3IN017 - TECHNOLOGIES DU WEB

Introduction – Architecture orientée service

14 janvier 2025

Gilles Chagnon

Organisation de l'UE

Objectifs de l'UE

SCIENCES SORBONNE UNIVERSITÉ

Acquisition de techniques pour le développement de sites Web « modernes »

→ Architectures des sites Web

→ Développement de services Web

→ Développement d'interfaces homme/machine

→ Traitement de grandes masses de données

Objectifs de l'UE

SCIENCES SORBONNE UNIVERSITÉ

Un enseignement « concret » basé sur la manipulation de technologies

→ Cours : Présentation des différentes technologies et de leur articulation, cours d'ouverture sur des sujets connexes

→ TD : Prise en main des technologies

→ TP : Mise en œuvre de ces technologies

TD et TP ont lieu sur machine. Les TPs sont tous structurés autour du développement d'un site web

Évaluation

SCIENCES SORBONNE UNIVERSITÉ

Évaluation de l'UE

- L'UE n'est pas une UE très difficile, mais **une UE dense** : une à deux technologies par semaine, aucun retour en arrière
- 50% : contrôle continu
  - Évaluation serveur à mi-parcours, document à rendre après le TME5
  - Soutenance de projet par vidéo
- 50% : Examen terminal sur feuille. L'examen est long...

Évaluation

Évaluation serveur à mi-parcours

- Document à rendre après le TME5
- Tableau de description des composants React et l'architecture *front*

Évaluation

Soutenance de projet

- Projet en binôme
- Réalisation du site côté Client + côté Serveur
- Implémentation de fonctionnalités **obligatoires** : voir cahier des charges (TD1)
- Implémentation de fonctionnalités **additionnelles**

Attention

Il y a du code existant, on le sait et on va vérifier :

- Vérification automatique de plagiat (avec projets existants sur le web et tous les projets de la promo)
- Vous devez connaître parfaitement votre code
- Vous devez savoir l'expliquer et y ajouter des modifications

Évaluation de projet

SCIENCES SORBONNE UNIVERSITÉ

À réaliser :

- Page d'accueil
- Formulaires de connexion / enregistrement
- Page de profil
- Fonctionnalités permettant de poster et consulter un message
- Plus toute autre fonctionnalité originale (par exemple agenda d'événements, messages privés, partie ouverte au public ou non...)

Concrètement

SCIENCES SORBONNE UNIVERSITÉ

Ce que l'on va utiliser :

- Côté client
  → HTML, CSS, Javascript/React JS

- Côté serveur
  → Node.js et ExpressJS (JavaScript)

- Communication client - serveur
  → React JS (librairie axios)

- Base de données
  → MongoDB

Planning de l'UE

S	Cours	TD/TME
1	Intro - AOS/Services	-
2	HTML/CSS	Modélisation et Prise en main / API REST
3	Javascript	HTML/CSS
4	React JS	Javascript
5	React JS	ReactJS
6	NodeJS	ReactJS
7	NodeJS	NodeJS
8	NoSQL	NodeJS
9	Accessibilité numérique	NoSQL
10	Mise en production, cloud	Travaux sur le projet
11	Cours d'ouverture	Travaux sur le projet
12	-	Travaux sur le projet

Attention :

On n'a que 11 semaines $\Rightarrow$ Il faut utiliser des technologies récentes et robustes.

Concrètement

Étude de cas

Le cours est structuré autour du développement « from scratch » d'un site Web incluant :

- Une interface Web pour les utilisateurs
- Une API disponible pour le développement d'applications
- Un serveur permettant le stockage de grandes masses de données dynamiques
- Une interface de traitement de données

En résumé...

SCIENCES SORBONNE UNIVERSITÉ

Les TPs sont tous structurés autour du développement d'un site Web

Cela implique :

→ Les TPs sont additifs ⇒ retard/absence à un TP doit être rattrapé avant le TP suivant

→ Les TDs introduisent les TPs ⇒ absence en TD = grosses difficultés en TP

→ Les cours présentent les technologies ⇒ absence en cours = retard en TP

En résumé...

Objectif double du cours

- Vous donner un point d'entrée pour un ensemble de technologies
- Vous « éclairer » sur les évolutions actuelles de l'informatique liée au Web

Ce que le cours n'est pas/ne vous apporte pas

- Une connaissance profonde de toutes les technologies
- Vous ne serez *pas* expert de développement web à la fin de ce cours. À vous de parfaire vos connaissances si vous le souhaitez
- Il y a des technologies différentes (Spring, Angular, Symfony...) qui demandent un niveau d'abstraction plus important. Si vous comprenez ce cours, vous saurez apprendre ces technologies.

Répétition : Cette UE est structurée autour d'un projet qui doit être la source de motivation de chacun. Les enseignants seront ouverts (et favorables) à toute proposition/personnalisation de l'UE. Pas de projet/implication $\Rightarrow$ pas de validation.

Pour vous aider : Forum de discussion

SCIENCES SORBONNE UNIVERSITÉ

- Utilisation d'une messagerie Mattermost pour les discussions entre étudiants + enseignants/étudiants
- Vous pouvez poser toutes les questions...
- Vous pouvez (devez) vous répondre entre vous (on répondra également)
- MAIS ON NE PARTAGE PAS DE CODE

Questions Cours/TD
Logistique - Examens
Grp1
Grp2
...

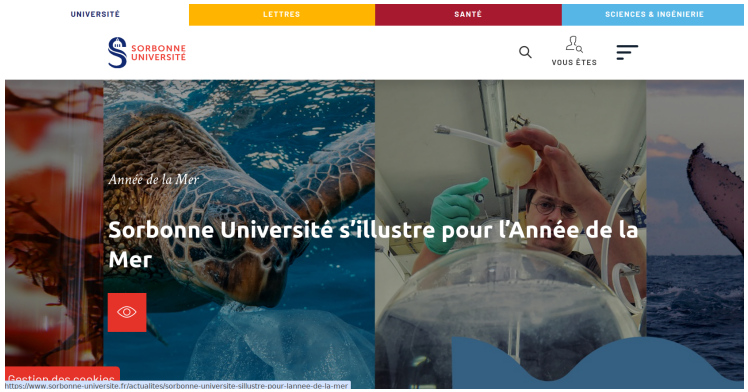
La team TechnoWeb !

- Gilles Chagnon : responsable d'UE, chargé de cours, TD/TME
- David Genova : chargé de cours
- Magali Milbergue : chargée de cours
- Cédric Geissert : chargé TD/TME
- Patrick Nollet : chargé TD/TME
- Yoann Poupart : chargé TD/TME

Questions ?

Développement web

Site web ?



Développer un site web ? Facile !

- Systèmes de gestion de contenu (Content Manager System CMS) : généralistes (Wordpress, Joomla, Drupal, Google sites, Wix, Mediawiki...) ou spécialisés (Prestashop, LMS comme Moodle, groupwares comme Nextcloud ou Microsoft 365...)
- Hébergeurs de site web : en France OVH, Hostinger, LWS...

Mais :

Développer un site web c'est aussi maîtriser son développement, sa maintenance et utiliser les technologies adaptées en fonction des besoins de ses utilisateurs !

Un peu d'histoire... La première page web

SCIENCES SORBONNE UNIVERSITÉ

- CERN – 1989 : "World Wide Web" (URL, HTML, HTTP)

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 r

What's out there?
: Pointers to the world's online information, subjects , W3 servers, etc.

Help
: on the browser you are using

Software Products
: A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical
: Details of protocols, formats, program internals etc

Bibliography
: Paper documentation on W3 and references.

People
: A list of some people involved in the project.

History
: A summary of the history of the project.

How can I help ?
: If you would like to support the web..

Getting code
: Getting the code by anonymous FTP , etc.

`http://info.cern.ch/hypertext/WWW/TheProject.html`

- premier navigateur web : `https://worldwideweb.cern.ch/`

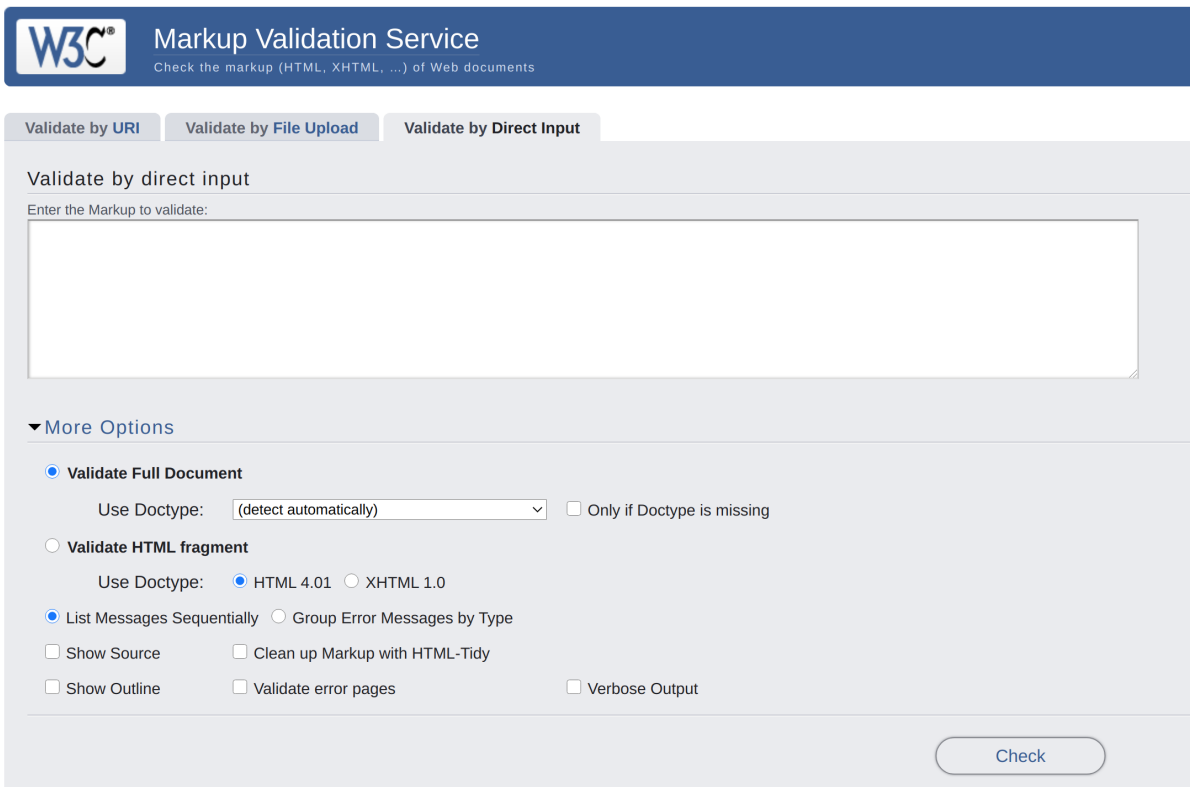
Un peu d'histoire...

- fin des années 60 : ARPANET
- 1971 : 23 ordinateurs reliés
- 1973 : définition du protocole TCP/IP
- 1983 : adoption de TCP/IP et du mot « Internet »
- 1990 : World Wide Web, HTTP, 100 000 ordinateurs
- 1992 : 1 000 000 d'ordinateurs
- 1993 : HTML
- 1994-1995 : Netscape, Yahoo !, Amazon, PHP, eBay, Internet Explorer, JavaScript, MySQL
- 1996 : CSS
- 1998 : Google
- 2000 : Éclatement de la Bulle Internet (368 000 000 ordinateurs)
- 2001 : Wikipédia
- 2002-2008 : Firefox, Wordpress, Facebook, YouTube, Twitter, Google Chrome...
- 2009 : MongoDB
- 2010 : Instagram, Pinterest
- 2014 : 1 milliard de sites Web
- 2016 : Tiktok
- 2021 : 4,6 milliards d'ordinateurs

Un peu d'histoire... 1994 : W3C

W3C

The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. Led by Web inventor and Director Tim Berners-Lee and CEO Jeffrey Jaffe, W3C's mission is to lead the Web to its full potential. (`https://www.w3.org/Consortium/`)



Site statique vs. dynamique

- Site statique : contenu de la page est figé (HTML, CSS)

- Site dynamique : contenu non figé
  - Adaptation à l'utilisateur, ses préférences
  - Scripts qui permettent de changer de contenu (PHP, JavaScript, Java, Python, Perl...)
  - Affichage varie (CSS, Javascript)
  - Syndication et agrégation de contenus (API...) : météo, actus, publicité...

→ Vers une réutilisation des composants : on crée des *applications* web !

Pourquoi est-ce compliqué de développer des sites web ?

SCIENCES SORBONNE UNIVERSITÉ

- De nombreuses technologies :
  - Document : DOM, HTML...
  - Mise en forme : CSS
  - Interaction côté client : Javascript
  - Interaction côté serveur : Java, PHP, Python, nodeJS...
  - Bases de données : MySQL, SQLite, Oracle... NoSQL, MongoDB, HBase...

- des technologies en constante évolution

- d'autres technologies : frameworks (Spring, Symfony, Laravel, Next.js... côté serveur ; (jQuery), Angular, AngularJS, VueJS, React, Svelte... côté client), CMS, sécurité, réseau...

→ **Veille technologique constante et des métiers qui se sont diversifiés (~~webmaster~~)**

Notions de base

Hypertext Transfer Protocol (HTTP)

- Protocole inventé en 1989 au CERN par Sir Tim Berners-Lee
- Il détermine comment les informations sont transmises entre le client et le serveur
- Plusieurs méthodes de transmission (requêtes) :
  - GET – la plus courante – demander une ressource.
  - POST – courante également – transmettre des données en vue d'un traitement à une ressource (création de nouvelle ressources et modification).
  - PUT – courante également – remplacer ou ajouter une ressource sur le serveur.
  - PATCH – modification partielle d'une ressource.
  - DELETE – supprimer une ressource du serveur
  - OPTIONS – options de communication d'une ressource ou du serveur en général.
  - CONNECT – utiliser un proxy comme un tunnel de communication.
  - TRACE – demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion.

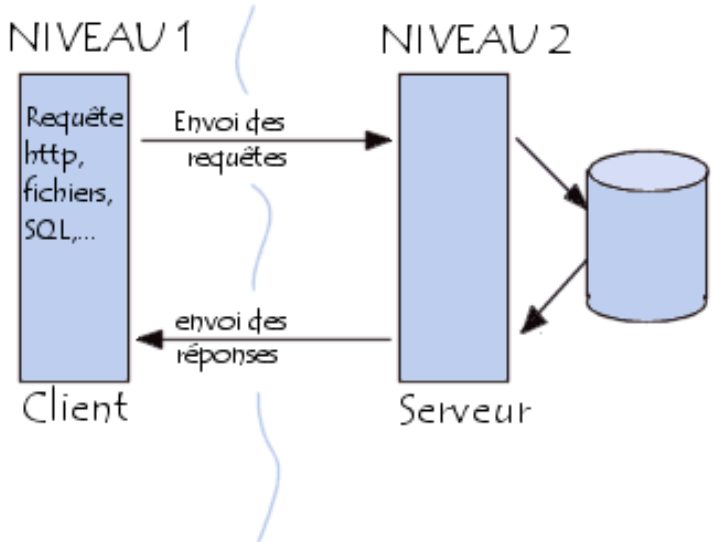
URL

- La transmission des informations est effectuée via l'URL (Uniform Resource Locator)
- Format standardisé :

Protocole	[Mot de passe]	Nom du serveur	[Port]*	Chemin
http ://	user :password@	www.monserveur.fr	:80	/login

(*) : facultatif si 80

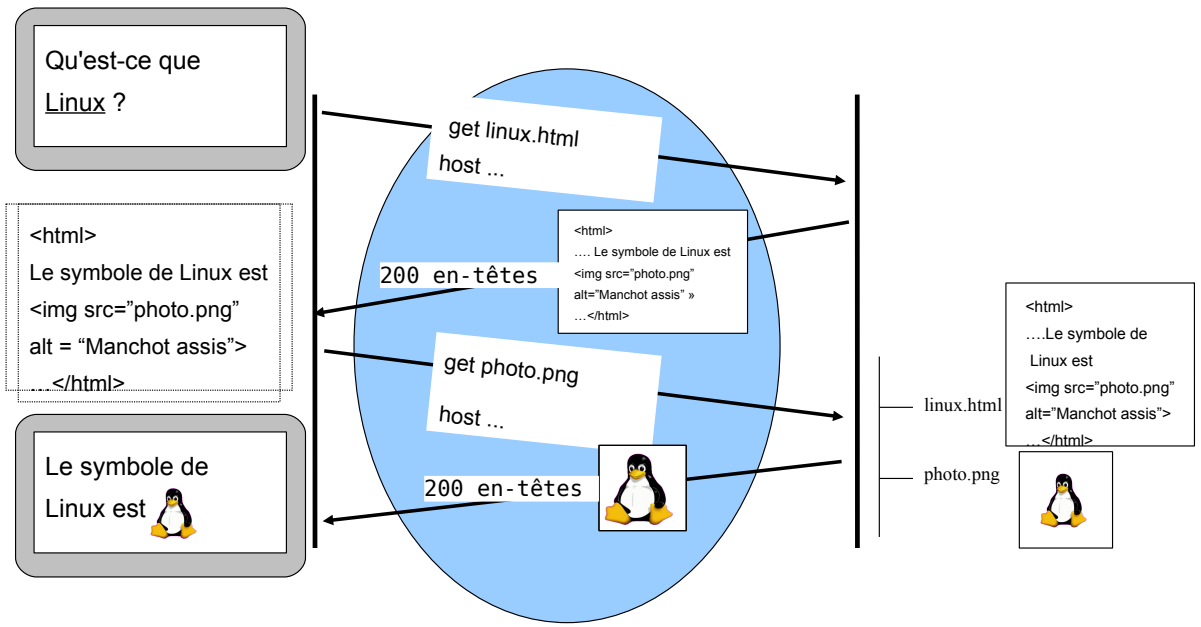
Architecture Client-Serveur



- Client (HTML, CSS, JavaScript...)
  - Il établit la connexion au serveur à destination d'un ou plusieurs ports réseau
  - lorsque la connexion est acceptée par le serveur, il communique /interroge le serveur au moyen de **requêtes**.
- Serveur
  - Il attend une connexion entrante sur un ou plusieurs ports
  - À la connexion d'un client sur le port, il communique avec le client au moyen de **réponses**.

Architecture Client-Serveur : chargement pages web

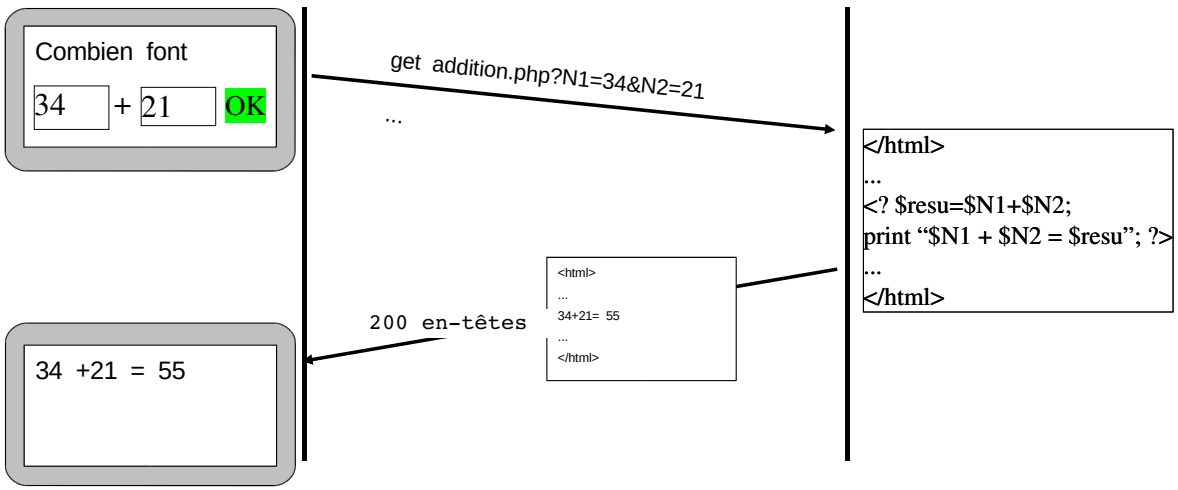
Pages Web stockées sur un serveur Web



Architecture client-serveur : web dynamique / interactions utilisateur

SCIENCES SORBONNE UNIVERSITÉ

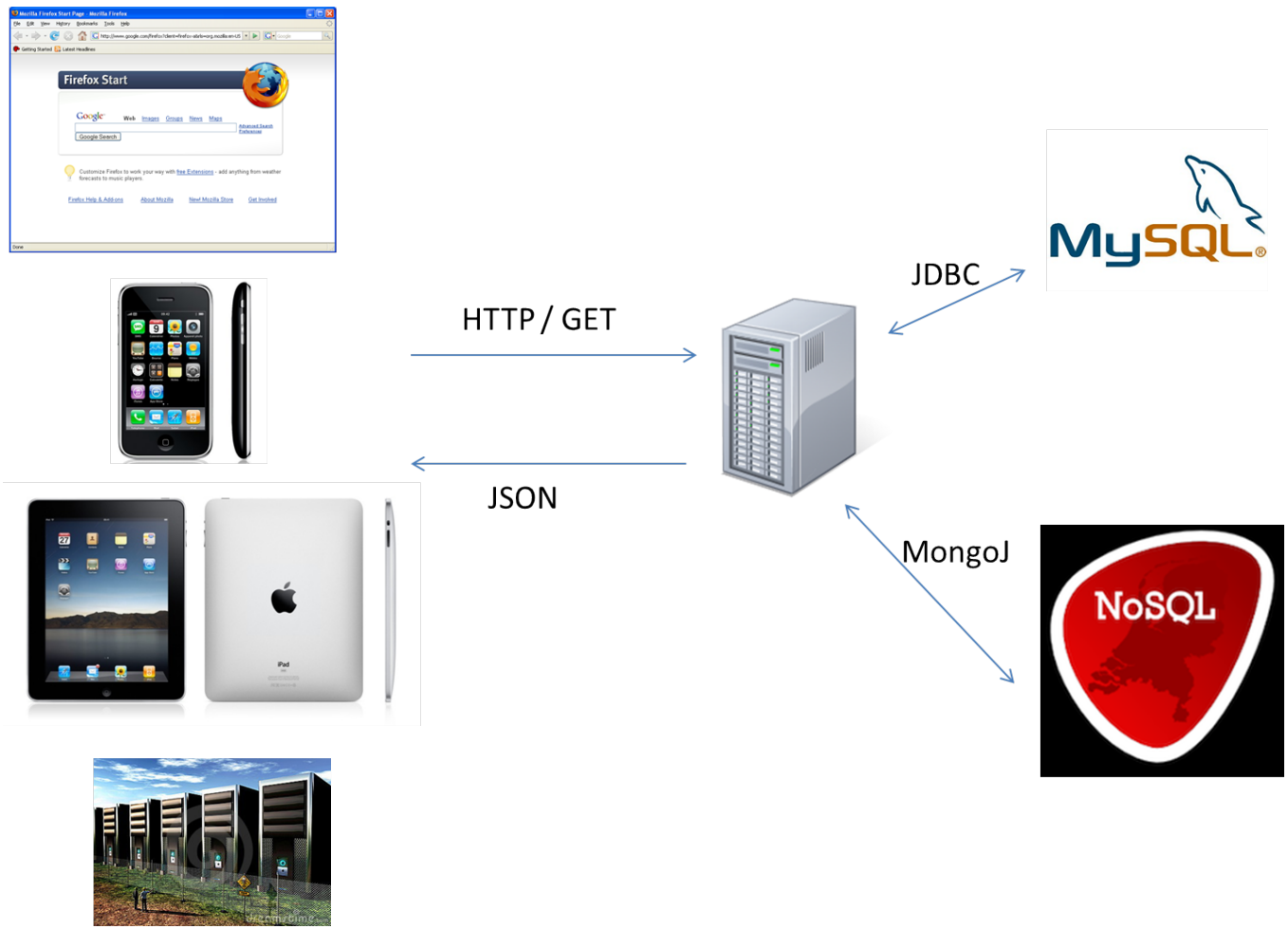
Mise en place de services Web



écran du navigateur | internet | disque dur et processeur du serveur

Web dynamique / Web API

Une interface *front* « vide » qui interroge le serveur (*back*) pour obtenir et mettre à jour le contenu dynamiquement



Vers une Architecture Orientée Service

Paradigmes de programmation : Programmation Orientée Services

SCIENCES SORBONNE UNIVERSITÉ

Architecture orientée Service (SOA pour *Services Oriented Architecture*)

- décomposition d'un problème en fonctionnalités basiques
- chaque fonctionnalité élémentaire est appelée *service*
- chaque service est caractérisé par des entrées et des sorties (résultats et messages d'information – erreurs ou autres)
- les services sont cloisonnés : un service n'en appelle pas un autre

Concevoir une application, c'est appeler des services et traiter leurs réponses pour en appeler d'autres ou restituer les résultats à l'utilisateur.
Lorsque l'architecture SOA s'appuie sur des web services, on parle alors de WSOA, pour ***Web Services Oriented Architecture***.

Avantages des *Web Services*

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feu sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

Inconvénients des WS

- Coûts de conception et de développement initiaux conséquents
- Les services Web souffrent de performances faibles pour des traitements faibles (notion de couches)

API REST

API REST (Wikipedia)

SCIENCES SORBONNE UNIVERSITÉ

REST

- REST = *REpresentational State Transfer*
- C'est une manière de construire une application pour les systèmes distribués
- REST n'est pas un protocole ou un format, c'est un style d'architecture
- Il est de plus en plus utilisé pour la réalisation d'architectures orientées services utilisant des services Web destinés à la communication entre machines.

API REST

Principes

- Repose sur une architecture client-serveur
- l'URI est important : connaître l'URI doit suffire pour nommer et identifier une ressource.
- HTTP fournit toutes les opérations nécessaires (GET, POST, PUT et DELETE, essentiellement).
- Chaque opération est auto-suffisante : il n'y a pas d'état.
- Système de couches : chaque composant voit uniquement les composants de la couche avec laquelle il interagit directement
- Utilisation des standards hypermedia : HTML ou XML ou **JSON**

Référence : RESTful Web Services, par Leonard Richardson et Sam Ruby

Avantages

- Simplicité
- Lisibilité par l'humain
- Évolutivité
- Repose sur les principes du Web
- Représentations multiples

Inconvénients

- Sécurité restreinte par l'emploi de HTTP (il faut ajouter le chiffrement TLS)
- Absence d'état : il faut trouver un moyen fiable externe de stocker les ressources et d'y accéder depuis le serveur d'API

Exemple d'accès aux ressources avec API REST (« naming »)

- Une ressource : équivalent de l'objet en java (raccourci très rapide...).
  Une « chose » et non une « action ». Par exemple : produit, utilisateur...
- Une ressource peut être un **singleton** (`customer`) ou une **collection** (`customers`).
- Accès à une ressource (appelé **document**) de la collection :
  `/customers/customerId`,
  `http://api.example.com/user-management/users/{id}`.
- Une ressource peut contenir une collection de ressources :
  `/customers/customerId/accounts`. On appelle cela un **store**.
- on peut appeler des actions (appelées **controllers**) sur la ressource : `http://api.example.com/song-management/users/{id}/playlist/play`

Pour plus d'information : `https://restfulapi.net/resource-naming/`

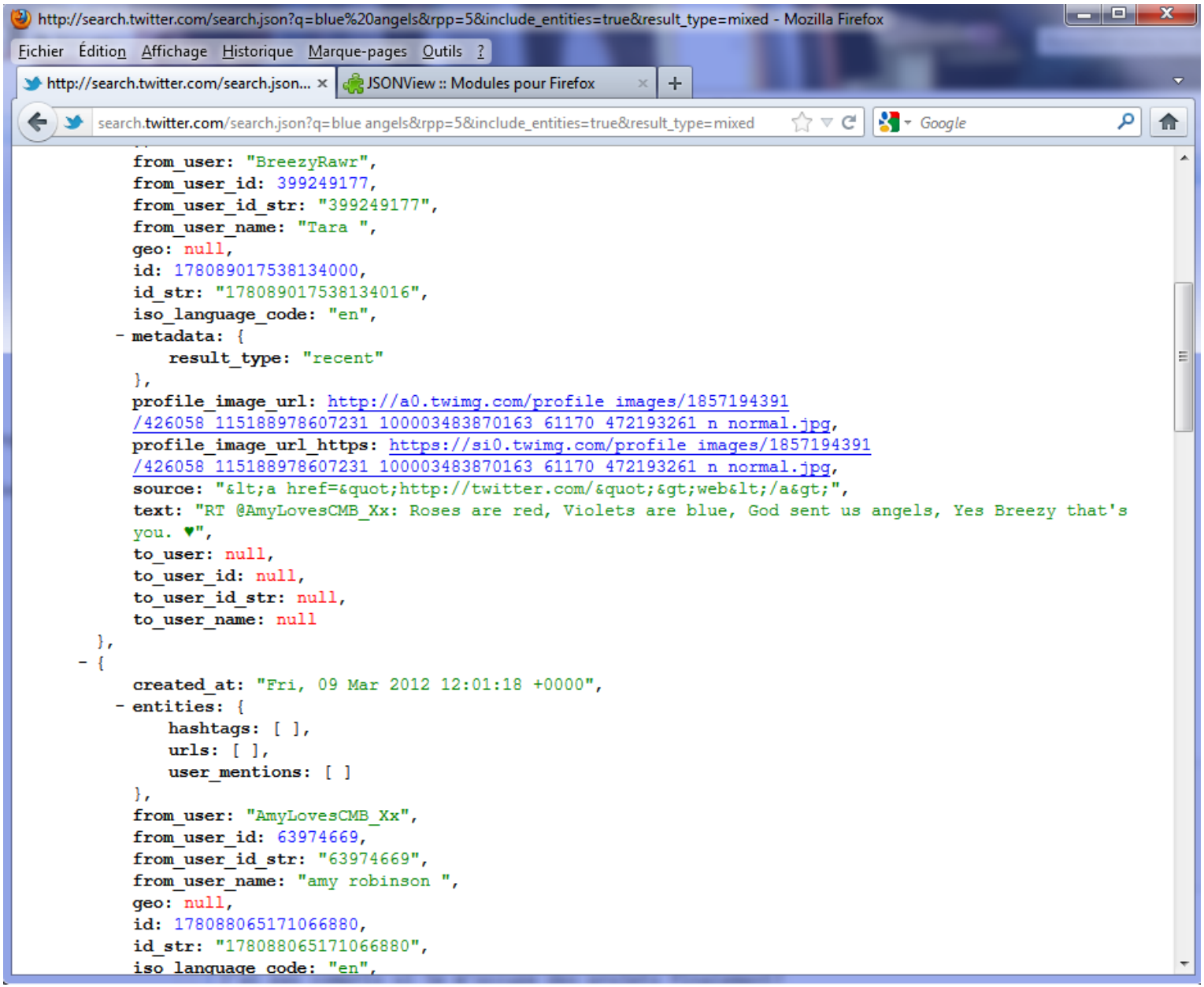
Exemple d'accès aux ressources avec API REST (« naming »)

- On utilise ensuite le protocole HTTP pour appeler les actions CRUD :

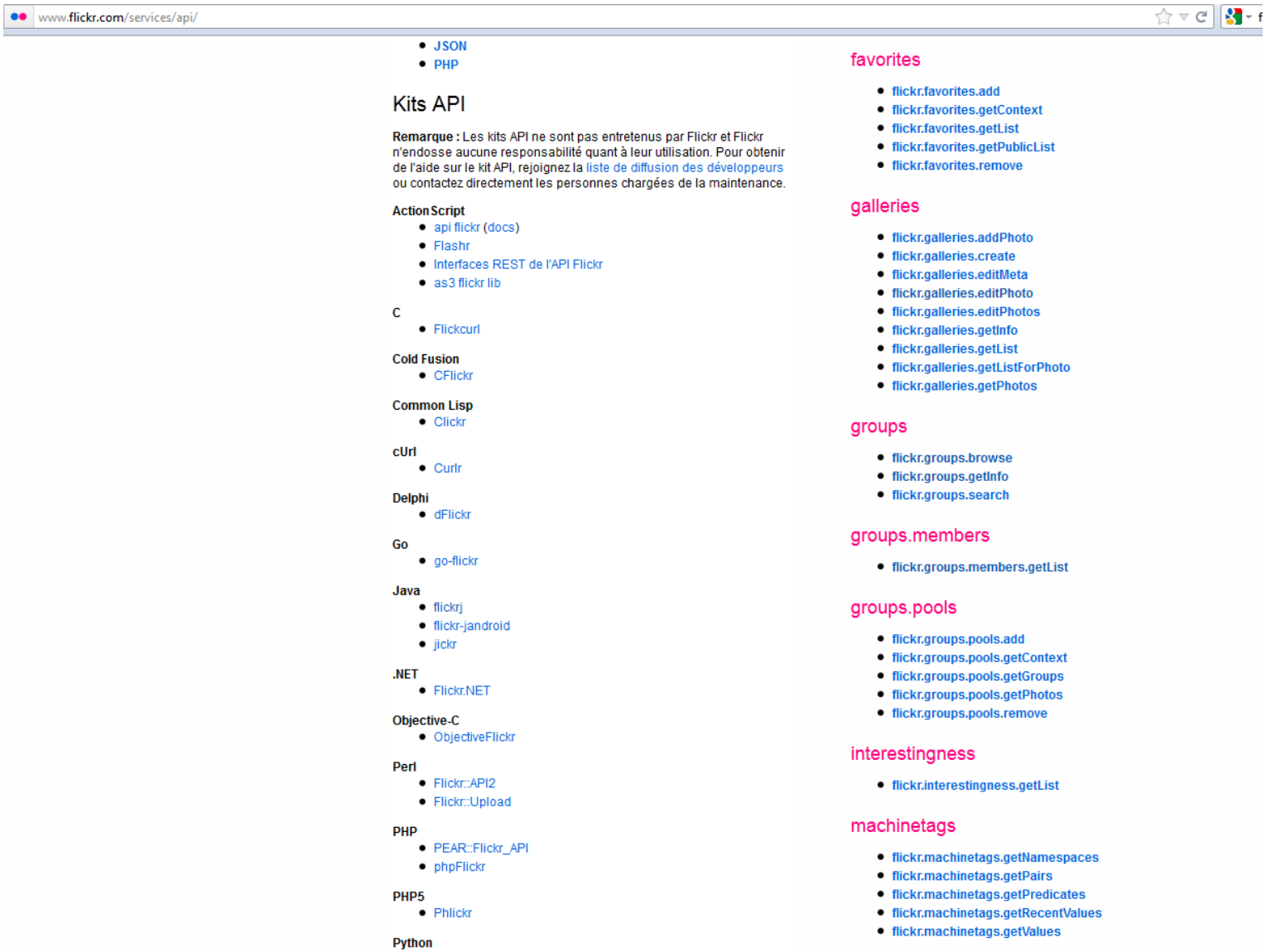
```
HTTP GET
http://api.example.com/device-management/managed-devices
//Get all devices
HTTP POST
http://api.example.com/device-management/managed-devices
//Create new Device
HTTP GET
http://api.example.com/device-management/managed-devices/id
//Get device for given Id
HTTP PUT
http://api.example.com/device-management/managed-devices/id
//Update device for given Id
HTTP DELETE
http://api.example.com/device-management/managed-devices/id
//Delete device for given Id
```

Pour plus d'information : `https://restfulapi.net/resource-naming/`

API REST : Exemple Twitter v1 (maintenant fermée)



API REST : Exemple Flickr



www.flickr.com/services/api/

- JSON
- PHP

Kits API

Remarque : Les kits API ne sont pas entretenus par Flickr et Flickr n'endosse aucune responsabilité quant à leur utilisation. Pour obtenir de l'aide sur le kit API, rejoignez la liste de diffusion des développeurs ou contactez directement les personnes chargées de la maintenance.

ActionScript
- api flickr (docs)
- Flashr
- Interfaces REST de l'API Flickr
- as3 flickr lib

C
- Flickcurl

Cold Fusion
- CFlickr

Common Lisp
- Clickr

cUrl
- Curlr

Delphi
- dFlickr

Go
- go-flickr

Java
- flickrj
- flickr-jandroid
- jickr

.NET
- Flickr.NET

Objective-C
- ObjectiveFlickr

Perl
- Flickr::API2
- Flickr::Upload

PHP
- PEAR::Flickr_API
- phpFlickr

PHP5
- Phlickr

Python

favorites
- flickr.favorites.add
- flickr.favorites.getContext
- flickr.favorites.getList
- flickr.favorites.getPublicList
- flickr.favorites.remove

galleries
- flickr.galleries.addPhoto
- flickr.galleries.create
- flickr.galleries.editMeta
- flickr.galleries.editPhoto
- flickr.galleries.editPhotos
- flickr.galleries.getInfo
- flickr.galleries.getList
- flickr.galleries.getListForPhoto
- flickr.galleries.getPhotos

groups
- flickr.groups.browse
- flickr.groups.getInfo
- flickr.groups.search

groups.members
- flickr.groups.members.getList

groups.pools
- flickr.groups.pools.add
- flickr.groups.pools.getContext
- flickr.groups.pools.getGroups
- flickr.groups.pools.getPhotos
- flickr.groups.pools.remove

interestingness
- flickr.interestingness.getList

machinetags
- flickr.machinetags.getNamespaces
- flickr.machinetags.getPairs
- flickr.machinetags.getPredicates
- flickr.machinetags.getRecentValues
- flickr.machinetags.getValues

API REST : Exemple Flickr

SCIENCES SORBONNE UNIVERSITÉ

Returns the comments for a photo

Authentification

Cette méthode n'exige pas d'authentification.

Arguments

api_key (Obligatoire)
Your API application key. See here for more details.
photo_id (Obligatoire)
The id of the photo to fetch comments for.
min_comment_date (Facultatif)
Minimum date that a a comment was added. The date should be in the form of a unix timestamp.
max_comment_date (Facultatif)
Maximum date that a comment was added. The date should be in the form of a unix timestamp.

Exemple de réponse

```
<comments photo_id="109722179">
  <comment id="6065-109722179-72057594077818641" author="35468159852@N01" authorname="Rev Dan Catt"
</comments>
```

Codes d'erreur

1: Photo not found
The photo id was either invalid or was for a photo not viewable by the calling user.
100: Invalid API Key
The API key passed was not valid or has expired.
105: Service currently unavailable
The requested service is temporarily unavailable.
111: Format "xxx" not found
The requested response format was not found.
112: Method "xxx" not found
The requested method was not found.
114: Invalid SOAP envelope
The SOAP envelope send in the request could not be parsed.

D'autres exemples d'API...

- `https://data.education.gouv.fr/pages/accueil/` : les API du Ministère de l'Éducation Nationale
- `https://portail-api.meteofrance.fr/web/fr/` : les API de Météo-France
- les API mises à disposition par les fournisseurs de contenus : médias, réseaux sociaux...

JSON

JSON

- JavaScript Object Notation
- Initialement créé pour la sérialisation et l'échange d'objets JavaScript
- Langage pour l'échange de données semi-structurées (et éventuellement structurées)
- Format texte indépendant du language de programmation utilisé pour le manipuler.
- Formellement proche du XML, mais moins verbeux

JSON

Un document JSON ne comprend que deux éléments structurels :

- des ensembles de paires nom / valeur ;
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- des objets ;
- des tableaux ;
- des valeurs génériques de type tableau, objet, booléen, nombre, chaine ou null.

JSON


```
{
    "menu":
    {
        "id": "file",
        "value": "File",
        "popup":
        {
            "menuitem":
            [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

JSON


```
{
  person: {name: "alan", phone: 3127786, email: "agg@abc.com"},
  person: &314
                {name: {first: "Sara", last: "Smith-Green"},
                phone: 2136877,
                email: "sara@math.xyz.edu",
                spouse: &443},
  person: &443
                { name: "Fred Green",
                  phone: 7786312,
                  Height: 183,
                  spouse: &314 }
}
```

Et maintenant ?

Semaine prochaine...

Prochain cours : HTML/CSS !

- TD 1 : Modélisation d'un site web
- TME1 : Installation de l'environnement, premiers codes...