

Introduction

IO2

Internet et outils

Cristina Sirangelo

IRIF, Université Paris Diderot

[cristina@irif.fr](mailto:cristina@irif.fr)

# Informations relatives au cours

Page Moodle du cours :

<http://moodlesupd.script.univ-paris-diderot.fr/course/view.php?id=6734>

pour :

Les supports de cours

Les sujets de TP

La soumission du projet

Les annonces et les informations de dernière minute

S'inscrire !

# Contrôle des connaissances

## Projet : en binôme

à réaliser en dehors des séances de TP

à soutenir en fin de semestre

soutenance sur machines

## Examen final:

sur papier

## Notation :

1ère session : max de

- examen final
- 35% projet + 65% examen final

2ème session : examen uniquement

le projet donne la note de CC

# Objectifs

Maîtriser les principaux concepts de base sous-jacents à la mise en œuvre d'une application Web

- quelques notions de base liées à Internet
- format HTML des documents
- CSS pour la mise en page / mise en forme des pages Web
- langages PHP (serveur), Javascript (client) pour dynamiser les pages Web
- bases de données relationnelles
- éléments de SQL pour l'interrogation des données
- système de gestion de bases de données MySQL

## Objectif des TPs

Mise en œuvre des connaissances pour réaliser une application Web sous Unix (FreeBSD)

# Quelle définition d'Internet ?

## Les définitions techniques

*“réseau international de communication entre utilisateurs d'ordinateurs”*

*“Réseau des réseaux à l'échelle mondiale qui effectue l'interconnexion d'un grand nombre de réseaux internationaux, régionaux et locaux, tous basés sur un jeu de protocoles commun à savoir TCP/IP”*

*“Internet est le réseau informatique mondial accessible au public. C'est un **réseau de réseaux**, sans centre névralgique, composé de millions de réseaux [...]. L'information est transmise par internet grâce à un ensemble standardisé de **protocoles** de transfert de données, qui permet l'élaboration d'**applications** et de **services** variés comme le courrier électronique, la messagerie instantanée, le pair-à-pair et le World Wide Web.”*

(source : [fr.wikipedia.org/wiki/Internet](https://fr.wikipedia.org/wiki/Internet))

# Quelle définition d'Internet ?

## Internet est un enchevêtrement

**Réseaux** : *Internetworking*, “réseau de réseaux” de machines

**Protocoles** : TCP/IP, HTTP, FTP, SMTP...

**Applications** : messagerie, forums, WWW (le Web), chat, etc.

**Services** : communication, recherche d'information, publications...

**Ressources** : sites web, blogs, bibliothèques numériques...

Confusions fréquentes entre Internet et ses applications (e.g. Web)

Internet != Web (World Wide Web)

Nécessité de bien distinguer les niveaux

**Web** :

une application *fonctionnant sur Internet*

un ensemble de documents et ressources + protocoles

# Réseaux

## Réseau :

Ensemble de machines « physiquement » connectées entre elles

## Connexion :

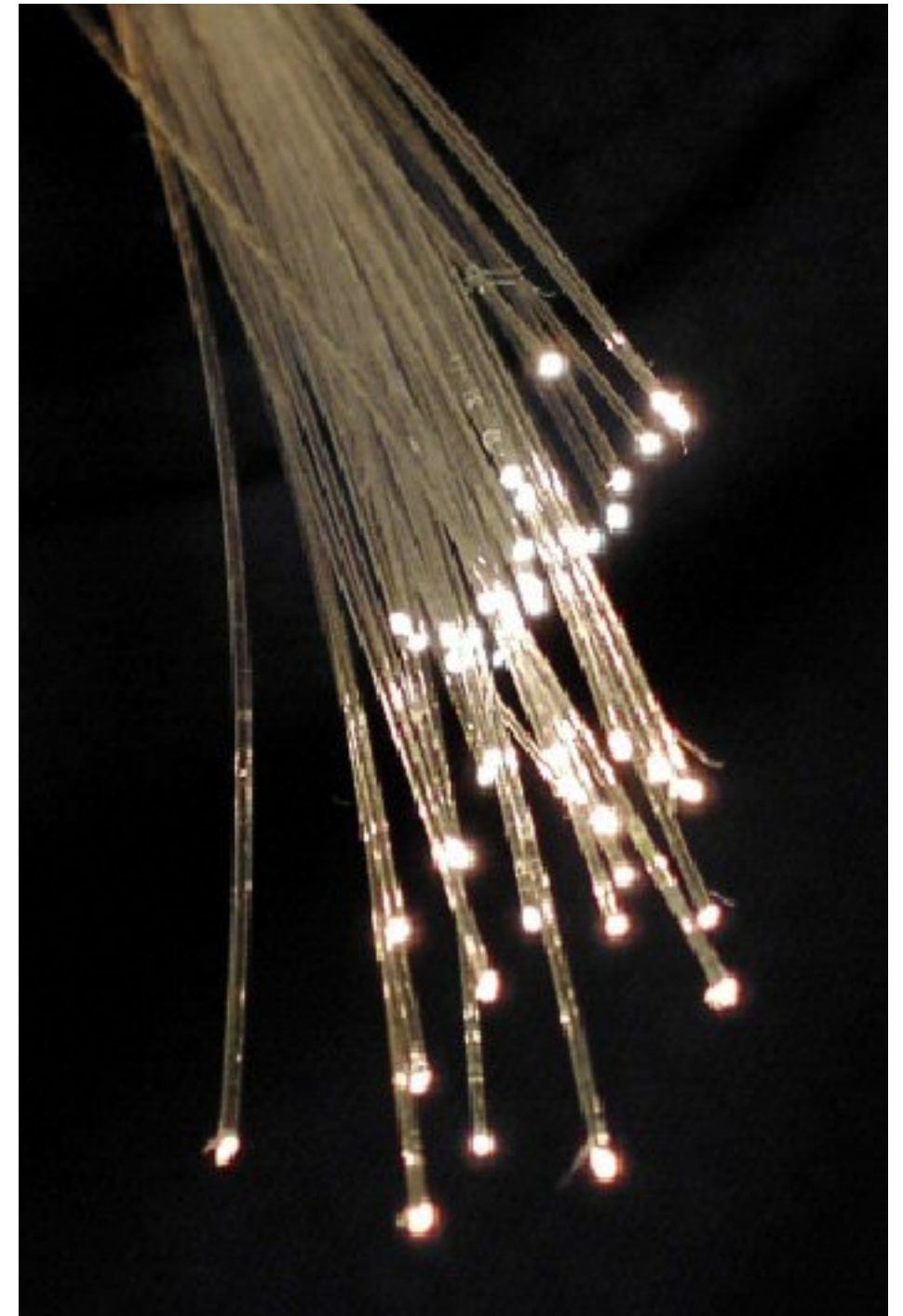
Cable relié à un boîtier spécialisé, ADSL,...

Cables coaxiaux

Fibre optique

Sans fil (WiFi, Bluetooth...)

Autre...



# Rappel : les différents aspects d'Internet

## Internet est un enchevêtrement

**Réseaux** : *Internetworking*, “réseau de réseaux” de machines

**Protocoles** : TCP/IP, HTTP, FTP, SMTP...

**Applications** : messagerie, forums, WWW (le Web), chat, etc.

**Services** : communication, recherche d'information, publications...

**Ressources** : sites web, blogs, bibliothèques numériques...



# Protocoles

- Il est nécessaire de coopérer !
- La mise en réseau d'ordinateurs ne consiste pas seulement à s'échanger des octets : c'est un système complexe avec différents outils et fonctionnalités
  - routage : établir des chemins de communications, acheminer de manière raisonnablement fiable et rapide les informations d'une machine à l'autre
  - détection d'erreurs
  - encryption, etc.
- Pour communiquer efficacement les entités de différents systèmes doivent parler la même langue
- Ces conventions et les règles associées sont appelées “**protocoles**”

# Protocoles

Différentes couches de protocoles assurent différentes fonctions

Réseau (**IP** - *Internet Protocol*)

assurer l'échange de paquets de données

Transport (**TCP** - *Transmission Control Protocol*

ou **UDP** - *User Datagram Protocol*)

assurer l'échange de messages entre services

Application (e.g. **SMTP** ou **HTTP**, ...)

opérer à plus haut niveau pour offrir un service à l'utilisateur

# Applications et Services

## **Quatre grands types d'applications et de services sur Internet**

*Messagerie* : messagerie individuelle et collective, chat, réseaux sociaux, forums,...

*Services de traitement informatique* : connexion à distance, transfert et partage de fichiers, peer-to-peer...

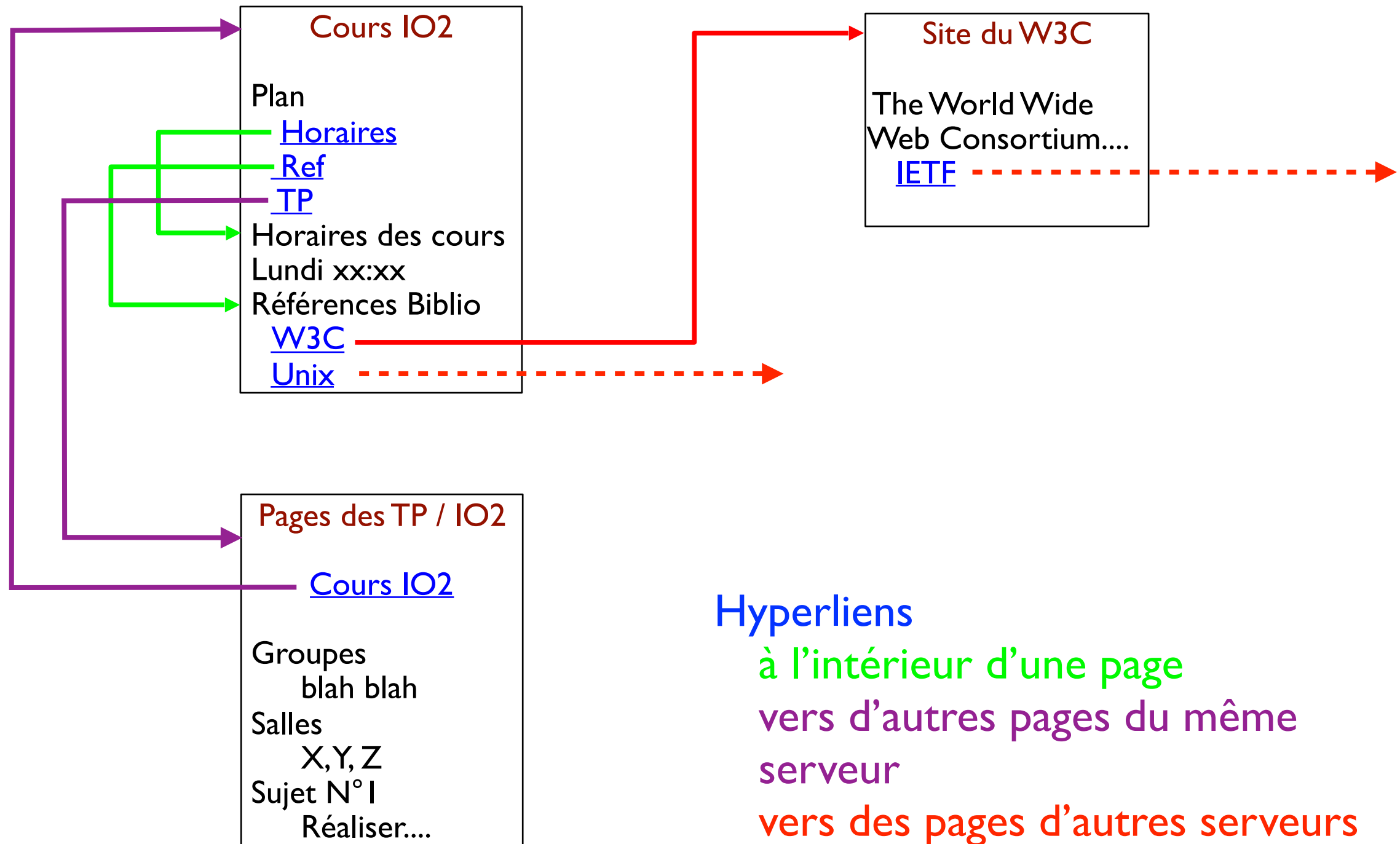
*Services de publication/information* : Web, sites collaboratifs (wiki), ...

*Services de communication directe* : téléphonie, visioconférence, ...

Pendant longtemps, applications bien distinctes ; aujourd'hui fusionnées sur le Web

**Dans ce cours** : développement d'applications pour le Web

# Web : Documents et Hypertexte



# Hypertexte

Texte / images / contenu multimedia

Avec directives de mise en page et mise en forme :

Couleur, taille, police, colonnes, tableaux...

**Liens** : textes/zones sensibles “au clic” référençant un autre document ou une partie d'un document

Les références doivent permettre de désigner un autre document / ressource de manière **non ambiguë**

## Contenu

Fixe : par exemple la description du cours IO2

Dynamique : par exemple les prévisions météo du jour

Identique pour tout le monde ou personnalisé

# Serveurs, Clients, Navigateurs

Les documents sont transférés, à la demande d'un utilisateur

Depuis une machine où ils sont stockés :

le “**serveur**”

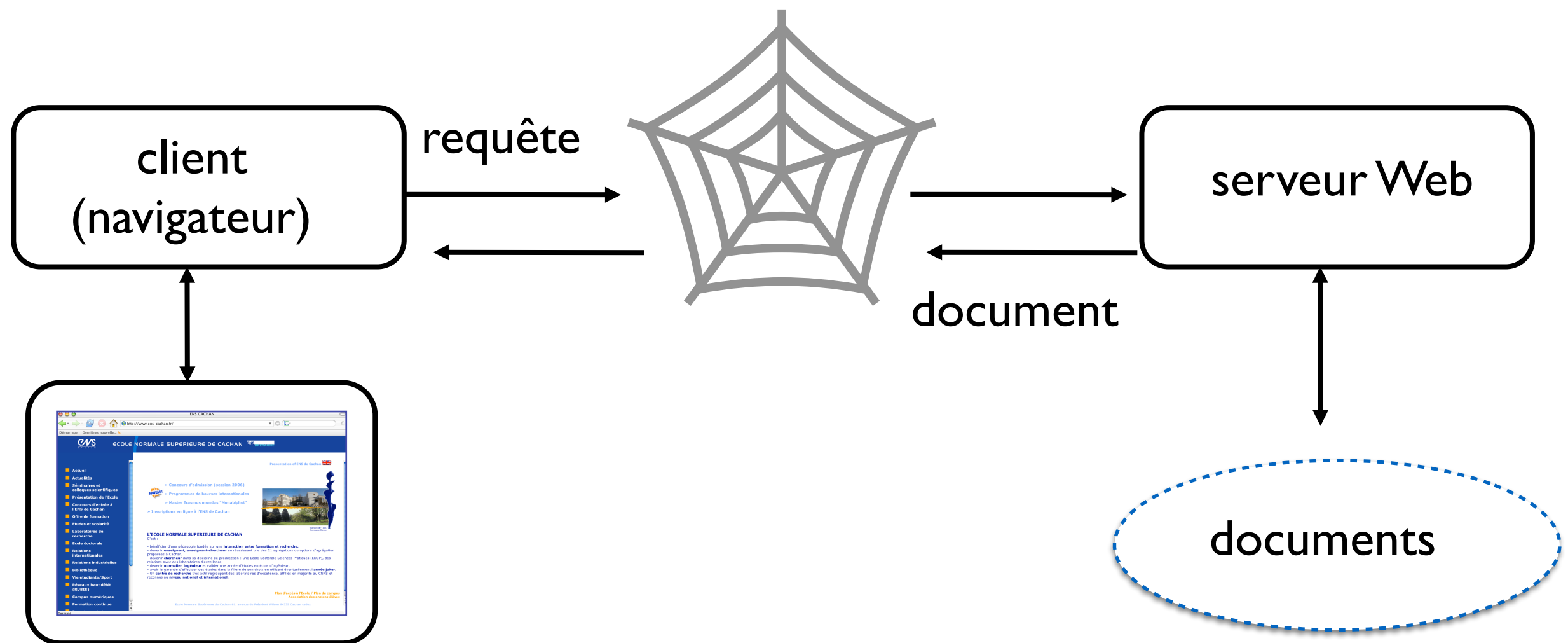
Vers la machine de l'utilisateur consultant le document :

le “**client**”

Et « affichés » par un logiciel spécifique :

le “**navigateur**”

# Serveurs, Clients, Navigateurs



# Problèmes

Faire communiquer les logiciels

qui s'exécutent sur les serveurs avec ceux qui s'exécutent sur les clients

Faire en sorte que les informations transférées soient compréhensibles à leur réception

Langage de description des documents



# Solutions

Faire communiquer client et serveur Web :

**HTTP** : *HyperText Transfer Protocol*

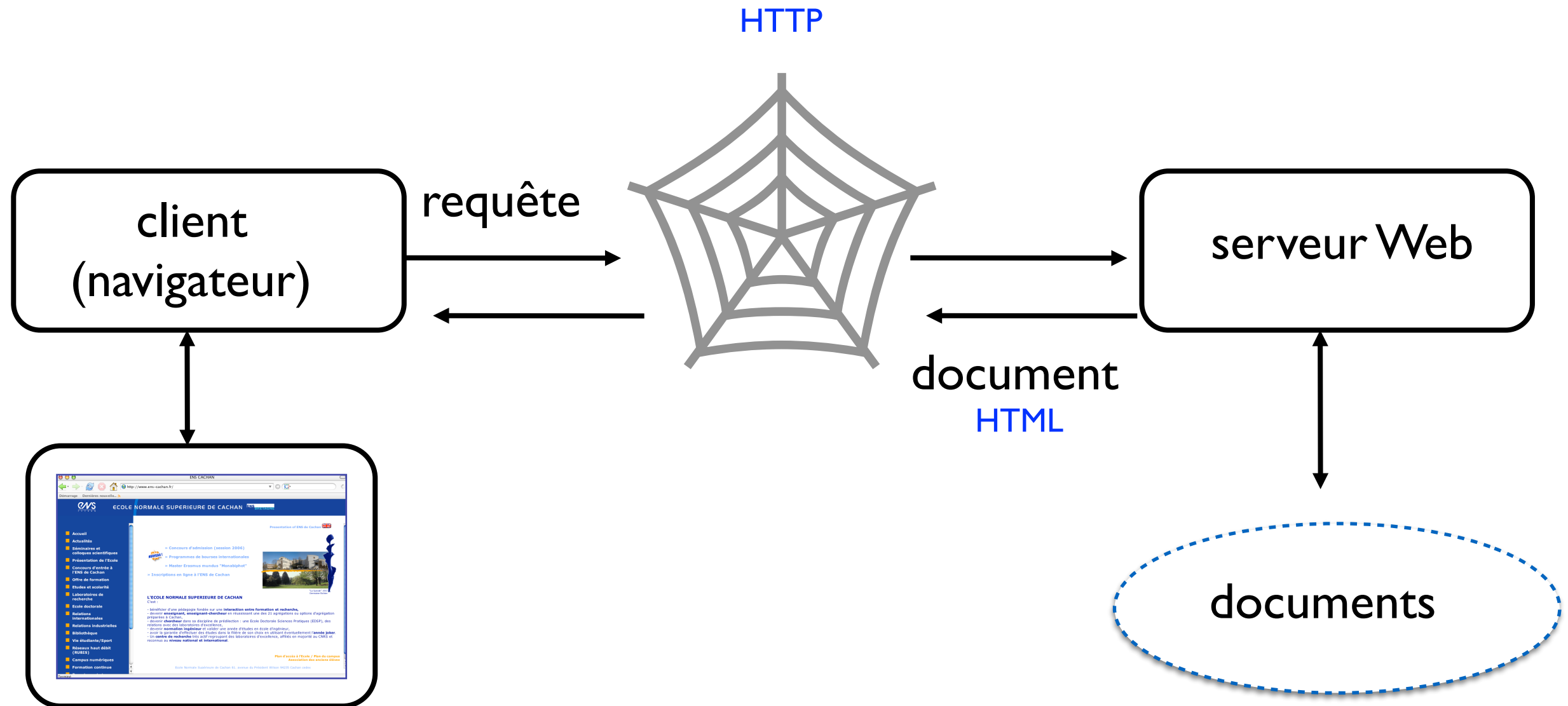
Règles de communication précises pour le transfert des documents

Transférer des informations compréhensibles :

**HTML** : *HyperText Markup Language*

Format standard de descriptions des hypertextes

# Serveurs, Clients, Navigateurs



# HTML : Origines et principe

HTML : un “*markup language*”

Issu de SGML (*Standard Generalized Markup Language*)

De la même famille que XML (*eXtensible Markup Language*)

Principes des “*markup languages*” : Enrichir le document (le texte) avec des informations complémentaires au document (informations sémantiques) :

- Utilisation de “**balises**” (*markup*)
  - Indiquent la fonction/structuration : titre, paragraphe, citation, légende, division...
- HTML permet ainsi de distinguer la fonction du contenu lui-même
- Depuis HTML5 : balises sémantiques additionnelles. Informations supplémentaires sur le rôle des différentes parties de la page
  - en-tête, section, article, etc...

# HTML et CSS

- Premières versions de HTML conçues pour décrire à la fois la structure, le contenu et la mise en forme du document
  - Mise en forme et mise en page : couleur, taille, police du texte, positionnement...
- Aujourd'hui :
  - un document HTML ne devrait contenir aucune information de mise en forme
  - toutes ces informations sont regroupées dans des feuilles de style (CSS) qui lui sont associées
  - CSS : Cascading Style Sheets

# HTML et CSS : les langages du Web

Toutes les pages Web sont donc décrites à la base à travers ces deux langages :

**HTML** : pour organiser le contenu. On décrit ce qui doit être affiché sur la page : du texte, des liens, des images...

**CSS** : pour gérer l'apparence de la page Web. On décrit l'agencement, le positionnement, la décoration, les couleurs, la taille du texte...

## Historique :

HTML a été créé en 1991 par Tim Berners-Lee

Il a également créé le [World Wide Web Consortium](#) (W3C) qui définit les nouvelles versions des langages liés au Web

Le langage CSS est venu compléter le HTML en 1996

En 2000, tentative de lancement de [XHTML](#), échec, abandon en 2009

HTML5 et CSS3 sont les dernières versions, non encore finalisées

# Documents HTML

**Document HTML** : le code qui décrit et spécifie une page Web

- Un fichier textuel “comme les autres” :
- Avec par convention un suffixe .html ou .htm
- Un contenu structuré pour être compréhensible et interprété par un navigateur (*browser*)

Le navigateur peut gérer d'autres types de fichiers (images, sons, vidéos, ...)

Le code HTML d'une page Web peut être visualisé dans chaque navigateur

Ex. code source de [www.airfrance.fr](http://www.airfrance.fr)

# Exemple

01-simple.html

```
<html>

<head>
  <title>
    Exemple simple de HTML
  </title>
</head>

<body>
  <h1>Premier paragraphe</h1>
  <p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit.</p>
</body>

</html>
```

un document HTML

## Premier paragraphe

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

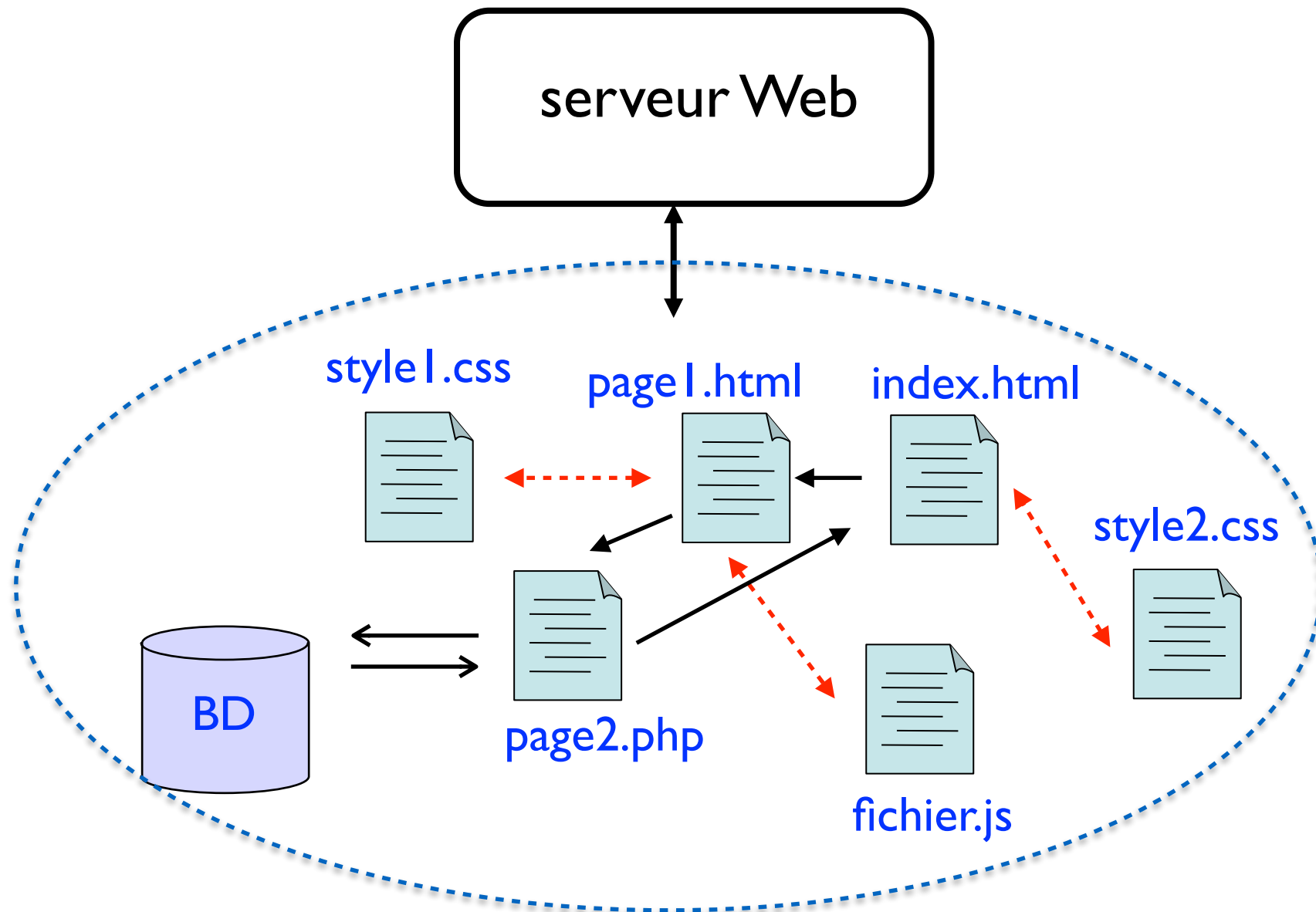
la page Web correspondante  
(affichée par un navigateur qui charge  
le document HTML)

# Structure de base d'une application Web

- Un ensemble de **documents HTML** (les pages Web du site) reliés par des liens
- Potentiellement enrichis avec
  - des **feuilles de style (CSS)** pour définir le style des pages
  - des programmes écrits dans un langage (typiquement **Javascript**) exécutable par le navigateur, qui s'occupent de dynamiser le contenu des pages Web une fois chargées par le navigateur (**HTML dynamique coté client**)
- Un ensemble de programmes écrits dans un langage de script (e.g. **PHP**) exécutable par le serveur Web, qui génèrent des documents HTML dynamiques (**HTML dynamique coté serveur**)
- Possiblement une **base de données** pour le stockage des données d'intérêt de l'application
- Tous ces documents et données : accessibles par un **serveur Web**



# Structure de base d'une application Web



le serveur Web est un logiciel qui tourne en permanence sur une machine (le *host*), pour mettre à disposition ces ressources

- Ex. sur le réseau du SCRIPT un serveur Web tourne sur la machine [pams.script.univ-paris-diderot.fr](http://pams.script.univ-paris-diderot.fr)

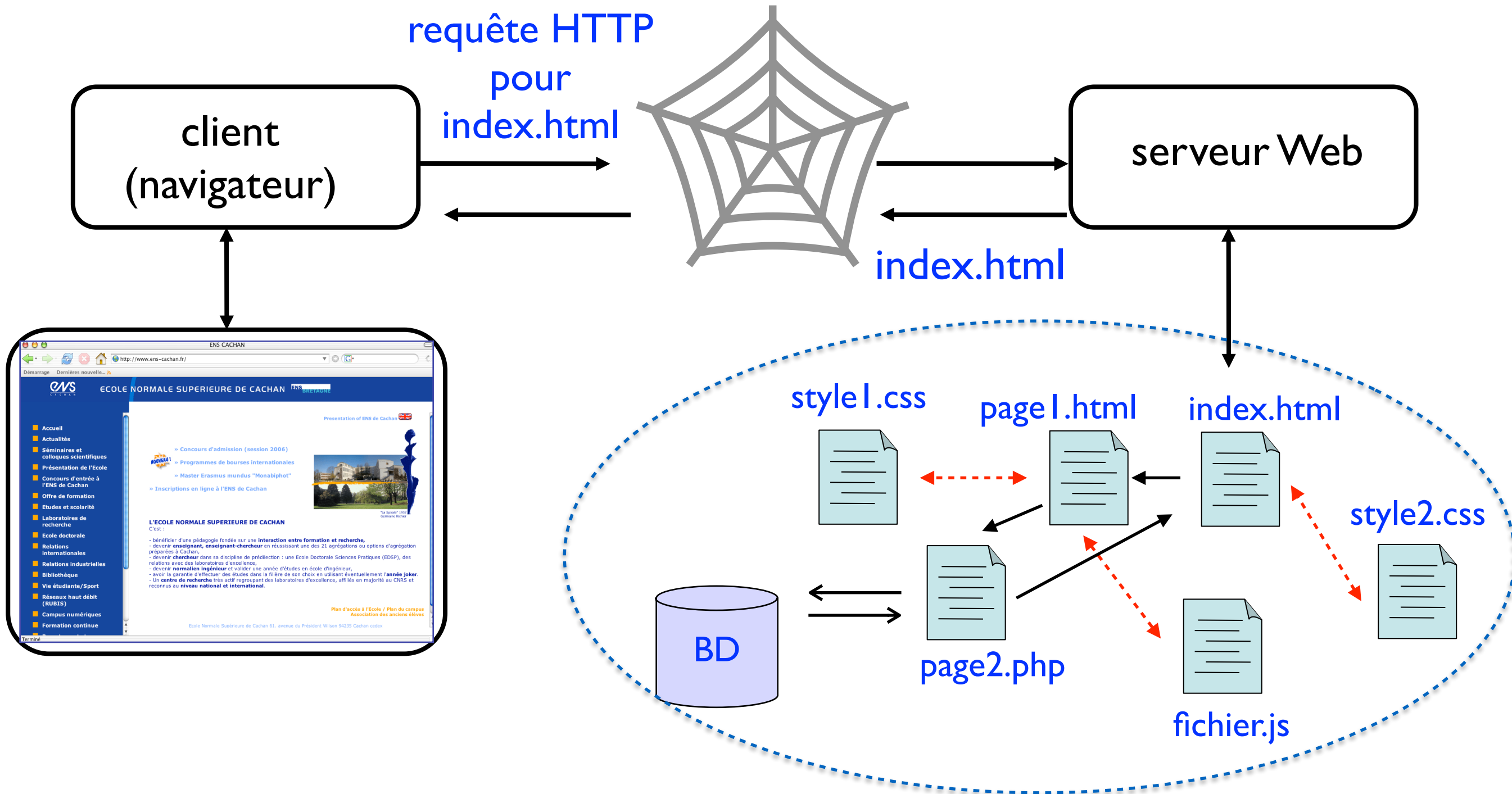
# Structure de base d'une application Web

Essayons de distinguer ces composantes sur un site Web réel

<http://dblp.org>

# Structure de base d'une application Web : le serveur et les ressources

Rappel : pour permettre la navigation sur le site Web, le serveur Web fournit les pages au fur et à mesure sur demande du navigateur (en suivant le protocole HTTP)





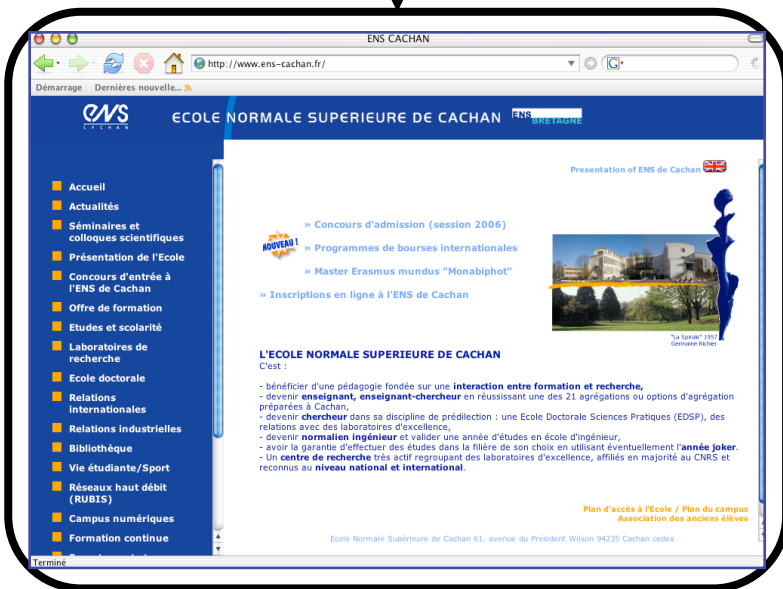
# Structure de base d'une application Web : le serveur et les ressources

<http://pams.script.univ-paris-diderot.fr/chemin/index.html>

client  
(navigateur)

serveur Web sur  
[pams.script.univ-paris-diderot.fr:80](http://pams.script.univ-paris-diderot.fr:80)

index.html



style1.css

page1.html

index.html

style2.css

page2.php

fichier.js

BD

# Plan général prévisionnel

(non contractuel)

- Introduction : Internet, Web, hypertextes, introduction à HTML
- HTML et CSS
- Développement serveur en PHP : bases du langage, gestion de formulaires, sessions, sécurité
- Introduction aux bases de données relationnelles : modèle relationnel, schémas, instances, bases du langage SQL, introduction au système MySQL
- Interaction entre serveur (PHP) et base de données (MySQL)
- (en fonction du temps) Introduction à Javascript
- ...

# Générer un document HTML

## Éditeurs de texte

On édite le texte et on insère les balises manuellement  
emacs, vi, gedit.....

## Outils spécifiques de type *WYSIWIG*

*What You See Is What You Get*

BlueFish (Linux-FreeBSD), Smultron (Mac), BlueGriffon (Linux/MAC/Windows),  
...

**CMS** (Content Management Systems) : systèmes complexes permettant de générer / gérer à la fois la présentation et le contenu dynamique d'un site Web, ainsi que les utilisateurs, les versions, etc...

Joomla, WordPress, Spip, ...

# Introduction à HTML : éléments, balises et attributs

**Élément** : portion bien formée d'un document HTML, avec un nom

e.g. le document : `html` , le corps du document : `body`, un lien : `a`

**Balise** : délimite (ouvre/ferme) un élément

exemple de balise ouvrante : `<html>`

exemple de balise fermante : `</html>`

**Attribut** : propriété d'un élément (nom, valeur)

syntaxe : nom = "valeur" dans la balise ouvrante de l'élément, e.g.

`<a href="http://www.univ-lyon2.fr">`



nom                      valeur



# Règles d'écriture HTML

1. Toute balise ouverte doit être fermée
2. Les chevauchements entre balises sont interdits (documents bien formés)
3. Les valeurs des attributs doivent être entre guillemets doubles

## Bonnes pratiques :

4. Les noms des balises doivent être écrits en minuscule
5. Les noms des attributs doivent être écrits en minuscules

# Règles d'écriture HTML

Chevauchements entre balise interdits :

<del><pre>&lt;body&gt;   &lt;h1&gt;Premier paragraphe &lt;/body&gt;   &lt;/h1&gt;</pre></del>	<pre>&lt;body&gt;   &lt;h1&gt;Premier paragraphe &lt;/h1&gt; &lt;/body&gt;</pre>
---	--

Dans l'exemple :

Puisque l'élément h1 s'ouvre dans l'élément body

(i.e. <h1> entre <body> et </body>)

h1 doit aussi se fermer à l'intérieur de body

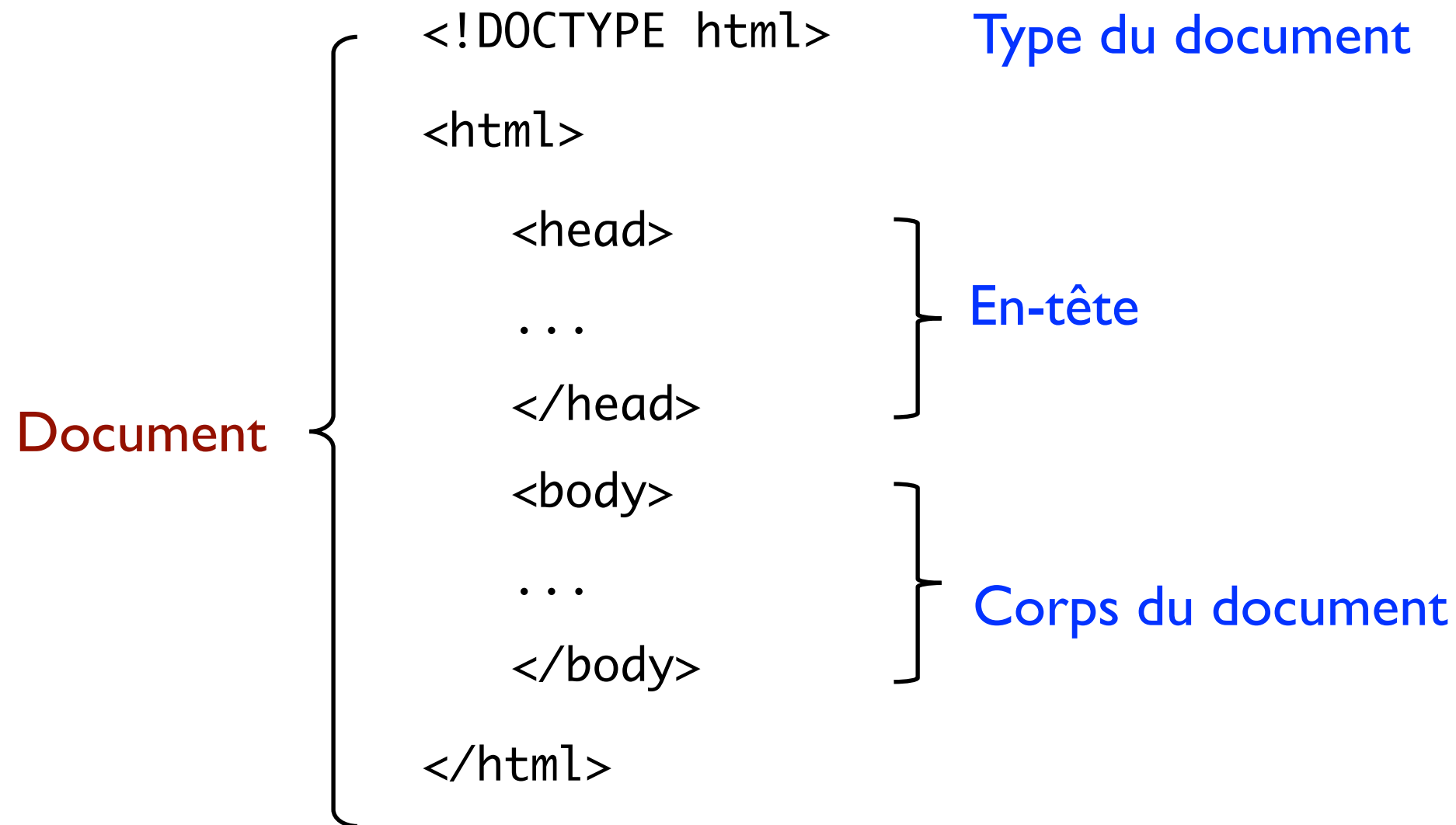
(i.e </h1> après <h1>, mais avant la balise fermante </body>)

# Règles d'écriture HTML

D'où la structure imbriquée (arborescente) d'un document HTML

```
<html>
  <head>
    <title>
      Exemple simple de HTML
    </title>
  </head>
  <body>
    <h1>Premier paragraphe</h1>
    <p>Lorem ipsum dolor sit amet</p>
  </body>
</html>
```

# Squelette d'un document HTML



## En-tête (<head> ...</head> )

Informations qui ne sont pas affichées mais qui sont utilisées à des fins diverses

**Titre** : <title>...</title>

**Métadonnées** : <meta>

Jeux de caractères, auteur, mots-clés, description de la page...

Autres éléments : <style>, <link>, <script> et <base>

**Note** : un élément vide peut se noter <balise></balise> ou <balise />, au choix;

En HTML5 il peut aussi se noter <balise>

Exemples : <meta>, <br>, <hr>, <img>, <input> ...

# Exemple d'en-tête

```
<head>
```

```
<!-- Jeu de caractères universel -->
```

```
<meta charset="utf-8">
```

```
<!-- Autres métadonnées -->
```

```
<meta name="author" content="Cristina Sirangelo">
```

```
<meta name="keywords" content="Enseignement, Informatique">
```

```
<meta name="description" content="Exemple de page HTML">
```

```
<!-- Titre -->
```

```
<title>Un premier exemple</title>
```

```
</head>
```

# Corps du document (<body> ...</body>)

Éléments qui apparaissent explicitement à l'affichage :

Texte

Images

Liens

...

Éléments de structuration du document :

Invisibles

Permettent le regroupement d'éléments à diverses fins :

- mise en forme ultérieure du document
- annotation sémantique

# Exemple du corps d'un document

```
<body>
```

```
<!-- Titre de 1er niveau -->
```

```
<h1>Exemple de page HTML</h1>
```

```
<!-- Paragraphe -->
```

```
<p>
```

```
<a href="http://www.lipsum.com/">Lorem ipsum</a> dolor sit amet,  
consectetur adipiscing elit. Phasellus a diam at purus efficitur  
ultricies tincidunt vel lacus.
```

```
</p>
```

```
<!-- Division -->
```

```
<div>
```

```
<!-- Image -->
```

```

```

```
<p>Phasellus a diam at purus efficitur ultricies tincidunt vel  
lacus.</p>
```

```
</div>
```

```
</body>
```