



Partie 1

0. Mythes sur le développement web

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Exemple d'application web

The screenshot shows a Microsoft Excel interface running in a web browser. The spreadsheet contains a table titled 'Sponsor List' with columns for Sponsor, Pledge, and Pledge Date. A bar chart titled 'Pledge Amount' is embedded in the spreadsheet, showing the total pledge amount for each sponsor. The chart has categories for Michael Schultz, Rachel Bondi, Uta Cebeci, Minesh Lad, Justin Hutchinson, Katie Madrid, Jennifer Pisani, and Anthony Sprause.

Sponsor	Pledge	Pledge Date
Michael Schultz	\$ 300.00	16-Aug
Rachel Bondi	\$ 350.00	17-Jul
Uta Cebeci	\$ 350.00	8-Aug
Minesh Lad	\$ 200.00	19-Jun
Justin Hutchinson	\$ 250.00	29-Aug
Katie Madrid	\$ 450.00	23-Aug
Jennifer Pisani	\$ 150.00	3-Sep
Anthony Sprause	\$ 45.00	5-Sep

Total Pledged: \$ 2,200.00
Goal: \$ 3,000.00

Microsoft Office Live

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Mythe #1:

Il n'y a pas grand chose à dire sur la façon d'écrire une page web.

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Mythe

Mythe #2:

Il y a des logiciels qui nous permettent d'écrire des pages web.

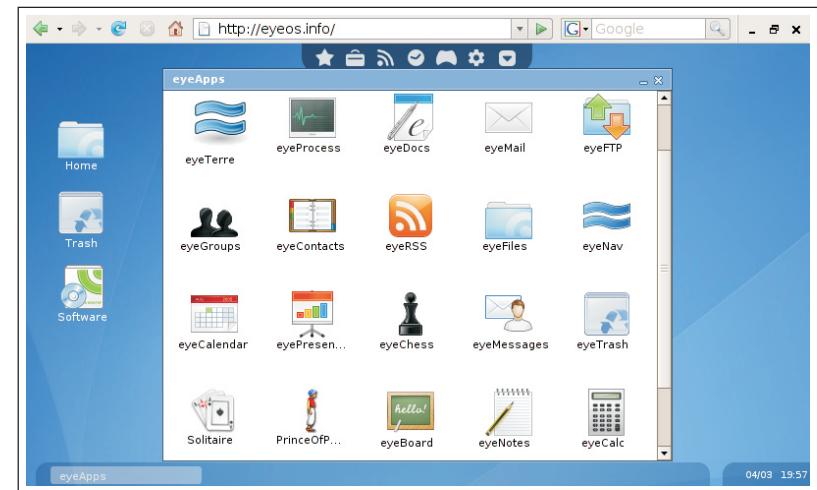
The screenshot shows the WordPress dashboard with a post editor open. The title of the post is 'Just Another WordPress Weblog'. The editor includes fields for Title, Content (with placeholder text), and Publish buttons. The left sidebar shows navigation links like Dashboard, Posts, Add New Post, and Plugins.

WordPress

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Mythe #3:

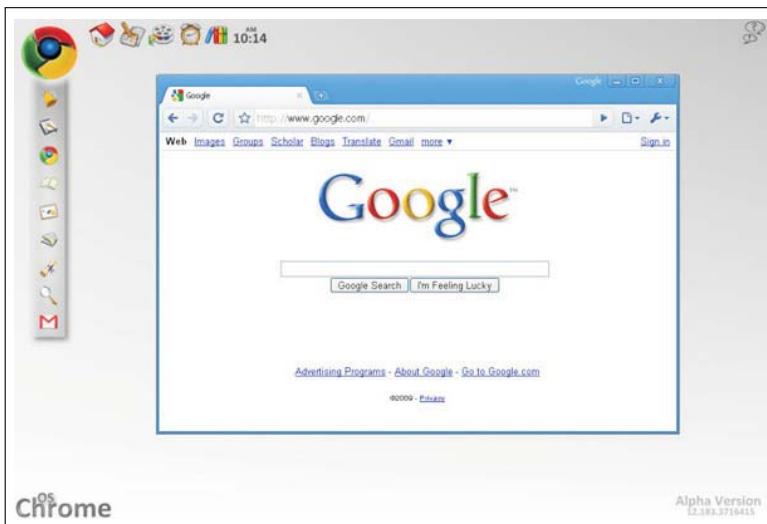
Pour mon travail, connaître les technologies web n'est pas aussi important que les vrais langages de programmation.



eyeOS



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Exemples d'applications web 

Chrome OS

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Partie 1

I. Les origines d'Internet et du web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Selon Ted Stevens (sénateur républicain, Alaska), Internet est « une série de tubes »...



Ted Stevens



W

...George W. Bush parlait « des Internets »

Plus sérieusement...



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Historique d'Internet 🕹️

Origine: la guerre froide (années 60-70)



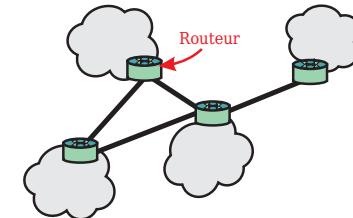
Le Département de la Défense américain imagine un réseau appelé ARPANET

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Internet

Réseau physique reliant des millions d'ordinateurs utilisant les mêmes protocoles pour partager et transmettre de l'information (TCP/IP)

En réalité, c'est un « réseau de réseaux »



💡 Exemple d'application Internet: le courriel. Pouvez-vous en nommer d'autres?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Historique d'Internet 🕹️

Objectif: relier les grands ordinateurs du pays pour survivre à une **attaque nucléaire**

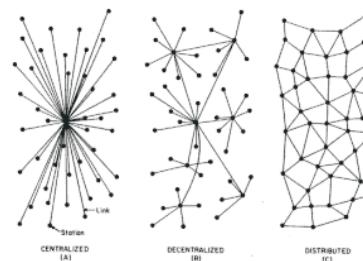
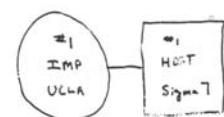


FIG. 1 – Centralized, Decentralized and Distributed Networks



THE ARPA NETWORK

SEPT. 1969

i NODE

Topologies étudiées

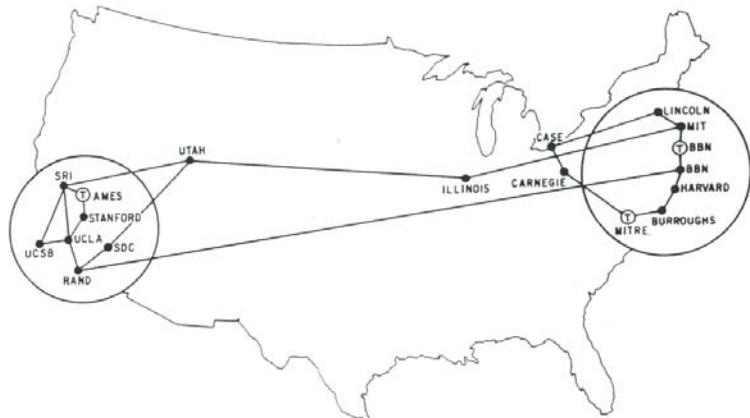
💡 Laquelle des trois a-t-on retenue?



ARPANET en sept. '69

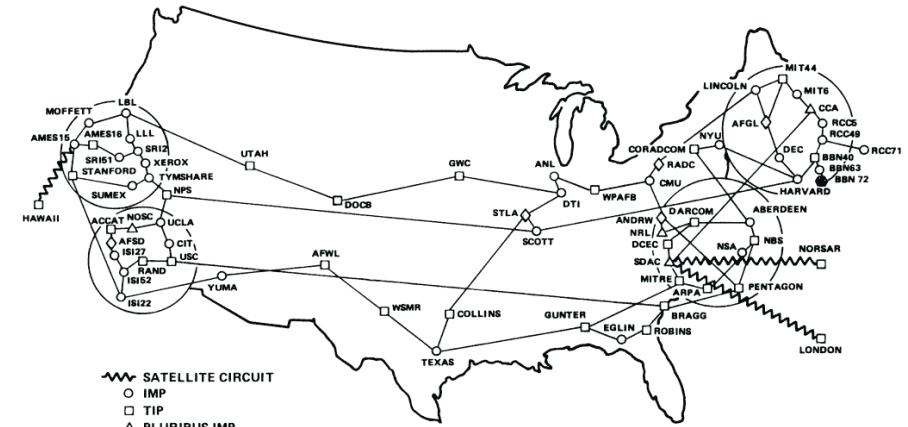
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

ARPANET en sept. '71



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

ARPANET en oct. '80



(NOTE: THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS)
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Protocole fourre-tout: **Telnet**. Permet d'échanger des commandes à travers une fenêtre de terminal (texte!)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un des premiers services: le **courriel**

```
Microsoft ESMTP Mail Service, Version: 6.0.3790.1830
ready at Sat, 19 Jan 2008 12:13:26 -0400
250 tazmania.production.com Hello [147.100.100.38]
mail from: brian@briensposey.com
250 2.0.0 Ok [147.100.100.38]... Sender OK
rcpt to: Brian_Posey@hotmail.com
250 1.5.1 Brian_Posey@hotmail.com
data
254 Start mail input now with CRLF+CRLF
Subject: This is a simple message to demonstrate how SMTP works.
This is just a simple demonstration message. I am trying to show how you can compose and transmit an SMTP message from a command line.

250 2.6.0 c7ZM0N1Ry6d1H9J09z00000001@tazmania.production.com Queued mail for delivery
quit
221 2.0.0 tazmania.production.com Service closing transmission channel
```

Envoyer un courriel: protocole **SMTP** dans une session Telnet

```
BIN-TENEX 1.34.17, BBN-SYSTEM-E EXEC 1.54.45
2LDG FRITS 260960
J08 28 DM TTY123 24-MAR-77 21:31
PREVIOUS LOGIN: 23-MAR-77 19:16
YOU HAVE NEW MAIL)
READEMAIL 2E(15)
TYPE ? FOR HELP
*
DATE: 24 MAR 1977 0935-EST
FROM: MCQUILLAN
To: FRITS

FRITS-
THE DINNER DATE IS FINE,
I HAVE WRITTEN THE LETTER TO BERKELEY; I'LL WRITE A
SIMILAR ONE TO IBM.
John.
```

Un des premiers courriels échangés avec les en-tête classiques

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour lire ses courriels: protocole POP(3) dans une session...
Telnet!

```

Ble Modifica Visualizza Terminali Schede Ajuda
+OK Gpop ready for requests From 11.22.33.44 p9zfqlf6gkh.20
user mario.rossi
+OK send PASS
pass 123456
+OK Welcome.
1) User
+OK 345 31611000
list
+OK 345 messages (31611000 bytes)
1 48372
2 63916
3 207654
4 22598
.
345 1475
retr 5
+OK message follows
Date: Thu, 15 Dec 2005 20:58:23 +0100
From: Jane Doe <jane.doe@gmail.com>
To: Mario Rossi <mario.rossi@gmail.com>
Subject: [REDACTED]
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
Watch this: http://searchlores.org
.
quit
+OK Bye p9zfqlf6gkh.20

```



Access your email without an email client

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

1991: invention du protocole **Gopher**

```

THE GOPHER PROJECT
-----
Welcome to GOPHER! Gopher is a slim, powerful, and
fast way to present information in a hierarchical catalog online.
Gopher actually predates the Web -- although most web browsers
make excellent gopher browsers too.

Good places to start are the "Why Gopher?" and "Using Gopher"
areas!

->[12] *** GOPHER TURNS 10 / GOPHER 3.0 (FurryTerror) RELEASED ***
[13] *** GOPHER TURNS 10 ..R 3.0 (FurryTerror) RELEASED *** [html] <HTML>
[14] A Brief Introduction to Gopherspace
[15] Clients, Servers, and Downloads/
[16] Home Gopher at UMN (a good place to browse)/
[17] Home Gopher at UMN [alternate]/
[18] Mailing List
[19] Mailing List Archives/
[20] Major Gopher Servers/
[21] Screenshots/

Press H for Help, Q to Quit Page: 1/2

```

Sites avec hiérarchie de pages et de sous-pages



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Autre service: File Transfer Protocol (FTP)... encore dans un terminal!

```

[C:\temp]ftp 127.0.0.1
Connected to 127.0.0.1.
220 BulletProof FTP Server ready ...
User (127.0.0.1:(none)): mmc
331 Password required for mmc.
Password:
230 User mmc logged in.
Ftp> dir
200 Port command successful.
150 Opening data connection for directory list.
drwxr-xr-x 1 ftp ftp 0 Jun 18 12:16 s_drive
226 Transfer ok
ftp: 64 bytes received in 0.00Seconds 64000.00Kbytes/sec.
Ftp> cd s_drive
250 CWD command successful. "/s_drive" is current directory.
Ftp> dir
200 Port command successful.
150 Opening data connection for directory list.
-rw-r--r-- 1 ftp ftp 320 Jun 18 12:16 index.html
-rw-r--r-- 1 ftp ftp 535 Jun 18 12:16 myspreadsheet.xls
-rw-r--r-- 1 ftp ftp 481 Jun 18 12:16 myworddocument.doc
-rw-r--r-- 1 ftp ftp 375 Jun 18 12:16 picture01.jpg
-rw-r--r-- 1 ftp ftp 425 Jun 18 12:16 picture02.jpg
226 Transfer ok
ftp: 356 bytes received in 0.00Seconds 356000.00Kbytes/sec.
Ftp>

```

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Gopher fonctionne toujours! Tapez:

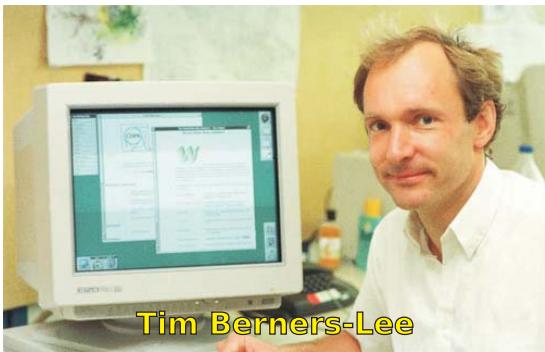
<gopher://gopherproject.org/>

dans votre navigateur et voyez ce qui se passe...



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

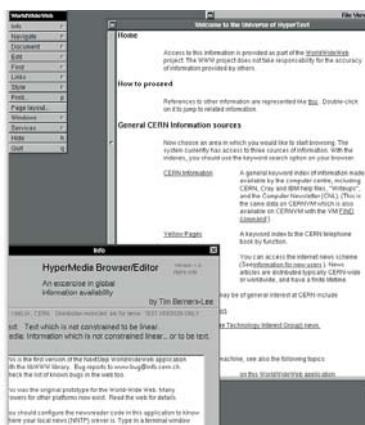
Invention du **World Wide Web** vers 1990-91



Notion d'hypertexte: un document peut contenir des liens vers d'autres documents situés sur **d'autres serveurs**



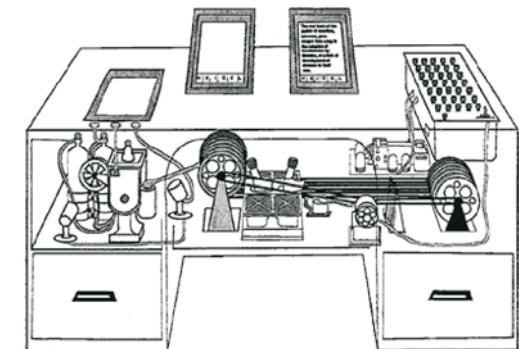
Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



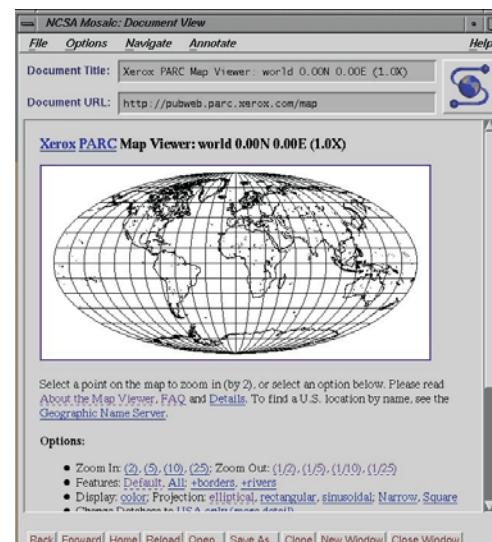
La notion d'hypertexte remonte en fait à 1945...



...le **memex** de Vannevar Bush



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

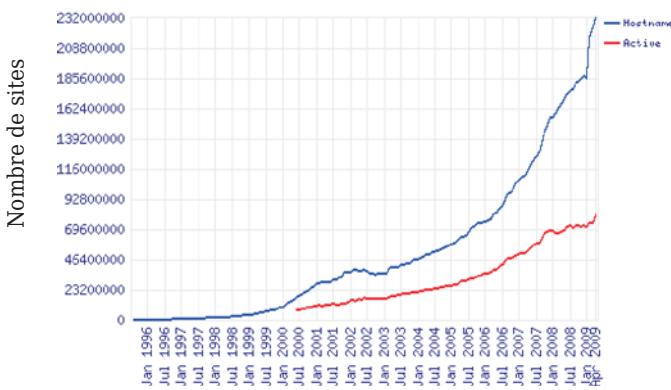


1992: **Mosaic** est le premier navigateur graphique populaire

Les pages peuvent mélanger texte formaté et graphiques via un langage appelé **HyperText Markup Language (HTML)**



Netscape 1.0 en 1994



Croissance du web depuis 1995



Welcome to Amazon.com Books!

*One million titles,
consistently low prices.*

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves EVERY day so please come often.

ONE MILLION TITLES

Search Amazon.com's [million title catalog](#) by author, subject, title, keyword, and more... Or take a look at the [books we recommend](#) in over 20 categories... Check out our [customer reviews](#) and the [award winners](#) from the Hugo and Nebula to the Pulitzer and Nobel... and [bestsellers](#) are 30% off the publishers list...

EYES & EDITORS, A PERSONAL NOTIFICATION SERVICE

Like to know when that book you want comes out in paperback or when your favorite author releases a new title? Eyes, our tireless, automated search agent, will send you mail. Meanwhile,

Amazon.com en 1995



(Essayez le **Wayback Machine**)

Aspects clés d'Internet

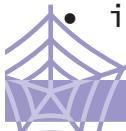
- But principal: partage de l'information
- Les sous-réseaux sont autonomes (ex.: étudiant de l'UQAC qui envoie un courriel à un autre étudiant de l'UQAC)
- Les ordinateurs peuvent se joindre/quitter le réseau de façon dynamique
- Construit autour de standards ouverts
- Presque aucun contrôle centralisé
- Tout le monde peut l'utiliser avec des logiciels simples et accessibles
- **Internet ≠World Wide Web!**
Internet = matériel (+ quelques protocoles de base)
WWW = logiciel + données (+ documents)



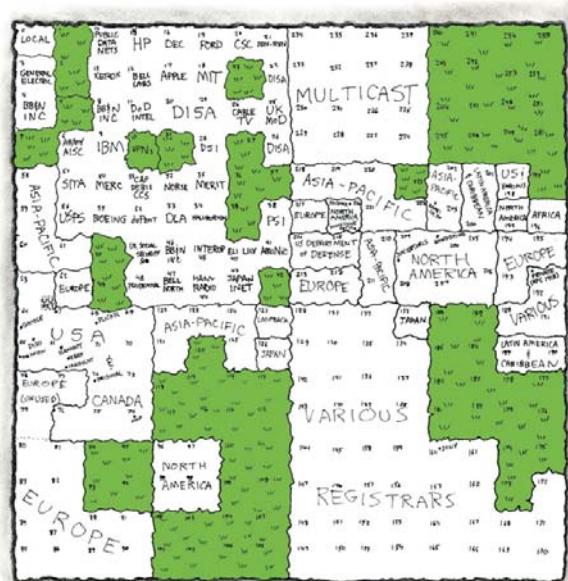
- Protocole simple pour envoyer des données entre deux ordinateurs
- Chaque machine possède une **adresse IP**: un nombre de 32 bits (IPv4) ou de 128 bits (IPv6) unique
- Une adresse (v4) s'écrit sous forme de 4 nombres de 8 bits (entre 0 et 255)

Ex.: 1001001 . 00001010 . 00100010 . 00000011
145.10.34.3

- Trouvez votre adresse IP: whatismyip.com
- ipconfig ou ifconfig dans un terminal



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- L'ICANN (*Internet Corporation for Assigned Names and Numbers*) attribue des blocs d'adresses IP à des organisations, qui sont responsables de les affecter
- Bloc de classe A: A.xxx.xxx.xxx, où $0 < A < 127$. Ex.: 18.x.x.x = MIT; 13.x.x.x = Xerox; 96.x.x.x = ARIN (American Registry for Internet Numbers), sous-traitant nord-américain chargé de redistribuer ces adresses (ex.: Vidéotron)
- Bloc de classe B: B1.B2.xxx.xxx, où $127 < B1 < 192$. Ex.: 132.203: Université Laval; 132.212: UQAC
- Bloc de classe C: C1.C2.C3.xxx, où $191 < C1 < 255$



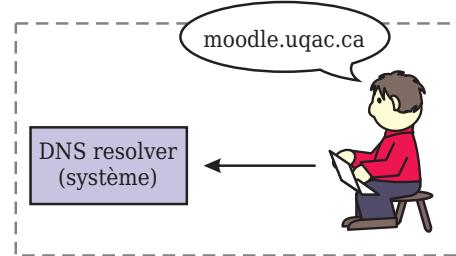
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Fonctionne « par-dessus » IP; ajoute le **multiplexage** et la **livraison garantie** des messages
- Multiplexage: permet à plusieurs programmes d'utiliser la même adresse IP
 - Port = numéro donné à chaque programme ou service
 - Port 80: navigateur web (HTTP)
 - Port 25: courriel (SMTP)
 - Port 23: Telnet
 - Port 70: Gopher
 - etc.
- Certaines applications utilisent plutôt UDP (plus simple) au lieu de TCP



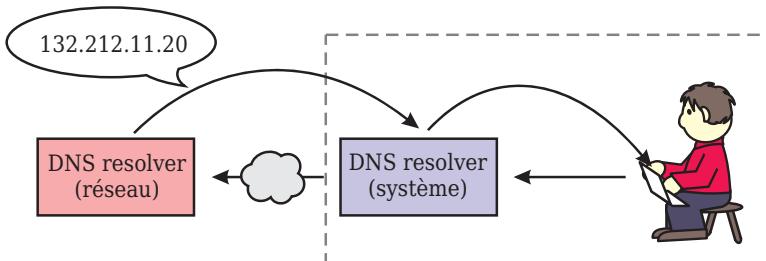
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Un serveur DNS établit une correspondance entre un nom de domaine et une adresse IP



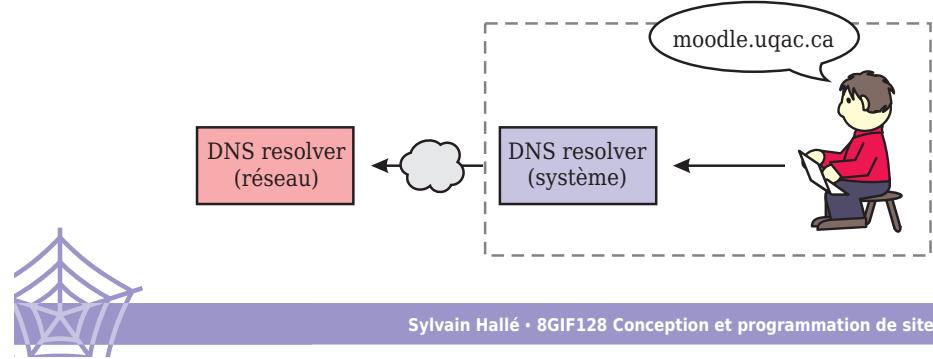
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Un serveur DNS établit une correspondance entre un nom de domaine et une adresse IP
- Le resolver envoie la requête au serveur DNS spécifié par la configuration du système (souvent celui du fournisseur)...
- ...lequel peut contacter d'autres serveurs DNS, jusqu'à l'obtention de l'adresse recherchée



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Un serveur DNS établit une correspondance entre un nom de domaine et une adresse IP
- Le resolver envoie la requête au serveur DNS spécifié par la configuration du système (souvent celui du fournisseur)...
- ...lequel peut contacter d'autres serveurs DNS, jusqu'à l'obtention de l'adresse recherchée



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Le résolveur place les entrées en **cache** (typiquement 60 sec.)

```
F:\>ipconfig /displaydns
Windows IP Configuration

1.0.0.127.in-addr.arpa
Record Name . . . . . : 1.0.0.127.in-addr.arpa.
Record Type . . . . . : 12
Time To Live . . . . . : 575706
Data Length . . . . . : 4
Section . . . . . : Answer
PTR Record . . . . . : localhost

technet2.microsoft.com
Record Name . . . . . : technet2.microsoft.com
Record Type . . . . . : 1
Time To Live . . . . . : 314
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . . : 207.46.196.114
```

- Le fichier cache est...
 - C:\WINDOWS\system32\drivers\etc\hosts (Windows)
 - /private/etc/hosts (Mac)
 - /etc/hosts (Linux)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



World Wide Web Consortium



Internet Engineering Task Force



Internet Corporation for Assigned Names and Numbers

Définit les standards comme HTML, CSS

Définit les protocoles comme IP, SMTP

Assigne les noms de domaines



Sylvain Hallé · 8GIF128 Conception et programmation de sites web

Navigateur (browser)

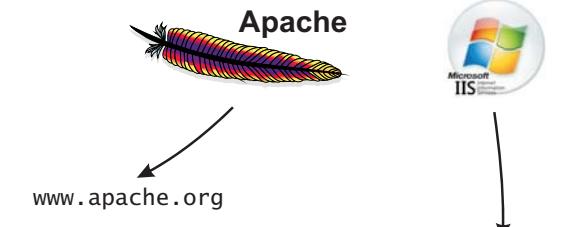


- Un logiciel qui affiche des pages web (et s'occupe de les récupérer des serveurs)
- Navigateurs populaires:



Sylvain Hallé · 8GIF128 Conception et programmation de sites web

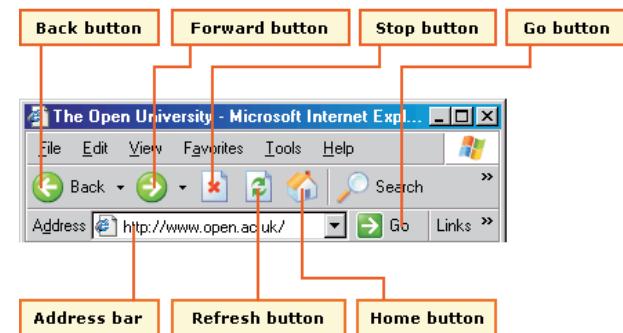
- Un ordinateur qui fait tourner un logiciel spécial qui écoute les requêtes sur le port TCP 80
- Logiciels de serveur web populaires:



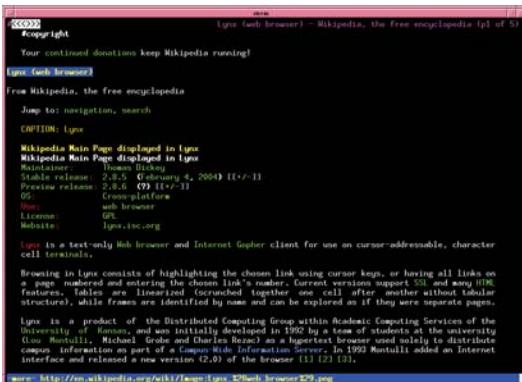
Sylvain Hallé · 8GIF128 Conception et programmation de sites web

Navigateur (browser)

- Éléments de navigation communs à la plupart des navigateurs:



Sylvain Hallé · 8GIF128 Conception et programmation de sites web



- Navigateur en mode texte (=Gopher!)
- Pas de JavaScript, pas d'images = sécuritaire + rapide
- Ce que Lynx voit = ce qu'un robot de moteur de recherche (spider) voit
- Un site que Lynx affiche mal = site mal structuré



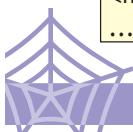
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

HyperText Transfer Protocol (HTTP)

- L'ensemble des commandes comprises par un serveur web et envoyées par un navigateur
- Quelques commandes HTTP (le navigateur les envoie sous le capot):
 - GET nomfichier: download
 - POST nomfichier: envoyer la réponse d'un formulaire
 - PUT nomfichier: upload
- On peut simuler le navigateur via une connexion... Telnet!

```
$ telnet example.com 80
Trying 128.208.3.88...
Connected to 128.208.3.88 (128.208.3.88).
Escape character is '^]'.
GET /beatles.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 ...">
<html>
...

```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



- Images et CSS de base
- Seul navigateur disponible dans certaines petites distributions de Linux



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Uniform Resource Locator (URL)

- Un identifiant pour l'emplacement d'un document sur un site web

Exemple:

http://www.example.com/artistes/beatles.html

Pouvez-vous identifier les trois parties de l'URL?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Le chargement d'une page 🕸️

10.1.1.2



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Le chargement d'une page 🕸️

10.1.1.2



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Le chargement d'une page 🕸️

10.1.1.2

1. On demande au navigateur d'afficher la page au bout d'une URL

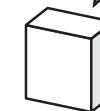
`http://example.com/artistes/beatles.html`



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

10.1.1.2

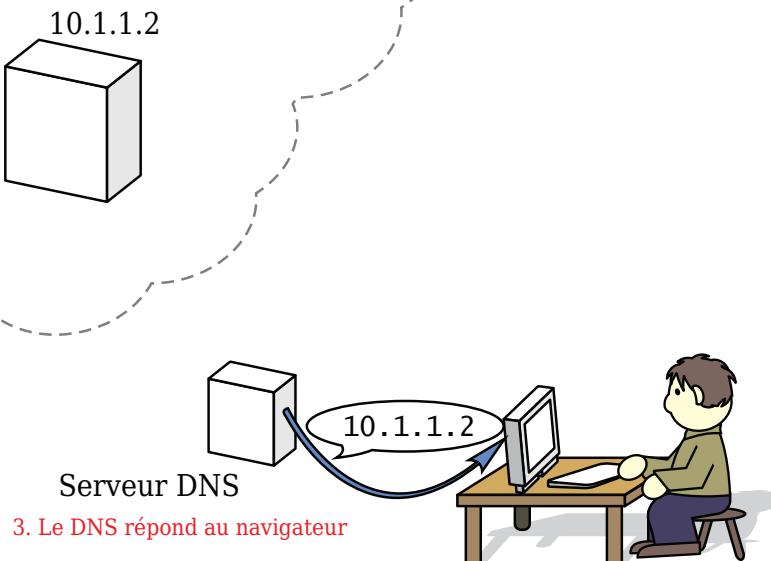
2. Le navigateur demande au DNS l'adresse IP du serveur en question



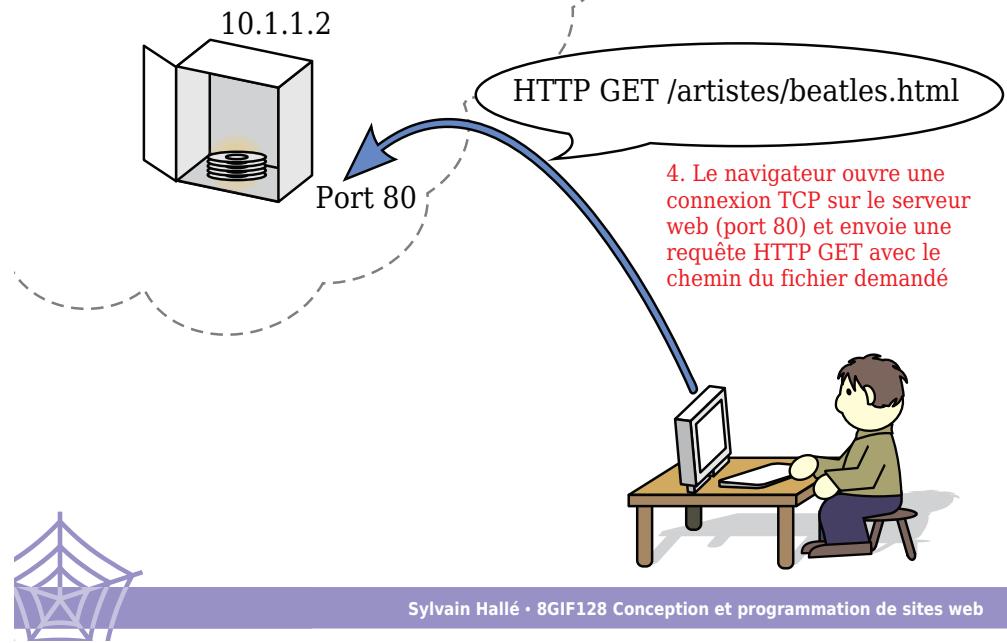
Serveur DNS

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

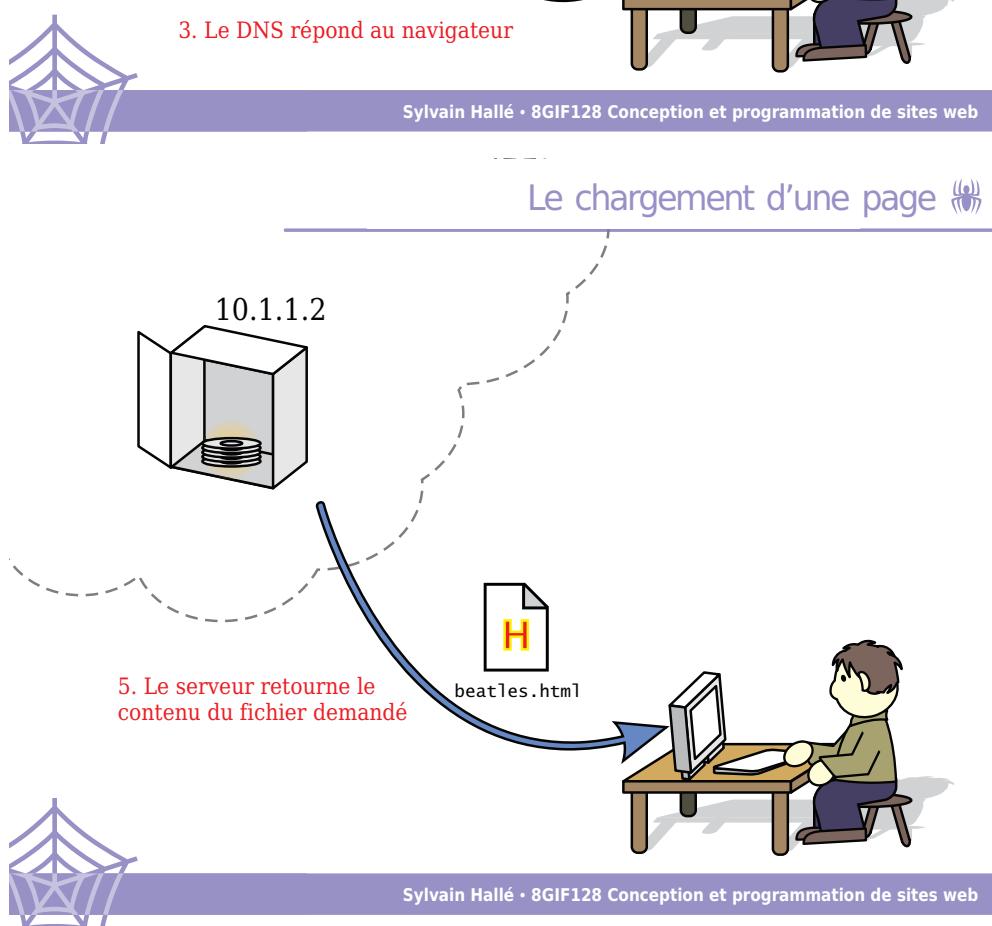
Le chargement d'une page 🕹️



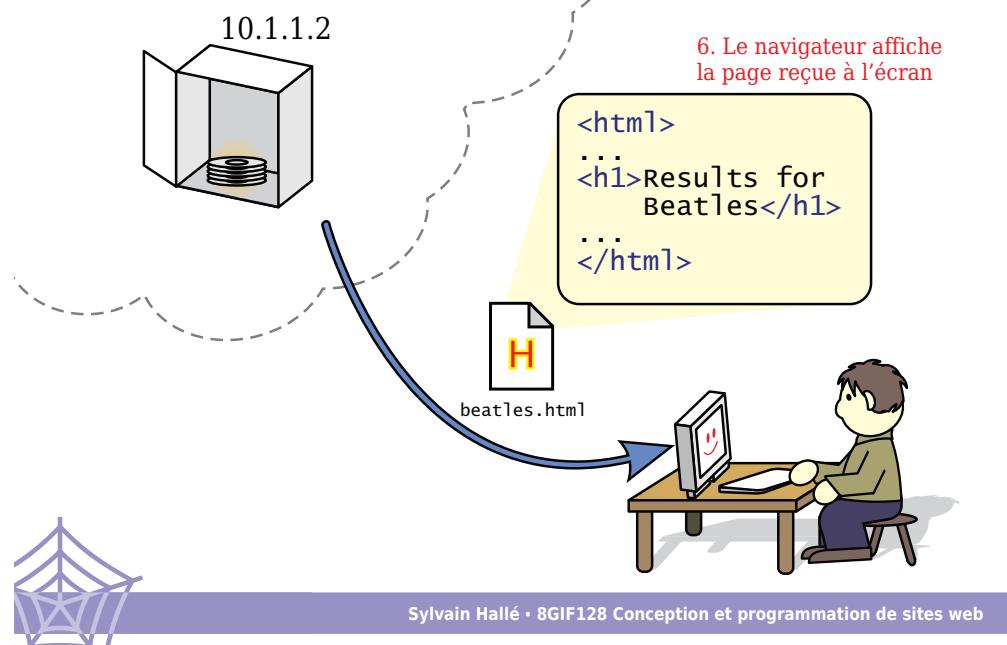
Le chargement d'une page 🕹️



Le chargement d'une page 🕹️



Le chargement d'une page 🕹️



- HyperText Markup Language (HTML): spécifie la structure des pages
- Cascading Stylesheets (CSS): spécifie la présentation des pages
- JavaScript: permet de programmer des pages interactives
- PHP Hypertext Processor (PHP): permet au serveur de créer des pages dynamiques
- Structured Query Language (SQL): interagit avec une base de données
- Asynchronous JavaScript and XML (AJAX): permet de récupérer des documents web en arrière-plan

- Le World Wide Web est une combinaison de protocoles et de technologies basées sur les réseaux:
 - TCP/IP (connexion, adresse, port)
 - DNS (nom de domaine)
 - HTTP (commandes/requêtes)
 - URL
- Serveur web = ordinateur qui écoute les requêtes sur le port TCP 80
- Navigateur = communique avec serveurs pour récupérer et afficher des pages



Pour la semaine prochaine: installez Firefox si vous ne l'avez pas déjà!



Partie 2

I. Le langage HTML: éléments de base
(Duckett, chap. 1)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Écrire une page web



(C) 2010 Gromit

H
T
M
L

C
S
S



(C) 2010 Gromit

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Écrire une page web

Écrire une page web = deux étapes

1. Décrire le **contenu** et la **structure** de l'information

- Écrire le contenu (i.e. le texte)
- Déclarer les éléments du contenu qui constituent un **titre**, un **paragraphe**, une **liste à puces**, un champ de formulaire, etc.

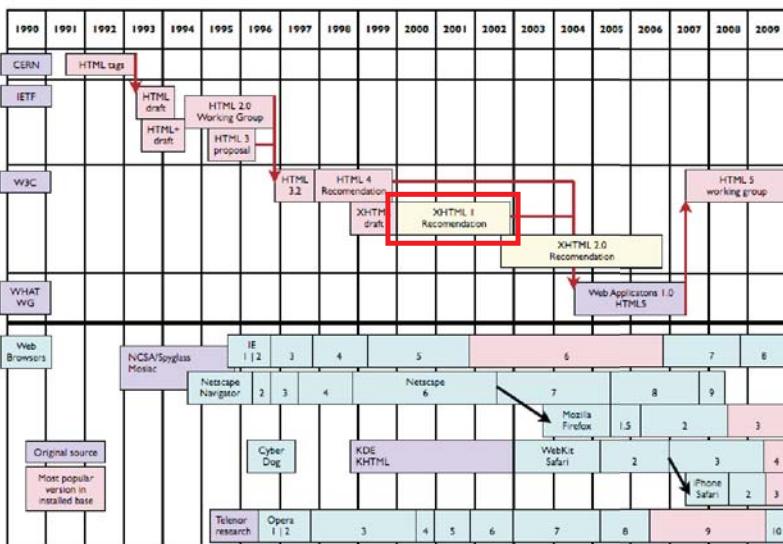
2. Définir la **présentation** de l'information

- Pour les éléments déclarés en 1), préciser les attributs de présentation: la **taille**, la **position**, la **couleur**, la **police de caractères**, l'**image de fond**, etc.



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

eXtended Hypertext Markup Language (XHTML) 🕸️

Dans le cours, on étudie **XHTML 1.0**, qui est une version plus stricte de HTML 4.01. Pourquoi?

- Le langage est plus rigide et plus strict (plus facile à lire pour un programme)
- Supporté par plus de navigateurs
- Plus de chances que nos pages soient affichées correctement dans le futur
- On peut interchanger XHTML avec d'autres formats XML: SVG (pour les graphiques), MathML, MusicML, etc.

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

HTML décrit le contenu et la structure de l'information dans une page web:

- On entoure des portions de texte avec des **tags** ouvrants et fermants
- Tag ouvrant + contenu + tag fermant = **élément**
 - Syntaxe: `<element>contenu</element>`
 - Exemple: `<p>Ceci est un paragraphe</p>`
- Les espaces (tabulations, sauts de ligne, etc.) n'ont pas de signification en HTML
 - Ils sont ignorés, ou considérés comme un seul espace
- Un navigateur applique une **présentation par défaut** à chaque élément... pas nécessairement la même pour chaque navigateur!

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Hypertext Markup Language (HTML) 🕸️

Un premier exemple de page web:

```

<html>
  <head>
    <title>Popular Websites: Google</title>
  </head>
  <body>
    <h1>About Google</h1>
    <p>Google is best known for its search engine, although
       Google now offers a number of other services.</p>
    <p>Google's mission is to organize the world's
       information and make it universally
       accessible and useful.</p>
    <p>Its founders Larry Page and Sergey Brin started
       Google at Stanford University.</p>
  </body>
</html>

```

ch_01_eg01.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un premier exemple de page web:

```
<html>
  <head>
    <title>Popular websites: Google</title>
  </head>
  Élément ↓
  <body>
    <h1>About Google</h1> Tag fermant
    <p>Google is best known for its search engine, although
       Google now offers a number of other services.</p>
    <p>Google's mission is to organize the world's
       information and make it universally
       accessible and useful.</p>
    <p>Its founders Larry Page and Sergey Brin started
       Google at Stanford University.</p>
  </body>
</html>
```



 ch_01_eg01.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un document XHTML possède toujours la structure suivante:

```
<html>
  <head> head: données à propos de la page
         (pas affichées dans le navigateur)
  </head>
  <body>
    body: informations affichées dans le
             navigateur (le « corps » de la page)
  </body>
</html>
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un premier exemple de page web:

```
<html>
  <head>
    <title>Popular websites: Google</title>
  </head>
  Élément ↓
  <body>
    <h1>About Google</h1> Tag ouvrant
    <p>Google is best known for its search engine, although
       Google now offers a number of other services.</p>
    <p>Google's mission is to organize the world's
       information and make it universally
       accessible and useful.</p>
    <p>Its founders Larry Page and Sergey Brin started
       Google at Stanford University.</p>
  </body> Tag fermant
</html>
```



 ch_01_eg01.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Quelques remarques:

- En XHTML, on écrit toujours les noms d'éléments en **minuscules**, i.e. **<body>** et non **<BODY>**
- Les tags sont comme des parenthèses: on doit les ouvrir/fermer correctement

 **<p>Bonjour les amis</p>** incorrect

 **<p>Bonjour les amis</p>** correct



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Certains tags contiennent de l'information additionnelle appelée **attributs**

- Syntaxe:

```
<element attribut1="val" attribut2="val">
    contenu
</element>
```

- Exemple: le tag `<a>` (pour « ancre »)

```
<a href="http://url...>un lien</a>
```

Nom de l'attribut

Valeur de l'attribut

 ch_01_eg02.html


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

XHTML valide 

Un document qui respecte les règles de syntaxe du (X)HTML est un document dit **valide**

La plupart des navigateurs tolèrent des fichiers invalides et essaient de corriger les erreurs (souvent assez bien)

Mais:

- Plus de travail à faire (CPU) avant d'afficher la page
- Deux navigateurs peuvent corriger l'erreur de façon différente
- Aucune garantie même que la page s'affiche
- Aucune garantie que les futurs navigateurs soient aussi tolérants

Règle d'or: **écrivez toujours* du XHTML valide!**

(* Sinon, sachez bien ce que vous faites!)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Quelques remarques:

- En XHTML, les noms d'attributs sont toujours en **minuscules**...
- ...ils sont toujours séparés de leur valeur par le caractère **=**
- ...la valeur est toujours comprise entre deux **guillemets doubles** (même si c'est la chaîne vide!)
- Un élément sans contenu n'a qu'un seul tag, qui se termine par `/>`

Ex.: ``
`<hr />`



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

XHTML valide 

Un premier module complémentaire pour Firefox: HTML Validator: <http://users.skynet.be/mgueury/mozilla/>

Type	Line	Column	HTML errors and warnings
Result	1	1	1 error / 34 warnings
Warning	12	1	missing before <p>
Warning	12	19	inserting implicit
Warning	12	26	discarding unexpected
Warning	23	11	discarding unexpected
Warning	28	5	nested emphasis
Warning	38	3	missing
Error	50	1	 is not approved by ...
Error	64	1	cabc is not recognized!
Warning	64	1	discarding unexpected
Warning	86	3	<form> should not be nested

The errors and warnings are generated by Tidy, originally developed by the W3C. Tidy tries to correct HTML errors by classifying them in 2 categories:
 Critical HTML errors that Tidy cannot fix or understand.
 HTML errors that Tidy can fix automatically.
 Or show the green icon, when there is no HTML errors.



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Éléments de XHTML pour marquer du texte:

- Titres et sous-titres: `<h1>`, `<h2>`, ..., `<h6>`
- Paragraphes `<p>`, sauts de ligne `
`
- Adresses `<address>`
- Éléments de présentation: ``, `<i>`, `<u>`, `<s>`, `<tt>`, `<sup>`, `<sub>`, `<strike>`, `<big>`, `<small>`, `<hr/>`
- Éléments de phrases: ``, ``, `<abbr>`, `<acronym>`, `<blockquote>`, etc.
- Listes: ``, ``, éléments de listes ``, définitions `<dl>`, `<dt>`, `<dd>`
- Édition: `<ins>` et ``

Groupements: `<div>` et ``



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Formatage de base

Pour le texte de base...

`
` introduit un saut de ligne forcé

 ch_01_eg08.html

Attention: ne pas l'utiliser pour positionner du texte!

`<pre>` affiche du texte « pré-formaté »

 ch_01_eg09.html

Les espaces et les sauts de ligne sont affichés tels quels.
Le navigateur utilise généralement une police « mono-space » (i.e. de type **machine à écrire**)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour le texte de base...

`<p>` débute un paragraphe de texte et `</p>` le termine

Rappel: les espaces (sauts de ligne, tabs) multiples sont affichés comme un seul espace par le navigateur

 ch_01_eg03.html

En général, le navigateur insère une nouvelle ligne après un paragraphe

 ch_01_eg07.html

`<h1>` débute un titre de niveau 1 et `</h1>` le termine

`<hn>` débute un sous-titre de niveau *n* et `</hn>` le termine

 ch_01_eg05.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Éléments de présentation

Mise en forme du texte:

<code>...</code>	Gras
<code><i>...</i></code>	Italiques
<code><u>...</u></code>	Souligné
<code><s>...</s></code>	Biffé
<code><tt>...</tt></code>	Dactylographié

 ch_01_eg10.html

<code><sup>...</sup></code>	En exposant
<code><sub>...</sub></code>	En indice
<code><big>...</big></code>	Plus gros
<code><small>...</small></code>	Plus petit
<code><hr/></code>	Ligne horizontale

 ch_01_eg11.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Éléments qui décrivent une propriété de leur contenu:

<code>...</code>	Accent sur un mot	 ch_01_eg12.html
<code>...</code>	Accent fort	
<code><address>...</code>	Adresse	 ch_01_eg13.html
<code><abbr>...</code>	Abréviation	
<code><acronym>...</code>	Acronyme	 ch_01_eg14.html
<code><dfn>...</code>	Définition	
<code><blockquote>...</code>	Longue citation	
<code><q>...</code>	Courte citation	 ch_01_eg15.html
<code><cite></code>	Référence (livre)	



XHTML définit trois types de listes:

Liste **ordonnée**: ``

- 1. Le premier élément de la liste justifié à gauche avec le numéro
- 2. Le deuxième élément
- 3. Le troisième élément avec du texte plus long

 ch_01_eg18.html

Liste **à puces**: ``

- Le premier élément de la liste justifié à gauche avec une puces
- Le deuxième élément
- Le troisième élément

 ch_01_eg17.html

Liste de **définitions**: `<dl>`

- | | |
|--|--|
| Nom du mot | |
| Définition du mot à la suite | |
| Nom du deuxième mot | |
| Définition de ce deuxième mot avec texte plus long | |

 ch_01_eg19.html

Éléments qui décrivent du texte « informatique »:

<code><code>...</code>	Afficher du code
<code><kbd>...</code>	Saisie au clavier
<code><var>...</code>	Nom de variable
<code><samp>...</code>	Output

 ch_01_eg16.html

Attention!

Les caractères `<` et `>` sont traités par HTML comme des délimiteurs de tags.

`if (x < 10) {` Début d'un tag nommé 10 `?`

Pour afficher `<`, on doit écrire `<`

Pour afficher `>`, on doit écrire `>`



Quelques remarques:

- Dans une liste ordonnée, on peut changer le type de numérotation avec l'attribut **type*** (`1`, `A`, `a`, `I`, `i`)
- On peut changer le numéro de départ avec l'attribut **start** (toujours un nombre peu importe le type)
- On peut imbriquer plusieurs listes, i.e. l'élément `` peut lui-même contenir une liste
 ch_01_eg20.html
- Erreur courante: oublier de fermer la liste (``, ``) ou un item de la liste (``): ce qui suit restera indenté



Erreur courante #2: fermer un item de liste trop tôt

```
<ul>
<li>Simpsons:</li>
  <ul>
    <li>Bart</li>
    <li>Lisa</li>
  </ul>
</li>
<li>Family Guy:
  <ul>
    <li>Peter</li>
    <li>Lois</li>
  </ul>
</ul>
```

La plupart des navigateurs l'afficheront correctement, mais le code XHTML est **invalidé** (voir remarque antérieure!)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Édition de texte

Éléments utilisés lors de révisions successives d'un document (ex.: le mode révision de Microsoft Word)

`<ins datetime="xxx">texte</ins>`

Indique que le texte a été inséré à la date xxx

`<del datetime="xxx">texte`

Indique que le texte a été retiré à la date xxx

Format de la date:

`aaaa-mm-jjThh:mm:ssTZD`



 ch_01_eg21.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Liste de définition: `<dl>, <dt>, <dd>`

`<dl>` représente une liste de définitions de termes
`<dt>` représente chaque terme
`<dd>` représente la définition du terme

```
<dl>
  <dt>newbie</dt>
  <dd>one who does not have mad
       skills</dd>
  <dt>own</dt>
  <dd>to soundly defeat
       (e.g. "I owned that
       newbie!")</dd>
  <dt>frag</dt>
  <dd>a kill in a shooting
       game</dd>
</dl>
```

 ch_01_eg19.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Entités HTML

Certains caractères sont **réservés** par le langage HTML et ne peuvent être affichés tels quels. Pour les afficher dans la page, on doit les remplacer par des séquences spéciales, appelées **entités HTML**.

<u>Pour afficher</u>	<u>Écrire</u>
<	<
>	>
&	&
"	"

Certains caractères spéciaux (accents, etc.) peuvent aussi être affichés au moyen d'entités, par exemple:



`Ñ`
`&ucirc`

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On peut insérer des **commentaires** dans un document HTML:

- Un commentaire débute par <!--
- Il se termine par -->
- Ce qu'il y a entre les deux est ignoré par le navigateur

```
<!-- Start of footnotes section added 04-24-04
by Bob Stewart -->

<!-- <h2>Character entities</h2> -->
<p><strong>Character entities</strong> can be used
to escape special...</p>

<!-- End of footnotes -->
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Éléments de niveau bloc vs. niveau texte

Quelques remarques:

- Chaque élément est soit de niveau bloc ou de niveau texte
- Un élément de niveau texte doit être contenu dans un bloc
- Un élément de niveau texte ne peut contenir d'élément de niveau bloc
- Pour qu'un document soit considéré valide, ces règles doivent être suivies (i.e. un validateur relèvera ces erreurs)!



 ch_01_eg22.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Éléments de niveau bloc (*block-level*)

Contiennent une grande région de texte complet

Exemple: paragraphes, listes, cellules de tableau

Le navigateur place une marge entre les éléments bloc pour les séparer (commencent sur leur propre ligne)

Éléments de niveau texte (*inline*)

Affectent une petite partie de texte et doivent être imbriqués dans un élément de niveau bloc

Exemple: texte gras, fragments de code, images

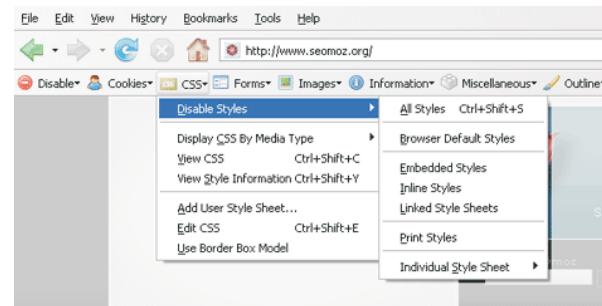
Le navigateur permet à plusieurs éléments de niveau texte d'apparaître sur une même ligne ou dans le même bloc



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

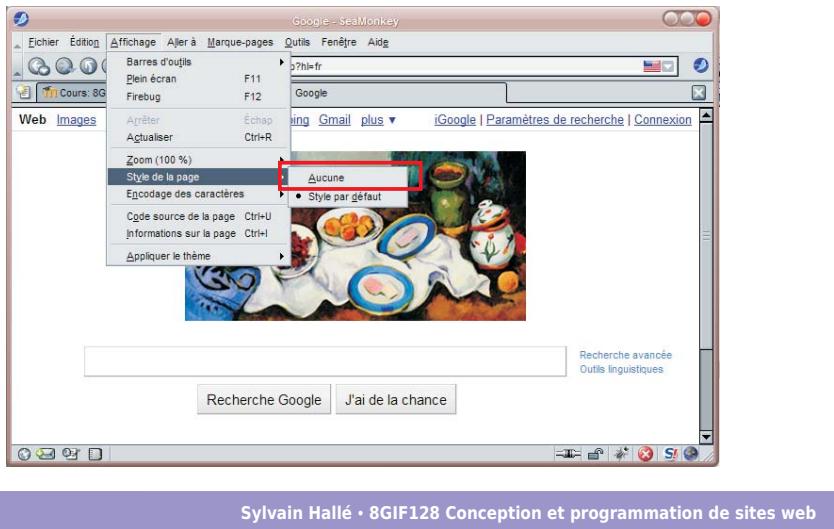
Web Developer Toolbar

Un autre module complémentaire pour Firefox: [Web Developer Toolbar](#)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On peut aussi **désactiver les styles** d'une page et forcer le navigateur à utiliser la présentation par défaut de chaque élément (dans Firefox et Opera; IE?)



DOCTYPE

XHTML est un sous-ensemble de XML. Un document XHTML peut inclure cette déclaration au tout début (rien d'autre avant!):

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Utilité principale: spécifier l'encodage des caractères

Il existe plusieurs versions de HTML et de XHTML. Chaque document bien formé doit inclure une déclaration de sa version avec l'élément DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.0 strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



Deux tags permettent de **grouper** des éléments:

<div>...</div>

...

Groupement de niveau bloc

Groupement de niveau texte

Exemple: Duckett p. 42

Ces éléments ne changent **rien** à la présentation par défaut des autres éléments (essayez)! À quoi servent-ils?

1. Ajouter une signification supplémentaire à des portions du document (ex.: délimite les notes de bas de page, ou les noms d'auteurs, etc.)
2. CSS peut appliquer une présentation particulière aux groupes définis par ces éléments



Encodage des caractères

American Standard Code for Information Interchange: **ASCII** (Duckett, annexe E)

Associe à chaque nombre de 0 à 127 un caractère de l'alphabet latin ou un code de contrôle

Un fichier texte = une suite de ces nombres, interprétés comme des lettres

< h t m l >

60 104 116 109 108 62



Un octet (8 bits) pour chaque caractère

0 NUL	32 espace	64 @	96 `
1 SOH	33 ¡	65 A	97 a
2 STX	34 ¢	66 B	98 b
3 ETX	35 §	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 *	71 G	103 g
8 BS	40 (72 H	104 h
9 HT	41)	73 I	105 i
10 LF	42 *	74 J	106 j
11 UT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 CS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EH	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ,	91 [123 {
28 FS	60 <	92 \	124 }
29 GS	61 =	93]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■

Que fait-on des accents, autres symboles, etc.?

Encodage de caractères

Association (table) entre des nombres (128 et plus) et des symboles dits "étendus"

- Dans les encodages, les caractères 0-127 restent les mêmes



Avantages d'UTF-8:

- 50% des pages sur le web sont maintenant encodées en UTF-8
- Les navigateurs le supportent depuis déjà près de 10 ans (peu de problèmes de compatibilité)
- Identique à ASCII si on utilise seulement les caractères 0-127

Désavantages:

- Les symboles ont une longueur variable (peu important pour nous)
- Peut prendre (**un peu**) plus d'espace que latin-1 pour le même texte



Nom de quelques encodages courants:

latin-1 / ISO 8859-1: caractères spéciaux des langues occidentales (français, allemand, ...), sur 1 octet

Ex.: À : 192, ç : 231, é : 233, etc.

cp1252 / windows-1252: idem, mais avec des différences!

Ex.: À : 192, ç : 231, é : 130, etc.

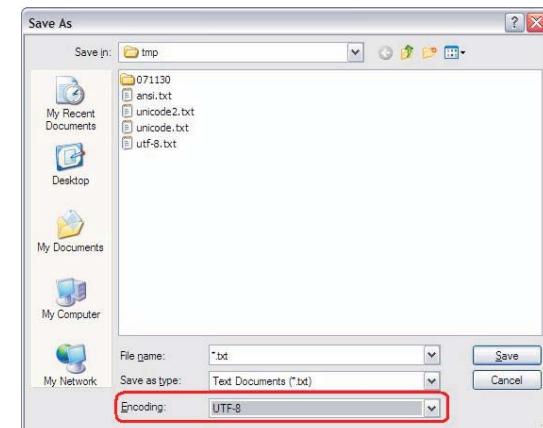
UTF-8: encodage de toutes les langues; chaque caractère peut utiliser de 1 à 4 octets

Ex.: é : les deux octets 195 169



⚠ Attention!

UTF-8 n'est pas l'encodage par défaut dans tous les éditeurs texte, en particulier sous Windows (cp-1252)!



⚠ Attention!

Préparez-vous à des surprises si...

1. L'encodage du fichier ne correspond pas à celui que vous déclarez
2. L'encodage du fichier n'est pas spécifié, et le navigateur en utilise un autre pour l'afficher

Quelques horreurs:

- Fichier UTF-8 sans **encoding** (le navigateur assume latin-1)
- Fichier latin-1 déclaré comme UTF-8



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Encodage des caractères 

En HTML, on peut convertir tous les caractères accentués en utilisant les **entités**:

é	devient	é
û	devient	û
À	devient	À

...et ainsi de suite!

Ce faisant:

- Le fichier HTML n'utilise plus que les caractères ASCII --qui sont identiques peu importe l'encodage
- C'est la solution « sécuritaire » (fonctionne toujours)
- Mais le fichier source est `légèrement dur à lire`



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Des sites **sérieux** éprouvent encore de **sérieux** problèmes à gérer les encodages correctement:

- Comparer "Sylvain Hallé", "Sylvain Halli" et "Sylvain Hallo" sur Google

The screenshot shows a list of notifications from Academia.edu. At the top, it says "Notifications 10". Below that, there are four items:

- 40 new page views since notifications were last cleared (3 months ago). [View Details](#) 16 days ago
- Someone searched for you on Google. [View Details](#) runtime... 15 days ago
- Someone searched for you on Google. [View Details](#) "valdmaker"... 30 days ago
- Someone searched for you on Google. [View Details](#) sylvain Hallé about 1 month ago

The fourth item, "Someone searched for you on Google. sylvain Hallé", has a red circle around the "Hallé" part of the search term.

The screenshot shows a list of notifications from Academia.edu. At the top, it says "Notifications 10". Below that, there are four items:

- 40 new page views since notifications were last cleared (3 months ago). [View Details](#) 16 days ago
- Someone searched for you on Google. [View Details](#) runtime... 15 days ago
- Someone searched for you on Google. [View Details](#) "valdmaker"... 30 days ago
- Someone searched for you on Google. [View Details](#) sylvain Hallé about 1 month ago

The fourth item, "Someone searched for you on Google. sylvain Hallé", has a red circle around the "Hallé" part of the search term.

Encodage des caractères 

Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Le langage HTML

II. Ajouter des liens
(Duckett, chap. 2)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Liens entre les pages

Règle de base pour les liens:

- Le contenu de l'élément `<a>` doit donner une **description claire et concise** de ce vers quoi pointe le lien
- Les moteurs de recherche utilisent le contenu de l'élément pour indexer les pages
- Certains logiciels (ex.: *screen readers*) dépendent du texte pour bien fonctionner (cf. dans Firefox)
- Éviter à tout prix les *ici*, *cliquez ici*, etc.
Mauvaise pratique dénoncée depuis au moins 10 ans!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour lier une page à une autre, on utilise le tag `<a>`. Exemple simple:

```
<body>
<p>Return to the
<a href="index.html">home page</a>.
</p>
</body>
```

ch_02_eg01.html

On donne un URL complet (`http://...`) à l'attribut `href` pour faire un lien à une page sur un autre serveur (lien **externe**)

ch_02_eg02.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Liens entre les pages

On peut ajouter une description à un lien avec l'attribut `title`

```
<a href="http://www.google.com/" title="Search the Web with Google">Google</a>
```

Un lien est affiché par défaut en souligné par le navigateur...

Erreur de débutant: laisser de l'espace avant ou après le texte du lien (l'espace sera souligné!)

ch_02_eg04.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Rappel: format d'un URL

`http://www.example.com/artistes/beatles.html`



Il y a d'autres protocoles que `http`

`gopher://` Lien vers un site Gopher

`ftp://` Lien vers un site FTP

`mailto:` Pour envoyer un courriel

`Envoyez-moi un courriel`



Q: Que se passe-t-il lorsqu'on tape

`http://www.google.ca`

Il n'y a pas de nom de fichier!

R: Le **serveur web** effectue les actions suivantes:

1. Cherche `index.html` (ou `index.php` ou `index.asp`) dans le répertoire
2. Si rien n'est trouvé, propose une liste de fichiers du répertoire
3. Sinon, renvoie un message indiquant que la page demandée est introuvable (404 Not Found)



On peut pré-remplir certains champs du courriel à l'avance dans le lien (p. 76):

`mailto:info@example.org?subject=Inquiry`

`mailto:info@example.org?subject=XHTML&cc=sales@example.org`

Attention!

1. S'il y a des espaces, on doit les écrire en `%20`
2. Si on veut sauter une ligne, on doit écrire `%0D%0A`
3. Le `&` doit lui-même être remplacé par `&`



Q: Pourquoi les liens des exemples précédents, comme

`index.html`

fonctionnent? Il n'y a pas de protocole et pas de serveur!

R: Le **navigateur** compose un URL complet à partir de l'URL de la page courante (p. 63-65)

On peut monter et descendre dans les répertoires du serveur en utilisant .. et /

On peut spécifier au navigateur la racine commune aux liens d'un document:

`<base href="http://www.examplesite2.com/" />`



Tous les liens vus jusqu'ici sont des liens **source**.

Un lien peut pointer vers une portion spécifique d'une page, par exemple:

```
<a href="mapage.html#section1">...</a>
    ↑
    Pointe vers la
    section1 de mapage.html
```

On déclare section1 en utilisant un lien **destination** dans `mapage.html`:

```
<a id="#section1"></a>
```



ch_02_eg06.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Comment structurer ses pages

L'étiquette d'un lien doit décrire clairement la cible du lien...
mais l'**URL lui-même doit être clair**.

La **structure** des pages d'un site (et le nom des documents) ne doit pas être choisie à la légère!

Exemple: site web de musique en ligne avec une page pour chaque artiste.

```
/index.html
page_01.html
page_02.html
page_03.html
...
...
```

URL pour Pascale Picard:
http://monsite.com/page_02.html

Quel est l'URL pour Pierre Lapointe?

Quel est l'URL de la liste des artistes?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

L'élément `<a>` peut avoir d'autres attributs (moins utilisés):

<code>accesskey</code>	Touche de raccourci clavier
<code>charset</code>	Encodage des caractères du fichier
<code>coords</code>	Position dans un <i>image map</i>
<code>hreflang</code>	Langue du fichier
<code>rel</code>	Relation avec la source
<code>rev</code>	Relation avec la cible (inutilisé)
<code>shape</code>	Forme du lien (lorsqu'il est une image)
<code>tabindex</code>	Ordre de visite des liens via la touche tab
<code>target</code>	Ouvre un lien dans une nouvelle fenêtre
<code>title</code>	Affiche un texte explicatif
<code>type</code>	Annonce le type de fichier du lien (inutilisé)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Comment structurer ses pages

```
/
index.html
artistes/
index.html
pascale_picard.html
eric_lapointe.html
...
...
```

- URL pour Pascale Picard:
http://monsite.com/artistes/pascale_picard.html
- Quel est l'URL pour Pierre Lapointe?
- Quel est l'URL de la liste des artistes?
- Quel est l'URL de la liste des spectacles de Pascale Picard? ...de Pierre Lapointe?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

/
index.html
artistes/
pascale_picard/
index.html
spectacles.html
...

URL pour Pascale Picard:
http://monsite.com/artistes/pascale_picard/

 Quel est l'URL de la liste des spectacles de 2009 de Pascale Picard?

/
index.html
artistes/
pascale_picard/
index.html
spectacles/
index.html
2009.html

URL pour Pascale Picard:
http://monsite.com/artistes/pascale_picard/

 Quel est l'URL de la liste des spectacles de 2009 de Pascale Picard?

Truc: si on sépare plusieurs éléments d'information dans un nom de fichier, comme:

artiste_pascale_picard_spectacles_2009_montreal.html
artiste_pierre_lapointe_spectacles_2010_france.html

...en général cette info devrait être représentée sous forme de hiérarchie de « répertoires »:

/artistes/pascale_picard/spectacles/2009/montreal.html



Les répertoires et noms de pages de l'URL doivent avoir un sens.

Liste des succursales sur le site de Desjardins:
http://www.desjardins.com/fr/votre_caisse/caisses-par-region.jsp

Liste des succursales sur le site de la Banque Nationale:
http://www.bnc.ca/bnc/cda/index/0,4229,divId-2_langId-2_navCode1000,00.html#

Autre exemple:

Liste des profs du département d'info de l'U. Laval:
<http://www.ift.ulaval.ca/departement/professeurs/>

...de l'UQAC:
http://wprod1.uqac.ca/programmes/personnel_dept.html?type=prof&noDept=8&menu=dim



/
index.html
artistes/
pascale_picard/
index.html
spectacles/
index.html
2009.html

URL pour Pascale Picard:
http://monsite.com/artistes/pascale_picard/

 Quel est l'URL de la liste des spectacles de 2009 de Pascale Picard?

Truc: si on sépare plusieurs éléments d'information dans un nom de fichier, comme:

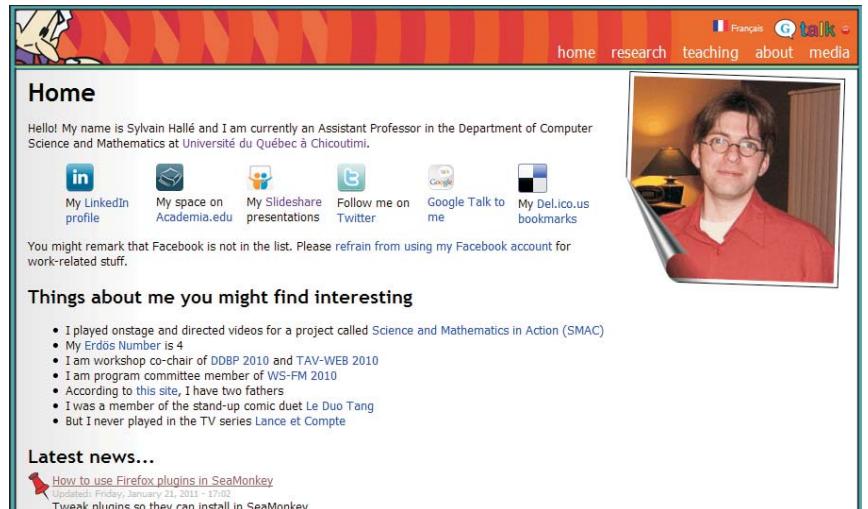
artiste_pascale_picard_spectacles_2009_montreal.html
artiste_pierre_lapointe_spectacles_2010_france.html

...en général cette info devrait être représentée sous forme de hiérarchie de « répertoires »:

/artistes/pascale_picard/spectacles/2009/montreal.html



Dernier exemple: la page web de votre professeur



Home

Hello! My name is Sylvain Hallé and I am currently an Assistant Professor in the Department of Computer Science and Mathematics at Université du Québec à Chicoutimi.

French 

home research teaching about media

 My LinkedIn profile  My space on Academia.edu  My Slideshare presentations  Follow me on Twitter  Google Talk to me  My Del.icio.us bookmarks

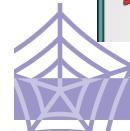
You might remark that Facebook is not in the list. Please refrain from using my Facebook account for work-related stuff.

Things about me you might find interesting

- I played onstage and directed videos for a project called Science and Mathematics in Action (SMAC)
- My Erdős Number is 4
- I am workshop co-chair of DDBP 2010 and TAV-WEB 2010
- I am program committee member of WS-FM 2010
- According to this site, I have two fathers
- I was a member of the stand-up comic duet Le Duo Tang
- But I never played in the TV series Lance et Compte

Latest news...

 How to use Firefox plugins in SeaMonkey
Tuesday, January 21, 2014 - 17:02
Tweak plugins so they can install in SeaMonkey
Read more...



Une fois un URL créé, il ne disparaîtra peut-être **jamais**: il reste toujours quelqu'un qui pointe dessus.

Morale: les bons URL **ne changent pas**



Exemple personnel:

Sur 81 universités canadiennes, **10** d'entre elles avaient changé l'URL de leur service des ressources humaines entre mai et novembre 2008...

...l'ancien lien menait à une page 404 Not Found



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Championne toutes catégories: l'Université Ste-Anne (Nouveau-Brunswick):

Mai 2008

<http://www.usainteanne.ca/emplois/combler.php>

Novembre 2008

<http://www.usainteanne.ca/emplois/emplois.php>

Octobre 2009

http://www.usainteanne.ca/index.php?option=com_content&view=article&id=62&Itemid=115

Janvier 2011

<http://www.usainteanne.ca/offres-dememploi/offres-dememploi>



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Championne toutes catégories: l'Université Ste-Anne (Nouveau-Brunswick):

Mai 2008

<http://www.usainteanne.ca/emplois/combler.php>

Novembre 2008

<http://www.usainteanne.ca/emplois/emplois.php>

Octobre 2009

http://www.usainteanne.ca/index.php?option=com_content&view=article&id=62&Itemid=115

Janvier 2011

<http://www.usainteanne.ca/offres-dememploi/offres-dememploi>



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Visitez des sites en désactivant les styles, ou en utilisant le Web Developer Toolbar pour encadrer les éléments des pages.

Notez le site qui fonctionne **le mieux** et le site le plus **illisible** en naviguant de cette façon. Partage mercredi prochain!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Le langage HTML

IV. Les images
(Duckett chap. 3)



Insérer des images

Attributs de l'élément img (cf. Duckett, pp. 80-ss)

src	URL de l'image
alt	Description textuelle (courte)
height	Hauteur de l'image (en pixels)
width	Largeur de l'image (en pixels)
align	Aligne l'image à gauche/droite
border	Définit une bordure autour de l'image
hspace	Espacement horizontal au texte qui entoure
vspace	Espacement vertical au texte qui entoure
longdesc	Description textuelle (longue)*



L'élément `` sert à ajouter des images dans un document, à travers le texte:

```

```



- L'attribut `src` spécifie l'URL de l'image
- XHTML **exige** l'attribut `alt`, dont la valeur est une description textuelle de l'image



Insérer des images

On peut faire d'une image un lien en l'encadrant d'un élément `<a>`:

ch03_eg06.html

Attention!

Si on veut simplement faire une icône avec un lien, il existe une meilleure manière de procéder (HTML + CSS) sans utiliser l'élément ``. On verra plus loin.

En général, réserver `` pour des photos, illustrations, mais PAS pour des éléments d'interface graphique.



Les navigateurs supportent tous trois formats d'images: JPEG, GIF, PNG. Les images peuvent aussi se retrouver dans deux autres formats courants: TIFF et BMP.

Chacun représente les images comme un tableau rectangulaire de **pixels**, et associe à chaque pixel une **couleur**.

Les formats diffèrent dans:

1. Le nombre de **couleurs** différentes pour les pixels (la *palette*)
2. La manière dont les données sont **compressées**
3. La présence d'information sur la **transparence** des pixels (*alpha channel*)



Pour chaque composante:

0 = absence de cette couleur

255 = intensité maximale de cette couleur

Exemples:

R	G	B	
255	0	0	
0	0	128	
255	255	0	
64	64	64	
128	128	128	
255	255	255	
239	155	160	



On définit une couleur par trois composantes (i.e. trois nombres):

R	G	B
Quantité de rouge	Quantité de vert	Quantité de bleu

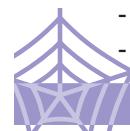
Selon le nombre de bits utilisé pour chaque composante, on obtient un spectre de couleurs différentes (*bit depth*)

On utilise souvent **8 bits par composante**:

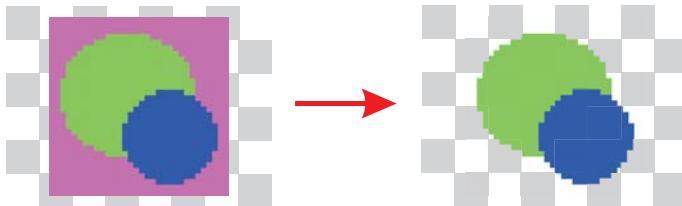
$$\begin{aligned} &= 24 \text{ bits par couleur} \\ &= 2^{24} = \text{environ 16 millions de couleurs possibles} \end{aligned}$$



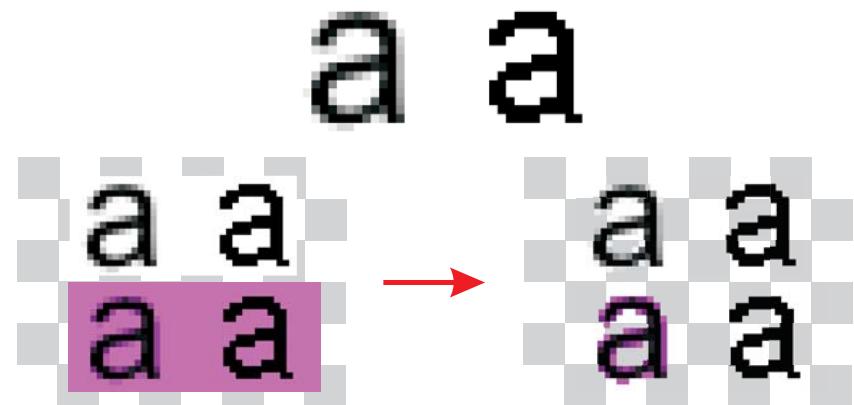
1. Palette: peut utiliser de 1 à 8 bits pour la couleur de chaque pixel
 - La palette contient de 2 à 256 couleurs
 - Chaque couleur de la palette est définie par 24 bits
 - Souvent beaucoup plus de couleur dans l'original: pertes
 - Cf. Figure 3-11
 - On peut choisir « intelligemment » les couleurs de la palette pour minimiser l'effet
2. Compression: utilise l'algorithme LZW (variante de Zip)
 - Zipper un GIF ne donne pas grand chose
 - LZW est une compression sans perte (lossless): restitue la même image, pixel pour pixel, que l'original*
 - Re-sauvegarder un GIF ne diminue pas sa qualité*
 - Peu efficace sur des images « photo-réalistes »



3. Transparence: on peut définir une couleur de la palette comme transparente
 - Tous les pixels de cette couleur dans l'image ne seront pas affichés, laissant paraître l'arrière-plan de l'image



Attention à l'*anti-aliasing*!



4. En bonus: GIF est le seul format d'image permettant de créer des **animations**



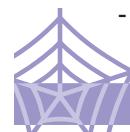
PNG = comme GIF à quelques exceptions...

1. Palette: peut avoir une palette de 256 couleurs comme GIF, ou donner à chaque pixel un nombre de 24 bits
2. Compression: LZW (sans pertes) comme GIF
3. Transparence: chaque pixel peut avoir une transparence entre 0 (opaque) et 255 (complètement transparent)
4. Pas d'animations en PNG



Format conçu pour les photographies et les images « photoréalistes »

1. Palette: pas de palette; chaque pixel a une couleur sur 24 bits (ou 8 bits si l'image est en noir et blanc)
2. Compression: avec **pertes**. Une image compressée en JPEG n'est pas identique, pixel pour pixel, à l'original.
 - JPEG introduit (volontairement) des « erreurs » dans l'image
 - On peut régler le niveau de fidélité de la compression de 100 (presque parfait) à 0 (affreux)
 - Plus l'image est compressée, plus elle contient d'erreurs et moins elle ressemble à l'original
 - On appelle ces erreurs **artéfacts de compression**



- Les artefacts de compression sont conçus pour être moins apparents à l'oeil lorsque l'image est **photo-réaliste**...
- ...mais beaucoup plus visibles sur du **texte**, des couleurs en aplat ou des **détails** à contraste élevé



not be inferred from established approach ts generalizations as McGill's interaction



Conséquence: compresser successivement une même image en JPEG **diminue sa qualité à chaque fois** (même si on garde le même paramètre de qualité!)



Morale: conserver l'original lors d'une sauvegarde en JPEG

3. Transparence: pas de transparence avec JPEG
4. Animations: pas d'animations non plus



La taille des fichiers

1. La résolution d'un écran est plus faible que celle d'une caméra ou d'un scanner

Exemple:

1600x1200 (écran 17")

2592x1944 (mon appareil photo)

2. Transférer une photo de 3 Mo (en 2011) est long
3. L'hébergement est chargé en fonction de la bande passante consommée

Conclusion: on doit redimensionner les photos pour en réduire la taille (en pixels et en kilo-octets).

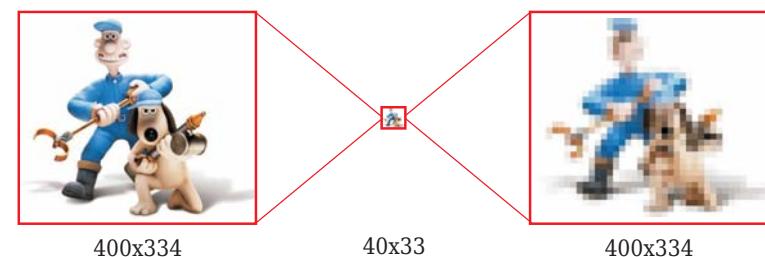


La taille des fichiers

Attention!

Une image qui a été réduite ne peut être agrandie à sa taille originale sans perte de qualité

Exemple extrême:



Morale: conserver l'original!



On utilise l'élément <object> pour insérer du Flash et des vidéos.

Voir Duckett, pp. 104-126





Les styles CSS

(Duckett, chap. 7-8)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Ajouter des styles

La mauvaise façon:

```
<p><font face="Arial">welcome to Greasy Joe's. You will
<b>never, <i>ever, <u>EVER</u></i></b> beat <font
size="+1" color="red">OUR</font> prices!</font></p>
```

Welcome to Greasy Joe's. You will
never, ever, EVER beat **OUR** prices!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Écrire une page web = deux étapes

1. Décrire le **contenu** et la **structure** de l'information

- Écrire le contenu (i.e. le texte)
- Déclarer les éléments du contenu qui constituent un **titre**, un **paragraphe**, une **liste à puces**, un champ de **formulaire**, etc.

2. Définir la **présentation** de l'information

- Pour les éléments déclarés en 1), préciser les attributs de présentation: la **taille**, la **position**, la **couleur**, la **police de caractères**, l'**image de fond**, etc.



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Règles CSS

Une feuille de style (Cascading Stylesheet ou CSS) est une série de **règles** définissant la présentation d'éléments d'un document:

```
sélecteur {
    propriété: valeur;
    propriété: valeur;
    ...
    propriété: valeur;
}
```

Déclaration

Exemple:

```
h1, h2, h3 {
    font-weight: bold;
    font-family: arial;
    color: #000000;
    background-color: #ffffff;
}
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Appliquons des styles au document suivant (Duckett p. 245):



- Le sélecteur spécifie le ou les éléments du document auxquels s'appliquent les déclarations de style
- L'attribut **class** permet d'associer des règles CSS aux éléments en fonction de sa valeur



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Héritage 

La règle

```
body {
    color: #000000;
    background-color: #ffffff;
    font-family: arial, verdana,
                sans-serif;
}
```

s'applique à l'élément **body** et à **tous les éléments qu'il contient**: c'est l'**héritage**. La règle

```
td.code {
    font-family:courier, courier-new, serif;
    font-weight: bold;
}
```

supplante la règle **body** pour les éléments **td.code**, car elle est plus spécifique.



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On étudie maintenant:

1. Les **propriétés** permettant de contrôler l'apparence des éléments et leurs **valeurs** possibles
2. Les **sélecteurs** permettant de spécifier à quels éléments les propriétés s'appliquent (il y en a plusieurs)
3. Le **modèle de boîtes** (*box model*) de CSS et la manière dont les éléments sont affichés et positionnés



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Les trois emplacements du CSS 

On peut spécifier des styles à trois endroits (Duckett pp. 250-253):

1. Dans un **fichier séparé**, référé en utilisant l'élément **<link>** dans la section **<head>** du document HTML
2. Dans un **élément <style>**, aussi dans la section **<head>**
3. Directement dans le tag ouvrant d'un élément, avec l'**attribut style**

On préfère la solution #1!

#3 est interdit dans XHTML 1.0 Strict



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Il existe une panoplie de propriétés CSS: cf. Duckett p. 254-255!

Pour la police de caractères:

<code>font-family</code>	Famille de polices
<code>font-size</code>	Taille de la police
<code>font-weight</code>	Gras ou non
<code>font-style</code>	Normal, <i>italique</i> ou <i>oblique</i>
<code>font-stretch</code>	Police étirée ou condensée
<code>font-variant</code>	Normal ou PETITES MAJUSCULES
<code>font-size-adjust</code>	Rapport d'aspect de la police
<code>font</code>	Tous ces attributs ensemble

```
strong {  
    font-weight: normal; color: red; }  
em {  
    font-style: normal;  
    background-color: #DDDDDD; }
```

Ces deux éléments n'ont pas besoin d'être affichés en gras ou en italiques:

- Décrivent des **attributs** du contenu (ceci est important)
- Alors que **b** et **i** décrivent la **présentation** (ceci est gras)



Pour le formatage du texte (p. 264):

<code>color</code>	Couleur du texte
<code>text-align</code>	Alignement horizontal
<code>vertical-align</code>	Alignement vertical dans l'élément
<code>text-decoration</code>	Souligné, biffé, etc.
<code>text-indent</code>	Alinéa au début d'un bloc
<code>text-transform</code>	Majuscules, minuscules
<code>text-shadow</code>	Avec ou sans ombre
<code>letter-spacing</code>	Espace entre les lettres
<code>word-spacing</code>	Espace entre les mots
<code>white-space</code>	Contrôle des espaces
<code>direction</code>	G-D ou D-G



Certaines portions d'un bloc de texte font partie d'une classe spéciale, sans qu'on doive les déclarer comme telles: ce sont des **pseudo-classes** de texte (p. 273)

<code>first-letter</code>	La première lettre d'un élément
<code>first-line</code>	La première ligne



Sélecteur universel

`*{}`

Sélecteur de type

`h1, h2, h3 {}`

Sélecteur de classe

`<p class="maclasse">Bonjour</p>``.maclasse {} ou p.maclasse {}` Tous les maclasse

Juste les p

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un même élément peut appartenir à plusieurs classes:

`<p class="classe1 classe2">Bonjour</p>``p.classe1 { Toutes les classe1 }``p.classe2 { Toutes les classe2 }``p.classe1.classe2 { Les deux ensemble }`

Sélecteur par ID

`<p id="monid">Bonjour</p>``p#monid {} ou #monid`

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Sélecteurs contextuels 

Sélecteur par enfant immédiat

`<td>Une cellule importante</td>``td>b { Tous les b enfants directs d'un td }`Attention: pourquoi `strong>p` n'a pas de sens?

Sélecteur par descendant

`table b { Tous les b à l'intérieur d'une table }`

Sélecteur par frère immédiat

`h1+p { Tous les p qui suivent directement un h1 }`

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Sélecteur par frère général

`h1~p { Tous les p qui suivent un h1 } (CSS3 seulement!)`Astuce: utiliser les sélecteurs au maximum pour éviter la prolifération de `class`

On peut aussi choisir des éléments en fonction de la valeur d'un de leurs attributs: seulement dans les navigateurs récents (IE7, FF2), cf. Duckett p. 281



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Plusieurs propriétés dans une règle CSS peuvent être héritées de l'élément parent, mais certaines ne le sont pas; par exemple:

border
margin
padding

Pour savoir si une propriété est sujette à l'héritage:

1. Essayer dans une page (préférable)
2. Consulter la référence de CSS



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Résolution de conflits

Que fait-on si plusieurs sélecteurs s'appliquent au même attribut? Des règles de précédence s'appliquent.

1. Une règle plus spécifique supplante une règle générale (ex.: body vs. td)
2. Avec deux règles aussi spécifiques, la dernière lue supplante la précédente
3. Un navigateur lit dans l'ordre:
 - a) les CSS liés
 - b) les règles dans la section style
 - c) les styles en ligne
 - d) applique ses styles par défaut au reste

Une cascade

Morale: **expérimenez** pour comprendre comment tout s'applique!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Les liens [<a>](#) possèdent quatre pseudo-classes spécifiques (Duckett p. 308):

link
visited
hover
active



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Spécificité d'un sélecteur

La spécificité d'un sélecteur est un nombre à 4 chiffres de la forme **abcd**

- On compte 1 si le style est appliqué via l'attribut style, et 0 sinon; ceci est la variable **a**
- On compte les attributs **id** dans le sélecteur; la somme est la variable **b**
- On compte les attributs, pseudo-classes et classes dans le sélecteur; la somme est la variable **c**
- On compte les noms d'éléments dans le sélecteur (**d**)
- Une règle avec **!important** passe avant tout



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Selector	Selector Type	Specificity
*	Universal Selector	0000 (a = 0, b = 0, c = 0, d = 0)
li	Element Name	0001 (a = 0, b = 0, c = 0, d = 1)
ul li	Element Name	0002 (a = 0, b = 0, c = 0, d = 2)
div h1 + p	Element Name	0003 (a = 0, b = 0, c = 0, d = 3)
input[type='text']	Element Name + Attribute	0011 (a = 0, b = 0, c = 1, d = 1)
.someclass	Class Name	0010 (a = 0, b = 0, c = 1, d = 0)
div.someclass	Element Name + Class Name	0011 (a = 0, b = 0, c = 1, d = 1)
div.someclass.someother	Element Name + Class Name + Class Name	0021 (a = 0, b = 0, c = 2, d = 1)
#someid	ID Name	0100 (a = 0, b = 1, c = 0, d = 0)
div#someid	Element Name + ID Name	0101 (a = 0, b = 1, c = 0, d = 1)
style (attribute)	style (attribute)	1000 (a = 1, b = 0, c = 0, d = 0)



Unités absolues

pt Point (#pixel! = 1/72 pouce)
 pc Pica (= 12 pt)
 in Pouce (= 2,54 cm)
 cm Centimètre
 mm Millimètre

Pourcentage...

...de la **même propriété** pour l'élément **parent**.

Ex.:

```
<p style="width:200px;font-size:  
24pt;">Bonjour  
<strong style="font-size:150%">  
toi</strong></p>
```

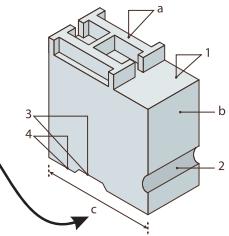


Certaines propriétés CSS font référence à des **longueurs** qu'on peut exprimer dans trois types d'unités (p. 283):

- Unités relatives
- Unités absolues
- Pourcentages

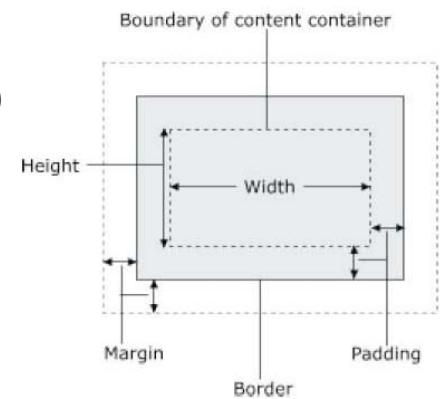
Unités relatives:

- px: pixels
 em: hauteur d'une lettre
 ex: largeur de la lettre x



Chaque élément est composé de (p. 285):

- Son contenu
- Une bordure autour
- Un remplissage (padding)
- Une marge extérieure



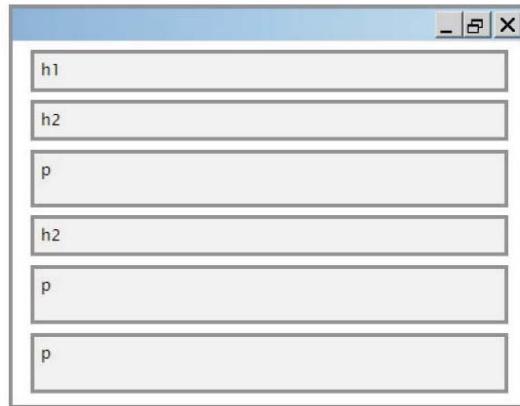
Largeur totale =
 largeur du contenu
 + padding gauche + droite
 + marge gauche + droite

Hauteur =
 contenu + padding h/b + marge h/b



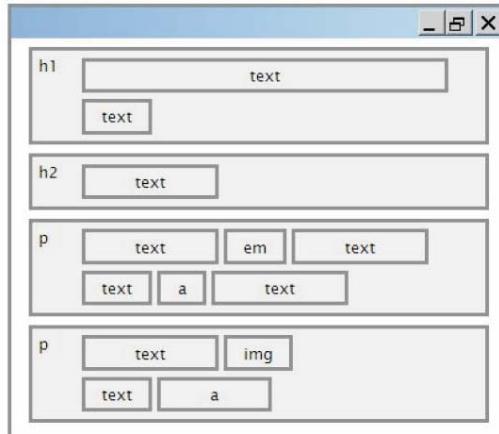
Le navigateur place les éléments les uns après les autres en suivant un **flot**

Flot des éléments de niveau **bloc**



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Exemple plus complexe:



 ch07_eg21.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Flot des éléments de niveau **texte**



Essayez d'encadrer tous les éléments d'une page!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On peut modifier:

- les bordures  ch07_eg22.html
- le remplissage  ch07_eg23.html
- les marges  ch07_eg24.html
- la taille de la boîte d'un élément  ch07_eg25.html
- l'interligne du texte  ch07_eg26.html
- la taille minimale et maximale de la boîte  ch07_eg27.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

La propriété **overflow** spécifie comment est traité le contenu qui dépasse de la boîte d'un élément



[ch07_eg29.html](#)

On peut centrer un élément de niveau bloc en utilisant **auto** comme valeur pour les marges horizontales:

```
p {  
    margin-left: auto;  
    margin-right: auto;  
    width: 750px;  
}
```

- Préférable de spécifier la largeur aussi (sinon, peut occuper toute la page)
- Pour centrer des éléments de niveau texte:
text-align: center;



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Couleur et image de fond

Les attributs

background-color [ch08_eg02.html](#)

background-image [ch08_eg03.html](#)

spécifient la couleur et l'image de fond d'un élément. On peut également modifier la position et la répétition du fond:

background-repeat [ch08_eg05.html](#)

background-position [ch08_eg08.html](#)

Une application: les **sprites CSS**



Tutoriel sur les sprites

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Au programme après cette pause:

- Le module FireBug de Firefox
- Variantes entre les navigateurs
- Solution d'exercices choisis



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Couleur et image de fond

Rappel d'un cours précédent...

Attention!

Si on veut simplement faire une icône avec un lien, il existe une meilleure manière de procéder (HTML + CSS) sans utiliser l'élément ``. On verra plus loin.

Comment feriez-vous?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour modifier l'apparence des listes:

list-style-type	 ch08_eg10.html
list-style-position	 ch08_eg11.html
list-style-image	 ch08_eg12.html

Un **contour** (*outline*) est comme une bordure, mais il n'occupe pas d'espace (Duckett p. 328-ss.)

outline-width	 ch08_eg19.html
outline-style	
outline-color	

Une application: un **menu** de navigation

-  Démo de menu CSS vertical
-  Démo de menu CSS horizontal



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Contenu généré 

On peut même utiliser CSS pour ajouter du contenu à un document HTML

- Avec les pseudo-classes **before** et **after**

 ch08_eg21.html

Une application: sections **auto-numérotées**

 ch08_eg22.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Autres propriétés 

La propriété **cursor** change le pointeur de la souris au-dessus des éléments choisis (à utiliser avec prudence)

 ch08_eg24.html

La propriété **display** spécifie si un élément est de niveau texte ou bloc et peut retirer un élément (**display:none**)

Une application: une **image-bouton avec du texte**

La propriété **visibility** permet d'afficher ou de cacher un élément (**l'élément occupe toujours le même espace**)

 ch08_eg25.html

La règle **@import** permet d'inclure un autre fichier CSS

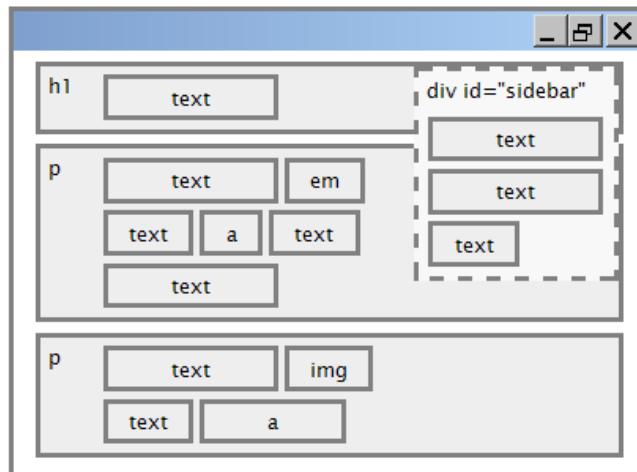


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On passe la main à 



Floating Elements Diagram



The CSS float Property

```
img.headericon {
  float: right; width: 130px;
}
```

CSS

[Borat Sagdiyev](#) (born July 30, 1972) is a fictional Kazakhstani journalist played by British-Jewish comedian Sacha Baron Cohen. He is the main character portrayed in the controversial and successful film Borat: Cultural Learnings of America for Make Benefit Glorious ...



output

property	description
float	side to hover on; can be left, right, or none (default)

- Removed from normal document flow; underlying text wraps around as necessary

Common float Bug: Missing Width

I am not floating, no width

I am floating right, no width

I am not floating, 45% width

I am floating right, 45% width

- Often floating block elements must have a width property value
 - If no width is specified, the floating element may occupy 100% of the page width, so no content can wrap around it

The clear Property

```
p { background-color: fuchsia; }
h2 { clear: right; background-color: yellow; } CSS
```

Homestar Runner is a Flash animated Internet cartoon. It mixes surreal humour with references to 1980s and 1990s pop culture, notably video games, classic television and popular music.



My Homestar Runner Fan Site

Property	Description
clear	disallows floating elements from overlapping this element; can be left, right, or none (default)

June 30, 2010

SUN YAT-SEN UNIVERSITY

6 / 40

Common Error: Container Too Short

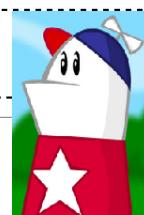
```
<p>
Homestar Runner is a Flash animated Internet cartoon.
It mixes surreal humour with ....</p>
```

HTML

```
p { border: 2px dashed black; }
img { float: right; }
```

CSS

Homestar Runner is a Flash animated Internet cartoon. It mixes surreal humour with



- We want the p containing the image to extend downward so that its border encloses the entire image

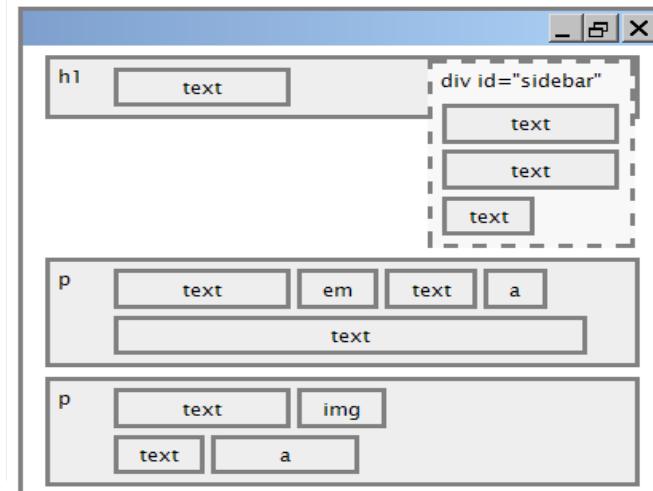
June 30, 2010

SUN YAT-SEN UNIVERSITY

8 / 40

Clear Diagram

```
div#sidebar { float: right; }
p { clear: right; } CSS
```



June 30, 2010

SUN YAT-SEN UNIVERSITY

7 / 40

The overflow Property

```
p { border: 2px dashed black;
overflow: hidden; }
```

CSS

Homestar Runner is a Flash animated Internet cartoon. It mixes surreal humour with



output

Property	Description
overflow	specifies what to do if an element's content is too large; can be auto, visible, hidden, scroll, or inherit

June 30, 2010

SUN YAT-SEN UNIVERSITY

9 / 40

Multi-column Layouts

```
<div>
  <p>first paragraph</p>
  <p>second paragraph</p>
  <p>third paragraph</p>
  Some other text that is important
</div>
```

HTML

```
p { float: right; width: 20%; margin: 0.5em;
    border: 2px solid black; }
div { border: 3px dotted green; overflow: hidden; }
```

CSS

Some other text that is important

third paragraph

second paragraph

first paragraph

Some other text that is important

output

June 30, 2010

SUN YAT-SEN UNIVERSITY

10 / 40

The position Property

```
div#ad {
  position: fixed;
  right: 10%;
  top: 45%;
}
```

CSS

Property	Value	Description
position	static	default position
	relative	offset from its normal static position
	absolute	a fixed position <i>within its containing element</i>
	fixed	a fixed position <i>within the browser window</i>
<u>top</u> , <u>bottom</u> , <u>left</u> , <u>right</u>		positions of box's corners

June 30, 2010

SUN YAT-SEN UNIVERSITY

12 / 40

Outline

- Floating Elements
- **Sizing and Positioning**
- Evil IE
- Thinking ...
 - Declarative programming
 - User Centric Design

11 / 40

SUN YAT-SEN UNIVERSITY

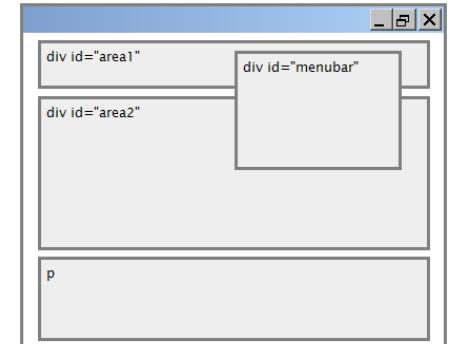
June 30, 2010

Absolute Positioning

```
#menubar {
  position: absolute;
  left: 400px;
  top: 50px;
}
```

CSS

- Removed from normal flow (like floating ones)
- Positioned relative to the block element containing them (assuming that block also uses **absolute** or **relative** positioning)
- Actual position determined by **top**, **bottom**, **left**, **right** values
- Should often specify a **width** property as well



13 / 40

SUN YAT-SEN UNIVERSITY

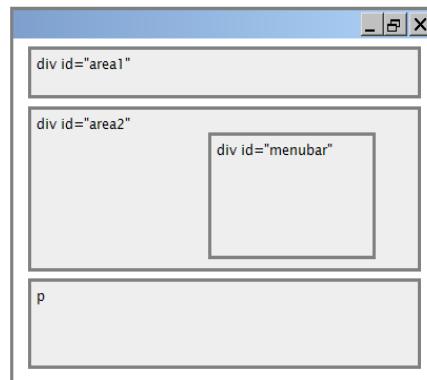
June 30, 2010

Relative Positioning

```
#area2 { position: relative; }
```

CSS

- Absolute-positioned elements are normally positioned at an offset from the corner of the overall web page
- To instead cause the absolute element to position itself relative to some other element's corner, wrap the **absolute** element in an element whose **position** is **relative**



June 30, 2010

SUN YAT-SEN UNIVERSITY

14 / 40

Alignment vs. Float vs. Position

- If possible, lay out an element by **aligning** its content
 - Horizontal alignment: **text-align**
 - Set this on a block element; it aligns the content within it (not only text, and not the block element itself)
 - Vertical alignment: **vertical-align**
 - Set this on an inline element, and it aligns it vertically within its containing element
- If alignment won't work, try **floating** the element
- If floating won't work, try **positioning** the element
 - absolute** / **fixed** positioning are a last resort and should not be overused
- [More position examples](#)

June 30, 2010

SUN YAT-SEN UNIVERSITY

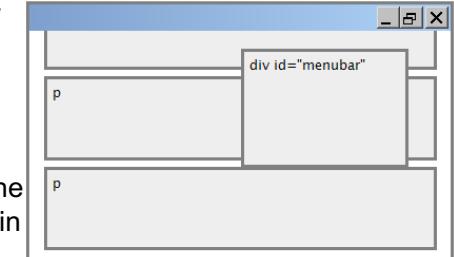
16 / 40

Fixed Positioning

```
#menubar {
  position: fixed;
  left: 400px;
  top: 50px;
}
```

CSS

- Removed from normal flow (like floating ones)
- Positioned relative to the browser window
 - Even when the user scrolls the window, element will remain in the same place



June 30, 2010

SUN YAT-SEN UNIVERSITY

15 / 40

Details about Inline Boxes

- Size properties (**width**, **height**, **min-width**, etc.) are ignored for inline boxes
- margin-top** and **margin-bottom** are ignored, but **margin-left** and **margin-right** are **not**
- The containing block box's **text-align** property controls horizontal position of inline boxes within it
 - text-align** does not align block boxes within the page
- Each inline box's **vertical-align** property aligns it vertically within its block box

June 30, 2010

SUN YAT-SEN UNIVERSITY

17 / 40

SUN YAT-SEN UNIVERSITY

The vertical-align Property

property	description
vertical-align	specifies where an inline element should be aligned vertically, with respect to other content on the same line within its block element's box

- Can be top, middle, bottom, baseline (default), sub, super, text-top, text-bottom, or a length value or %
- Baseline means aligned with bottom of non-hanging letters



June 30, 2010

SUN YAT-SEN UNIVERSITY

18 / 40

Common bug: Space under Image

<p style="background-color: red; padding: 0px; margin: 0px">

</p>

HTML



- Red space under the image, despite padding and margin of 0
- This is because the image is vertically aligned to the baseline of the paragraph (not the same as the bottom)
- Setting vertical-align to bottom fixes the problem (so does setting line-height to 0px)

June 30, 2010

SUN YAT-SEN UNIVERSITY

20 / 40

vertical-align Example

```
<p style="background-color: yellow;">
<span style="vertical-align: top; border: 1px solid red;">
Don't be sad! Turn that frown
 upside down!

Smiling burns calories, you know.

Anyway, look at this cute puppy; isn't he adorable! So cheer up,
and have a nice day. The End.
</span></p>
```

HTML

Don't be sad! Turn that frown  upside down!  Smiling
burns calories, you know.  Anyway, look at this cute puppy;
isn't he adorable! So cheer up, and have a nice day. The End.

output

June 30, 2010

SUN YAT-SEN UNIVERSITY

19 / 40

The display Property

h2 { display: inline; background-color: yellow; }

CSS

This is a heading This is another heading

output

Property	Description
display	sets the type of CSS box model an element is displayed with

- Values: none, inline, block, run-in, table, table-caption, ...
 - Not all values supported by all browsers (check out at <http://www.quirksmode.org/css/display.html>)
- Use sparingly, because it can radically alter the page layout

June 30, 2010

SUN YAT-SEN UNIVERSITY

21 / 40

Displaying Block Element as Inline

```
<ul id="topmenu">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

HTML

```
#topmenu li {
  display: inline;
  border: 2px solid gray;
  margin-right: 1em;
}
```

CSS

Item 1 Item 2 Item 3

output

- Lists and other block elements can be displayed inline
 - Flow left-to-right on same line
 - Width is determined by content (block elements are 100% of page width)

June 30, 2010

SUN YAT-SEN UNIVERSITY

22 / 40

Outline

- Floating Elements
- Sizing and Positioning
- Evil IE**
- Thinking ...
 - Declarative programming
 - User Centric Design

June 30, 2010

SUN YAT-SEN UNIVERSITY

24 / 40

The visibility Property

```
p.secret { visibility: hidden }
```

CSS

Property	Description
visibility	sets whether an element should be shown onscreen; can be <code>visible</code> (default) or <code>hidden</code>

- hidden** elements will still take up space onscreen, but will not be shown
 - To make it not take up any space, set `display` to `none` instead
- Can be used to show/hide HTML content on the page in response to events

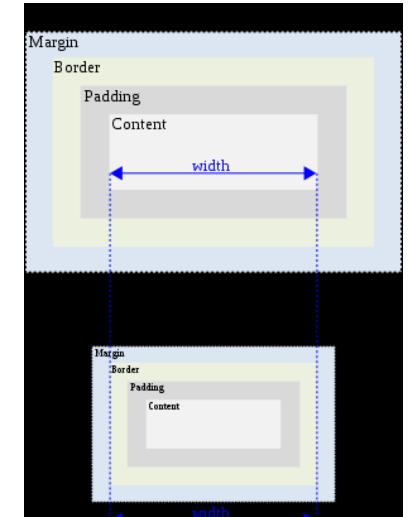
June 30, 2010

SUN YAT-SEN UNIVERSITY

23 / 40

Evil IE

- IE is painful for Web designer and developer, since it doesn't compatible with W3C standards, and mostly deliberately ...
- Weird IE Box model
- Double margin with float
- Block has width floats when beneath a float element
- Transparent png (IE 6.0)



June 30, 2010

SUN YAT-SEN UNIVERSITY

25 / 40

Evil IE

- A lot of workarounds available, but the best is loading a specific style sheet for IE with **conditional comment**

```
<!--[if IE 7]>
  According to the conditional comment this is Internet Explorer
<![endif]-->
<!--[if gte IE 5]>
  According to the conditional comment this is Internet Explorer 5 and up
<![endif-->
```

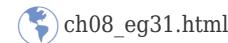
XHTML

- gt, lt, gte, lte**

La propriété **z-index** permet de changer l'ordre d'**empilement** des éléments.

- Amener en avant-plan
- Envoyer en arrière-plan

Plus la valeur de **z-index** est élevée, plus l'élément est « au-dessus »



Pour récapituler:

Un exemple: Duckett p. 351-354





JavaScript

I. Introduction au langage

- Inventé par Netscape
- S'est appelé **Mocha**, puis **LiveScript**, puis **JavaScript** (1995) lorsque Sun Microsystems s'est joint au projet
- La European Computer Manufacturers Association (ECMA) a défini la première norme: ECMA-262
- ECMA-262 version 3 = Netscape JavaScript 1.5
- Les « JavaScript » de Firefox (3) et d'IE (8) sont des **variantes** (dialectes) de ce « ECMAScript »
 - Conformes à ECMA-262, mais ajoutent des fonctionnalités hors-standard



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

JavaScript

C'est un langage de programmation formé de trois parties:

1. Le noyau (*Core JavaScript*)

Opérateurs, expressions, sous-programmes

2. JavaScript côté client (*Client-side JS*)

Collection d'objets permettant le contrôle du navigateur et ses interactions avec les utilisateurs

3. JavaScript côté serveur (*Server-side JS*)

Collection d'objets utiles pour l'exécution sur un serveur (connexion avec une BD, etc.). On n'en parlera pas!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

JavaScript vs. Java

JavaScript ≠ Java

- On ne **compile pas** un script JavaScript (langage purement interprété)
- JS permet de définir des « **classes** », mais selon un modèle très différent de celui de Java (pas vraiment un langage OO)
- En JS, le nombre de méthodes et champs d'une classe peut changer durant l'exécution (?!?)
- Beaucoup d'erreurs en JS sont **silencieuses**
- Différences de **syntaxe**
- Moins de types de données; typage **faible**, déclarations facultatives



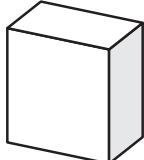
Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pourquoi utiliser JavaScript? 🕸️

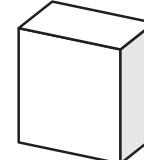
Serveur



1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



Serveur



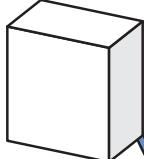
1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



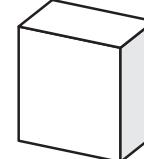
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pourquoi utiliser JavaScript? 🕸️

Serveur



1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



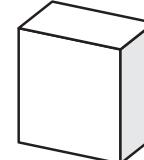
1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pourquoi utiliser JavaScript? 🕸️

Serveur



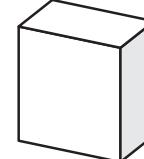
1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pourquoi utiliser JavaScript? 🕸️

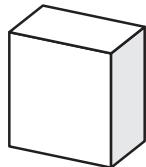
Serveur



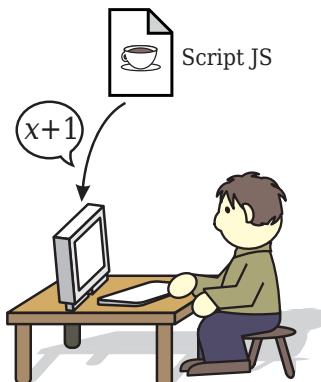
1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



1. Déplace la charge de travail dans le navigateur plutôt que sur le serveur



Par contre, JavaScript ne permet **pas**:

- La gestion de fichiers sur disque
- Les communications (directes*) avec une base de données
- Fonctions avancées* de réseaux

2. Permet d'**interagir** avec l'utilisateur

- Capter le mouvement de la souris...
- Intercepter les clics sur des boutons, ou même sur des éléments (HTML) de la page...

...et réagir à ces événements via du code exécuté par le navigateur

3. Permet de **modifier le contenu** de la page après son chargement depuis le serveur



L'élément HTML **<script>** permet d'inclure du code JavaScript à une page



On utilise le type MIME **text/javascript** pour le représenter

Dans cet exemple:

- L'objet **document** représente la page web
- La méthode **write()** écrit le texte au point courant
- Le code inséré dans **<body>** s'exécute à mesure que la page se charge
- Le code inséré dans **<head>** s'exécute lorsque les événements appropriés se déclenchent



On peut inclure du code JavaScript depuis un **fichier externe**:

```
<script type="text/javascript"
src="monfichier.js"></script>
```

Quand ce code est-il exécuté?

- Comme pour <head> si l'élément est dans <head>
- Comme pour <body> si l'élément est dans <body>



Rappel: un document XHTML ne doit contenir que du XHTML valide.

Quelles sont les erreurs XHTML de ce bout de code?

```
...
<script type="text/javascript">
document.write("<p>");
if (a == 1 && b < 1)
    document.write("Hello!</p>");
else
    document.write("Good bye!</p>");
</script>
...
```



Rappel: un document XHTML ne doit contenir que du XHTML valide.

Quelles sont les erreurs XHTML de ce bout de code?

```
...
<script type="text/javascript">
document.write("<p>");
if (a == 1 && b < 1)
    document.write("Hello!</p>");
else
    document.write("Good bye!</p>");
</script>
...
```



Bonne pratique: inclure le code JavaScript entre des balises CDATA:

```
<script type="text/javascript">
//<![CDATA[
...
]]>
</script>
```

Tout ce qui se trouve entre `<![CDATA[...]]>` est « ignoré » par le parser XHTML

Référez à un fichier externe « règle » le problème, y compris pour les vieux navigateurs qui ne comprennent pas JS... autrement on peut utiliser l'élément `<noscript>`



Mots réservés par JavaScript: il y en a 25

break	delete	function	return	typeof
case	do	if	switch	var
catch	else	in	this	void
continue	finally	instanceof	throw	while
default	for	new	try	with

Commentaires: comme en Java

// ou /* ... */

Noms de variables:

Expression régulière: [A-z_][A-z_0-9]*
Sensibles à la casse!



Un « objet » est une **collection de propriétés** qui sont soit:

- Une **donnée** (ou propriété)
- Une **fonction** (ou méthode)

Exemple: l'objet **document**

```
document.write("abc");
document.title
```

write est une méthode
title est une propriété



JS n'est pas un langage **orienté-objet**...

...mais un langage **basé sur** les objets

- Pas de classes, pas d'héritage
- Concept fondamental: la **fonction**

...on peut mettre une fonction dans une variable:

```
a = function(x) { return x+1; }
```



JavaScript définit 5 **types primitifs** (i.e. non objets):

- Number
- String
- Boolean
- Undefined
- Null

JavaScript fournit des objets correspondant à ces types, appelés **wrapper objets**:

- Number
- String
- Boolean

Fournissent propriétés/méthodes pour manipuler les variables de types primitifs correspondants



Nombres

- Entiers et réels = **même** type (pas de `int` vs. `double`)
- Opérateurs classiques: + - * / % ++ -- = += ...

Strings

Chaînes de caractères

- Délimitées par guillemets simples ('bonjour les amis') ou doubles ("bonjour les amis")
- Aucune différence, SAUF pour les guillemets:
 - "Il dit: \"Attention à l'apostrophe\""
 - 'Il dit: "Attention à l'\\' apostrophe'"
- On peut utiliser \r, \n (comme en Java)
- Concaténées avec +



L'objet **Math** contient des méthodes/propriétés pour manipuler des objets de type **Number** (pp. 523-4)

```
Ex.: y = Math.PI;           y = 3.1415927
      y = Math.sin(y);     y = -1
      y = Math.round(10.5); y = 11
      y = Math.min(8, 10);  y = 8
      y = Math.abs(-2);    y = 2
```

L'objet **Number** possède des propriétés:

MIN_VALUE, MAX_VALUE, NaN
 NEGATIVE_INFINITY, POSITIVE_INFINITY
 PI, E

`Number.toString(x)` convertit le nombre x en String



Null

Pas de valeur (`x=null`). Essayer de lire une valeur null provoque une (rare!) erreur

Undefined

Seule valeur possible de ce type: `undefined`. Ex.:

```
var x;
document.write(x); ← Écrira la chaîne "undefined"
```

Boolean

Valeurs `true` et `false` (résultat d'opérations booléennes)

On peut vérifier le type d'une variable avec le mot réservé `typeof` (retourne une String avec le nom du type):

```
typeof(1) == "number"
```



JavaScript convertit automatiquement le type des données lorsque c'est possible

- Si l'une des opérandes de + est une String, toutes les autres opérandes sont converties en String

Ex.: "Août" + 1977 ← évalue à "Août 1977"

- Si l'une des opérandes de *, /, - est un nombre, tout est converti en Number

Ex.: 7 * "3" ← évalue à 21

- On obtient NaN quand la conversion échoue

Ex.: 7 * "Août" ← évalue à NaN



Les opérateurs `==` et `!=` sont des tests d'égalité stricte:
vérifient la valeur et le type (pas d'auto-conversion)

Ex.: `x = 5 == "5"` ← évalue à `false`

Une String peut être convertie en nombre avec le constructeur de l'objet Number

Ex.: `y = Number("250");` ← `y = 250`

Fonctionne seulement si x ne contient que des nombres (i.e.
pas avec "250 bonjour" ou "250,"

On peut utiliser `parseInt()` et `parseFloat()` pour une conversion
plus robuste

Ex.: `y = parseFloat("2.54E+3");` ← `y = 2540`

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Méthodes de String

On peut effectuer des opérations sur des Strings via les méthodes suivantes:

```
s = "Bonjour toi";
x = s.substring(0, s.indexOf(" "));
x = s.charAt(3);
x = s.toLowerCase();
x = s.toUpperCase();
k = s.lastIndexOf("o");
t = s.split(" ");
x = s.replace("toi", "vous");
```

Longueur d'une String: `s.length` ← Propriété,
pas méthode

Accès aux lettres individuelles: `s[3]` Pas dans IE!

Cf. Duckett, pp. 516-519

Un nombre peut être converti en String avec la méthode `toString()`

Ex.: `x = 6;`
`y = x.toString();` ← `y = "6"`
`z = x.toString(2);` ← `z = "110"` (en binaire)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

L'objet Date

L'objet Date permet de manipuler la... date (pp. 520-523)

On instancie d'abord une date...

```
var d = new Date();
```

Méthodes possibles:

<code>d.toLocaleString();</code>	Date en String
<code>d.getDate();</code>	Jour du mois (1-31)
<code>d.getMonth();</code>	Mois (0-11)
<code>d.getDay();</code>	Jour de la semaine (0-6)
<code>d.getFullYear();</code>	Année
<code>d.getTime();</code>	ms depuis 1970-01-01
<code>getHours(), getMinutes(),</code>	
<code>getSeconds(), getMilliseconds();</code>	



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Les instructions de contrôle de flot sont semblables à Java
(Duckett pp. 508-ss.)

```
if (condition)
{
    // Code si vrai
}
else
{
    // Code si faux
}

switch (expr)
{
    case val1:
        // Code
        break;
    case val2:
        // Code
        break;
    default:
        // Code
}
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Tableaux

On crée des tableaux avec l'objet Array (pp. 525-6):

```
var maliste = new Array(1, 2,      ← Crée tableau avec les
    "trois", "quatre");

var maliste2 = new Array(100);   ← Crée tableau de
                                taille 100

var maliste3 = [1, 2, "trois",   ← Crée tableau avec les
    "quatre"];
```

- Les indices de tableau commencent à zéro
- Les indices sont consécutifs, donc:
maliste[47] = "blabla";
rend maliste de taille 48 (et non 5)
- On peut donner une valeur à la propriété length d'un tableau (ce qui le redimensionne!)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Les instructions de contrôle de flot sont semblables à Java
(Duckett pp. 508-ss.)

```
while (condition)
{
    ...
}

do
{
    ...
} while (condition);

for (init; ctrl; inc)
{
    ...
}
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Tableaux

Les éléments d'un tableau sont alloués dynamiquement, i.e.

```
var maliste2 = new Array(100);
```

crée un tableau de 100 « cases », mais ne réserve aucune mémoire pour les éléments eux-mêmes

Les cellules d'un même tableau n'ont pas besoin de contenir des éléments du même type:

```
maliste2[0] = 1000;
maliste2[1] = "Abc";
maliste2[2] = function(x) {return x+1;};
```

insert_names.html



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Les tableaux possèdent plusieurs méthodes de manipulation:

```
var noms = ["Pierre", "Jean", "Jacques"];
var s = noms.join(" et ");
    ↗
    s contient "Pierre et Jean et Jacques"

noms.concat("Luc", "Luce");
    ↗
    noms contient ["Pierre", "Jean", "Jacques", "Luc", "Luce"]

noms.sort();
    ↗
    noms contient ["Jacques", "Jean", "Luc", "Luce", "Pierre"]

var noms2 = noms.slice(1,3);
    ↗
    noms2 contient ["Jean", "Luc"]
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Un élément de tableau peut être... un autre tableau

 nested_arrays.html

Crée un tableau en deux dimensions

Les tableaux possèdent plusieurs méthodes de manipulation:

```
noms2.push("Julie");
    ↗
    noms2 contient ["Jean", "Luc", "Julie"]

var x = noms2.pop();
    ↗
    x = "Julie" (retiré de noms2)

noms2.shift("Julie");
    ↗
    noms2 contient ["Julie", "Jean", "Luc"]

x = noms2.unshift();
    ↗
    x = "Julie" (retiré de noms2)
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On a vu `document.write()`...

On peut aussi effectuer:

```
var x = "1234";
alert("La valeur est " + x);

var question = confirm("Voulez-vous
continuer?");

var nom = prompt("Quel est votre nom?", "");
```

Ces méthodes sont appelés sur l'objet par défaut: `window`

Autres propriétés de `window`: Duckett pp. 526-8



Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Une variable qui est déclarée **sans** le mot-clé **var** est automatiquement créée lors de sa première occurrence dans le script
 - La variable est **toujours globale** (visible dans tout le document), même si elle apparaît dans une fonction!
- Une variable déclarée avec **var** est **locale**
- Une variable locale cache une variable globale du même nom

Conseils:

- 1) déclarer/initialiser les variables comme en Java
- 2) toujours utiliser var à moins d'une exception



- Les propriétés sont comme des indices d'un « tableau », donc ces trois expressions sont équivalentes:

```
var x = my_car.brand;
var y = my_car["brand"];
var b = "brand"; var z = my_car[b];
```

- Les propriétés d'un objet peuvent être... un autre objet:

```
my_car.engine = new Object();
my_car.engine.config = "V6";
my_car.engine.hp = 200;
```

- On peut supprimer une propriété:

```
delete my_car.model;
```



- Rappel: un objet est une collection de propriétés et de méthodes.
- L'objet **Object** est l'objet vide
- Rappel: on peut ajouter des propriétés/méthodes à un objet dynamiquement
- Donc, on peut faire ceci:

```
var my_car = new Object();
my_car.brand = "Ford";
my_car.model = "F-150";
```

- Ou ceci:

```
var my_car = {make: "Ford", model: "F-150"};
```



- On peut itérer sur les propriétés d'un objet:

```
for (var prop in my_car)
{
  document.write("Nom: ", prop, " valeur ",
    my_car[prop]);
}
```

À suivre...



Pour déclarer une fonction:

```
function mafct(x, y, z)
{
    ...
}
```

- Pas de types d'entrée ou de sortie!
- Toujours le mot **function**
- Possiblement un **return;** ou un **return valeur;**
- En JS, une fonction est aussi un objet!
- Une variable peut donc contenir une **référence** à une fonction:

```
y = mafct;
y(1,2,3); ← Même chose que mafct(1,2,3)
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Une fonction peut avoir un **nombre variable d'arguments**, accessibles via le tableau (Array) **arguments**, local à chaque fonction

params.js



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Autre différence avec Java: l'interpréteur doit lire la définition d'une fonction avant toute invocation de la fonction

```
document.write(
    mafct(0));
function mafct(x)
{
    return x+1;
}
```

Incorrect

```
function mafct(x)
{
    return x+1;
}
document.write(
    mafct(0));
```

Correct

En général: les fonctions sont dans le **<head>**, les appels dans le **<body>**



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Syntaxe alternative pour déclarer une fonction:

```
mafct = function(x, y, z)
{
    ...
}
```

Les types primitifs sont passés **par valeur**; essayer ceci:

```
function myfunction(x)
{
    x = 5;
}
var x = 4;
alert(x);
myfunction(x);
alert(x);
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Par contre, les objets sont passés **par référence**:

```
function monobjet()
{
    this.value = 5;
}

var o = new monobjet();
alert(o.value);

function changement(fnc)
{
    fnc.value = 6;
}

changement(o);
alert(o.value);
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Objets (II)

Puisque les fonctions sont des objets, une **méthode** n'est qu'une propriété d'un objet à laquelle on affecte une fonction:

```
var my_car = new Object();
my_car.brand = "Ford";
my_car.model = "F-150";
my_car.toString = function(yr)
{
    return this.brand + " " +
        this.model + yr;
};
document.write(my_car.toString("2010"));
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour trier autre chose que des chaînes en ordre alphabétique, on écrit une **fonction** qui compare deux objets et on la passe à la méthode `sort` d'un tableau

La fonction doit retourner:

- un nombre négatif si l'ordre est OK
- zéro si les deux arguments sont égaux
- un nombre positif si les arguments doivent être échangés

 Le tableau `t` contient des nombres. Comment trie-t-on ces nombres en ordre croissant? (La solution a deux lignes.)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Constructeurs

Un **constructeur** est une fonction qui initialise un objet vide (possiblement avec des arguments d'entrée):

```
function car(mk, md, yr) {
    this.make = mk;
    this.model = md;
    this.year = yr;
}
my_car = new car("Ford", "F-150", 1983);
```

Le constructeur peut même inclure des fonctions, qui agissent comme les **méthodes** de l'objet:

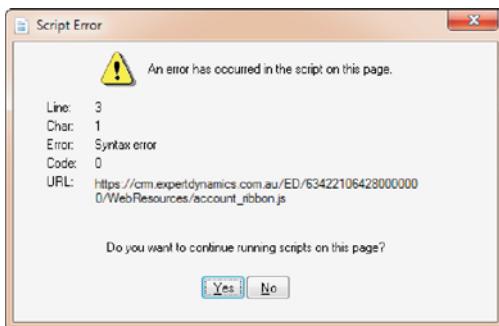
```
function car(mk, md) {
    ...
    toString = function(yr) {
}
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

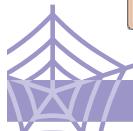
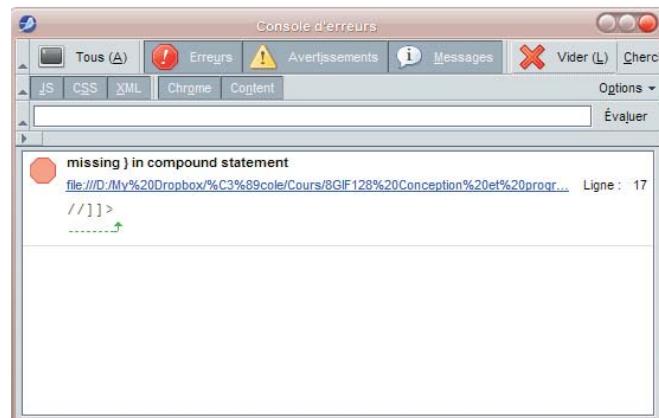
JavaScript produit peu de messages d'erreur visibles. Exemple:

```
document.write("Bonjour");
for (var i = 0; i <= 3; i++) {
    document.write(i);
X
```



Certaines versions
d'IE affichent
l'erreur sur-le-champ

Firefox accumule ces erreurs dans une **Console d'erreurs** qu'on doit faire afficher



On peut déboguer son code JavaScript en utilisant...

- Le **débogueur** JavaScript d'IE8 (dans le menu *Tools/Development Tools*)
- Le module **Firebug** dans Firefox





JavaScript

II. Document Object Model



Chaque objet Document possède:

- **forms**: un tableau de références aux formulaires du document. Chaque objet **Form** possède un tableau **elements**, qui a des références aux éléments du formulaire
- Document possède aussi **anchors**, **links**, **images**



Comment interagit-on avec le document HTML? On a vu `document.write()`, mais cela semble un peu limité!

- L'objet **window** représente la fenêtre dans laquelle le navigateur affiche les documents

Toutes les variables globales sont des propriétés de Window

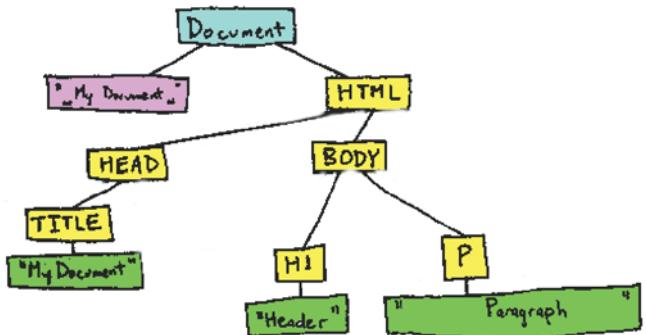
- **window** déclare implicitement certaines propriétés:

- **document**: référence à l'objet Document que la fenêtre affiche
- **frames**: tableau de référence aux frames du document



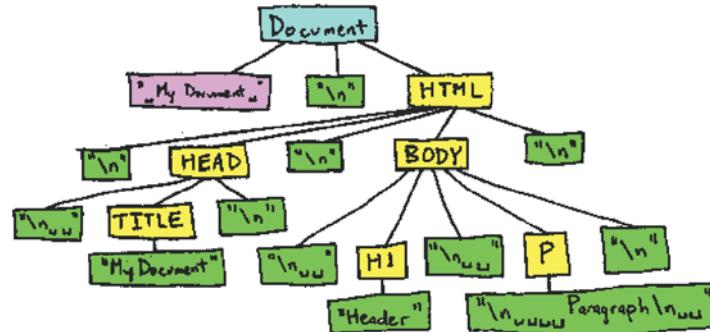
Rappel: un document HTML est un arbre

```
<!-- My document -->
<html>
<head>
  <title>My Document</title>
</head>
<body>
  <h1>Header</h1>
  <p>
    Paragraph
  </p>
</body>
</html>
```



Même les espaces blancs sont des éléments!

```
<!-- My document -->\n<html>\n<head>\n  <title>My Document</title>\n</head>\n<body>\n  <h1>Header</h1>\n  <p>\n    Paragraph\n  </p>\n</body>\n</html>\n
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Document Object Model 🕸️

Document Object Model (DOM)

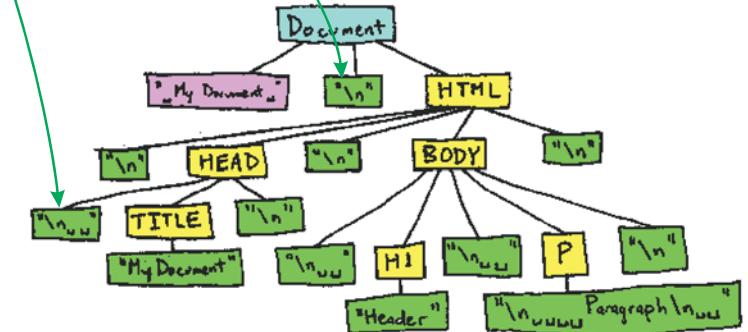
Ensemble d'interfaces normalisées permettant d'exposer le contenu d'un document (X)HTML à un langage de programmation (i.e. une API)

- Les documents dans le DOM ont une structure en arbre
- Dans la version JavaScript du DOM...
 - les éléments sont représentés sous forme d'**objets**
 - les attributs des éléments sont représentés sous forme de **propriétés** de ces objets
- Ex.: <input type="text" name="address"> sera représenté par un objet avec deux propriétés, **type** et **name**, dont les valeurs sont "text" et "address"

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Même les espaces blancs sont des éléments!

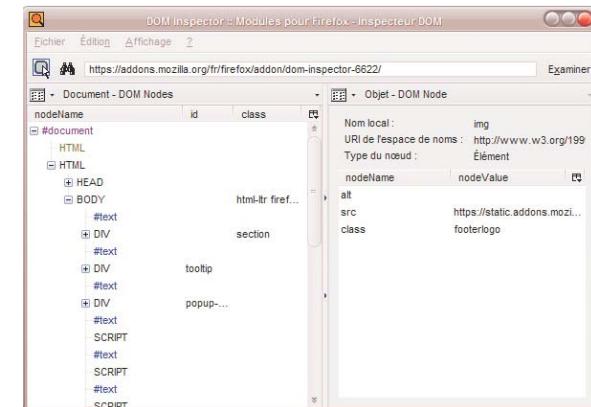
```
<!-- My document -->\n<html>\n<head>\n  <title>My Document</title>\n</head>\n<body>\n  <h1>Header</h1>\n  <p>\n    Paragraph\n  </p>\n</body>\n</html>\n
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

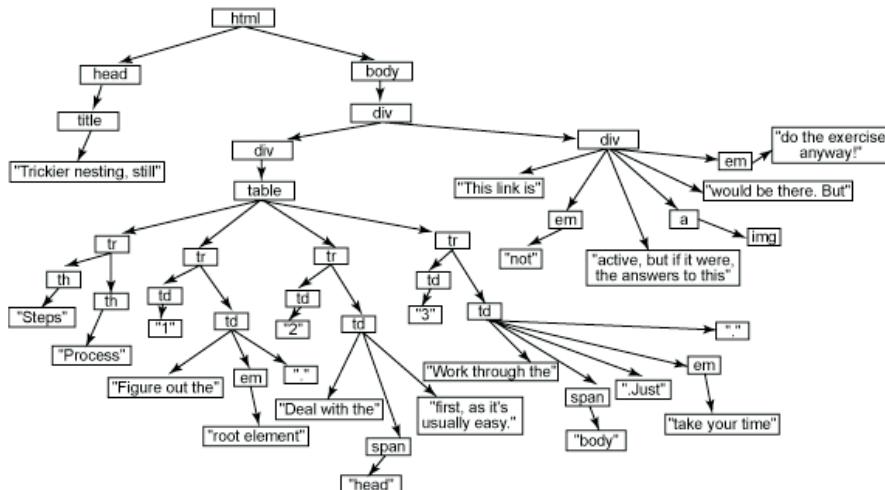
Inspecter le DOM 🕸️

Firefox et IE peuvent afficher l'arbre DOM d'un document. Dans Firefox, on utilise le module complémentaire DOM Inspector:



Exemple: essayer le DOM Inspector avec 🌎 table2.html

Sylvain Hallé • 8GIF128 Conception et programmation de sites web



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Accéder aux éléments en JavaScript

Méthode #2

Si tous les éléments sur le chemin de l'élément à accéder ont un attribut name (sauf body), on tape le chemin:

Ex.: `<body>
<form name="myForm" action=""
 <input type="button" name="myButton" />
</form>
</body>`

```
var btn = document.myForm.myButton;
```

Problèmes:

- 1) Si le document change, le chemin peut changer
- 2) Il faut prendre soin de tout nommer
- 3) XHTML 1.1 interdit l'attribut name dans les formulaires (entre autres)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Il existe plusieurs méthodes pour accéder à un élément d'une page à l'intérieur de JavaScript

Méthode #1

On écrit le chemin retourné par le DOM Inspector (« l'adresse DOM ») directement

Ex.: `var e = document.body.childNodes[1];`

Réfère au tableau
dans l'exemple précédent

Problème: et si le document change?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Accéder aux éléments en JavaScript

Méthode #3

On donne un attribut id à l'élément à accéder, et on utilise `getElementById`

Ex.: `<body>
<form action=""
 <input type="button" id="myButton" />
</form>
</body>`

```
var btn = document.
getElementById("myButton");
```

Problème:

Si on veut faire la même chose sur plusieurs éléments, il faut y accéder un par un avec leur id...

Il n'y a pas de `getElementsByClass`!



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Méthode #4

On donne un nom de balise et on utilise
`getElementsByName`

Ex.: `var t_liens = document.getElementsByName("a");`

⚠ Attention!

La méthode retourne toujours un tableau d'éléments, même s'il n'y en a qu'un!

Problème:

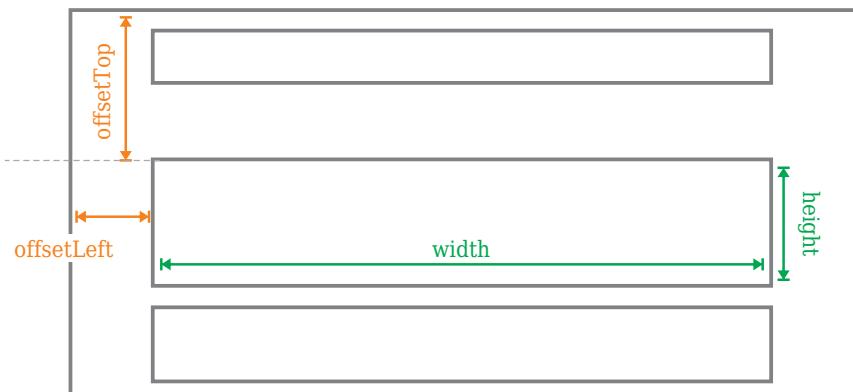
Après avoir ainsi filtré les éléments de la page, on se ramène au problème de la méthode #1 (lequel veut-on dans le tableau?)



Propriétés des éléments

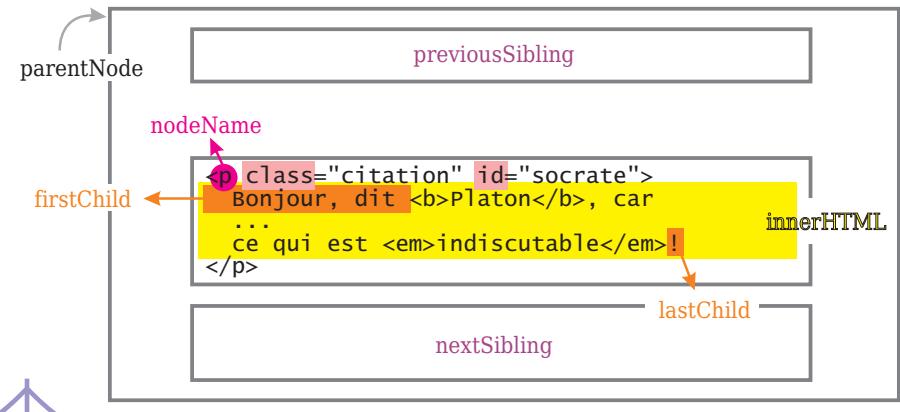
La plupart des éléments auxquels on accède sont des objets de type `HTMLElement`

Quelques `propriétés` universelles d'un `HTMLElement`:



La plupart des éléments auxquels on accède sont des objets de type `HTMLElement`

Quelques `propriétés` universelles d'un `HTMLElement`:



Propriétés des éléments

La plupart des éléments auxquels on accède sont des objets de type `HTMLElement`

Quelques `propriétés` universelles d'un `HTMLElement`:

<code>attributes []</code>	Tableau des attributs de l'élément
<code>childNodes []</code>	Tableau des enfants de l'élément
<code>className</code>	Nom de la classe CSS de l'élément
<code>style</code>	Tous les styles appliqués à l'élément

`innerHTML`
Contenu HTML de l'élément. Utilisé en écriture, `remplace` le contenu de l'élément!

```
e.innerHTML = "<p>Bonjour <em>tous</em></p>";
```

+ une dizaine d'autres moins utilisées



La propriété **style** permet de modifier les attributs CSS **en ligne** (i.e. dans le tag) d'un élément

- **style** possède des propriétés qui ont le même nom que l'attribut CSS, ex. **style.background** correspond à la propriété **background** de CSS

```
var p = document.getElementById("paragraphe");
p.style.color = "red";
p.style.top = "30px";
if (p.style.background == "yellow")
...

```



Attention!

4. Dernière exception: **float** est un mot réservé en JS **et** un nom d'attribut CSS...

Pour y accéder, on doit utiliser

```
style.cssFloat
ou
style.styleFloat dans IE
```



Attention!

1. Si un élément a des attributs CSS définis dans la balise **<style>** ou dans un CSS externe, on ne les verra pas en lisant la propriété **style**
2. Par contre, si on **définit** un attribut dans la propriété **style**, il a priorité sur les autres (application de la cascade)
3. Si un **nom** d'attribut CSS possède un trait d'union, son équivalent DOM est écrit en CamelCase
Ex.: **font-family** devient **fontFamily**
margin-top devient **marginTop**

Ne s'applique **pas aux valeurs**:

```
e.style.fontFamily = "sans-serif";
```



Quelques **méthodes** universelles d'un **HTMLElement**:

Les plus évidentes...

toString()	Le nom le dit
hasChildNodes()	Idem
getElementsByName()	On l'a déjà vu

Les autres...

cloneNode(b)
Retourne copie du noeud. Si **b** = true, recopie les enfants du noeud; si **b** = false, juste le noeud lui-même

appendChild()
Ajoute un nouvel enfant à la fin de la liste des enfants de l'élément



Exemple: recopier un mot d'un paragraphe à la fin de celui-ci...

```
var p = document.getElementById("paragraphe");
var e = p.childNodes[1].cloneNode(true);
p.appendChild(e);
```

...contrôle plus fin qu'avec `document.write!` Mais si on veut ajouter un élément ailleurs qu'à la fin?

`insertBefore(noeud, p)`
Ajoute noeud avant l'enfant p (p est un noeud)

```
var p = document.getElementById("paragraphe");
var e1 = p.childNodes[1];
var e2 = e1.cloneNode(true);
p.insertBefore(e2, e1);
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

`getAttribute(n) / setAttribute(n, v)`
Lit / écrit la valeur d'un attribut de l'élément

```
var p = document.getElementById("paragraphe");
var e = p.getAttribute("lang");
p.setAttribute("lang", "fr");
```

`removeAttribute(n)`
Supprime l'attribut n de l'élément

Les deux expressions suivantes sont équivalentes:

```
e = p.getAttribute("class");
p.setAttribute("id", "abc");
e = p.class;
p.id = "abc";
```

⚠️ Attention! Les attributs accessibles par `xxxAttribute()` sont les attributs HTML qui vont dans le tag...
...pas les attributs CSS (comme background, etc.)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

`removeChild(c)`

Retire le noeud c

```
var p = document.getElementById("paragraphe");
var e = p.childNodes[1].cloneNode(true);
p.removeChild(e);
```

`replaceChild(c, d)`
Remplace le noeud d par le noeud c

```
var p = document.getElementById("paragraphe");
var e1 = p.childNodes[1];
var e2 = p.childNodes[2].cloneNode(true);
p.replaceChild(e2, e1);
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Une fois qu'on a trouvé l'élément qui nous intéresse, on peut modifier ses propriétés.

Quelles propriétés?

- a) Consultez le DOM Inspector!
- b) Consultez une référence DOM en ligne

Exercice

1. Écrivez un bout de code JavaScript qui permet d'aller lire le texte à l'intérieur de la première cellule de tableau
2. Remplacez cette cellule par le texte "Bonjour" en caractères gras



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

On peut créer de nouveaux éléments comme suit:

```
var txtNode = document.createTextNode("Hello. This is
a new node.");
var link = document.createElement('a');
link.setAttribute('href', 'mypage.htm');
```

Le modèle DOM définit aussi des **événements**: notifications que quelque chose s'est produit dans le document...

...soit avec le navigateur

...soit par une intervention de l'utilisateur

Un **gestionnaire d'événement** (*event handler*) est un bout de code qui est exécuté en réponse à l'occurrence d'un événement

Connecter un gestionnaire d'événement à un événement se dit **enregistrer**



Exemple d'événement DOM: **click**

- Se produit chaque fois qu'un élément de la page est cliqué par l'utilisateur, i.e. tous les éléments produisent un événement **click** lorsqu'on clique dessus
- Pour réagir à l'événement **click** d'un élément, il faut lui enregistrer un gestionnaire (*handler*)
- On peut enregistrer un gestionnaire de **deux** manières...



Manière #1

L'attribut HTML **onclick** permet de spécifier la fonction qui doit être appelée lors d'un événement **click**

```
function mafonction()
{
    // Fonction appelée lorsqu'on
    // clique sur le paragraphe
}

...
<p id="par" onclick="mafonction();">
    Le paragraphe
</p>
```



Exercice: écrire un bout de code qui fait alterner les items d'une liste entre rouge et noir en cliquant dessus



Problème de cet exemple?

On doit donner un `id` à chaque item, et passer le nom de l'`id` lorsqu'on enregistre le gestionnaire dans chaque élément...

...une excellente **recette pour se tromper** quelque part!



[handlers-dom-soln2.html](#)

(Indice: la solution utilise `this`)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Enregistrement d'un gestionnaire

Problème de cet exemple?

Le code qui enregistre le gestionnaire doit être écrit à la toute fin du `<body>`... s'il est placé plus haut, les éléments n'existent pas encore lorsque le code s'exécute!



[handlers-dom-soln4.html](#)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Manière #2

Les éléments DOM ont un attribut `onclick`. On peut assigner comme valeur de cet attribut la **fonction** qui gère l'événement.

```
function mafonction() {...}
var p = document.getElementById("monelement");
p.onclick = mafoonction;
```



[handlers-dom-soln3.html](#)

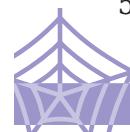


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Séparation des fonctionnalités

On peut complètement éliminer le code JavaScript du `body`. Il suffit de...

1. Enregistrer les gestionnaires via la **propriété `onclick`** plutôt que l'attribut HTML `onclick`
2. Placer tout le code dans la balise `<script>` de la section `<head>`
3. Placer tout le code d'initialisation à la fin de la balise `<script>` dans une fonction `initialise()`
4. Ajouter à la toute fin du code JavaScript l'instruction `window.onload = initialise;`
5. Encore mieux: envoyer tout ce code dans un fichier `.js` externe et ne faire qu'une inclusion



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Avantages:

- La page web est formée de trois composantes **séparées**:

1. Le contenu (HTML)
2. La présentation (CSS)
3. Le comportement (JS)



- Le code de chaque composante est centralisé: évite le couplage et les erreurs

En anglais: **separation of concerns** (intraduisible en français!)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Autres événements DOM

Autres événements**change**

Lorsque le contenu d'un élément est modifié (par l'utilisateur ou autrement)

focus, blur

Lorsqu'un élément obtient/perd le focus

keydown, keyup, keypress

Pression de touches clavier

load, unload

Lorsqu'un élément est chargé/déchargé de la page

reset, select, submit

Pour éléments de formulaires



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Événements souris**click, dblclick**

Réponse aux clics de souris

mousedown, mouseup

Lorsqu'un bouton est pressé/relâché

mouseover, mouseout

Lorsque la souris arrive au-dessus / quitte un élément

mousemove

Mouvement de la souris au-dessus d'un élément

 L'attribut HTML de chaque événement est précédé de "on" (ex.: onclick, onmousemove, etc.)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Événement implicite

Rappel: tous les événements sont des objets de type **Event**...

- Lorsqu'un événement se produit, JavaScript génère implicitement un objet **Event**
- On peut y référer dans l'enregistrement du gestionnaire via la variable **event**

Ex.: `<body onclick="mafonction(event);">`

- Le gestionnaire (ici **mafonction**) peut référer aux propriétés de l'événement

 Liste des propriétés



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Quelques exemples d'utilisation des propriétés des événements...

1. Déetecter la position de la souris lors d'un clic

 where.html

2. Afficher un message sous le pointeur lorsqu'on appuie sur le bouton

 anywhere.html

3. Déetecter la touche pressée au clavier



Bouger des éléments

Un exemple: déplacer un bloc de texte de la position (100,100) jusqu'à la coordonnée (300,300) de la page

 moveText.html

On doit:

- Utiliser l'attribut `onload` de l'élément `body` pour récupérer la position initiale du bloc
- Technicalité: les attributs `top` et `left` sont des `String`, qui contiennent les unités (ex.: "150px"): il faudra les « convertir »
- Utiliser une fonction `move` pour changer les attributs `top` et `left` d'un pixel dans la direction de la cible
- Rappeler `move` à intervalles de 1 ms jusqu'à l'atteinte de la cible



On peut déplacer un élément de la page en modifiant ses propriétés `top` et `left`, mais ce déplacement est **instantané**... peut-on le rendre progressif?

Il suffit de le déplacer légèrement, à plusieurs reprises sur un intervalle de temps!

La fonction `setTimeout` appelle une fonction après un délai

 Appel de la fonction
 Délai en ms
Ex.: `setTimeout("mafonction()", n)`



Attention!

L'appel de la fonction est une `String`, qui peut contenir des paramètres, ex.: "mafonction(" + x + ")"



DOM 0 vs. DOM 2

Le modèle d'événements présenté jusqu'ici appartient au niveau « zéro » de la norme DOM

- **Supporté** par les navigateurs, mais **absent** des dernières spécifications
- DOM niveau 2 offre un modèle d'événements plus complet
- DOM 2 n'est pas supporté par IE, au moins jusqu'à IE8...

...IE le remplace par son propre modèle d'événements différent 

On n'en parlera pas



Un événement est un objet de type Event, qui possède les propriétés et méthodes suivantes:

target

Cible de l'événement (i.e. élément DOM concerné)

type

Type d'événement: "focus", "load", etc.

stopPropagation

Arrête la propagation de l'événement après l'élément concerné

preventDefault

Si on appelle cette méthode, on empêche l'action par défaut de l'événement (ex.: empêche un formulaire d'être envoyé)

currentTarget

Noeud sur lequel l'événement se trouve actuellement



Pour intercepter un événement DOM 2 dans un élément, on doit lui attacher un gestionnaire d'événement avec la méthode `addEventListener()`

```
function envoiform(event) {
  if (this.elements.adresse.value === "") {
    alert("L'adresse est vide.");
    event.preventDefault();
  }
}

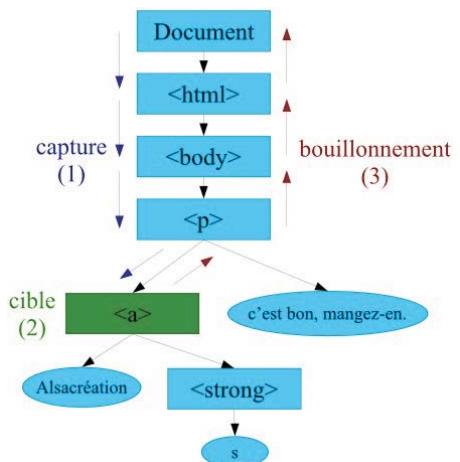
var formulaire = document.getElementById("coordonnees");
formulaire.addEventListener("submit", envoiform, false);
```

Type de l'événement à intercep... Gestionnaire d'événement true si traité durant la phase de capture, false durant bouillonnement



Un événement DOM 2 se propage en trois phases. Par exemple, si on clique sur `<a>` dans le document suivant:

```
<!DOCTYPE html ...>
<html>
<head>
  <title>Publicité</title>
</head>
<body>
  <p>
    <a href="page.html">
      Alsacrération
      <strong>s</strong></a>
      c'est bon, mangez-en.
    </p>
  </body>
</html>
```

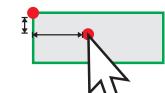


Exemple: les aimants de frigo

dragNDrop.html

On doit enregistrer l'événement `mousedown` de chaque mot. Lorsqu'il se produit, il faut...

- Déterminer l'élément cliqué, sa position et son décalage avec celle du pointeur (pourquoi?)
- Enregistrer un gestionnaire pour `mousemove` et `mouseup` sur cet élément
- Dans `mousemove`: place l'élément vis-à-vis du curseur*
- Dans `mouseup`: détruit les gestionnaires pour `mousemove` et `mouseup` de l'élément



L'objet `navigator` indique le... navigateur utilisé

- La propriété `appName` contient le nom du navigateur
- La propriété `appVersion` contient le numéro de version

Mais...

- Dans IE8, `appVersion` retourne "4.0"
- Dans Firefox 3, `appVersion` retourne "5.0" et `appName` retourne "Netscape"



`navigate.html`

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Bibliothèques JavaScript

JavaScript fournit des fonctions **suffisantes** mais **primitives** pour manipuler les éléments et les événements DOM

Il existe des bibliothèques de fonctions et d'objets JavaScript pré-programmés pour...

1. simplifier les tâches courantes
2. étendre les fonctionnalités de base de JS

Quelques bibliothèques populaires:



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

- Le modèle DOM est une interface qui permet d'accéder à un document HTML via un langage de programmation (= **API**)
- Les éléments d'une page sont exposés sous forme d'**objets** que l'on peut manipuler via leurs propriétés et méthodes
- Manière rapide d'obtenir un élément: `getElementById()`
- Pour modifier le style (CSS) d'un élément, on utilise la propriété **style** de l'élément
- On réagit à événement dans un objet lui enregistrant un **gestionnaire d'événements**
- La manière la plus propre consiste à passer par la **propriété** de l'objet (plutôt que dans la balise HTML)



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Bibliothèques JavaScript

Choisir un élément

- Il n'y a pas de `getElementsByClassName` en JS...
- Dans Prototype, on peut utiliser `$(x)`, où `x` est un sélecteur CSS quelconque

Animer des éléments

- La bibliothèque `script.aculo.us` prédéfinit des dizaines d'effets applicables sur des éléments

Certaines bibliothèques (YUI, GWT, Cappuccino) vont encore plus loin et recréent des widgets (boutons, fenêtres, etc.) similaires à une application bureau classique!

Google Web Toolkit in Action



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

JavaScript est un langage interprété: le navigateur doit récupérer tout le code source en clair ⇒ gros fichiers!

Ex.: jquery.js fait 155 ko
dojo.js fait 123 ko

Solution 1: demander au serveur de compresser sa réponse
HTTP ⇒ nécessite configuration du serveur

Solution 2: supprimer les espaces et commentaires inutiles
dans la source + renommer les variables et les fonctions par
des chaînes les plus courtes possible: on appelle le procédé
la **minification** d'un fichier JS

jquery.js minifié fait 72 ko
dojo.js minifié fait 65 ko



JavaScript

III. Asynchronous JavaScript and XML (Ajax)



Historiquement...

- IE4 et Netscape4 ont développé un élément (non-standard) appelé iframe
- Un iframe invisible permettait d'envoyer des requêtes à un serveur via JavaScript et en récupérer la réponse
- Microsoft a introduit les objets XmlDocument et XMLHTML dans IE5; tous les navigateurs modernes possèdent un objet similaire
- Intérêt accru de la technique à cause de...
 1. L'apparition de Google Maps et Google Mail
 2. JJ Garrett lui donne le nom Ajax



Aperçu d'Ajax spider

Objectif:

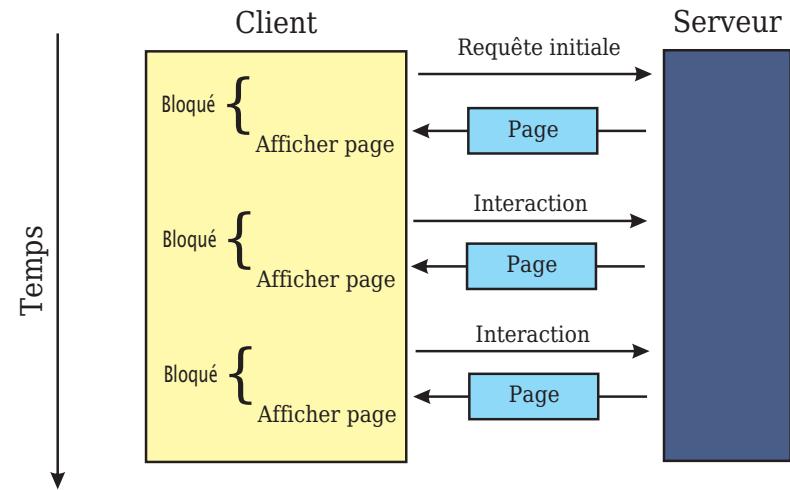
- Donner aux applications web une vitesse de réponse s'approchant de celle d'une application bureau
- Améliore surtout les applications qui interagissent fréquemment avec un serveur

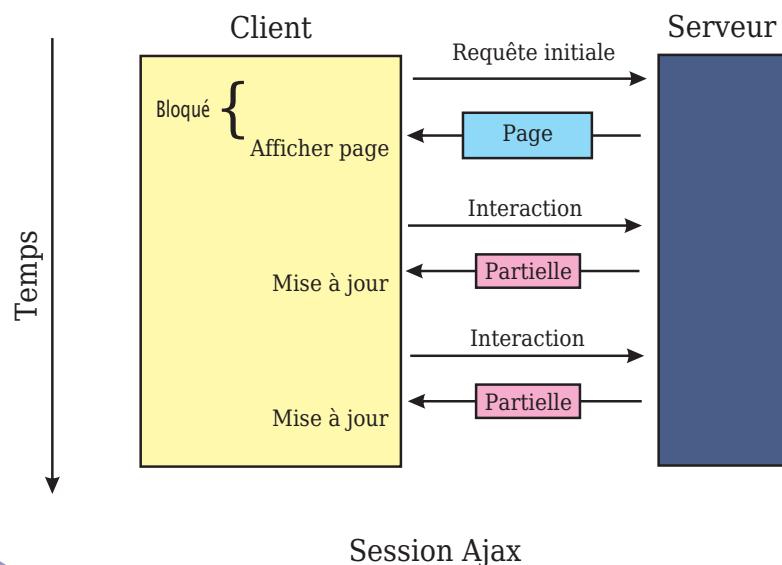
L'objectif est atteint en...

- Traitant les requêtes de manière **asynchrone**: l'application continue de fonctionner en attendant la réponse à une requête
- **Mettant à jour** seulement une petite partie du document à chaque interaction



Ajax vs. page traditionnelle spider

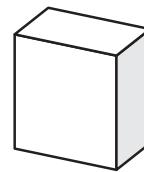




Session Ajax

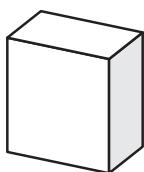
Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Ajax vs. page traditionnelle

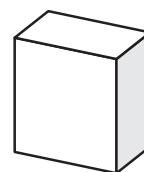


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Ajax vs. page traditionnelle

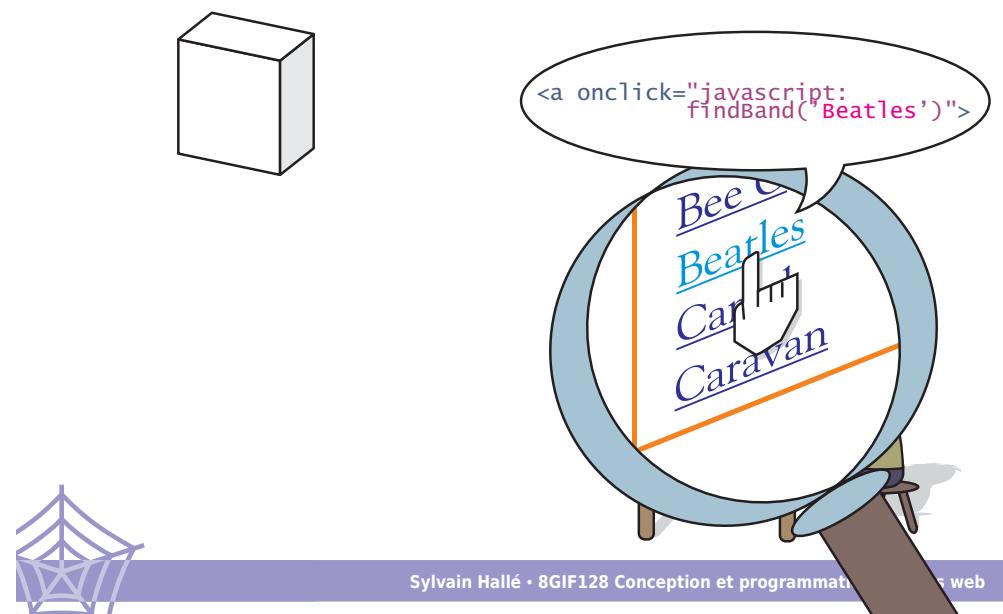


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

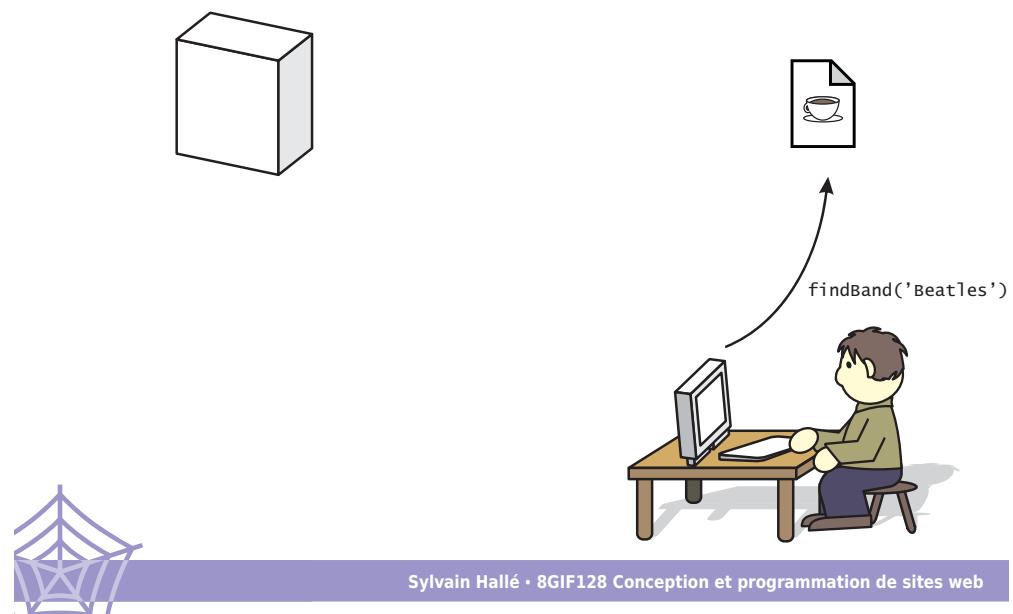


Sylvain Hallé • 8GIF128 Conception et programmation de sites web

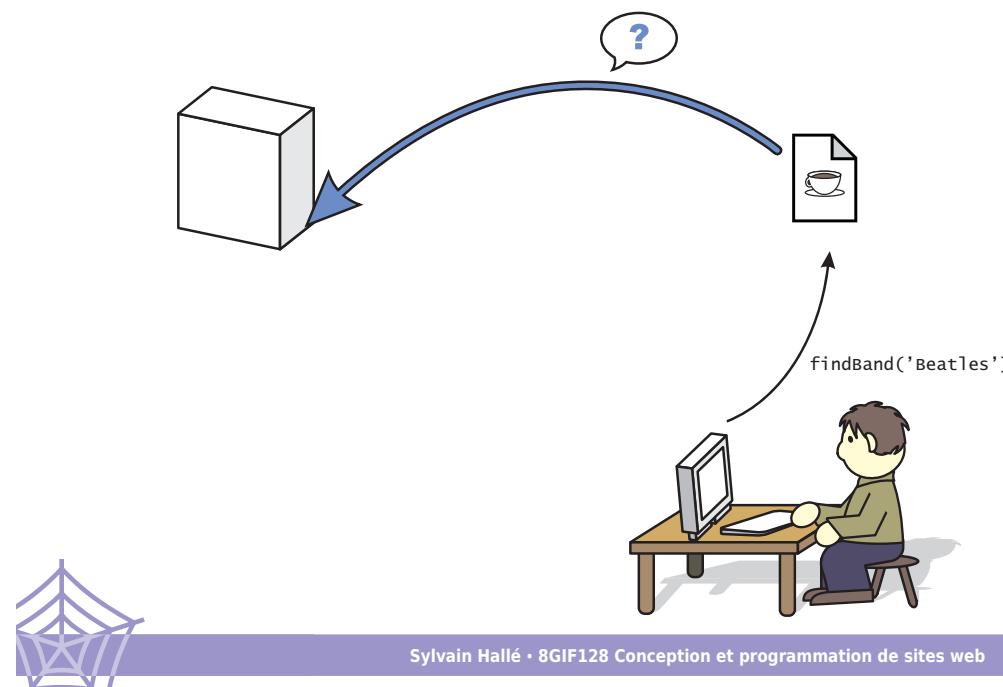
Ajax vs. page traditionnelle



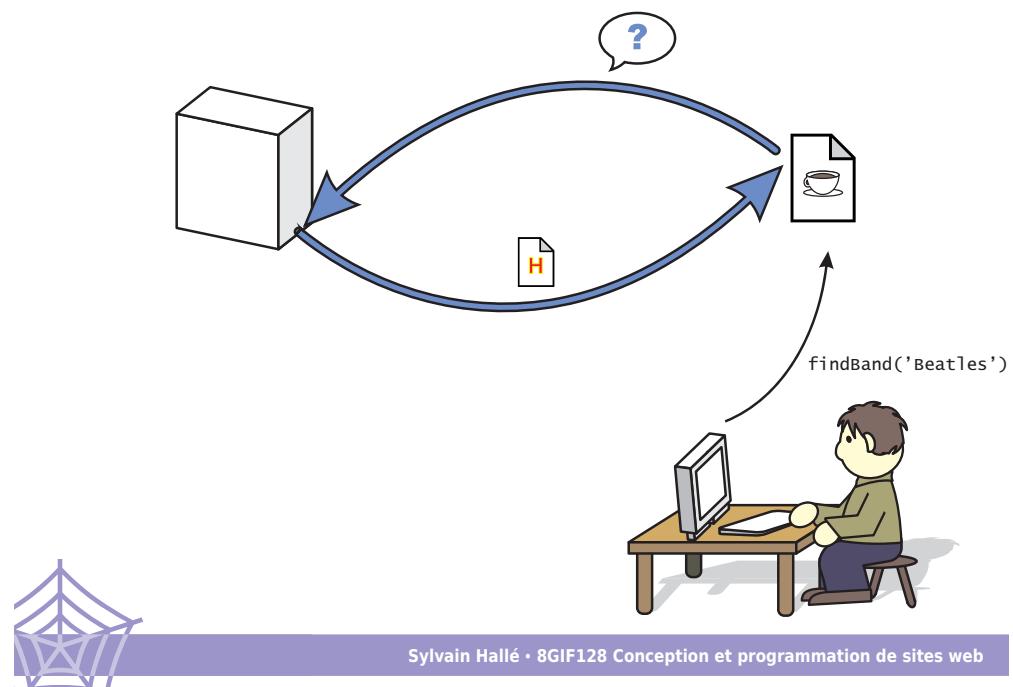
Ajax vs. page traditionnelle

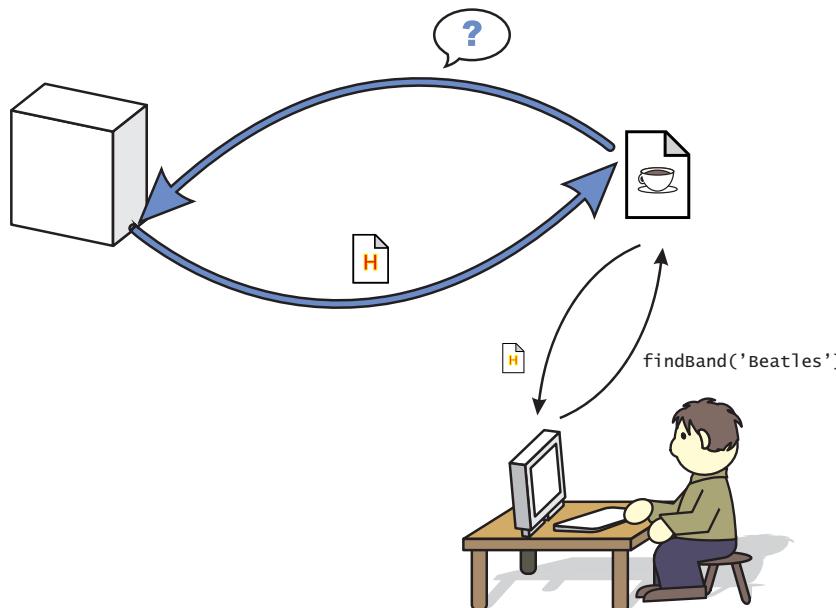


Ajax vs. page traditionnelle



Ajax vs. page traditionnelle





Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Une première application Ajax

Une application Ajax est formée de quatre parties:

1. Un document XHTML (+ CSS) pour produire l'affichage initial
2. Un script JavaScript pour envoyer une requête au serveur
3. Un script sur le serveur pour recevoir la requête et produire la réponse
4. Un script JavaScript pour recevoir la réponse et l'intégrer au document original (déjà affiché)

Un exemple: le formulaire Popcorn



Ajax n'utilise aucun nouveau langage de programmation:

- Sur le client: JavaScript, XML, XHTML DOM, CSS
- Sur le serveur: n'importe quoi!

JavaScript fournit un objet appelé **XMLHttpRequest** pour gérer les communications côté client...

...mais on peut aussi utiliser des **bibliothèques JS** (jQuery, Prototype)

...et des **frameworks** (JavaServer Faces, Rails, Adobe Flex)

Commençons par **XMLHttpRequest**!

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Une première application Ajax

Le document XHTML

- Quelques éléments de formulaire
- Gestionnaire d'événement **blur** enregistré pour la boîte de texte Zip code
- Appelle fonction JS **getPlace** avec le texte saisi

Le script JS

- Contient...
- Le gestionnaire d'événement **blur** (**getPlace**)
- La fonction pour envoyer la requête
- La fonction pour recevoir la réponse et mettre à jour l'affichage

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Pour envoyer une requête:

1. On initialise un objet XMLHttpRequest (XHR)

```
var xhr = new XMLHttpRequest();
```

Lorsque XHR envoie une requête, son état change plusieurs fois. On peut interroger l'état via la propriété `readyState`:

- 0: requête pas initialisée
- 1: connexion au serveur établie
- 2: requête reçue
- 3: requête en traitement
- 4: requête terminée, réponse prête



3. On prépare l'envoi de la requête en appelant la méthode `open` de XHR:

```
xhr.open(methode, url, async);
```

methode

Chaîne de caractères, soit "GET" ou "POST"

url

Chaîne de caractères correspondant à l'URL de la requête, ex.: "http://www.xyz.com/page.php"

async

Valeur booléenne; `true` si la requête est asynchrone, `false` si synchrone (on utilise très rarement `false`)



La réponse est arrivée lorsque `xhr.readyState == 4`... mais comment faire pour savoir lorsque c'est le cas?

Un objet XHR appelle une fonction chaque fois que son état change (*callback function*).

2. On lui spécifie la fonction à appeler via la propriété `onreadystatechange`:

```
mafonction = function() {...};  
xhr.onreadystatechange = mafonction;
```



Si la méthode choisie est GET, on peut passer des paramètres directement dans l'URL:

`http://www.xyz.com/page.php?param1=valeur1¶m2=valeur2`

Sépare la page des paramètres

Sépare le paramètre de sa valeur

Sépare les couples param=valeur

Dans l'exemple Popcorn, on doit passer un code postal via le paramètre zip:

`getCityState.php?zip=80908`



4. On envoie la requête au serveur en appelant la méthode `send` de XHR:

```
xhr.send(null);
```



Paramètres passés avec la requête

On peut passer à `send...`

- Une chaîne de caractères contenant une liste de couples paramètre=valeur
- Un objet DOM

...seulement si on utilise POST comme méthode!



...définir la fonction directement dans le scope de `xhr`, comme ceci:

```
var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function() {
    // Ici, on peut référencer à xhr
    if (xhr.readyState == 4 && xhr.status == 200)
    {
        // Traitement de la réponse
    }
};
```

- Si la réponse du serveur est en format texte, on peut l'obtenir avec la propriété `responseText`
- Si la réponse est en format XML, on peut l'obtenir avec la propriété `responseXML`



(Ici, le serveur traite la requête et prépare sa réponse. On verra plus tard comment.)

5. Lorsque la réponse est arrivée, XHR appelle la fonction qu'on lui a définie à l'étape 2

- Sa propriété `readyState` est égale à 4
- Sa propriété `status` est égale à 200 (HTTP OK) ou à 404 (HTTP Not Found)

Problème: comment la fonction peut-elle accéder à ces propriétés?

On peut définir `xhr` comme variable globale (pas propre), ou...



Une fois la réponse obtenue, on extrait de la chaîne ou du XML reçu les informations nécessaires à la mise à jour dynamique de certains éléments de la page.

Dans l'exemple Popcorn, le serveur retourne une réponse texte de la forme "nom d'une ville, état"

Ex.: Aspen, Colorado

On casse la réponse en deux pour remplir les champ City et State du formulaire

 Si le texte retourné contient plusieurs éléments de données dans un format particulier, quelle est la meilleure technique pour les extraire? (Indice: on l'a vu)



On peut observer les requêtes/réponses envoyées par les objets XMLHttpRequest dans le panneau Réseau du module Firebug

URL	Status	Domaine	Poids	Chronologie
GET getCityState.j	200 OK	localhost	22 B	
GET getCityState.j	200 OK	localhost	22 B	

Réponse

```

Date: Wed, 23 Mar 2011 20:22:26 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_...
X-Powered-By: PHP/5.3.1
Content-Length: 22
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

Requête
Host: localhost
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; fr; rv:1.9.1.17) Gecko/20110123 SeaMonkey/2.0.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr-ca,en-ca;q=0.8,fr;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Keep-Alive: keep-alive
Connection: keep-alive
Referer: http://localhost:8GIF128/w6code10/popcornA.html?name=&street=&zip=80908&city=Black+Forest&s...
Referer: rado
  
```

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Formats de retour

Dans l'exemple précédent, le serveur retourne une chaîne simple de texte clair. D'autres formats de retour sont possibles:

1. Le serveur retourne du code XHTML

```

<h2>2007 US Champions – baseball</h2>
<ul>
  <li>Boston Red Sox</li>
  <li>Colorado Rockies</li>
</ul>
  
```

- Approche classique: on crée un div vide dans le document original. Ex.:
- On utilise sa propriété innerHTML pour y affecter la chaîne retournée par le serveur

```

var divDom = document.getElementById("la-div-vide");
divDom.innerHTML = xhr.responseText;
  
```

Sylvain Hallé • 8GIF128 Conception et programmation de sites web

La méthode décrite précédemment ne fonctionne pas avec les navigateurs IE avant la version 7 😞

En fait, ils offrent un objet identique, mais qui s'instancie différemment:

```
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
```

On peut tester cette condition via le code suivant:

```

if (window.XMLHttpRequest)
  // vrai si XMLHttpRequest existe, i.e. IE7+ et FF
  xhr = new XMLHttpRequest();
else
  // vrai dans IE5 et 6
  xhr = new ActiveXObject("Microsoft.XMLHTTP");
  
```



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

Formats de retour

Désavantages:

- Utile seulement si le HTML reçu peut être utilisé **tel quel**
- Si on ne veut que les données, on doit les extraire du HTML (compliqué)
- **Couplage** entre la présentation (client) et la fonctionnalité (serveur): si le code réfère à des classes CSS définies par le client?



Sylvain Hallé • 8GIF128 Conception et programmation de sites web

2. Le serveur retourne du code XML

```
<champions>
  <annee>2007</annee>
  <sport>baseball</sport>
  <equipes>
    <equipe>Boston Red Socks</equipe>
    <equipe>Colorado Rockies</equipe>
  </equipes>
</champions>
```

Avantage: le serveur retourne des **données**; le client s'occupe de les présenter comme il le souhaite

Inconvénient: il faut extraire les données de la réponse!

- On peut utiliser des méthodes DOM (un peu comme pour XHTML): varie selon les navigateurs
- On peut utiliser une feuille XSLT (on verra)



Règles de syntaxe JSON:

- Un **objet** est entouré d'accolades et contient des couples paramètre-valeur séparés par des virgules
- Un **couple** paramètre-valeur est de la forme **a : b**, où a est une chaîne de caractères et b est soit...
 - un objet
 - un tableau
 - une String ou un nombre
- Un **tableau** est encadré par des crochets et contient des objets séparés par des virgules
- L'élément de niveau supérieur est un objet



3. Le serveur retourne un objet JSON

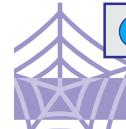
```
{"champions" :
  {
    "annee" : "2007",
    "sport" : "baseball",
    "equipes" :
    [
      {"equipe" : "Boston Red Socks"},
      {"equipe" : "Colorado Rockies"}
    ]
  }
}
```

JSON = JavaScript Object Notation

Manière textuelle de représenter un objet JavaScript en utilisant deux types de structures: des collections de paires attribut-valeur ET des tableaux de valeurs



Un objet JSON est une String structurée



JSON et XML se ressemblent beaucoup...

```
{"champions" :
  {
    "annee" : "2007",
    "sport" : "baseball",
    "equipes" :
    [
      {"equipe" : "Boston Red Socks"},
      {"equipe" : "Colorado Rockies"}
    ]
  }
}
```

JSON

```
<champions>
  <annee>2007</annee>
  <sport>baseball</sport>
  <equipes>
    <equipe>Boston Red Socks</equipe>
    <equipe>Colorado Rockies</equipe>
  </equipes>
</champions>
```

XML

...mais JSON peut être converti directement en un objet JavaScript!



Première méthode: utiliser la fonction JavaScript `eval`:

```
var json = '{"champions" : ...}';  
var objet = eval(json);
```

On peut ensuite référer aux propriétés de l'objet facilement:

```
var e = objet.champions.equipes[0].equipe;  
↑  
e = "Boston Red Socks"
```

Danger!

La fonction `eval` évalue n'importe quelle chaîne JS. Si la chaîne contient un script, il sera exécuté! (Sécurité)

Deuxième méthode: utiliser un parser JSON

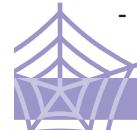
 <http://www.json.org/json2.js>

```
var json = '{"champions" : ...}';  
var objet = JSON.parse(json);
```

`JSON.parse` s'assure que la chaîne est bien un objet JSON valide

Avantages de JSON:

- Plus petit que XML (en général): transmission plus rapide
- `JSON.parse` est bien plus rapide qu'un traitement manuel de XML...
- ...et plus simple!



Bibliothèques JavaScript

Les bibliothèques JavaScript peuvent aussi simplifier l'utilisation d'Ajax

Exemple avec Prototype:

```
new Ajax.request("getCityState.php",  
{  
    method: "get",  
    parameters: "zip=" + zip,  
    onSuccess: function(request)  
    {  
        var x = request.responseText;  
        //...  
    },  
    onFailure: function(request)  
    {  
        alert("Requête sans succès");  
    }  
});
```

