

## Fiche

Le Web est une application qui utilise le réseau Internet. C'est un système hypertexte géant et public, un immense ensemble de ressources (page web, vidéo, son, image...) dont certaines sont reliées entre elles par des liens cliquables (hyperliens ou liens hypertexte) et stockées sur des millions de serveurs répartis sur toute la planète. Certains serveurs web sont capables de créer une page web. Un site web est un ensemble composé de plusieurs pages web.

### I. Définitions et repères historiques

- Le *World Wide Web* signifie « la toile d'araignée à l'échelle mondiale ». On l'appelle de manière raccourcie le Web. Pour « naviguer » sur le Web grâce à un appareil (ordinateur, smartphone...), on utilise très souvent un **navigateur**. C'est un logiciel qui est chargé de communiquer avec les serveurs du Web afin de pouvoir présenter les ressources à l'utilisateur : afficher une page web, jouer un son...
- On peut assimiler le Web à un graphe orienté dont les sommets représentent les ressources et les arcs représentent les hyperliens sortants.
- L'organisme W3C (*World Wide Web Consortium*) a été créé en 1994 afin de standardiser les formats.
- Quelques dates :  
1965 : création du terme hypertexte.  
**1989 : naissance du Web, lancé par le britannique Tim Berners Lee au CERN près de Genève.**  
1993 : apparition du premier navigateur grand public, Mosaic.  
1995 : début du web interactif et du web dynamique avec les langages Javascript et PHP.

### II. Notions juridiques

#### Droit d'usage

- Il permet de réutiliser des ressources sans autorisation. **Exemples** : citation, revue de presse, caricature, parodie...

#### Droit d'auteur

- Il englobe un droit moral (confirme la paternité de l'auteur sur son œuvre) et les droits patrimoniaux (monopole de l'exploitation économique). Le droit d'auteur s'applique sur le Web.
- Pour toute exploitation (diffusion, modification) d'un contenu il faut l'accord de l'auteur.
- Un auteur peut céder tout ou une partie de ses droits patrimoniaux par une **licence**.

#### Licence

- Il existe des licences libres et des licences propriétaires.
- La licence Creative Commons est une licence libre célèbre qui précise si un contenu est modifiable, à usage commercial ou non et si des contenus dérivés doivent être placés sous la même licence (CC-0, CC-BY...).
- On peut par exemple lire dans les conditions d'utilisation de Facebook : « Lorsque vous partagez, publiez ou téléchargez du contenu couvert par des droits de propriété intellectuelle (comme des photos ou des vidéos) sur ou en rapport avec nos produits, **vous nous accordez une licence non exclusive, transférable, sous-licenciable, gratuite et mondiale pour héberger, utiliser, distribuer, modifier, exécuter, copier, réaliser publiquement ou afficher publiquement, traduire et créer des œuvres dérivées de votre contenu**. Vous pouvez mettre fin à cette licence à tout moment en supprimant votre contenu ou votre compte. Vous devez savoir que, pour des raisons techniques, le contenu que vous supprimez peut être conservé pendant une durée limitée dans des copies de sauvegarde (même s'il ne sera pas visible pour les autres utilisateurs). De plus, le contenu que vous supprimez peut continuer d'apparaître si vous l'avez partagé avec d'autres personnes qui ne l'ont pas supprimé. »

### III. Hypertexte

- Un **hyperlien** (ou lien hypertexte) permet d'associer à une chaîne de caractères (ou à une image) l'adresse d'une ressource à charger quand l'utilisateur clique sur le lien.
- Par défaut, au survol d'un hyperlien, le curseur se transforme en une petite main. Une fois que l'on clique dessus une nouvelle ressource se charge (dans le cas d'une ancre, on se déplace sur la même ressource).

- Un **hypertexte** est un réseau constitué par un ensemble de ressources liées entre elles grâce à des hyperliens. Ce système est utilisé par le Web et permet ainsi de naviguer d'une ressource à une autre en cliquant sur des hyperliens.

## IV. HTML et CSS

- Durant l'élaboration d'une page web on sépare le contenu (fond) du style graphique (mise en forme). On utilise deux langages standardisés par le W3C : le HTML et le CSS.

### HTML

- Créé en 1990, le **HTML** (*HyperText Markup Language*) peut se traduire par « langage de balisage pour hypertexte ». C'est un langage de description qui permet de structurer un texte à l'aide de **balises**.
- Voici un exemple du code source d'une page web inventée.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>ASP</title>
  </head>
  <body>
    <h1>Les organismes internationaux</h1>
    Voici deux organismes :
    <ul>
      <li><a href="www.un.org">ONU</a></li>
      <li><a href="www.iaea.org">AIEA</a></li>
    </ul>
    <p>Un paragraphe est souvent utile pour créer ce qu'on appelle un bloc, c'est à dire un ensemble, que l'on va pouvoir mettre en forme à l'aide d'un fichier CSS</p>
  </body>
</html>
```

- On peut créer une page web en utilisant un éditeur de texte classique ou un éditeur spécialisé dans le HTML qui pourra par exemple mettre en couleur les balises, vérifier si les balises sont fermées...
- Si l'on souhaite de l'aide (page préconstruite) on peut aussi utiliser un CMS (*content management system* ou système de gestion des contenus) qui fonctionne en général sur le principe du WYSIWYG (*What you see is what you get* ou ce que l'on voit est ce qui sera affiché).
- Dans tous les cas, on crée un fichier texte dont l'extension sera « html ». C'est ce fichier qui est à ouvrir avec un navigateur afin d'afficher la page web que l'on a créée.
- La balise <head> permet de préciser des informations comme le type d'encodage, le titre de la page...
- La balise <body> contient le « corps » de la page web.
- La balise <h1> permet de créer un titre d'importance 1 à 5 (1 étant très important).
- La balise <ul> permet de créer une liste d'éléments.
- La balise <p> permet de créer un paragraphe.
- La balise <iframe> permet d'obtenir une page web intégrée dans la page courante.
- Il existe de nombreuses balises et chacune possède une utilité particulière.
- Il ne faut pas oublier de fermer une balise que l'on a ouverte. Certaines balises, cependant, sont dites auto-fermantes (ou orphelines) et n'ont pas besoin d'être refermées : <br> pour un retour à la ligne, <img> pour insérer une image...

- Si l'on ouvre le fichier html créé avec un navigateur, alors on obtient par exemple ceci :

# Les organismes internationaux

Voici deux organismes :

- [ONU](#)
- [AIEA](#)

**Un paragraphe est souvent utile pour créer ce qu'on appelle un bloc, c'est à dire un ensemble, que l'on va pouvoir mettre en forme à l'aide d'un fichier CSS**

- Une même page web peut être affichée différemment par deux navigateurs différents.

## CSS

• Un fichier **CSS** (*Cascading Style Sheets* ou feuilles de style en cascade) est un fichier texte qui contient des règles de mise en forme et précise le style graphique d'une page web. On peut créer un fichier CSS à partir d'un éditeur de texte classique, il faudra juste préciser l'extension après enregistrement : « css ».

• **Exemple** : On souhaite que les titres de niveau 1 soient affichés en rose et en police Times. On écrira :

```
h1
{
color : pink
font-family : Times
}
```

- On dit que h1 est un sélecteur, color est une propriété et pink est une valeur.
- Il existe beaucoup de sélecteurs différents afin que l'utilisateur puisse cibler le bloc ou la partie d'un bloc qu'il souhaite mettre en forme. Une très grande quantité de propriétés sont également disponibles : background, border, cursor, flex, font-size, justify-content...
- Le but de la création d'un fichier CSS est d'homogénéiser la présentation d'un site web. De plus, il sera plus facile d'effectuer des modifications de présentation dans un seul fichier.
- Il ne faut pas oublier de **lier son fichier CSS à son fichier HTML** à l'aide de la balise <link> que l'on place dans le <header>.
- On peut accéder au code source (HTML et CSS) de n'importe quelle page web. Il suffit d'ouvrir la page avec un navigateur, d'effectuer un clic droit n'importe où sur la page et de sélectionner « Examiner l'élément » ou « Inspecter l'élément » ou encore « Code source de la page ».

## Javascript

- Le code Javascript est exécuté côté client.
- Une page web qui contient du Javascript est dite page web interactive.

## PHP

- Le code PHP est exécuté côté serveur.
- Une page web qui contient du PHP est dite page web dynamique.

## V. URL

• Une ressource du Web possède une adresse unique : son URL qui signifie Uniform Resource Locator ou localisateur uniforme de ressource. Elle indique l'emplacement de la ressource sur le serveur. C'est grâce à l'URL que l'on accède à la ressource ; elle pourra ainsi être référencée.

**Exemple :**

- <https://www.un.org/fr/member-states/index.html> est une URL ;
- https est le **protocole** utilisé. D'autres existent comme http, mailto, ftp, file... ;
- www est le serveur web (sous-domaine, ou préfixe) ;
- un est le domaine de second niveau ;

- org est le **domaine de premier niveau** (*top level domain* ou TLD) ;
  - un.org est le **nom de domaine** ;
  - www.un.org est le nom de serveur ;
  - /fr/member-states/ est le chemin d'accès (le répertoire sur le serveur) ;
  - index.html est le fichier.
- On peut éventuellement trouver une ancre dans une URL, à l'aide du symbole #, pour qu'une page web (souvent longue) s'ouvre à un certain endroit. L'ancre est également nommée dans le code HTML.

- Dans l'URL d'une page dynamique sont parfois placés des paramètres après un « ? », liés entre eux par un « & ».

**Exemple :**

`https://search.un.org/results.php?ic=utf8&output=xml_no_dtd&oe=utf8&Submit=Search&query=france&tpl=un&lang=fr&rows=10`

- Dix paramètres ont été renseignés ici.
- Un utilisateur (particulier, entreprise, association...) loue un nom de domaine à un distributeur agréé de l'ICANN. Ce dernier procède alors à l'enregistrement du nom sur au moins deux serveurs DNS.
- C'est à un nom de domaine que les serveurs DNS vont associer une adresse IP.

## VI. HTTP

### Protocole HTTP

- Les ressources du web sont accessibles via Internet grâce au protocole HTTP ou à sa version sécurisée : le protocole HTTPS. Ce dernier crypte les échanges à l'aide de certificats SSL.
- HTTP est un protocole de la couche application dont les données vont transiter via TCP. Il précise quel message le navigateur doit envoyer par le réseau Internet pour demander une ressource à un serveur et dans quel format il attend une réponse.

### Requête HTTP

- Pour que le navigateur de l'appareil d'un utilisateur puisse obtenir une ressource, elle doit avoir fait l'objet d'une **requête**.
- Le navigateur demande au serveur qui héberge la ressource de la lui envoyer et le serveur envoie une réponse.
- Pour consulter la page test.html du site web www.exemple.fr, le navigateur doit envoyer la requête **GET** www.exemple.fr/test.html.
- Le mot-clé GET signifie que le client demande une information au serveur.
- Le mot-clé POST signifie que le client souhaite envoyer une information au serveur (identifiant, mot de passe,...). Il en existe d'autres comme PUT, DELETE et PATCH.
- Le protocole HTTP définit également des codes de statut que le serveur va envoyer au client pour lui dire comment la requête s'est passée. Les deux codes les plus utilisés sont :
  - Tout s'est bien passé : 200 Ok ;
  - Lien mort : 404 Not Found.

- Prenons l'exemple concret de la page web dont l'URL est :

`http://www.archives.nantes.fr/PAGES/DOSSIERS_DOCS/monument_aux_morts/pagel.html`

- Lorsque l'on entre cette URL dans la barre d'adresse d'un navigateur, celui-ci va envoyer la requête suivante au serveur :

```

▶ Frame 9: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on interface 0
▶ Ethernet II, Src: [redacted], Dst: Sagemcom_0e:93:54 (b8:d9:4d:0e:93:54)
▶ Internet Protocol Version 4, Src: 192.168.0.13, Dst: 80.82.231.240
▶ Transmission Control Protocol, Src Port: 51531, Dst Port: 80, Seq: 1, Ack: 1, Len: 414
▼ Hypertext Transfer Protocol
  ▼ GET /PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html HTTP/1.1\r\n
    ▼ [Expert Info (Chat/Sequence): GET /PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html HTTP/1.1\r\n]
      [GET /PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
      Request Method: GET
      Request URI: /PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html
      Request Version: HTTP/1.1
      Host: www.archives.nantes.fr\r\n
      User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:67.0) Gecko/20100101 Firefox/67.0\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3\r\n
      Accept-Encoding: gzip, deflate\r\n
      DNT: 1\r\n
      Connection: keep-alive\r\n
      Upgrade-Insecure-Requests: 1\r\n
      \r\n
      [Full request URI: http://www.archives.nantes.fr/PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html]
      [HTTP request 1/6]
      [Response in frame: 21]
      [Next request in frame: 24]

```

```

0040 00 00 47 45 54 20 2f 50 41 47 45 53 2f 44 4f 53  GET /PAGES/DOS
0050 53 49 45 52 53 5f 44 4f 43 53 2f 6d 6f 6e 75 6d  SIERS_DO CS/monum
0060 65 6e 74 5f 61 75 78 5f 6d 6f 72 74 73 2f 70 61  ent_aux_morts/pa
0070 67 65 31 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e  ge1.html HTTP/1.
0080 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 61 72 63  l·Host: www.arc
0090 68 69 76 65 73 2e 6e 61 6e 74 65 73 2e 66 72 0d  hives.nantes.fr·
00a0 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a  ·User-Agent: Moz

```

- On peut observer qu'en plus du message GET, on précise au serveur le type d'appareil de l'utilisateur, le navigateur qu'il utilise et sa version.
- Voici la réponse reçue :

```

▶ Frame 21: 748 bytes on wire (5984 bits), 748 bytes captured (5984 bits) on interface 0
▶ Ethernet II, Src: Sagemcom_0e:93:54 (b8:d9:4d:0e:93:54), Dst: [redacted]
▶ Internet Protocol Version 4, Src: 80.82.231.240, Dst: 192.168.0.13
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 51531, Seq: 9577, Ack: 415, Len: 682
▶ 18 Reassembled TCP Segments (10258 bytes): #10(1368), #11(1368), #13(1368), #15(1368), #16(1368), #19(1368), #20(1368), #21(682)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    ▼ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    ▼ Content-Length: 10005\r\n
      [Content length: 10005]
      Content-Type: text/html\r\n
      Last-Modified: Tue, 05 Jun 2018 11:34:53 GMT\r\n
      Accept-Ranges: bytes\r\n
      ETag: "9b8e8738c1fcd31:13424"\r\n
      Server: Microsoft-IIS/6.0\r\n
      X-Powered-By: ASP.NET\r\n
      Date: Sun, 23 Jun 2019 19:48:19 GMT\r\n
      \r\n
      [HTTP response 1/6]
      [Time since request: 0.079850000 seconds]
      [Request in frame: 9]
      [Next request in frame: 24]
      [Next response in frame: 33]
      [Request URI: http://www.archives.nantes.fr/PAGES/DOSSIERS_DOCS/monument_aux_morts/page1.html]
      File Data: 10005 bytes
    ▶ Line-based text data: text/html (118 lines)

```

```

0000 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d  HTTP/1.1 200 OK·
0010 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a  Content-Length:
0020 20 31 30 30 30 35 0d 0a 43 6f 6e 74 65 6e 74 2d  10005· Content-

```

Frame (748 bytes)    Reassembled TCP (10258 bytes)

- Le message reçu est un 200 Ok, donc le fichier page1.html est envoyé. Il comporte 118 lignes dont voici les premières :

```

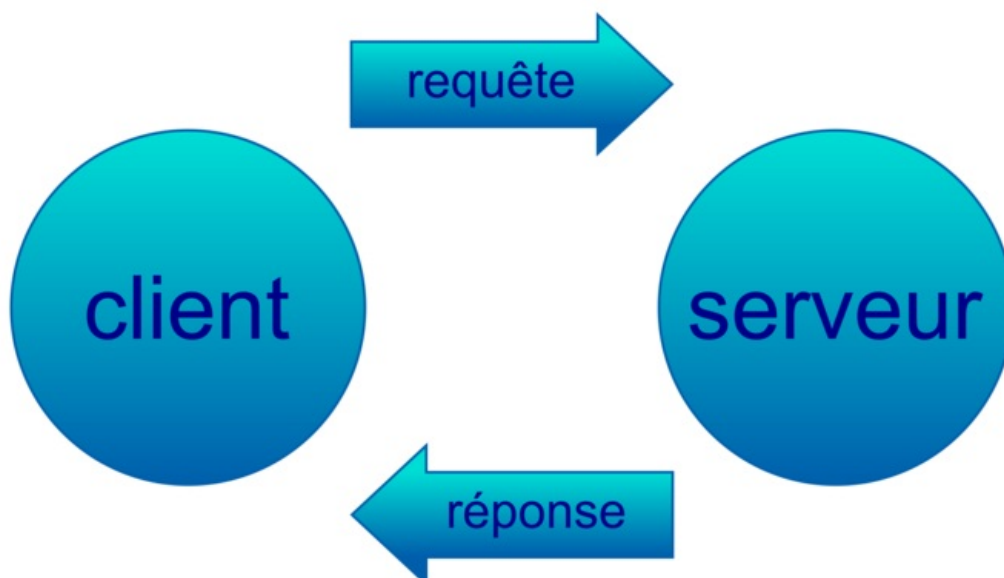
v Line-based text data: text/html (118 lines)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">\r\n
<html xmlns="http://www.w3.org/1999/xhtml">\r\n
\r\n
<head> \r\n
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> \r\n
<title>Archives de Nantes : le monuments aux morts</title>\r\n
<link href="style.css" rel="stylesheet" type="text/css" />\r\n
<script type="text/javascript" src="highslide/highslide.js"></script>\r\n
<link rel="stylesheet" type="text/css" href="highslide/highslide.css" />\r\n
<script type="text/javascript">\r\n
  hs.graphicsDir = 'highslide/graphics/';\r\n
</script>\r\n
<style type="text/css">\r\n
<!--\r\n
.Style1 {\r\n
\tfont-family: Verdana, Arial, Helvetica, sans-serif;\r\n
\tfont-weight: bold;\r\n
\tpadding-bottom: 20px;\r\n
}\r\n
.Style2 {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: x-small; }\r\n
-->\r\n
</style>\r\n
</head> \r\n
\r\n
<body> \r\n
<div align="left"></div>\r\n
<div id="conteneur">\r\n
\r\n
<div id="menusite">\r\n
<ul>\r\n
<li class="espace"><a href="../../default.htm">Page d'accueil</a></li>\r\n
<li class="espace"><a href="../../CONTACT/page1.htm">Contacts</a></li>\r\n

```

- L'échange va se poursuivre entre le client et le serveur puisque le client va avoir besoin du fichier style.css, d'un fichier en Javascript, des photos... Au total 33 requêtes et 33 réponses juste pour le chargement de cette unique page.

## VII. Modèle client/serveur

- Dans l'architecture du Web, les appareils sont répartis en deux types : les **clients** (ordinateurs, smartphones...) et les **serveurs** (ordinateurs connectés constamment à l'Internet) qui fournissent l'information au client.
- Les ressources sont stockées sur un serveur ou bien elles sont créées dynamiquement.
- Initiative du client : il envoie une requête HTTP au serveur.
- Dans un cas sans encombre, le serveur envoie une réponse positive et la ressource au client.
- L'affichage des pages (lecture du HTML et du CSS) est réalisé chez l'utilisateur par le navigateur. Une page peut contenir des codes exécutables (en Javascript par exemple) et sera alors dite interactive.
- Les échanges client-serveur se résument ainsi à ceci :



- Si le serveur diffuse les mêmes ressources à tous les utilisateurs, alors on parle de site web statique.
- En revanche, si le serveur est capable de personnaliser le contenu en fonction de la demande de l'utilisateur, alors on parle de site web dynamique (réservation d'un billet d'avion, consultation de son compte en banque...).

## VIII. Moteur de recherche

### Définition

Un moteur de recherche permet à un utilisateur de **trouver des ressources sur le Web** à l'aide de **mots-clés**. À la suite d'une demande, le moteur donne une **page de résultats** (SERP) qui contient les URL des ressources jugées pertinentes pour l'utilisateur. Tout est calculé à l'avance, ce qui permet une réponse très rapide.

### Indexation

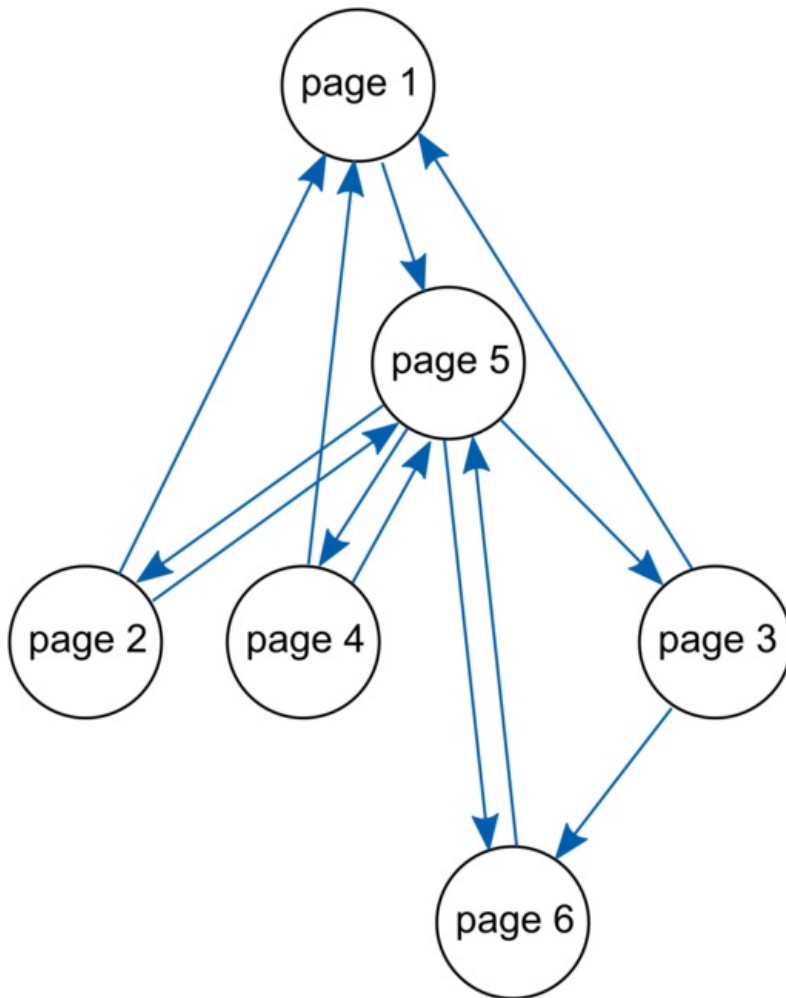
- Certains livres contiennent à la fin un index qui permet à un lecteur qui souhaite trouver rapidement une notion d'obtenir les pages du livre où cette notion est mentionnée.
- Un moteur de recherche utilise des programmes (robots, *crawlers*) qui parcourent le Web constamment. Ces robots analysent les ressources du Web pour y détecter des mots-clés (contenus dans une base de données), puis fabriquer un **index** permettant de retrouver ces ressources à l'aide de leurs URL. Une fois qu'un robot a fini de « travailler » sur une page web, il se rend sur tous les liens proposés par cette page, et il recommence.
- Le procédé d'indexation est le même utilisé par Windows Search sur Windows (ou Spotlight sur macOS) lorsque l'on recherche un fichier sur son ordinateur personnel.

### Classement

- La difficulté réside dans le **classement** pertinent des ressources contenant les mots-clés demandés par l'utilisateur. C'est sur ce point essentiel que les différents moteurs se concurrencent.
- Google utilise entre autres son algorithme PageRank (depuis 1998) dont le principe est d'attribuer à chaque ressource une note de popularité proportionnelle au nombre de fois que passerait par cette page un utilisateur parcourant le Web en cliquant au hasard sur un des hyperliens apparaissant sur chaque page. Plus la somme des notes des ressources qui pointent vers une ressource est grande, plus la note de la ressource sera élevée.
- Le classement final prend également en compte le nombre de fois où le mot-clé est mentionné dans la page web, sa présence dans des balises de type titre du code HTML et le nombre de fois qu'un utilisateur a choisi une certaine URL en réponse à une recherche.
- Un utilisateur qui possède un site web peut améliorer le référencement de ce dernier de deux manières principales :
  - le **référencement naturel** (SEO) avec des méthodes particulières, comme celles qui consistent à placer les mots-clés de manière stratégique (répétition, placement dans les titres), ou à augmenter la vitesse de chargement... ;
  - le **référencement payant** (SEA) en payant un (ou plusieurs) moteur(s) de recherche pour apparaître dans les publicités. Il est alors écrit à gauche de l'URL « **Annnonce** » pour le signaler.
- Deux moteurs de recherche n'ayant pas les mêmes algorithmes de classement, ils ne vont pas fournir les mêmes pages de résultats.
- **4 % des ressources du Web seulement sont indexées**. Le reste est le *Deep Web*.

### Modélisation à l'aide d'un graphe

- Considérons six pages web dont certaines possèdent un hyperlien renvoyant vers une autre page.
- Pour simuler un robot on choisit une page de départ, puis on choisit au hasard un hyperlien pour arriver sur la page suivante.
- En répétant l'expérience un très grand nombre de fois et en notant le nombre de visites effectuées sur chaque page, on observe que la fréquence de visite de la page 5 est supérieure à celles des autres pages et qu'elle semble se stabiliser autour de 40 %.



- On peut vérifier ce que l'on affirme à l'aide d'un programme comme celui-ci :

```

import random
Les_pages=["Page1", "Page2", "Page3", "Page4", "Page5", "Page6"]
Hyperlien={}
Hyperlien["Page1"]=["Page5"]
Hyperlien["Page2"]=["Page1", "Page5"]
Hyperlien["Page3"]=["Page1", "Page6"]
Hyperlien["Page4"]=["Page1", "Page5"]
Hyperlien["Page5"]=["Page2", "Page3", "Page4", "Page6",]
Hyperlien["Page6"]=["Page5"]

def Simulation(durée_vie, Nombre_repetition):
    for i in range(Nombre_repetition):
        Nombre_visites={}
        Nombre_visites["Page1"]=0
        Nombre_visites["Page2"]=0
        Nombre_visites["Page3"]=0
        Nombre_visites["Page4"]=0
        Nombre_visites["Page5"]=0
        Nombre_visites["Page6"]=0
        page_au_hasard=random.choice(Les_pages)
        for j in range(durée_vie):
            Nombre_visites[page_au_hasard]=Nombre_visites[page_au_hasard]+1
            page_au_hasard=random.choice(Hyperlien[page_au_hasard])
        print("\n")
        for page in Les_pages:
            print("Pour ", page, " il y a eu ", Nombre_visites[page], "visites")
            print("Soit une fréquence de environ",
                round((Nombre_visites[page]/durée_vie)*100,2)
                , "% d'arriver sur ", page)
  
```



- Les variables « Hyperlien » et « Nombre\_visites » sont des dictionnaires alors que la variable « Les\_pages » est une liste classique.
- On a créé une fonction « Simulation » afin de pouvoir répéter facilement l'expérience. L'utilisateur devra choisir la durée de vie d'un robot et le nombre de fois pour lequel il souhaite répéter l'expérience.
- Une fois le programme compilé, on a entré « Simulation(1000000,2) » dans le *shell* et on a obtenu ceci :

**In [2]: Simulation(1000000,2)**

```
Pour Page1 il y a eu 149809 visites
Soit une fréquence de environ 14.98 % d'arriver sur Page1
Pour Page2 il y a eu 100053 visites
Soit une fréquence de environ 10.01 % d'arriver sur Page2
Pour Page3 il y a eu 99716 visites
Soit une fréquence de environ 9.97 % d'arriver sur Page3
Pour Page4 il y a eu 100187 visites
Soit une fréquence de environ 10.02 % d'arriver sur Page4
Pour Page5 il y a eu 400193 visites
Soit une fréquence de environ 40.02 % d'arriver sur Page5
Pour Page6 il y a eu 150042 visites
Soit une fréquence de environ 15.0 % d'arriver sur Page6
```

```
Pour Page1 il y a eu 149737 visites
Soit une fréquence de environ 14.97 % d'arriver sur Page1
Pour Page2 il y a eu 99813 visites
Soit une fréquence de environ 9.98 % d'arriver sur Page2
Pour Page3 il y a eu 99865 visites
Soit une fréquence de environ 9.99 % d'arriver sur Page3
Pour Page4 il y a eu 100273 visites
Soit une fréquence de environ 10.03 % d'arriver sur Page4
Pour Page5 il y a eu 400050 visites
Soit une fréquence de environ 40.01 % d'arriver sur Page5
Pour Page6 il y a eu 150262 visites
Soit une fréquence de environ 15.03 % d'arriver sur Page6
```

- Cela a pris quelques secondes à l'ordinateur, mais on observe bien que lorsque l'on répète deux fois l'expérience avec des robots qui ont une durée de vie d'un million de pages, alors la page la plus visitée est bien la page 5 avec environ 40 % du trafic.
- Les pages 2, 3 et 4 semblent obtenir environ 10 % du trafic alors que les pages 1 et 6 semblent recueillir environ 15 % du trafic.
- On pourra vérifier qu'en changeant un arc dans notre graphe, les fréquences changent.

## IX. Paramètres de sécurité d'un navigateur

Ces paramètres sont accessibles dans le menu « Préférences » du navigateur et dans l'onglet « Sécurité » et/ou « Vie privée ».

**Général** Vie privée

**Accueil**

**Recherche**

**Vie privée et sécurité**

**Sync**

**Blocage de contenu**

Bloquez les contenus tiers qui vous pistent sur le Web. Contrôlez le niveau d'activité en ligne qui est enregistrée et partagée avec des sites web. [En savoir plus](#) [Gérer les exceptions...](#)

**Standard**  
Bloque uniquement les traqueurs connus dans les fenêtres de navigation privée.

**Strict**  
Bloque tous les traqueurs détectés par Firefox. Peut provoquer le dysfonctionnement de certains sites.

- Les traqueurs connus dans toutes les fenêtres
- Les cookies tiers utilisés pour le pistage

- « Envoyer aux sites web un signal "Ne pas me pister" indiquant que vous ne souhaitez pas être pisté » : Il est plus prudent de cocher « Toujours ».

Envoyer aux sites web un signal « Ne pas me pister » indiquant que vous ne souhaitez pas être pisté

[En savoir plus](#)

**Toujours**

Seulement quand Firefox est paramétré pour bloquer les traqueurs connus

**Cookies et données de sites**

Le stockage des cookies, du cache et des données de sites utilise actuellement 0 octets d'espace disque. [En savoir plus](#)

[Effacer les données...](#)

[Gérer les données...](#)

[Gérer les permissions...](#)

En mode de navigation privée permanent, les cookies et les données de site sont toujours effacés à la fermeture de Firefox.

Supprimer les cookies et les données des sites à la fermeture de Firefox

- En effet, la plupart des sites web suivent les habitudes de leurs visiteurs (puis vendent ou fournissent ces informations à d'autres sociétés). La prise en compte de la demande résulte néanmoins d'une **démarche volontaire** de la part des sites web : ils ne sont pas tenus de la respecter.
- Des sites web peuvent conserver des informations, stockées dans des fichiers sur l'appareil de l'utilisateur. Ces fichiers ne peuvent être supprimés que manuellement par l'utilisateur. Cela permet au site web d'être plus rapide et d'empêcher la perte d'informations si votre connexion est perdue.
- Un **cookie** (témoin de connexion) est une information conservée, en général sous la forme d'un petit fichier texte, sur l'appareil d'un utilisateur par un site web qu'il a visité.
- Les cookies contiennent la langue préférée, la localisation, toutes les informations données par l'utilisateur sur le site, les pages consultées dans le site et leur ordre...
- De plus, si un élément de la page visitée est hébergé dans un serveur différent de celui du site web consulté alors ce site tiers peut, lui aussi, envoyer des cookies à l'utilisateur.
- La gestion de l'historique est une étape importante avant de surfer sur le Web.

**Historique**

Règles de conservation **Ne jamais conserver l'historique**

Firefox utilisera les mêmes paramètres que pour la navigation privée et ne conservera aucun historique de votre navigation. [Effacer l'historique...](#)

- Enfin il faut vérifier les permissions que l'on donne aux sites web pour qu'il puisse accéder à du matériel de l'appareil.

## Permissions

|  |               |
|--|---------------|
|  Localisation                                  | Paramètres... |
|  Caméra  | Paramètres... |
|  Microphone                                    | Paramètres... |
|  Notifications <a href="#">En savoir plus</a>  | Paramètres... |
| <input checked="" type="checkbox"/> Empêcher la lecture automatique de contenus sonores  | Exceptions... |
| <input checked="" type="checkbox"/> Bloquer les fenêtres popup   | Exceptions... |
| <input checked="" type="checkbox"/> Prévenir lorsque les sites essaient d'installer des modules complémentaires                | Exceptions... |
| <input type="checkbox"/> Empêcher les services d'accessibilité d'avoir accès à votre navigateur <a href="#">En savoir plus</a> |               |

 [Exercice n°1](#)

 [Exercice n°2](#)

 [Exercice n°3](#)

 [Exercice n°4](#)

 [Exercice n°5](#)