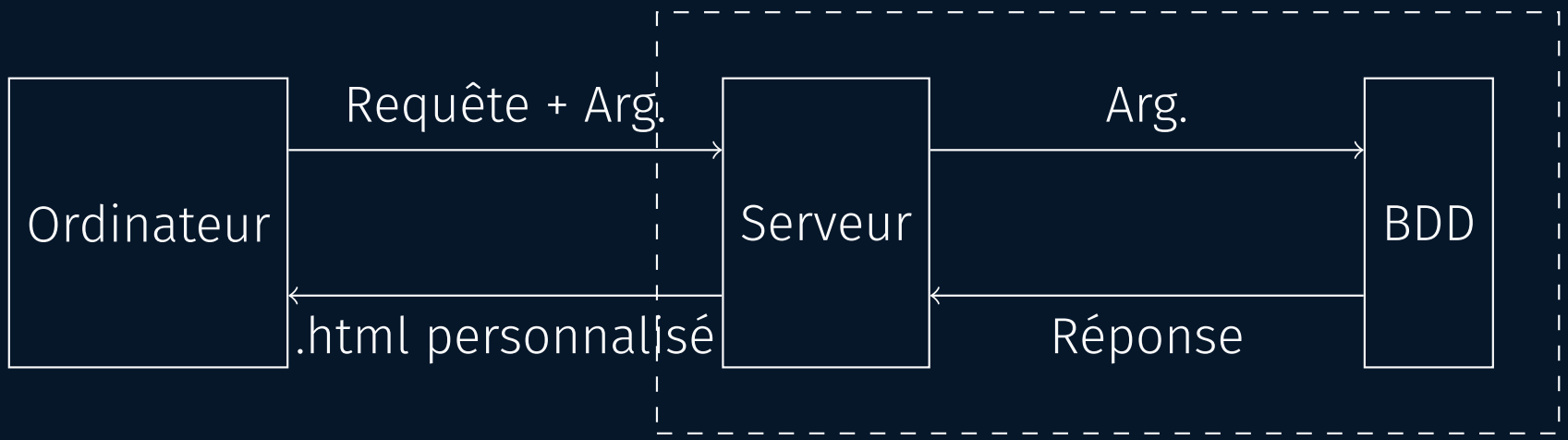
[1] Baromètre du numérique 2021

Site statique/dynamique : le cas statique



- Réponse identique, peu importe qui émet la requête et ce qu'elle contient
- Fichier `.html` codé en dur

Site statique/dynamique : le cas dynamique



- Envoi d'arguments

```
<p>https://monsite.com?nom=infomath
```

- Traitement des arguments, possiblement avec une base de données

```
<p>Bonjour <?php echo $_GET['nom']; ?> !</p>
```

- Réponse avec un fichier `.html` personnalisé

```
<p>Bonjour infomath !</p>
```

Un mot sur la cyber-sécurité[1]

Dynamique ($\Longrightarrow$ BDD) $\Longrightarrow$ Attaque serveur (vol, destruction, ...) :

Nom	#Données	Année
Dockerhub	190 000	2019
L'Express	700 000	2018
Uber	57 Millions	2017
Facebook	87 Millions	2014
Marriott	500 Millions	2018
Yahoo!	3 Milliards	2013 (admis en 2017)

Statique = Sûr ? Non : attaque possible côté client (par exemple javascript qui mine en cachette)

Rappel : L'hébergeur est responsable de la sécurité des données !

[1] On en reparlera dans un prochain infomath !

Handicap et accessibilité

Les non-voyants

- Navigateur spécialisé (lit/transcrit uniquement le contenu)
- Séparation du contenu de la forme
- Texte alternatif pour les images, problème de captcha, ...

Handicap et accessibilité

Les non-voyants

- Navigateur spécialisé (lit/transcrit uniquement le contenu)
- Séparation du contenu de la forme
- Texte alternatif pour les images, problème de captcha, ...

Pensons aussi aux personnes...

Daltoniennes : pensez à vos figures ![3][4]

[3] Sélecteur de couleurs : ColorBrewer
[4] Test de sites web pour personnes daltoniennes : Toptal

Handicap et accessibilité

Les non-voyants

- Navigateur spécialisé (lit/transcrit uniquement le contenu)
- Séparation du contenu de la forme
- Texte alternatif pour les images, problème de captcha, ...

Pensons aussi aux personnes...

Daltoniennes : pensez à vos figures ![3][4]

Sourdes : sous-titres de vidéos, ...

[3] Sélecteur de couleurs : ColorBrewer

[4] Test de sites web pour personnes daltoniennes : Toptal

Handicap et accessibilité

Les non-voyants

- Navigateur spécialisé (lit/transcrit uniquement le contenu)
- Séparation du contenu de la forme
- Texte alternatif pour les images, problème de captcha, ...

Pensons aussi aux personnes...

- Daltoniennes : pensez à vos figures ![3][4]
- Sourdes : sous-titres de vidéos, ...
- Handicapées moteur : navigation possible sans souris

[3] Sélecteur de couleurs : ColorBrewer
[4] Test de sites web pour personnes daltoniennes : Toptal

Handicap et accessibilité

Les non-voyants

- Navigateur spécialisé (lit/transcrit uniquement le contenu)
- Séparation du contenu de la forme
- Texte alternatif pour les images, problème de captcha, ...

Pensons aussi aux personnes...

- Daltoniennes : pensez à vos figures ![3][4]
- Sourdes : sous-titres de vidéos, ...
- Handicapées moteur : navigation possible sans souris

$\Longrightarrow$ Un site doit être accessible et
ne doit jamais dépendre d'un navigateur[5] !

[3] Sélecteur de couleurs : ColorBrewer
[4] Test de sites web pour personnes daltoniennes : Toptal
[5] Plus jamais de : "Ce site est optimisé pour Chrome/Firefox/~~Internet Explorer~~"

Conclusion intermédiaire

Le web

- Développement (très) rapide
- Devenu (presque) indispensable dans nos métiers (et nos vies) en moins de 30 ans
- L'accès au web change : smartphone $\approx$ ordinateur

Développer un site web aujourd'hui

- Plusieurs métiers : programmation, design, contenu, ...
- Cybersécurité : ne soyons pas naïf
- Accessibilité : ne les oublions pas
- Vie privée (RGDP) : attention en cas d'ajout de tracking

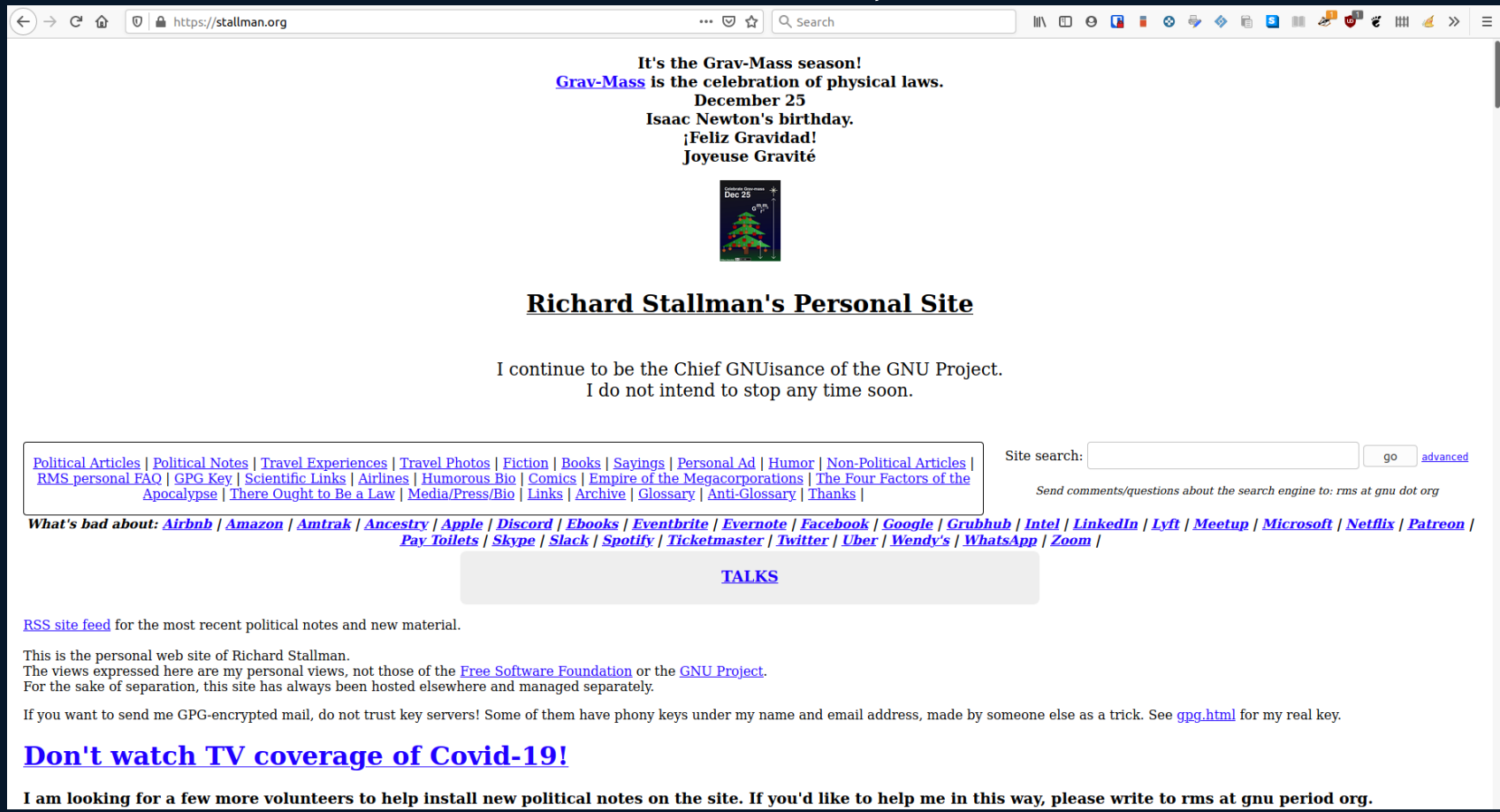
En pratique : quelques outils

Quelques outils

- Par copier/coller du site d'un(e) collègue
  Tu peux copier/coller mon site, je n'y comprends rien, je l'ai moi même copier/coller d'un collègue maintenant à la retraite. Un peu comme mes préambules LaTeX...

Quelques outils

- Par copier/coller du site d'un(e) collègue
- Statique from scratch : *simple mais (très) efficace !* La page web de Richard Stallman en est un bel exemple !



Quelques outils

- 😐 Par copier/coller du site d'un(e) collègue
- 😀 Statique from scratch : *simple mais (très) efficace !*
- 😐 Dynamique from scratch : *Haut risque de faille de sécurité (critique)* !

Quelques outils

- Par copier/coller du site d'un(e) collègue
- Statique from scratch : *simple mais (très) efficace !*
- Dynamique from scratch : *Haut risque de faille de sécurité (critique)* !
- Dynamique avec Framework : *Minimise le risque de faille sous réserve de maintenance !*
  - Symfony, Laravel, ...
  - Flask, Bottle, Django, ...

Quelques outils

- Par copier/coller du site d'un(e) collègue
- Statique from scratch : *simple mais (très) efficace !*
- Dynamique from scratch : *Haut risque de faille de sécurité (critique)* !
- Dynamique avec Framework : *Minimise le risque de faille sous réserve de maintenance !*
  - Symfony, Laravel, ...
  - Flask, Bottle, Django, ...
- All Inclusive : Wordpress.com, Google Site, ...

Quelques outils

- Par copier/coller du site d'un(e) collègue
- Statique from scratch : *simple mais (très) efficace !*
- Dynamique from scratch : *Haut risque de faille de sécurité (critique)* !
- Dynamique avec Framework : *Minimise le risque de faille sous réserve de maintenance !*
  - Symfony, Laravel, ...
  - Flask, Bottle, Django, ...
- All Inclusive : Wordpress.com, Google Site, ...
- Content Management System (CMS) : *minimise le risque de faille sous réserve de maintenance !*
  - Spip, Drupal, Joomla, Wordpress, ...

Quelques outils

- Par copier/coller du site d'un(e) collègue
- Statique from scratch : *simple mais (très) efficace !*
- Dynamique from scratch : *Haut risque de faille de sécurité (critique)* !
- Dynamique avec Framework : *Minimise le risque de faille sous réserve de maintenance !*
  - Symfony, Laravel, ...
  - Flask, Bottle, Django, ...
- All Inclusive : Wordpress.com, Google Site, ...
- Content Management System (CMS) : *minimise le risque de faille sous réserve de maintenance !*
  - Spip, Drupal, Joomla, Wordpress, ...
- **Générateurs de sites statiques** (opt. + `git`)
  - Hugo
  - Jekyll
  - Pelican
  - Next.js, Gatsby, Nuxt, ...

JAMStack

JAMStack ?

J : Javascript (pour dynamiser)

A : API (réutilisation et scalabilité)

M : MarkUp (facilité d'écriture de page)

Stack : ...Pile 😐 ...

Principes

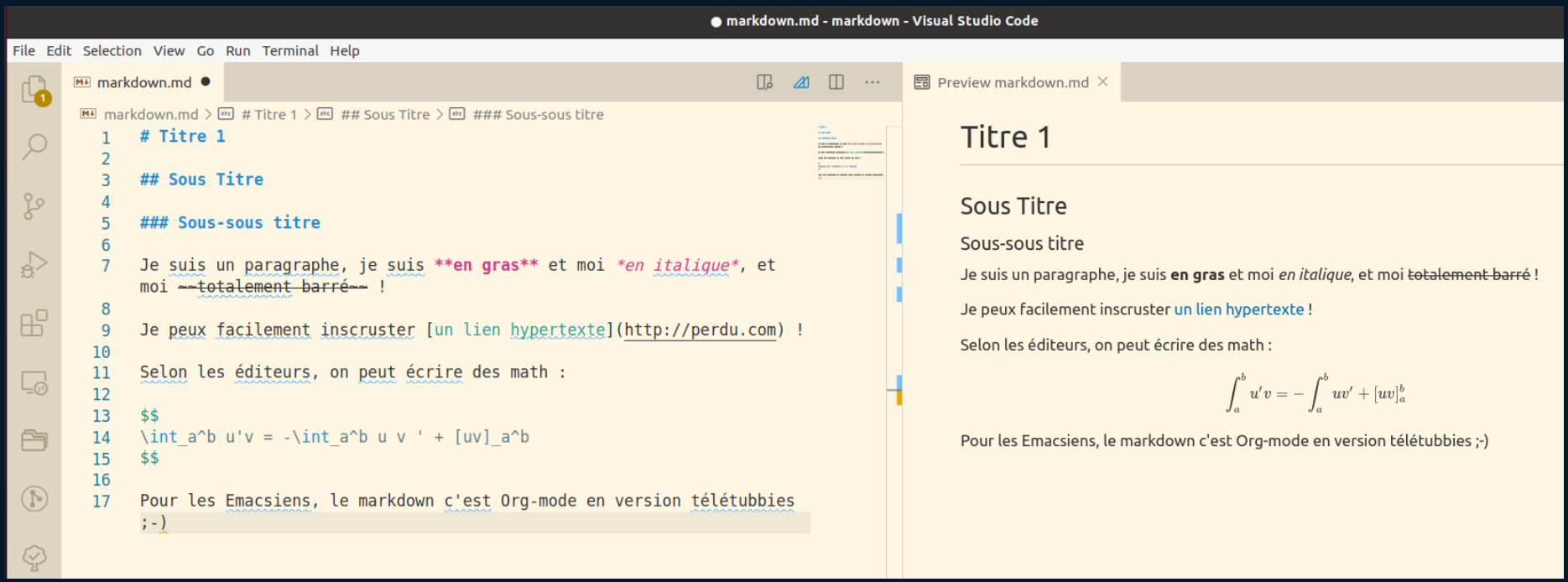
Contenu écrit en Markdown (≃ LaTeX simplifié)

Conversion en HTML via un Générateur de sites statiques

Utilisation d'un thème pour le design

Pas de base de données. Non modifiable en ligne. Que des fichiers texte. Site web statique.

Markdown


markdown.md - markdown - Visual Studio Code
File Edit Selection View Go Run Terminal Help
markdown.md
markdown.md > # Titre 1 > ## Sous Titre > ### Sous-sous titre
1 # Titre 1
2
3 ## Sous Titre
4
5 ### Sous-sous titre
6
7 Je suis un paragraphe, je suis **en gras** et moi *en italique*, et moi ~~totalement barré~~ !
8
9 Je peux facilement inscruster [un lien hypertexte](http://perdu.com) !
10
11 Selon les éditeurs, on peut écrire des math :
12
13 $$
14 \int_a^b u'v = -\int_a^b u v ' + [uv]_a^b
15 $$
16
17 Pour les Emacsiens, le markdown c'est Org-mode en version télétubbies ;-)
Preview markdown.md
Titre 1
Sous Titre
Sous-sous titre
Je suis un paragraphe, je suis en gras et moi en italique, et moi totalement barré !
Je peux facilement inscruster un lien hypertexte !
Selon les éditeurs, on peut écrire des math :
$$\int_a^b u'v = -\int_a^b uv' + [uv]_a^b$$
Pour les Emacsiens, le markdown c'est Org-mode en version télétubbies ;-)

Quelques générateurs de sites statiques

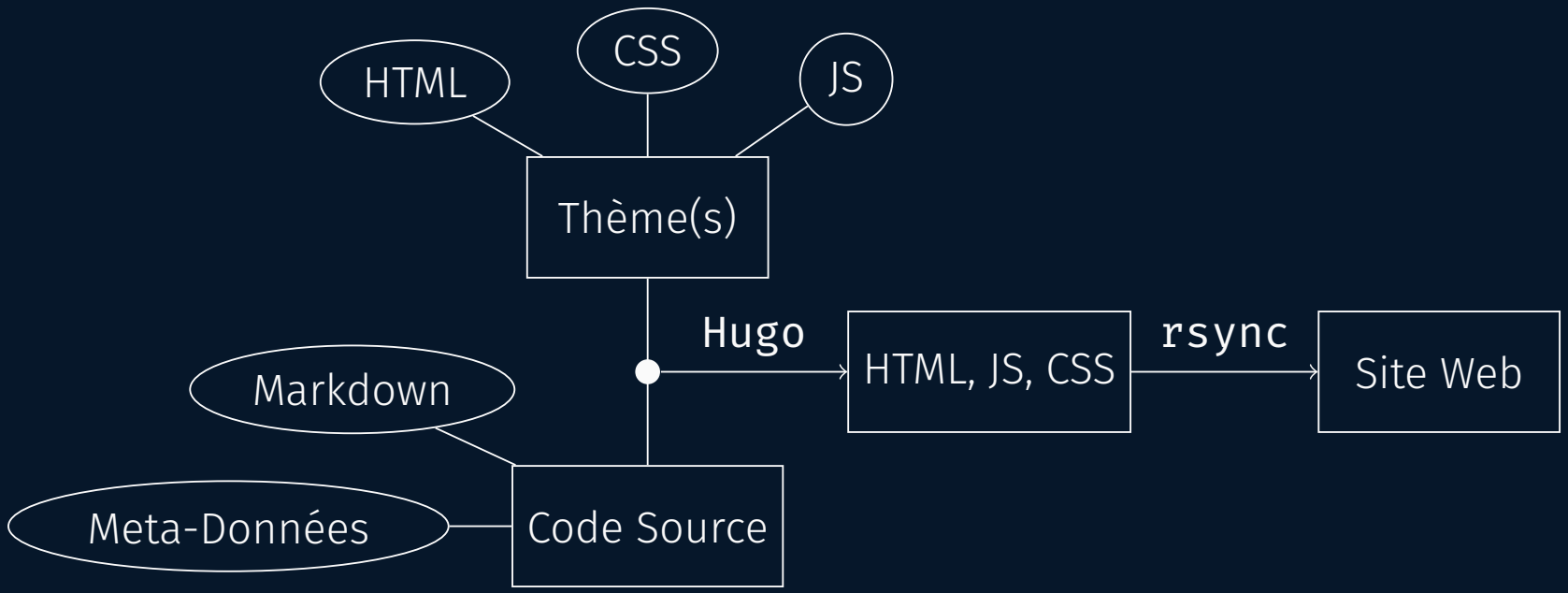
Généraliste

Nom	Langage	Type		Vitesse
Jekyll	Ruby (on Rails)	interprété	+++	Lent
Pelican	Python	interprété	++	?
Hugo	Go	Compilé	++	Rapide

Spécialement conçu pour une documentation

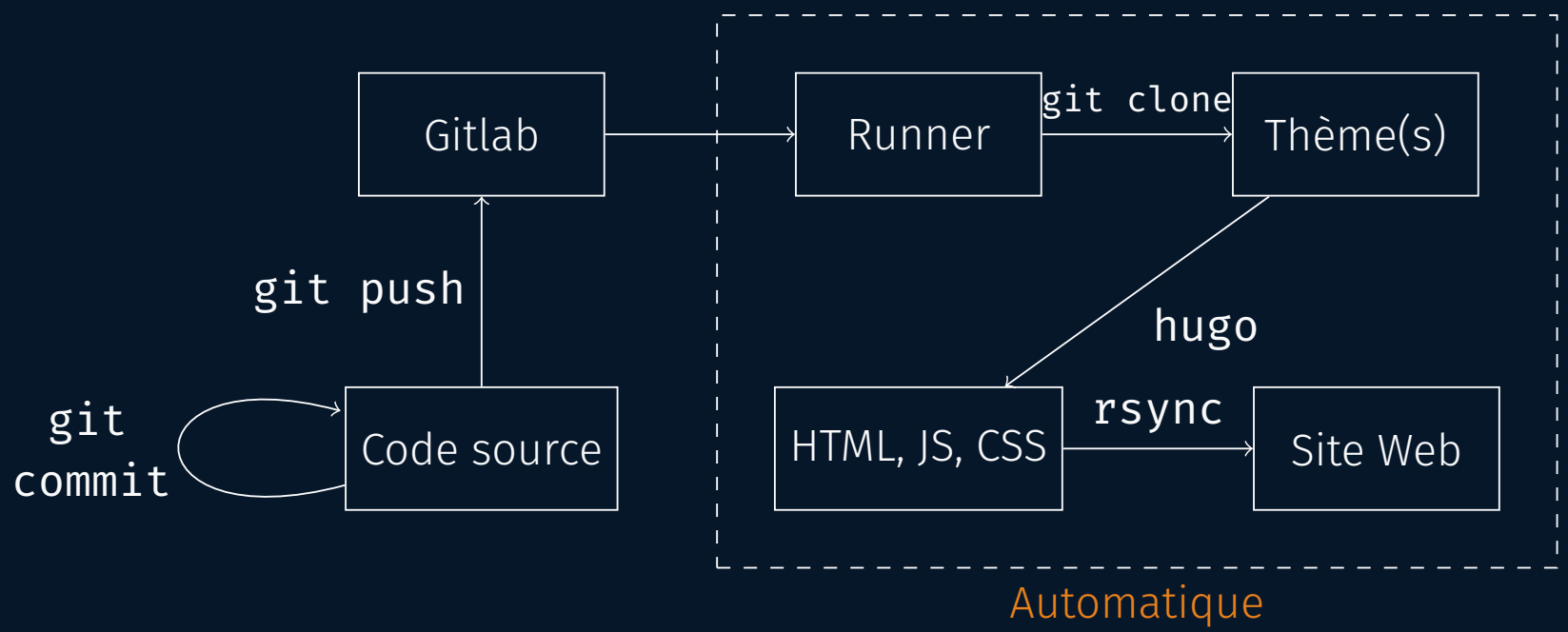
Nom	Langage/Framework
Docusaurus	ReactJS
Docsify	JS
Docz	Gatsby
Sphinx	Python

Hugo : schéma de fonctionnement



- Code source du contenu écrit en markdown
- **hugo** convertit le code source en HTML en suivant les règles établies par un thème
- Code source et code généré ont la même arborescence
- Génération de fichiers de (méta-)données : sitemap, ...
- Déploiement sur un serveur (`rsync`)
- Travailler localement : `hugo server`

Déploiement automatique avec `git` et `Gitlab page`[1]



- ✚ Déploiement automatique, sauvegarde de votre code source, versionning
- ━ Apprendre `git` (mais est-ce un désavantage 😉?)

[1] Voir https://infomath.pages.math.cnrs.fr/tutorial/website/automatisation/

Hugo en pratique : arborescence

Code source

```
/
├─ config/default/
│  ├─ config.toml
│  └─ params.toml
├─ content
│  ├─ home
│  │  └─ index.md
│  ├─ publication
│  │  └─ acta2000
│  │     └─ index.md
│  └─ talk
│     └─ superconf
│        └─ index.md
├─ layouts
├─ static
│  └─ some static files
```

Code généré

```
/public/
├─ index.html
├─ publication
│  └─ acta2000
│     └─ index.html
├─ talk
│  ├─ infomath
│  │  └─ index.html
│  └─ superconf
│     └─ index.html
├─ assets
│  ├─ css
│  │  └─ style.css
│  └─ js
│     └─ script.js
└─ some static files
```

Hugo en pratique : exemple de fichier markdown

```
---
title: Construire un Site Internet
publishDate: 2021-09-01T16:00:00+01:00
date: 2021-11-04T14:00:00+01:00
date_end: 2021-11-04T15:00:00+01:00
authors: [Julien Guillod]
location: 15-25-309
address:
  street: Sorbonne Université, 4 place Jussieu
  postcode: 75005
  city: Paris
  country: France
  country_code: FR
tags: [website, hugo, wowchemy]
categories: [Grand Public]
url_pdf: ""
url_slides: ""
links:
  - name: Tutoriel
    url: /tutorial/website
---
Dans notre communauté, il est courant de disposer de son propre site web
professionnel afin d'améliorer sa visibilité en listant, par exemple, ses
travaux : publications et exposés (avec leurs documents), CV, codes source
de logiciel, etc.
```

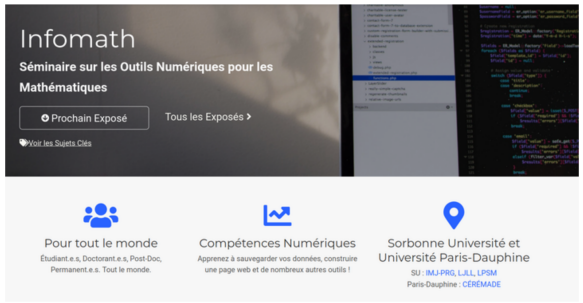
Hugo en pratique : résultat

Accueil Prochain Exposé Logiciels Équipe Contact Calendrier

Construire un Site Internet

Julien Guillod
4 Nov 2021 · Grand Public

Tutoriel



Date	4 Nov 2021 14:00 — 15:00
Lieu	15-25-309 Sorbonne Université, 4 place Jussieu 75005 Paris

Dans notre communauté, il est courant de disposer de son propre site web professionnel afin d'améliorer sa visibilité en listant, par exemple, ses travaux : publications et exposés (avec leurs documents), CV, codes source de logiciel, etc.

website hugo wowchemy

Julien Guillod
Maître de Conférences
Maître de conférences en mathématiques appliquées à Sorbonne Université au Laboratoire Jacques-Louis Lions.

Sur le même sujet

- Construire un Site Internet
- Build a professional website
- Hugo + Academic Theme

Choix du thème : Wowchemy[6]

- + Un des thèmes **Hugo** les plus avancés pour une page académique
- + Différents contenus types : publications, exposés, projets, enseignement, documentation, ...
- + Mises à jour régulières (parfois cassantes...)
- + Documentation exhaustive et communauté active (Discord)
- + Personnalisation relativement facile (inhérent à **Hugo**)
- + Page de garde par "widget" (personnalisable, ...)
- − Porté par une personne (dictateur bienveillant): arrêt brutal possible, choix parfois contestables, ...

Exemple : https://infomath.pages.math.cnrs.fr

D'autres thèmes sont disponibles.

[6] https://wowchemy.com

Time for action!

Un tutoriel est aussi proposé sur infomath.

- Installer Go et Hugo
- Cloner (ou télécharger) le dépôt : https://github.com/wowchemy/starter-hugo-academic
- Lancer le serveur local : `hugo server`
- Ouvrir `localhost:1313/` dans un navigateur
- Remplir les méta-données de configuration
- Ajouter ce qu'on veut (enseignement, blog, ...)
- Ajouter ses publications (script Python mais reste laborieux)
- Déployer le site à chaque modification
- Déploiement automatique via Gitlab/Github pages, Netlify,...

La documentation du thème est très fournie.

Conclusion

Construire un site web

- C’est compliqué mais il existe (plein) d’outils (open-source)
- N’oubliez pas la cybersécurité et l’accessibilité
- Ne perdez pas trop de temps…
- …Mais ça permet de mieux comprendre Internet !

Thank you for your attention!