



Ressources pour le cycle terminal général et technologique

Informatique et Sciences du Numérique

Autour des pages « Web » dynamiques

Ces documents peuvent être utilisés et modifiés librement dans le cadre des activités d'enseignement scolaire, hors exploitation commerciale.

Toute reproduction totale ou partielle à d'autres fins est soumise à une autorisation préalable du Directeur général de l'enseignement scolaire.

La violation de ces dispositions est passible des sanctions édictées à l'article L.335-2 du Code de la propriété intellectuelle.

Juin 2012

Présentation / Autour des pages « Web » dynamiques

1 / Thème abordé

1.1 Problématique, situation d'accroche

De nombreux sites comme les sites commerciaux, les forums de discussion, moteurs de recherche ou réseaux sociaux invitent l'utilisateur à rentrer quantités d'informations (informations personnelles, bancaires, discussions de forum, critères de recherche etc.). Au vu des pages obtenues en retour des informations saisies par l'internaute, on se doute bien que du côté serveur web un programme informatique prend en charge l'information transmise et la traite avant de renvoyer une réponse adéquate au navigateur.

Notre vécu d'internaute, comme le ciblage publicitaire dont nous sommes parfois victimes, nous amènent aussi à nous interroger sur les traces de notre passage sur les différents sites visités, stockées dans le disque dur de notre ordinateur personnel, et dont les données sont susceptibles d'être collectées voire détournées à notre insu.

L'expérience proposée par le site de démonstration de la C.N.I.L. (<http://www.cnil.fr/vos-libertes/vos-traces/experience>) nous éclaire davantage sur ces aspects.

Les activités menées dans cette séquence visent à répondre au questionnement suivant :

- Comment rendre un site Web interactif ?
- Comment le serveur Web peut-il reconnaître mon ordinateur ?
- Comment maintenir des données temporairement ou de manière plus persistante pendant la navigation ?
- Quels sont les risques encourus concernant ces données collectées pendant la navigation et pouvant porter préjudice à la personne (morale ou physique) sur le Net ?

1.2 Frontières de l'étude et prolongements possibles

Les langages et technologies pouvant être utilisés côté « client » (comme JavaScript) ne sont pas abordés dans cette séquence. La persistance des données, usuellement gérée côté serveur à l'aide d'un S.G.B.D. (Système de Gestion de Base de Données), ainsi que l'architecture Client/Serveur ne font pas l'objet d'activités dans cette séquence mais pourront être évoquées. La plupart des applications web reposent (partiellement) sur une base de données au niveau du serveur, mais l'approche de ce domaine représente un assez gros investissement.

2 / Objectifs pédagogiques

2.1 Disciplines impliquées

Un enseignant de philosophie peut très bien prolonger l'aspect technique de cette ressource en faisant réfléchir les élèves sur les problèmes éthiques et philosophiques causés par la persistance des données sur les espaces numériques interconnectés.

2.2 Prérequis

- les bases du langage de description HTML ;
- les bases de la programmation dans le langage retenu (le langage Python par exemple) ;
- un serveur pouvant exécuter des programmes dans ce langage (on peut éventuellement se contenter d'un serveur en local).

2.3 Éléments du programme

- Langages et programmation / Langage de description HTML
- Représentation de l'information / persistance de l'information.

2.4 Quelques compétences et capacités mises en jeu

Concevoir et réaliser une solution à un problème :

- mettre un programme au point en le testant, en l'instrumentant ;
- créer et analyser une page Web en langage HTML.

Décrire et expliquer une situation :

- maîtriser l'adressage sur un réseau pour identifier des machines distantes.

Faire un usage responsable des sciences du numérique :

- prendre conscience de la persistance de l'information sur les espaces numériques interconnectés ;
- savoir se comporter de manière responsable par rapport au droit des personnes ;
- prendre conscience que les données personnelles peuvent être collectées, qu'elles peuvent rester disponibles sur le web pendant une très longue période.

3 / Modalités de mise en œuvre

3.1 Durée prévue pour la partie se déroulant en classe

Trois séances de deux heures.

3.2 Type de l'animation

En classe entière puis en binôme pour les travaux dirigés ou pratiques.

3.3 Projet ?

Cette séquence est un prolongement d'une précédente séquence sur le langage HTML pour insister sur le fait que ce langage peut aussi servir de cible à des générateurs de pages. Un projet est toujours possible mais il faut prévoir des séances supplémentaires.

3.4 Recherches documentaires

La recherche documentaire concerne d'une part les spécifications des bibliothèques logicielles utilisées («framework») et d'autre part les aspects juridiques, éthiques et philosophiques sur les problèmes liés au droit à la vie privée et au droit à l'oubli.

3.5 Production des élèves

Programmes complétés et répondant au cahier des charges, accompagnés de synthèses de réflexions sur l'éthique et le comportement à adopter par rapport aux droits des personnes dans les espaces numériques.

4 / Outils logiciels et références

- Éditeur de texte avec coloration syntaxique, par exemple NotePad++ : <http://notepad-plus-plus.org> (charger le complément NppExec pour avoir une console de dialogue et de contrôle d'exécution) ou encore Geany : <http://www.geany.org> ou Bluefish : <http://bluefish.openoffice.nl>
- Langage Python : <http://www.python.org/getit>
On peut aussi trouver Python avec un éditeur de texte adapté sur SIALLE sous le nom « AmiensPython » : <http://www.cndp.fr/sialle/fiche-detaillee-amiens-python-399.php>
- Bibliothèque HTTP CherryPy pour Python : <http://www.cherrypy.org>
- Gerard Swinnen, *Apprendre à programmer avec Python*, livre en téléchargement gratuit : à l'adresse <http://www.inforef.be/swi/python.htm> (chapitre 17 – Applications Web)
- « How to use cookies with CherryPy » : <http://www.cherrypy.org/wiki/Cookies>

5 / Auteurs

Fabien Le Bris, professeur de STI, académie de Nantes

André Guigot, professeur de philosophie, académie de Nantes

Autour des pages « Web » dynamiques

1 / Créer des pages « web » dynamiques

1.1 Contexte

Classiquement, une application Web tourne autour d'un serveur Web accompagné d'un serveur de données. Un logiciel applicatif connecté au serveur Web se charge de générer dynamiquement les pages Web afin de créer l'interactivité avec le « client ».

Cette séquence vise à montrer le principe de la création dynamique des pages à l'aide d'un langage de programmation côté serveur. On en profite alors pour glisser dans le code de quoi créer ces fameux « cookies » qui donnent l'impression à l'internaute de laisser des « traces » plus ou moins persistantes de son passage à chaque visite.

1.2 Dynamisme des pages

*Les premières pages HTML étudiées sont **statiques** au sens où elles sont, une fois déposées sur le serveur, invariables et donnent toujours le même affichage à la consultation avec une transmission des données qui se fait essentiellement du serveur vers le navigateur.*

On fait constater aux élèves que dans le cas d'applications Web interactives, les échanges sont plus complexes et se font en deux temps. A partir de la requête du client et des arguments passés, le serveur reconstruit la page Web avant de la retourner à celui-ci.

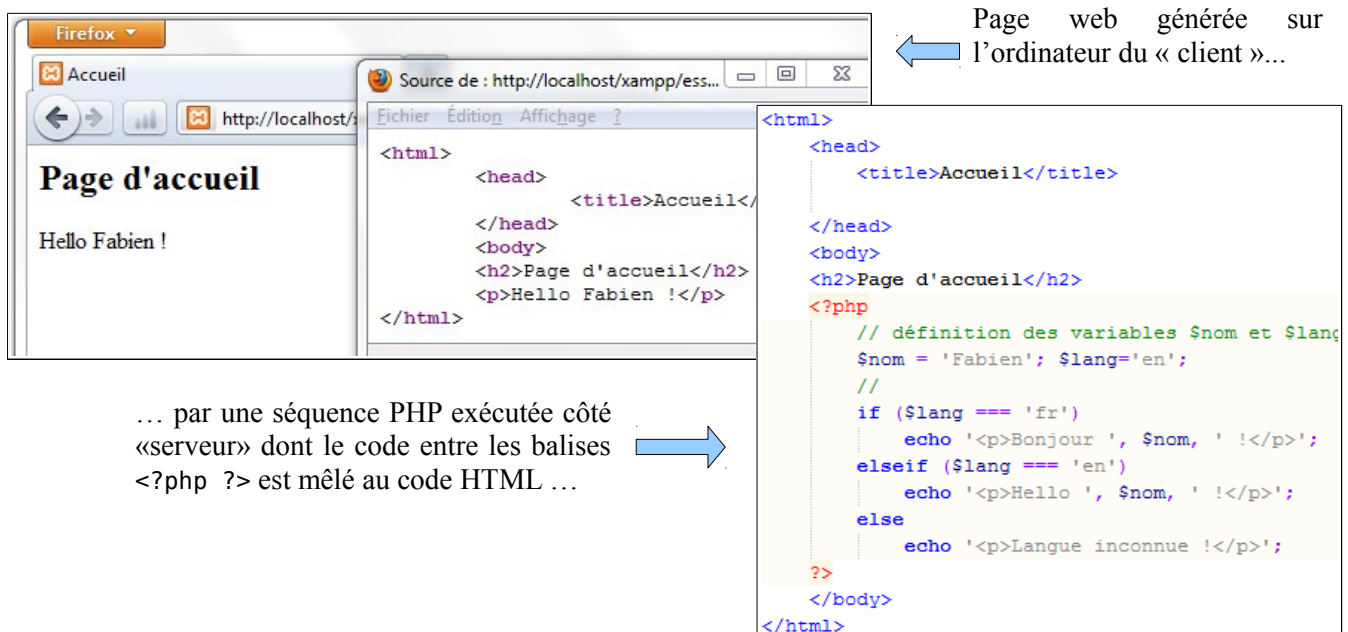
1.3 Principe de la génération dynamique des pages Web

Cette partie vise à décrire les principes et les outils côté serveur Web visant à « générer » des pages codées en HTML par programmation.

On peut évoquer avec les élèves le populaire langage PHP, très majoritairement installé sur les serveurs *Apache*. Usuellement, un script PHP (généralement identifié par l'extension `.php`) est imbriqué dans le code HTML entre des balises spécifiques. Lorsque la page réclamée par le client est un script PHP, le serveur appelle l'interpréteur PHP pour traiter les instructions entre les balises `<?php . . . ?>` afin de générer le code HTML/CSS qui va être retourné au navigateur.

Le langage *Python* dispose également de modules HTTP comme *CherryPy*. Le principe est ici très différent puisque dans la logique de *CherryPy*, un site Web est vu comme une classe (au sens de la programmation objet) dont les méthodes donnent la structure des pages.

À cette étape et quel que soit le langage serveur retenu, le principe de la génération dynamique des pages peut être décrit en classe entière au vidéo-projecteur.



Page web générée sur l'ordinateur du « client »...

... par une séquence PHP exécutée côté «serveur» dont le code entre les balises `<?php ?>` est mêlé au code HTML ...

```
<html>
<head>
  <title>Accueil</title>
</head>
<body>
  <h2>Page d'accueil</h2>
  <p>Hello Fabien !</p>
</html>
```

```
<?php
// définition des variables $nom et $lang
$nom = 'Fabien'; $lang='en';
//
if ($lang === 'fr')
  echo '<p>Bonjour ', $nom, ' !</p>';
elseif ($lang === 'en')
  echo '<p>Hello ', $nom, ' !</p>';
else
  echo '<p>Langue inconnue !</p>';
?>
```

On montre ici une page générée en langage Python associée à la bibliothèque HTTP CherryPy. La méthode « index » retourne une chaîne de caractères composée en Python (concaténation et/ou formatage) comprenant le code HTML de la page.

La méthode « QuickStart » de l'objet « CherryPy » prend en charge la création et la gestion du serveur Web.

Si le langage Python a déjà été utilisé par les élèves au cours de l'année, c'est celui-ci, associé à une bibliothèque HTTP, que nous retiendrons comme langage côté serveur par la suite. Pour les activités de T.P., on veillera dans ce cas à fournir aux élèves des gabarits de programmes préparés en amont afin qu'ils n'aient plus qu'à se concentrer sur les traitements essentiels sans faire appels à des concepts trop poussés autour de la programmation orientée objet que requiert l'utilisation de ces bibliothèques.

```
import cherrypy

class MonSiteWeb(object):
    def index(self):
        nom='Fabien'
        lang='en'
        if lang=='fr':
            msgbonjour='Bonjour ' + nom + ' !!'
        elif lang=='en':
            msgbonjour='Hello ' + nom + ' !!'
        else:
            msgbonjour='Langue inconnue !!'
        return '''<html>
                <head>
                    <title>Accueil</title>
                </head>
                <body>
                    <h2>Page d'accueil</h2>
                    <p> %s </p>
                </body>
            </html>''' %msgbonjour

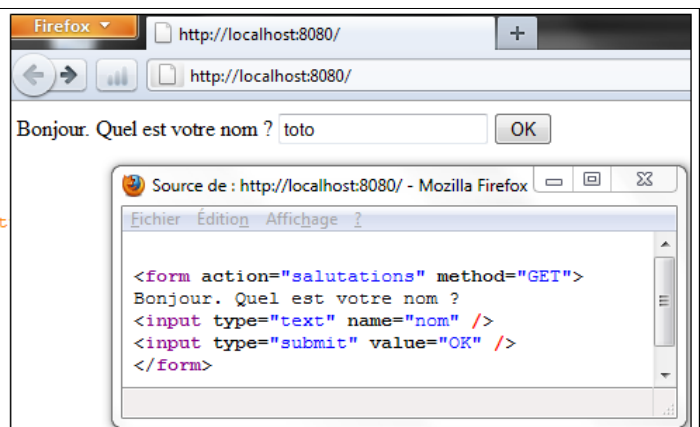
index.exposed=True

##### Programme principal : #####
cherrypy.quickstart(MonSiteWeb(), config ="D:\\ma
```

1.4 Formulaires web

Les données saisies par l'utilisateur seront encodées dans les formulaires rédigés en HTML entre les balises <form> ... </form> et comprenant des zones de texte, des listes déroulantes, des cases à cocher, etc. On met en évidence le fait que ce formulaire permet d'envoyer au serveur Web une requête accompagnée des informations saisies (méthodes GET ou POST). Il faudra alors exposer aux

```
1 import cherrypy
2
3 class MonSiteWeb(object):
4     # page principale
5     def index(self):
6         return '''
7     <html>
8         <head> <title>Connexion</title>
9         <link rel="stylesheet" type="text/css" href="st
10        </head>
11
12        <body>
13            <form action="salutations" method="GET">
14                <p>Bonjour. Quel est votre nom ? </p>
15                <input type="text" name="nom" />
16                <input type="submit" value="OK" />
17            </form>
18        </body>
19    </html>
20
21        ...
22
23        index.exposed=True
24
25    # page de bienvenue
26    def salutations(self, nom=None):
27        return '''<h2>bienvenue %s </h2>''' %nom
28        salutations.exposed=True
29
30    ##### Programme principal : #####
31    cherrypy.quickstart(MonSiteWeb(), config ="D:\\pr
```



élèves comment on peut récupérer les arguments passés par la méthode GET, dans le langage serveur pour y effectuer les traitements nécessaires.

L'URL invoquée par la méthode Get (HTTP) comprend les arguments saisis par l'utilisateur :

http://localhost:8080/salutations?nom=toto

Dans la logique de CherryPy, celle-ci est convertie en un appel de méthode.

1.5 Maintien de données à l'aide de cookies

On constate que les échanges HTTP ne permettent pas jusqu'à présent de mémoriser un « état » et que l'on peut revenir indéfiniment sur la même page sans que le serveur en garde « le souvenir ». Les cookies ont été inventés

justement pour donner une « mémoire » de l'état des échanges serveur-navigateur (client).

Après avoir défini les principes généraux de fonctionnement des cookies et des mécanismes d'échanges client-serveur associés, ainsi que les cas classiques d'utilisation (gestion des sessions, personnalisation et même pistage), on peut explorer les cookies stockés dans le navigateur.

En installant l'extension **Firebug** dans le navigateur Firefox (<https://addons.mozilla.org/fr/firefox/addon/firebug>), on parvient à mettre en évidence la présence et le rôle des cookies, extension à laquelle on peut encore rajouter le complément Firecookie (<https://addons.mozilla.org/fr/firefox/addon/firecookie>) pour aller plus loin dans la gestion des cookies.

Deux autres extensions sont particulièrement spectaculaires de ce point de vue (on peut donc répartir les élèves en deux ou plusieurs groupes et les inviter à expérimenter un peu) :

Collusion (<https://addons.mozilla.org/fr/firefox/addon/collusion>) permet d'afficher un graphique suggérant indiquant les liens entre les divers serveurs déposant des cookies.

Ghostery (<http://www.ghostery.com>) permet d'autoriser sélectivement le dépôt des cookies et de repérer ceux qui ne servent qu'à pister les comportements de internautes.

Dans un second temps, les activités de programmation en TP peuvent conduire les élèves à répondre au questionnement suivant :

- quelles sont les données à maintenir ? (information de connexion, contenus de paniers d'achats électroniques, préférences de navigation,...)
- comment utiliser ces données pour gérer l'interactivité du site ? (également l'ergonomie en termes de navigation)
- quel est le degré de persistance des données à maintenir ? (maintien en mémoire vive le temps de la visite du site, sauvegarde sur le disque de l'ordinateur « client » le temps de la session ou pour une durée déterminée)
- que se passe-t-il si le visiteur interdit la présence de cookies sur son poste ?

Expérience : le message d'accueil est personnalisé selon le pseudo et la langue saisis dans un formulaire de connexion. Les informations de connexion doivent être maintenues grâce aux cookies tant que l'utilisateur ne s'est pas déconnecté du site.

The image shows two screenshots of a Firefox browser. The left screenshot shows a login form titled 'Connexion' with fields for 'Entrez votre pseudo : Fabien' and 'Votre nationalité : Allemand'. A red arrow points from this form to the right screenshot. The right screenshot shows the browser displaying a personalized message 'Guten Tag Fabien !!' and the Firebug cookie table. The table lists two cookies: 'nationalite' and 'pseudo'.

Nom	Contenu	Hôte	Taille	Chemin	Expire
nationalite	ALL	localhost	14 B	/	dimanche 10 ju
pseudo	Fabien	localhost	12 B	/	dimanche 10 ju

1.6 Comment ne pas aller trop loin ?

Les activités autour du cookie sont également une occasion de soulever les problèmes causés par les données personnelles laissées sur le Net.

Dans un modèle économique largement financé par la publicité, le cookie stocké par le navigateur Web de l'utilisateur est souvent chargé de jouer les « espions ». L'internaute se retrouve alors « ciblé » à son insu dans ses goûts et ses habitudes de navigation à des fins marketing (voir l'[expérience du CNIL](#)).

Dans ce flux de données sensibles et difficiles à maîtriser, il est bien dans cette séquence d'évoquer avec les élèves les quelques gardes-fous permettant de créer une première forme de protection de la vie privée (à commencer par les options des assistants de confidentialité proposés dans les navigateurs).

Sur le plan législatif, on peut montrer les efforts menés sur les notions de droit à l'oubli (voir la récente charte du droit à l'oubli numérique¹). Malgré les chartes de déontologie (qui n'offrent pas la même valeur juridique que les lois) et face aux progrès des nouvelles technologies, souvent en avance sur les lois et habilement exploitées par les sociétés du Web (par exemple la possibilité de ciblage publicitaire géo-adapté grâce aux récents services de géolocalisation), la première garantie de protection de la vie privée reste l'éducation et la prévention en adoptant des comportements simples comme :

- la lecture des règles de confidentialité des sociétés du Web (ou « privacy policy ») et leurs maîtrises (exercice pouvant être réalisé en classe ou comme travail à la maison) et ainsi aller en faveur de la transparence des sites;
- l'usage de la navigation anonyme, le contrôle des cookies (il est difficile de les interdire complètement étant donné que la plupart des services proposés sur Internet requiert leur usage).

Le respect des lois actuelles (adaptées à l'ère du numérique) sur la diffamation et le respect de la vie privée n'est pas une chose facile compte tenu du volume des échanges qui se déroulent quotidiennement sur le Net. Il s'agit de faire comprendre qu'en l'état actuel de la législation, sans prise de conscience de l'utilisateur, celui-ci donne par défaut son accord tacite à une éventuelle utilisation commerciale des cookies installés sur son poste par son navigateur.

1.7 Activités de TP

En TP, on peut fournir une application Web simple et fonctionnelle aux pages générées dynamiquement (en langage Python par exemple). L'élève analyse la structure du programme et l'enchaînement des pages du site en testant l'application en local dans le navigateur.

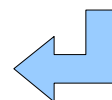
On lui propose une ou plusieurs évolutions parmi :

- modification du traitement sur une donnée saisie dans un formulaire ;
- donnée supplémentaire à encoder dans un formulaire et programmation du traitement associé.
- Implanter un cookie pour apporter une nouvelle fonctionnalité ou une amélioration (compteur de session, maintien en mémoire d'un identifiant, d'une préférence personnelle de navigation), prévoir sa construction avec ses attributs, sa destruction. Programmer le traitement associé.

Voir des exemples d'activités en annexe.

Nom	Contenu	Hôte	Taille	Chemin	Expire
moncookie	"Marseille, Paris, Nantes"	localhost	35 B	/	Session

Dans cet exercice, on simule le comportement d'un moteur de recherche. Le serveur crée un cookie mémorisant les derniers critères renseignés par l'internaute.
À quelles fins pourrait-on exploiter ce cookie ?



2 / À propos du « droit à l'oubli numérique »

2.1 Problématique

L'autre question à soulever concerne la durée de conservation des données collectées. Grâce aux capacités d'indexation des moteurs de bases de données et des possibilités quasi-illimitées de stockage de celles-ci, les données personnelles renseignées aussi de façon active par l'internaute (comme dans le cas emblématique des messages postés dans les forums des réseaux sociaux) risquent de rester indéfiniment sur la toile.

¹ Lire ici : http://fr.wikipedia.org/wiki/Chartes_du_droit_%C3%A0_l%27oubli_num%C3%A9rique

On rappelle que si les cookies peuvent être supprimés physiquement du poste de l'utilisateur, les données inscrites dans les bases de données sont stockées sur un serveur tiers et peuvent échapper au contrôle de l'utilisateur.

Ce sont bien de ces données personnelles collectées par le biais de cookies ou dans les gigantesques bases de données renseignées par l'internaute lui-même et du risque de la perte de la maîtrise de ces données qu'est née la revendication du droit à l'oubli numérique.

On propose de prolonger le travail de cette séquence en faisant réfléchir sur les conditions et les limites philosophiques concernant la mémorisation des données issues des échanges sur la toile.

Dans cette optique, on pourra proposer un travail alternant des (petits) passages théoriques, quelques lectures et recherches personnelles, des échanges par petits groupes avec un « porte-parole » et une synthèse globale assurée par un enseignant quelle que soit sa discipline de spécialisation.

Un enseignant de Philosophie pourra accompagner la réflexion des élèves en leur proposant des textes comme ceux extraits de Bergson sur la mémoire (extrait de son *Essai sur les données immédiates de la conscience*) et de Foucault sur le pouvoir de normalisation par la mémoire. Le but de ces lectures est de montrer que la mémoire est à la fois psychologiquement et moralement nécessaire (individuellement et collectivement), mais dangereuse pour les libertés individuelles si on l'utilise comme un moyen d'emprisonner les individus dans leur passé.

Le travail de groupe, à partir de ces textes et d'autres sources documentaires (CDI, sites Internet, ou pourquoi pas à partir de faits divers) mettant en évidence la nécessité d'une sorte de « droit à l'oubli », doit permettre aux élèves d'élaborer une sorte de fiche-mémoire pouvant s'intituler « Éthique et droit à l'oubli dans les espaces numériques ».

2.2 Pistes de travail possibles

Partir d'un constat : l'inachèvement juridique du « droit à l'oubli » et les nombreux problèmes que cela pose. Les spécialistes comme les usagers du numérique ne peuvent échapper à cette exigence d'interrogation au sujet des implications éthiques de leur pratique. En somme ...

La mémoire est une fonction nécessaire, tout comme l'oubli.

Rappel (rapide) de la nécessité psychologique de la mémoire

La « mémoire » n'est pas seulement un meuble dans lequel nous rangeons des souvenirs et nous puisons comme dans des tiroirs. Elle est une fonction constitutive de notre conscience, comme accumulation et anticipation permanente. Sans mémoire, pas de conscience de soi, donc pas d'identité. C'est pourquoi nous tenons tant (trop?) à afficher ce que nous avons fait de nous-mêmes (cf. Facebook, MySpace etc.). Cette mémoire affichée n'est que l'illustration de l'importance de cette fonction. Sans la présence à soi de son propre passé, l'être humain ressent comme une perte de maîtrise de lui-même. Notons que cette fonction est de fait inséparable de l'acquisition du langage. C'est pourquoi « conscience de soi », « langage » et « mémoire » sont liés. On ne parvient jamais à « remonter » plus loin que l'âge de trois ans dans son passé, âge à partir duquel la mémoire s'appuie sur du langage pour associer l'identité aux activités : on commence à dire « Je ».

Illustration morale : le « courage » est fait du souvenir et d'une image de soi, d'une identification de « soi présent » à cette image qui donne confiance en soi.

La mémoire est moralement et juridiquement nécessaire pour constituer une responsabilité

On ne saurait rendre pareillement responsable un individu qui aurait effectivement « oublié » la totalité de ses actes. Non pas, évidemment, qu'il n'aurait pas malgré tout à en « rendre compte », voire à réparer, à être puni, mais comme tel, « l'oubli » (réel) modifie le discours que l'on peut lui tenir. L'accusation, par exemple, devient dans ce cas un rappel des faits, et le jugement un rappel de l'exigence à assumer devant les autres les conséquences de ses actes. Être l'auteur réel implique une responsabilité de fait. Pour que cette responsabilité ait un sens, il faut que la mémoire, la peine encourue, la punition éventuelle et la prise de conscience des torts causés mette en jeu une mémoire.

On peut, à partir de là, réfléchir ensemble sur le rapport de la responsabilité au temps.

Exemple : un condamné a bien effectué sa peine, son « existence numérique » ne cesse pas pour autant. Comment le problème se pose-t-il et comment y remédier ? Voir, à ce sujet, l'article suivant :

https://wiki.inria.fr/sciencinfolycee/Droit_%C3%A0_l%27oubli_%28CNIL%29

Apparition du droit à l'oubli en lien avec la question de la mémoire. Les risques d'enfermer un individu dans des images passées.

Il n'y a pas de liberté sans le pouvoir légitime sur soi que constitue le « droit à l'oubli ». Si celui-ci paraît difficilement applicable, il n'est pas interdit de tenter par tous les moyens de faire prendre conscience aux futurs spécialistes que la technologie numérique est un pouvoir, non seulement un savoir-faire neutre, mais un ensemble de techniques pouvant durablement impacter la vie des individus.

À titre d'illustration, on peut se référer aux nombreux cas d'images circulant sur le web tout en échappant au contrôle des personnes concernées, avec des conséquences parfois dramatiques.

3 / Références

3.1 Cookies

- « All about cookies » : <http://www.allaboutcookies.org/>
- Cookie(informatique) : http://fr.wikipedia.org/wiki/Cookie_%28informatique%29
- Vos traces sur internet : ça n'est pas virtuel ! : <http://www.cnil.fr/vos-libertes/vos-traces>
- La norme technique sur les cookies : <http://www.ietf.org/rfc/rfc2109.txt>

3.2 Droit à la vie privée, droit à l'oubli

- Proposition de loi visant à mieux garantir le droit à la vie privée à l'heure du numérique (article 6) : <http://www.senat.fr/leg/ppl09-093.html>
- Directive 2002/58/CE du Parlement européen et du Conseil de l'UE du 12 juillet 2002 concernant le traitement des données à caractère personnel et la protection de la vie privée dans le secteur des communications électroniques (directive vie privée et communications électroniques) :
http://europa.eu/legislation_summaries/information_society/legislative_framework/l24120_fr.htm
voir notamment le point (25) de la directive, qui concerne les *cookies*.
- Fiches pratiques de la CNIL :
[http://www.cnil.fr/recherche/?tx_indexedsearch\[ext\]=1&tx_indexedsearch\[sword\]=publicit%C3%A9+cibl%C3%A9&x=18&y=9](http://www.cnil.fr/recherche/?tx_indexedsearch[ext]=1&tx_indexedsearch[sword]=publicit%C3%A9+cibl%C3%A9&x=18&y=9)
- Publicité ciblée sur Internet : vers un consentement préalable de l'internaute (article de la CNIL) :
<http://www.cnil.fr/la-cnil/actu-cnil/article/article/publicite-ciblee-sur-internet-vers-un-consentement-prealable-de-linternaute/>
- Charte du droit à l'oubli numérique signée du 13 Octobre 2010 sous l'égide de M^{me} Nathalie Kosciusko-Morizet (secrétaire d'Etat chargée de la Prospective et du Développement de l'économie numérique):
http://fr.wikisource.org/wiki/Charte_du_droit_%C3%A0_l%27oubli_dans_les_sites_collaboratifs_et_les_moteurs_de_recherche
- Quelques liens sur le « droit à l'oubli » (SILO) :
https://wiki.inria.fr/sciencinfolycee/Droit_%C3%A0_l%27oubli
- La Commission propose une réforme globale des règles en matière de protection des données pour accroître la maîtrise que les utilisateurs ont sur leurs données, et réduire les coûts grevant les entreprises :
<http://europa.eu/rapid/pressReleasesAction.do?reference=IP/12/46&format=HTML&aged=0&language=FR&guiLanguage=en>

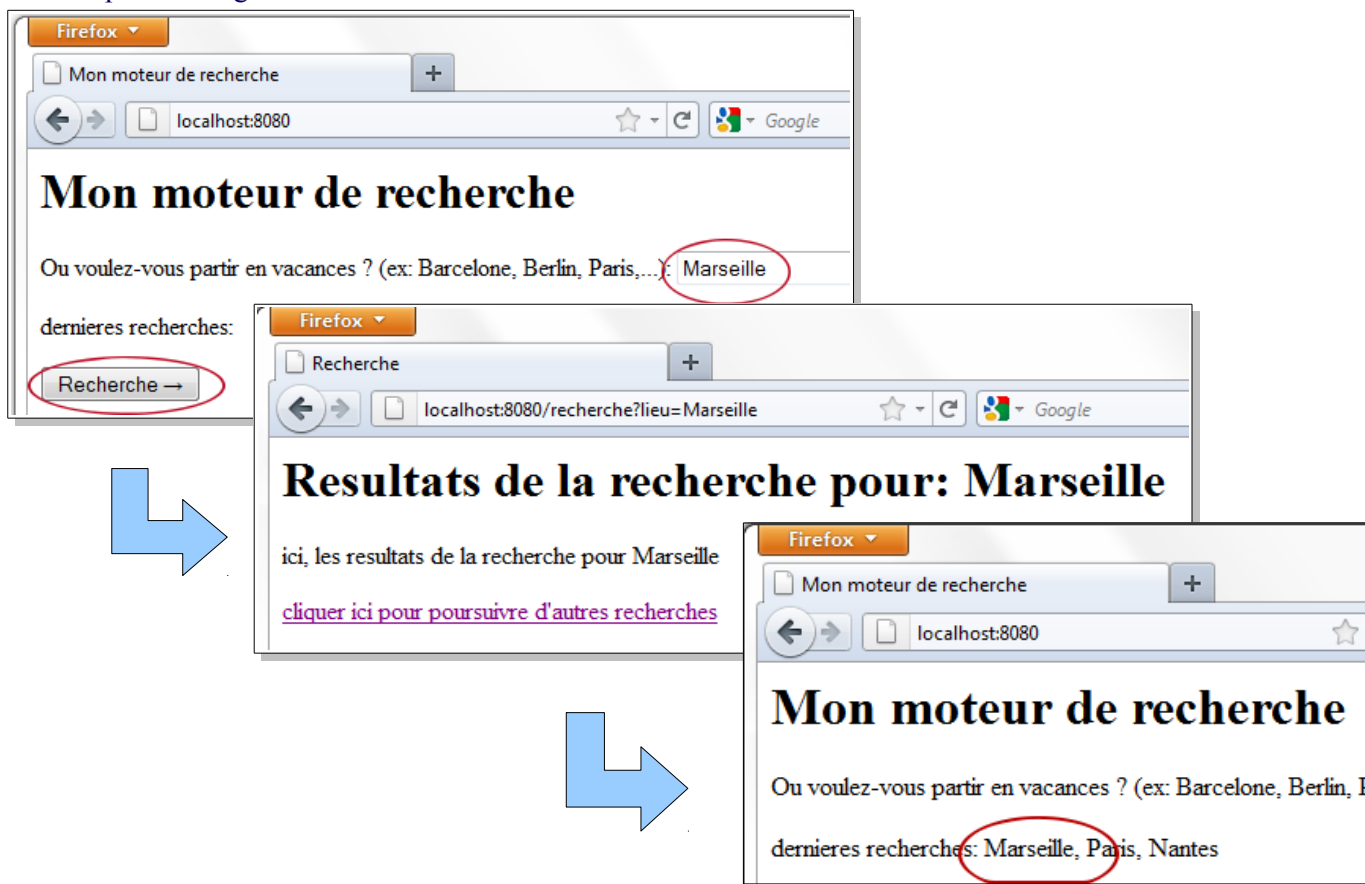
Annexe: Autour des pages « Web » dynamiques

1 / But de cette activité

On essaye de découvrir dans cet exercice comment les moteurs de recherche utilisent et conservent nos recherches afin de mieux nous cibler par la suite (voir la démonstration du CNIL <http://www.cnil.fr/vos-libertes/vos-traces/experience>).

2 / Principe

Le contenu dynamique des pages est géré en langage *Python* associé à la bibliothèque HTTP *CherryPy*. Le maintien des données est obtenu par un *cookie*. Le comportement de notre prototype de moteur de recherche est illustré par les images ci-dessous.



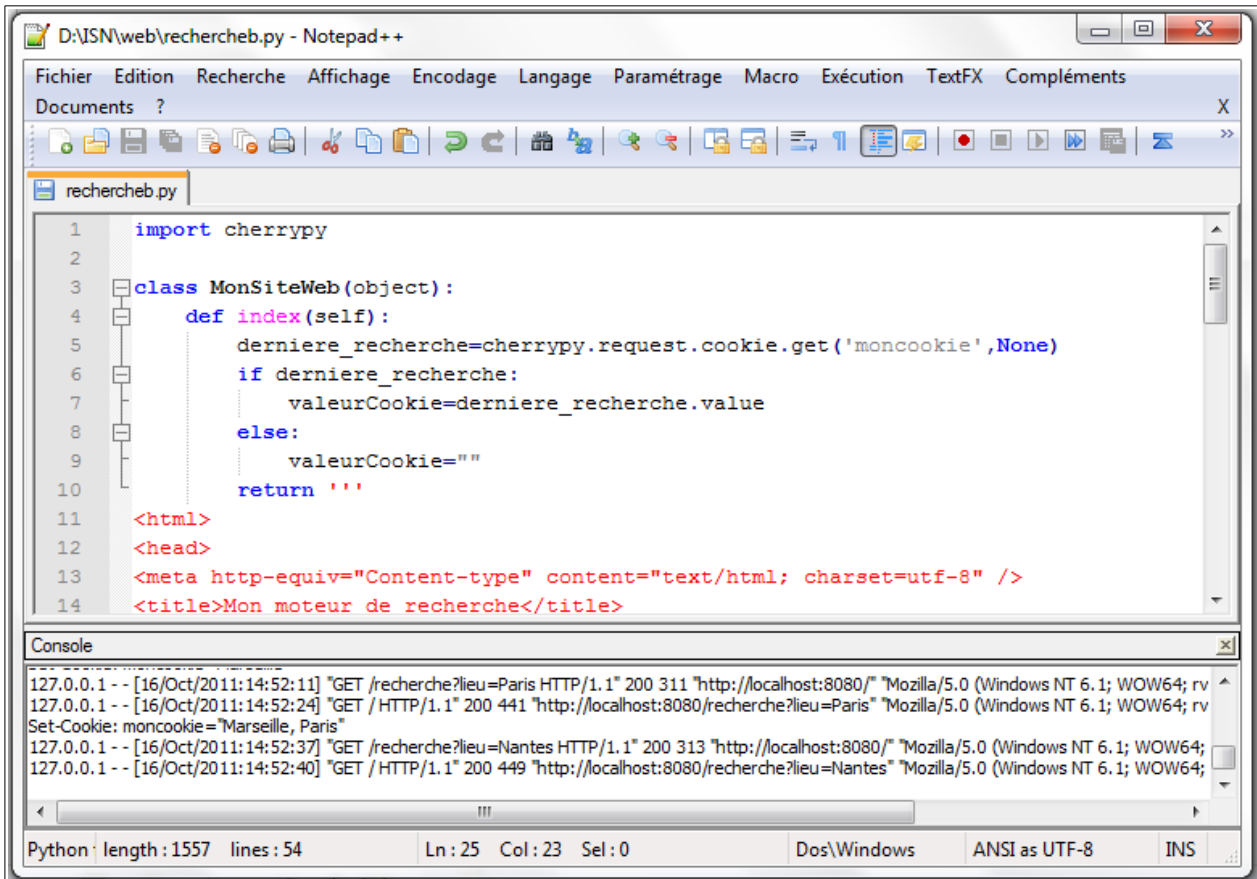
À chaque retour sur la page principale, les recherches précédentes sont mémorisées et affichées le temps de la session.

3 / Outils

Le langage de programmation utilisé est *Python* (v2.6) associé au framework HTTP *CherryPy* (<http://www.cherrypy.org/>). L'environnement de programmation comportera un éditeur avec coloration syntaxique comme *NotePad++* (<http://notepad-plus-plus.org>) avec en complément la console *NppExec*. Pendant le développement, le site sera testé sur un serveur Web local (localhost).

On pourra consulter le livre de Gerard Swinnen « Apprendre à programmer avec Python » en téléchargement libre: <http://www.inforef.be/swi/python.htm> (chapitre 17 – Applications Web)

ainsi que l'article « How to use cookies with CherryPy » : <http://www.cherrypy.org/wiki/Cookies>



La console permet de lancer l'exécution du programme et de rendre compte des échanges HTTP.



Le navigateur Internet pourra comporter un module utilisé par les développeurs Web pour visualiser le code HTML généré ou l'état des cookies (par exemple les extensions *Firebug*/*Firecookie* pour *Firefox*).

L'emploi d'un deuxième navigateur (qui peut être *Opera*, *Chromium* ou tout autre) est recommandé pour simuler différentes sessions de navigation.

4 / Description de l'activité

Après avoir expliqué le but du prototype aux élèves, on peut leur proposer le gabarit de programme suivant :

```

import cherrypy

class MonSiteWeb(object):
    def index(self):
        return '''
<html>
<head>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<title>Mon moteur de recherche</title>
</head>
<body>
<h1>Mon moteur de recherche</h1>
<form action="recherche" method="GET">
<p>Ou voulez-vous partir en vacances ? (ex: Barcelone, Berlin, Paris,...):
<input type="text" name="lieu" /></p>
<p>dernieres recherches: ...</p>
<p><input type="submit" value="Recherche &rarr;" /></p>
</form> </body>'''

    index.exposed=True

    def recherche(self, lieu):
        return '''
<html>
<head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <title>Recherche</title>
</head>
<body>
    <h1>Resultats de la recherche pour: ...</h1>
    <p>...</p>
    <p><a href="/">cliquer ici pour poursuivre d'autres recherches</a>
    </p>
</body>
'''

    recherche.exposed=True

if __name__ == '__main__':
    cherrypy.quickstart(MonSiteWeb())

```

Au démarrage du programme, la méthode *quickstart* prend en charge la gestion du serveur Web (serveur local à l'adresse *localhost:8080* par défaut).

```

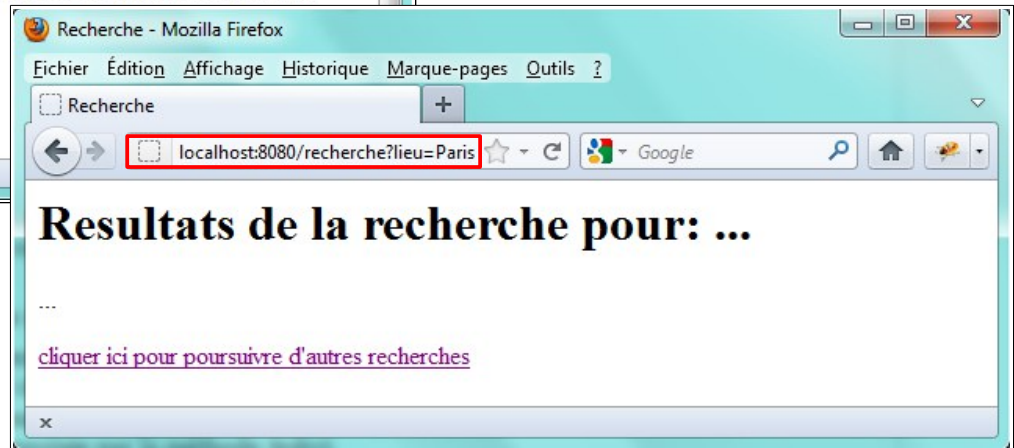
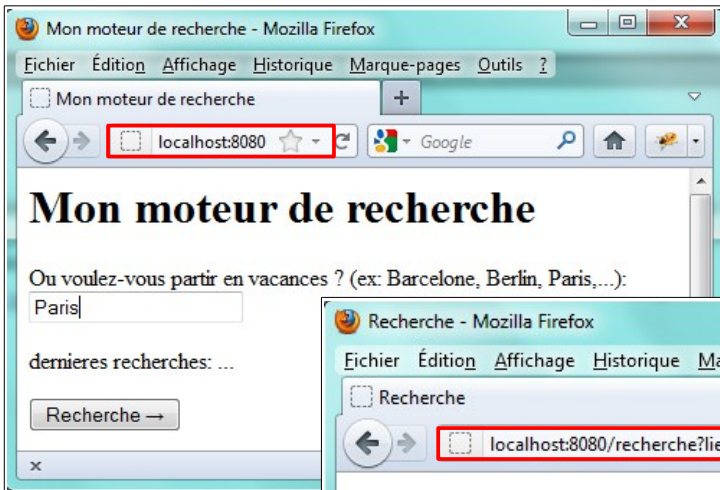
Process started >>>
[28/Apr/2012:21:41:23] ENGINE Listening for SIGTERM.
[28/Apr/2012:21:41:23] ENGINE Bus STARTING
CherryPy Checker:
The Application mounted at " has an empty config.

[28/Apr/2012:21:41:23] ENGINE Started monitor thread '_TimeoutMonitor'.
[28/Apr/2012:21:41:23] ENGINE Started monitor thread 'Autoreloader'.
[28/Apr/2012:21:41:23] ENGINE Serving on 127.0.0.1:8080
[28/Apr/2012:21:41:23] ENGINE Bus STARTED

```

L'exécution du programme dans un navigateur en local permet déjà de découvrir la structure du site et l'enchaînement des deux seules pages qui le compose. Il découvre entre-autres qu'après soumission du formulaire, la requête HTTP comporte la ville saisie dans la zone de texte comme argument :

<http://localhost:8080/recherche?lieu=Paris>



L'analyse du code HTML généré dans le navigateur (clic droit puis « code source de la page » avec *Firefox*) et du code *Python* permet de comprendre le principe de la génération de code HTML par programmation avec la bibliothèque *CherryPy*.

L'élève comprend que les deux méthodes *index* et *recherche* structurent les deux pages web.

On fait constater comment *CherryPy* transforme automatiquement les paramètres *action* et *name* de la méthode HTTP GET en appel de méthode classique après soumission du formulaire (code surligné en **jaune**). De même le paramètre *href* (surligné en **bleu**) permet automatiquement la redirection vers la racine du site (la racine étant la page retournée par la méthode *index*).

La suite du développement du prototype peut commencer, en particulier la mise en œuvre du cookie pour le maintien des données sur les différentes recherches de villes saisies.

Lors de l'affichage du formulaire, il faut tester l'existence de recherches précédentes en récupérant le contenu du cookie si celui-ci existe (voir le lexique en fin de document). Puis on modifie dynamiquement la chaîne HTML de retour afin d'afficher les dernières recherches par concaténation ou formatage (attention aux nouvelles syntaxes de formatage des chaînes de caractères depuis la version 3 de Python).

```
def index(self):
    derniere_recherche=cherrypy.request.cookie.get('moncookie',None)
    if derniere_recherche:
        valeurCookie=derniere_recherche.value
    else:
        valeurCookie=""
    return '''
```

```
<html><head>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<title>Mon moteur de recherche</title>
</head>
<body>
<h1>Mon moteur de recherche</h1>
<form action="recherche" method="GET">
<p>Ou voulez-vous partir en vacances ? (ex: Barcelone, Berlin, Paris,...):
<input type="text" name="lieu" /></p>
<p>dernieres recherches: %s</p>
<p><input type="submit" value="Recherche &rarr;" /></p>
```

```
</form> </body></html>''' %(valeurCookie)
```

Dans la page donnant le résultat de la recherche (méthode *recherche*), il faut aussi recomposer dynamiquement la chaîne retournée. S'il n'y a pas de recherches précédentes, il faut créer le cookie. Si le cookie existe déjà, il faut modifier son contenu en concaténant la dernière recherche saisie suivant le principe à tester suivant :

```
>>> derniere_recherche="Nantes"
>>> print ", ".join([derniere_recherche, "Bordeaux"])
Nantes, Bordeaux
>>>
```

Ce qui donne par exemple :

```
def recherche(self, lieu):
    resultats=" ".join(["ici, les resultats de la recherche pour", lieu])
    derniere_recherche=cherry.py.request.cookie.get('moncookie', None)
    if derniere_recherche:
        valeurCookie=", ".join([derniere_recherche.value, lieu])
    else:
        valeurCookie=lieu
    cherry.py.response.cookie['moncookie']=valeurCookie
    return'''
```

```
<html><head>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<title>Recherche</title>
</head>
<body>
    <h1>Résultats de la recherche pour: %s</h1>
    <p>%s</p>
    <p><a href="/">cliquer ici pour poursuivre d'autres recherches</a>
    </p>
</body></html>''' % (lieu, resultats)
```



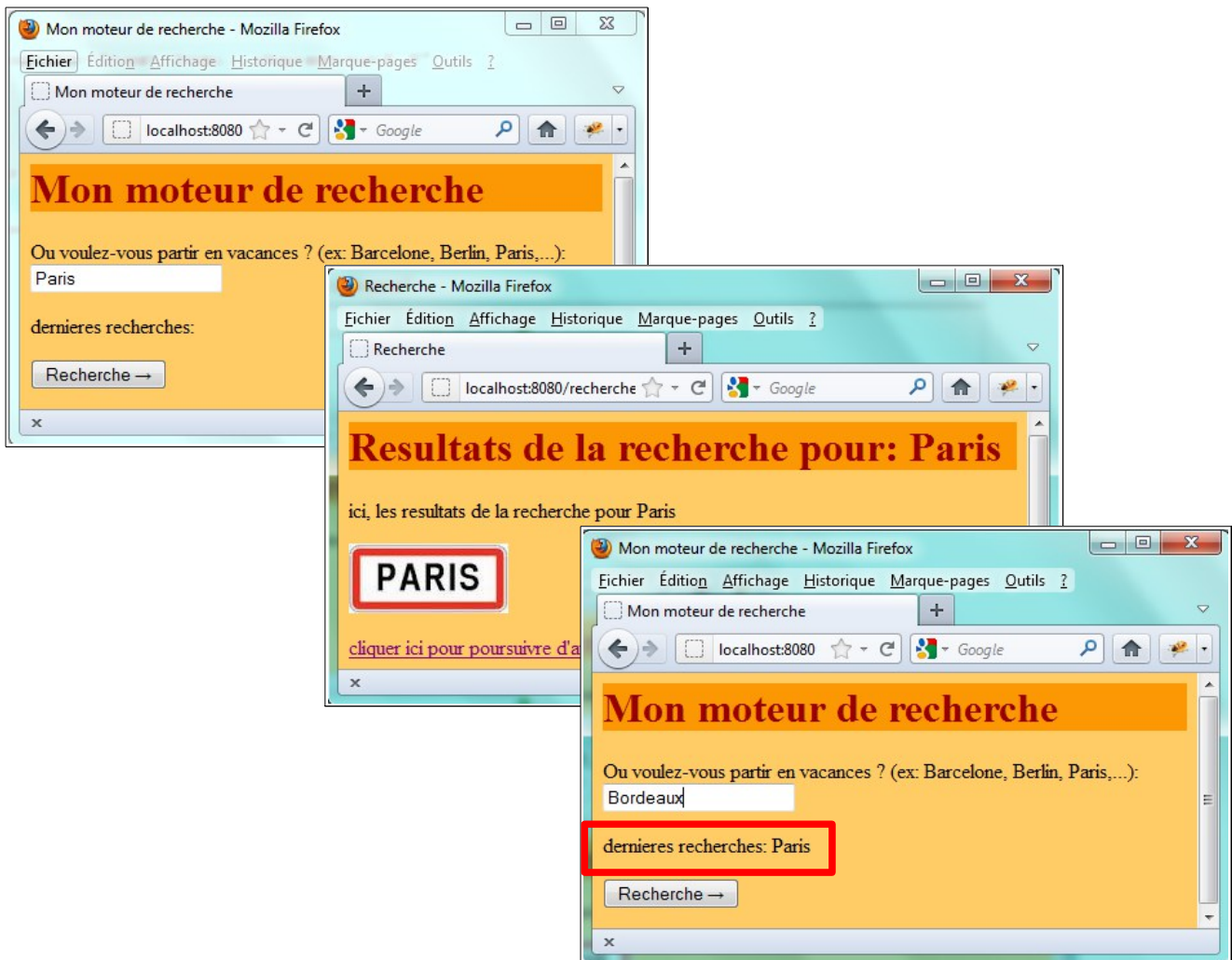
Le contenu de la page restituée peut comporter des variantes selon l'avancement et la motivation de l'élève ou du groupe et le temps disponible (séquence courte ou mini-projet).

On peut se contenter d'une simple restitution de la ville saisie dans un paragraphe de texte HTML (minimaliste mais suffisant pour comprendre le principe) mais aussi recomposer une page plus complexe avec une image de la ville recherchée par exemple, page mise en forme grâce à une feuille de style CSS séparée. Dans ce cas il peut-être nécessaire de préciser certains paramètres pour le stockage des images ou du fichier CSS dans un fichier de configuration appelé par la méthode *quickstart* :

```
cherry.py.quickstart(MonSiteWeb(), config='D:\\maconfig.conf')
```

Exemple de contenu du fichier de configuration *maconfig.conf*, (voir les options <http://www.cherrypy.org>):

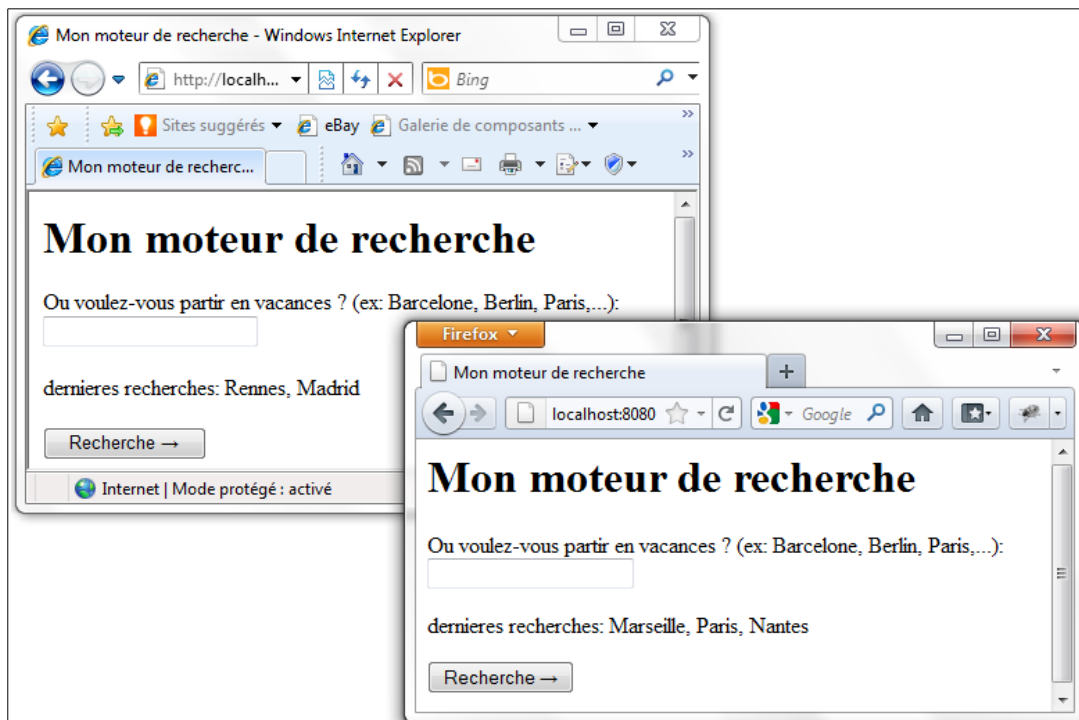
```
[/]  
tools.staticdir.root = "D:"  
  
[/style.css]  
tools.staticfile.on= True  
tools.staticfile.filename="D:\\style.css"  
  
[/images]  
tools.staticdir.on= True  
tools.staticdir.dir="D:\\images"
```



Sur chaque page retournée, on rajoute le lien vers la feuille de style CSS dans l'entête :

```
<head>  
...  
<link rel="stylesheet" type="text/css" href="style.css" />  
...  
</head>
```

Enfin, afin de simuler différentes sessions de navigation dans un fonctionnement local, on peut ouvrir et tester le site dans plusieurs navigateurs internet :



On peut enfin se poser la question de l'accès aux *cookies* par d'autres serveurs que le serveur initial ; c'est évidemment une question de sécurité mais la chose est envisageable, lire ici : http://www.codeguru.com/print/csharp/csharp/cs_internet/article.php/c19417/Sharing-Cookies-Across-Domains.htm

5 / Lexique *CherryPy* pour la gestion des cookies

- Pour créer un cookie avec *CherryPy* :

```

MonCookie=cherry.py.response.cookie           # Déclaration d'un cookie
MonCookie['NomDuCookie']=ContenuDuCookie      # Contenu du cookie
# Les paramètres qui suivent sont facultatifs, par défaut la durée de vie du cookie est
# le temps de la session
MonCookie['NomDuCookie']['path']='/'           # Chemin
MonCookie['NomDuCookie']['max-age']=30         # Durée de vie en secondes

```

- Pour tester l'existence d'un cookie:

```

MonCookie=cherry.py.request.cookie.get('NomDuCookie',None)
if MonCookie:                                # le cookie existe
    print MonCookie.value                     # afficher le contenu du cookie dans la console
else:
    print 'Cookie supprimé'                  # le cookie n'existe pas/plus

```

- Pour supprimer tous les cookies:

```

reqcookie=cherry.py.request.cookie
rescookie=cherry.py.response.cookie
for name in reqcookie.keys():
    rescookie[name]=name
    rescookie[name]['path']='/'
    rescookie[name]['max-age']=0

```
