Autour du Web

Gaëlle Loosli

Polytech, GMM

2011 - v1.1

1. Internet
   - Définitions
   - Données personnelles
   - Architecture des réseaux
     - Architecture client/serveur (2-tiers)
     - Architecture 3-tiers
     - Les clients
     - Architecture Peer to peer

2. Les bases de la publication Web
   - Introduction au webmastering
   - Langage HTML
   - Langage CSS

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web
- HTTPS: version sécurisée

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web
- HTTPS: version sécurisée
- FTP: transférer des fichiers

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web
- HTTPS: version sécurisée
- FTP: transférer des fichiers
- SFTP : version sécurisée

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web
- HTTPS: version sécurisée
- FTP: transférer des fichiers
- SFTP : version sécurisée
- SSH: se connecter à une machine distante

Internet?

Concept de réseau

Relier des ordinateurs pour échanger des données : nécessité d'un langage de communication : les protocoles. Des réseaux hétérogènes se sont développés aux quatre coins du globe. Les protocoles permettent de les réunir pour former le réseau de tous les réseaux : Internet.

Protocoles

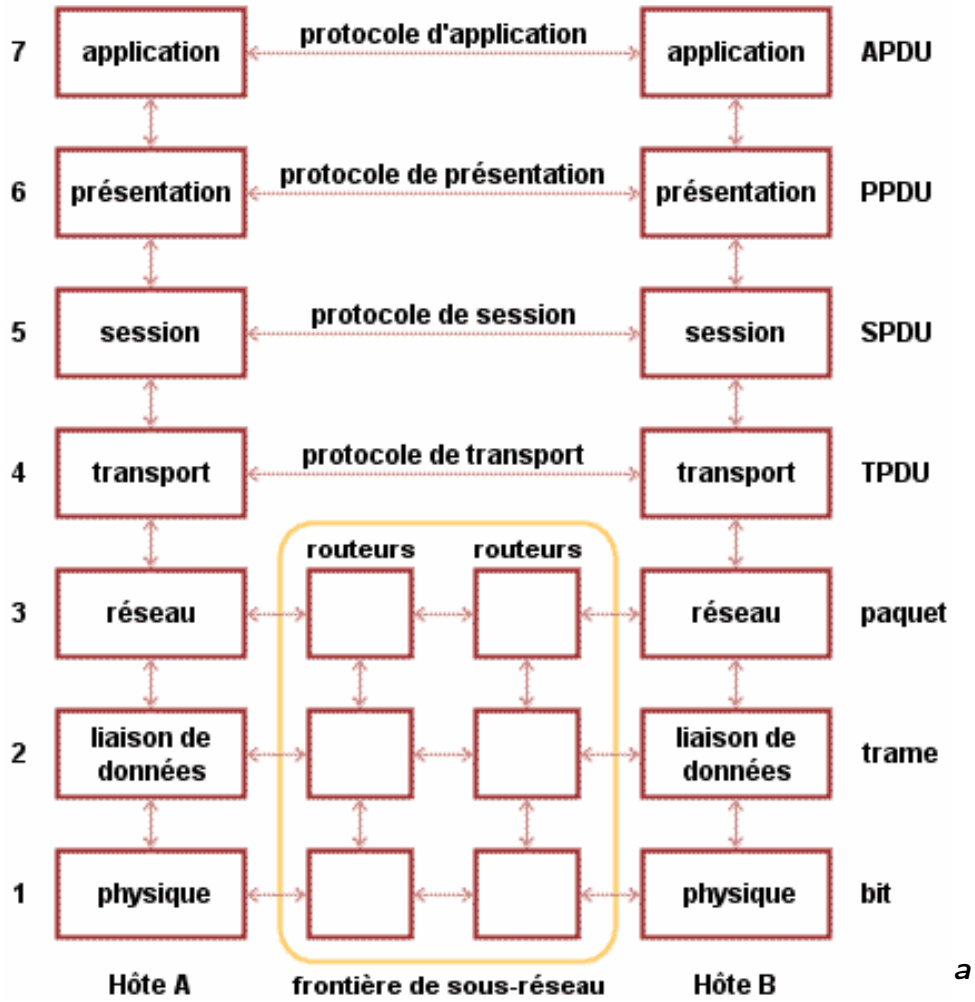
Ceux utilisés sur Internet font partie de la suite TCP/IP, basée sur le repérage de chaque ordinateur par une adresse IP. On a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

- IRC: discuter en direct
- HTTP: regarder des pages web
- HTTPS: version sécurisée
- FTP: transférer des fichiers
- SFTP : version sécurisée
- SSH: se connecter à une machine distante

A chaque protocole est assigné un numéro : le port (exemple : HTTP : 80)

Protocoles

Modèle OSI : 7 niveaux



[a]

[a] source:www.frameip.com

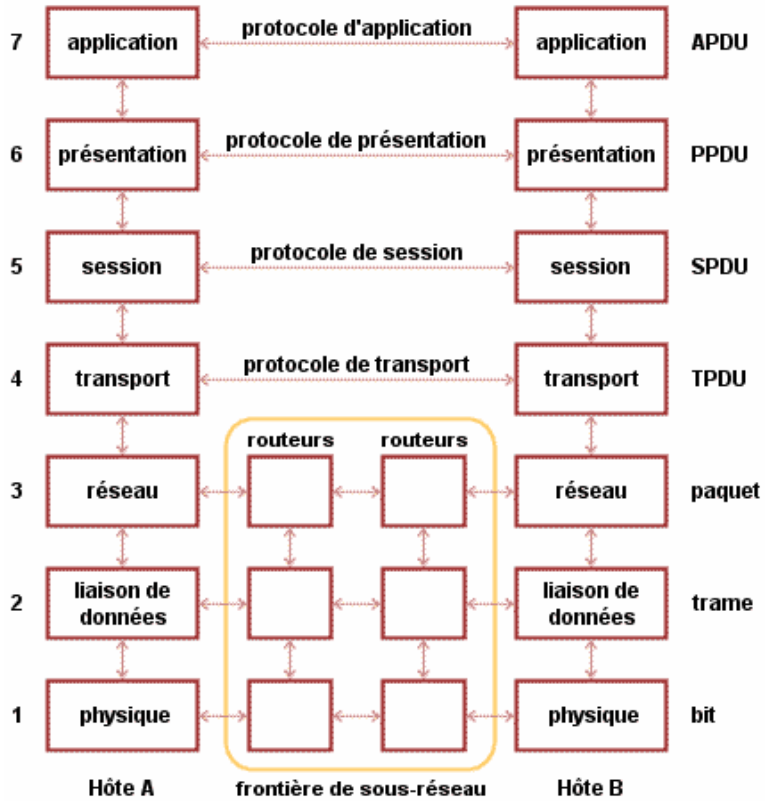
Protocoles

Modèle TCP/IP

- Couche Application : application (7), présentation (6) et session (5)

Couche application

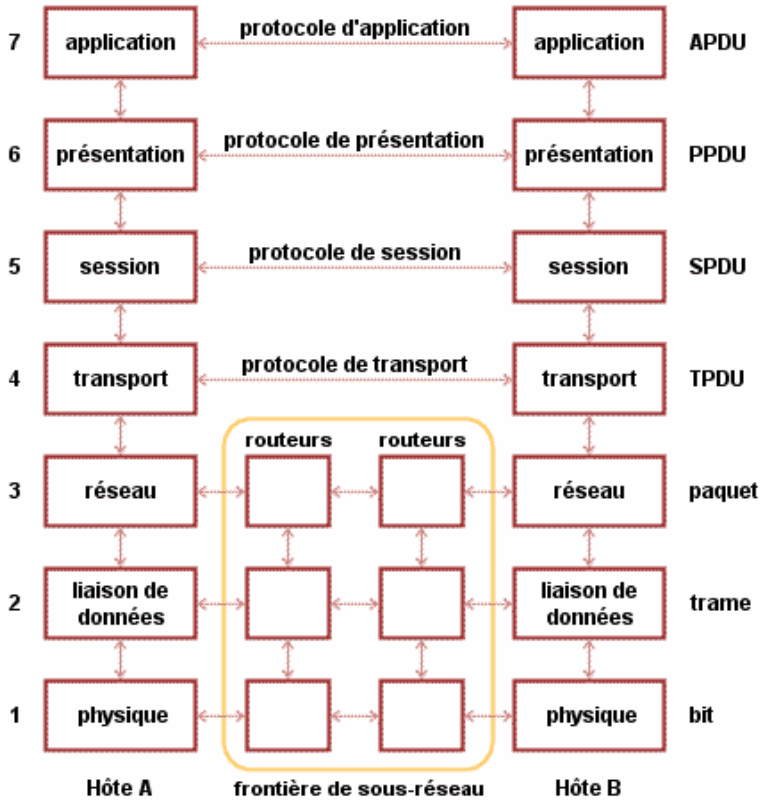
elle englobe les applications standard du réseau (Telnet, SMTP, FTP...)



Protocoles

Modèle TCP/IP

- Couche Application : application (7), présentation (6) et session (5)
- Couche Transport (TCP) : transport (4)



Couche transport

elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission (TCP ou UDP)

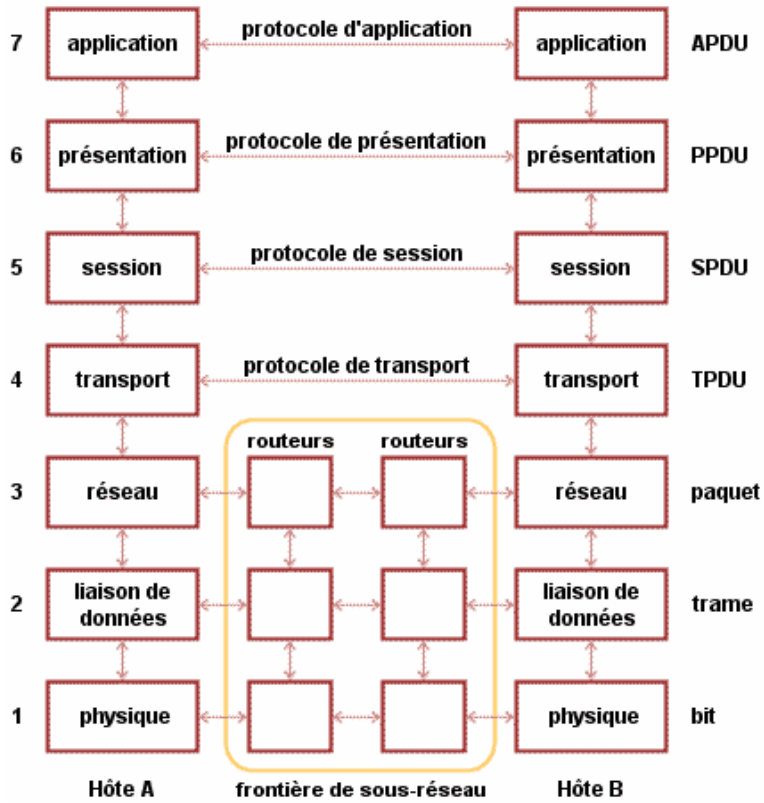
Protocoles

Modèle TCP/IP

- Couche Application : application (7), présentation (6) et session (5)
- Couche Transport (TCP) : transport (4)
- Couche Internet (IP) : Couche réseau (3)

Couche internet

elle est chargée de fournir le paquet de données (IP (Internet Protocol), ARP (résolution d'adresses), ICMP (erreurs)...)



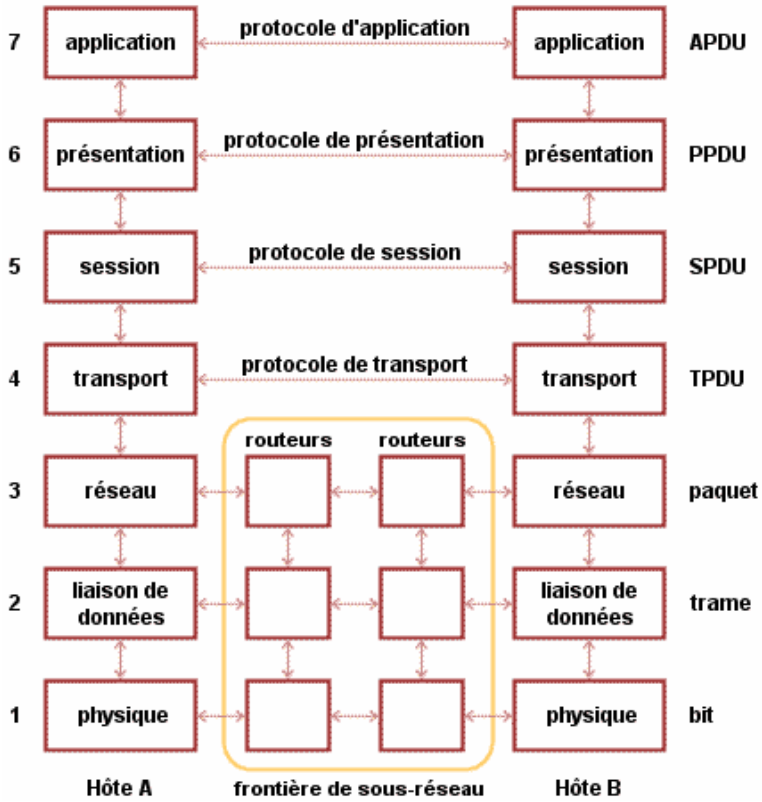
Protocoles

Modèle TCP/IP

- Couche Application : application (7), présentation (6) et session (5)
- Couche Transport (TCP) : transport (4)
- Couche Internet (IP) : Couche réseau (3)
- Couche Accès réseau : liaison donnée (2) et physique (1)

Couche accès réseau

spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé (PPP, Ethernet, Token ring...)



Adresse IP

Le protocole IP v4 (Internet Protocol)

utilise des adresses numériques, appelées adresses IP, composées de 4 nombres entiers (4 octets) entre 0 et 255 et notées sous la forme xxx.xxx.xxx.xxx. Par exemple, 194.153.205.26 est une adresse IP donnée sous une forme technique.
Adresses réservées :

- 127.0.0.1 (localhost)
- 192.168.xxx.xxx (réseau privé)
- ...

ICANN (Internet Corporation for Assigned Names)

chargée d'attribuer des adresses IP publiques, c'est-à-dire les adresses IP des ordinateurs directement connectés sur le réseau public internet.

Adresse IP - v6

IPv4 est presque obsolète : nous passons à IPv6

- IPv4 n'a plus d'adresses libres
- IPv6 est optimisé pour accélérer les vitesses de routage
- IPv6 est prévu pour mieux tenir compte des types de données transportées

La notation IPv6

Elle comprend 8 groupes de 4 chiffres hexadécimaux séparés avec le symbole deux-points.
Par exemple :
8000:0000:0000:0000:0123:4567:89AB:CDEF

- les premiers zéro d'un groupe peuvent être omis
- un ou plusieurs groupes de 4 zéros consécutifs peuvent être remplacés par un double deux-points

8000::123:4567:89AB:CDEF

DNS

Associer des noms en langage courant aux adresses numériques

On appelle résolution de noms de domaines (ou résolution d'adresses) la corrélation entre les adresses IP et le nom de domaine associé.

Domain Name System

système de gestion des noms hiérarchisé et administrable. Ce système propose :

DNS

Associer des noms en langage courant aux adresses numériques

On appelle résolution de noms de domaines (ou résolution d'adresses) la corrélation entre les adresses IP et le nom de domaine associé.

Domain Name System

système de gestion des noms hiérarchisé et administrable. Ce système propose :

- un espace de noms hiérarchique permettant de garantir l'unicité d'un nom dans une structure arborescente, à la manière des systèmes de fichiers d'Unix.

DNS

Associer des noms en langage courant aux adresses numériques

On appelle résolution de noms de domaines (ou résolution d'adresses) la corrélation entre les adresses IP et le nom de domaine associé.

Domain Name System

système de gestion des noms hiérarchisé et administrable. Ce système propose :

- un espace de noms hiérarchique permettant de garantir l'unicité d'un nom dans une structure arborescente, à la manière des systèmes de fichiers d'Unix.
- un système de serveurs distribués permettant de rendre disponible l'espace de noms.

DNS

Associer des noms en langage courant aux adresses numériques

On appelle résolution de noms de domaines (ou résolution d'adresses) la corrélation entre les adresses IP et le nom de domaine associé.

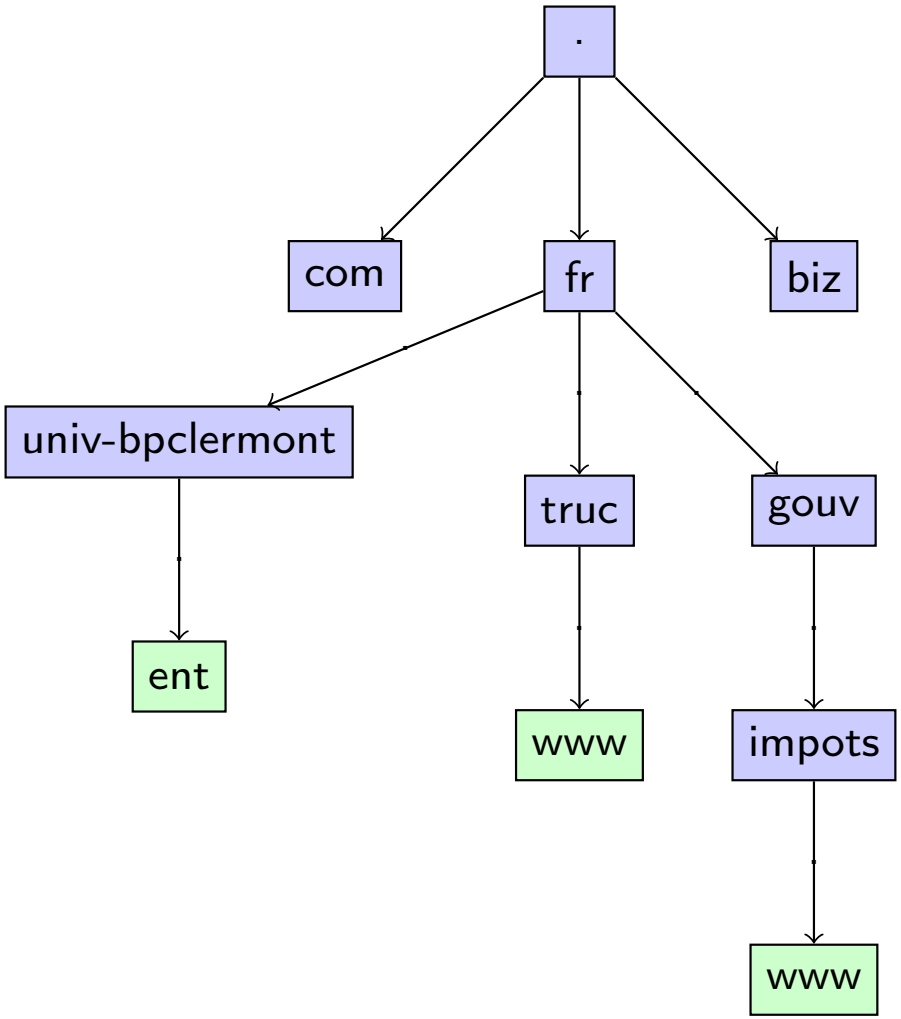
Domain Name System

système de gestion des noms hiérarchisé et administrable. Ce système propose :

- un espace de noms hiérarchique permettant de garantir l'unicité d'un nom dans une structure arborescente, à la manière des systèmes de fichiers d'Unix.
- un système de serveurs distribués permettant de rendre disponible l'espace de noms.
- un système de clients permettant de « résoudre » les noms de domaines, c'est-à-dire interroger les serveurs afin de connaître l'adresse IP correspondant à un nom.

DNS : Domain Name Server

L'espace de noms : un arbre



Les ports

De nombreux programmes TCP/IP peuvent être exécutés simultanément

L'ordinateur doit pouvoir distinguer les différentes sources de données : chaque application se voit attribuer une adresse unique sur la machine : un port (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée socket).

- adresse IP : identifier de façon unique un ordinateur sur le réseau

Les ports

De nombreux programmes TCP/IP peuvent être exécutés simultanément

L'ordinateur doit pouvoir distinguer les différentes sources de données : chaque application se voit attribuer une adresse unique sur la machine : un port (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée socket).

- adresse IP : identifier de façon unique un ordinateur sur le réseau
- numéro de port : indiquer l'application à laquelle les données sont destinées (65536 ports)

Les ports

De nombreux programmes TCP/IP peuvent être exécutés simultanément

L'ordinateur doit pouvoir distinguer les différentes sources de données : chaque application se voit attribuer une adresse unique sur la machine : un port (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée socket).

- adresse IP : identifier de façon unique un ordinateur sur le réseau
- numéro de port : indiquer l'application à laquelle les données sont destinées (65536 ports)
- les ports 0 à 1023 sont les «ports reconnus» ou réservés («Well Known Ports»). Ils sont, de manière générale, réservés aux processus système (démons) ou aux programmes exécutés par des utilisateurs privilégiés. Un administrateur réseau peut néanmoins lier des services aux ports de son choix.

Les ports

De nombreux programmes TCP/IP peuvent être exécutés simultanément

L'ordinateur doit pouvoir distinguer les différentes sources de données : chaque application se voit attribuer une adresse unique sur la machine : un port (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée socket).

- adresse IP : identifier de façon unique un ordinateur sur le réseau
- numéro de port : indiquer l'application à laquelle les données sont destinées (65536 ports)
- les ports 0 à 1023 sont les «ports reconnus» ou réservés («Well Known Ports»). Ils sont, de manière générale, réservés aux processus système (démons) ou aux programmes exécutés par des utilisateurs privilégiés. Un administrateur réseau peut néanmoins lier des services aux ports de son choix.
- les ports 1024 à 49151 sont appelés «ports enregistrés» («Registered Ports»).

Les ports

De nombreux programmes TCP/IP peuvent être exécutés simultanément

L'ordinateur doit pouvoir distinguer les différentes sources de données : chaque application se voit attribuer une adresse unique sur la machine : un port (la combinaison adresse IP + port est alors une adresse unique au monde, elle est appelée socket).

- adresse IP : identifier de façon unique un ordinateur sur le réseau
- numéro de port : indiquer l'application à laquelle les données sont destinées (65536 ports)
- les ports 0 à 1023 sont les «ports reconnus» ou réservés («Well Known Ports»). Ils sont, de manière générale, réservés aux processus système (démons) ou aux programmes exécutés par des utilisateurs privilégiés. Un administrateur réseau peut néanmoins lier des services aux ports de son choix.
- les ports 1024 à 49151 sont appelés «ports enregistrés» («Registered Ports»).
- les ports 49152 à 65535 sont les «ports dynamiques et/ou privés» («Dynamic and/or Private Ports»).

Les ports

Ports reconnus les plus couramment utilisés

- 21 FTP
- 22 SSH
- 23 Telnet
- 25 SMTP
- 53 Domain Name System
- 63 Whois
- 80 HTTP
- 110 POP3

URL

URL (Uniform Resource Locator)

Format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties :

- protocole (HTTP,FTP,News,Mailto,Gopher,...)

Exemple

http://

URL

URL (Uniform Resource Locator)

Format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties :

- protocole (HTTP,FTP,News,Mailto,Gopher,...)
- Identifiant et mot de passe : permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL

Exemple

http://user:password@

URL

URL (Uniform Resource Locator)

Format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties :

- protocole (HTTP,FTP,News,Mailto,Gopher,...)
- Identifiant et mot de passe : permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL
- serveur (il est possible d'utiliser l'adresse IP du serveur)

Exemple

http://user:password@www.monserveur.fr

URL

URL (Uniform Resource Locator)

Format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties :

- protocole (HTTP,FTP,News,Mailto,Gopher,...)
- Identifiant et mot de passe : permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL
- serveur (il est possible d'utiliser l'adresse IP du serveur)
- port (par défaut au protocole est le port numéro 80)

Exemple

http://user:password@www.monserveur.fr:80

URL

URL (Uniform Resource Locator)

Format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties :

- protocole (HTTP,FTP,News,Mailto,Gopher,...)
- Identifiant et mot de passe : permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL
- serveur (il est possible d'utiliser l'adresse IP du serveur)
- port (par défaut au protocole est le port numéro 80)
- chemin d'accès à la ressource

Exemple

http://user:password@www.monserveur.fr:80/site/index.html

Données personnelles sur Internet

Droits et devoirs sur Internet : la CNIL

La mission essentielle de la CNIL est de protéger la vie privée et les libertés dans un monde interconnecté. Toutes les données personnelles stockées doivent être autorisée par les usagers et déclarées à la CNIL. Par ailleurs, elles doivent avoir une durée de vie limitée.

Petit test sur les informations enregistrées

Beaucoup de choses sont tracées lors d'une navigation sur Internet. Petit essai sur le site de la CNIL : `http://www.cnil.fr/vos-libertes/vos-traces/`

Présentation de l'architecture d'un système client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur

Cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur.

Présentation de l'architecture d'un système client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur

Cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur.
Serveur

- machine généralement très puissante en terme de capacités d'entrée-sortie
- fournit des services
- les services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Présentation de l'architecture d'un système client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur

Cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur.

Serveur

- machine généralement très puissante en terme de capacités d'entrée-sortie
- fournit des services
- les services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Clients

- les services sont exploités par des programmes : clients
- exemple de clients : client FTP, client de messagerie, etc.
- désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur

Avantages et inconvénients du client/serveur

Avantages de l'architecture client/serveur

- des ressources centralisées : ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- une meilleure sécurité : car le nombre de points d'entrée permettant l'accès aux données est moins important
- une administration au niveau serveur : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- un réseau évolutif : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure

Avantages et inconvénients du client/serveur

Avantages de l'architecture client/serveur

- des ressources centralisées : ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- une meilleure sécurité : car le nombre de points d'entrée permettant l'accès aux données est moins important
- une administration au niveau serveur : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- un réseau évolutif : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure

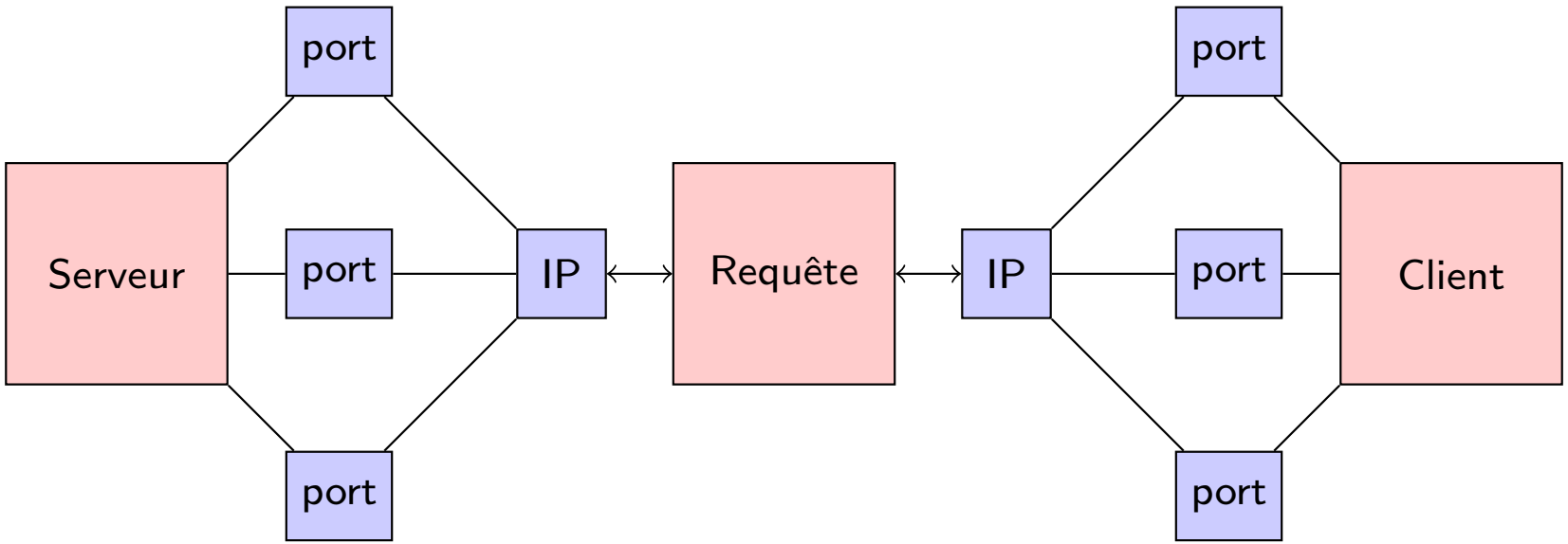
Inconvénients du modèle client/serveur

- un coût élevé dû à la technicité du serveur
- un maillon faible : le serveur est le seul maillon faible du réseau client/serveur (mais le serveur a une grande tolérance aux pannes, notamment grâce au système RAID)

Fonctionnement d'un système client/serveur

Requête

- Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur (Un serveur possède des numéros de port fixes généralement compris entre 0 et 1023)
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port (Les ports du client ne seront jamais compris entre 0 et 1023 car cet intervalle de valeurs représente les ports connus)



Architecture mainframe

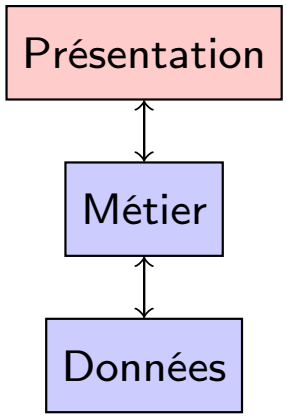
Les premiers réseaux informatiques étaient architecturés autour d'un ordinateur central, appelé « mainframe »

- le mainframe représente un ordinateur central de grande puissance chargé de gérer les sessions utilisateurs des différents terminaux qui lui étaient reliés
- la performance du système tout entier repose sur les capacités de traitement de l'ordinateur central : « informatique lourde »
- les terminaux du réseau ne peuvent voir que le serveur central

Architecture 3-tiers

L'architecture logique du système est divisée en trois niveaux ou couches

- couche présentation



Définition

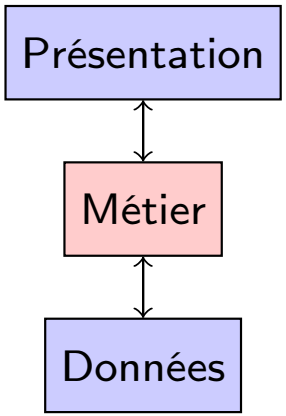
- partie visible de l'application
- partie interactive avec les utilisateurs
- Interface Homme Machine (IHM)

L'IHM peut consister en une interface graphique, textuelle, ou encore du HTML pour l'utilisation par un navigateur web.

Architecture 3-tiers

L'architecture logique du système est divisée en trois niveaux ou couches

- couche présentation
- couche métier



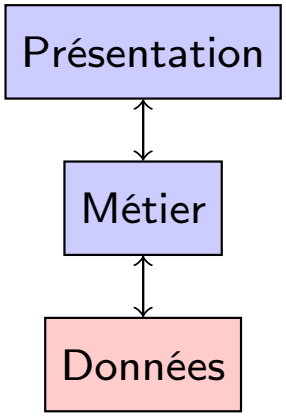
Définition

- partie fonctionnelle de l'application
- implémente la « logique »
- décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs

Architecture 3-tiers

L'architecture logique du système est divisée en trois niveaux ou couches

- couche présentation
- couche métier
- couche accès aux données



Définition

- partie gérant l'accès aux gisements de données du système
- les données peuvent être propres au système
- les doonées peuvent être gérées par un autre système

Les données peuvent être stockées indifféremment dans de simples fichiers texte, ou eXtensible Markup Language (XML), ou encore dans une base de données. Quel que soit le support de stockage choisi, l'accès aux données doit être le même.

Comparaison des deux types d'architecture

Serveurs polyvalents/ serveurs spécialisés

- architecture à deux niveaux : le serveur est polyvalent
- architecture à trois niveaux : chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple)

Comparaison des deux types d'architecture

Serveurs polyvalents/ serveurs spécialisés

- architecture à deux niveaux : le serveur est polyvalent
- architecture à trois niveaux : chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple)

A la faveur du 3-tiers

- plus grande flexibilité/souplesse
- sécurité accrue (définie indépendamment pour chaque service, et à chaque niveau)
- meilleures performances

Architecture N-tiers

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux...

Client lourd

Application cliente graphique exécutée sur le système d'exploitation de l'utilisateur

- possède généralement des capacités de traitement évoluées

Client lourd

Application cliente graphique exécutée sur le système d'exploitation de l'utilisateur

- possède généralement des capacités de traitement évoluées
- peut posséder une interface graphique sophistiquée

Client lourd

Application cliente graphique exécutée sur le système d'exploitation de l'utilisateur

- possède généralement des capacités de traitement évoluées
- peut posséder une interface graphique sophistiquée
- demande un effort de développement

Client lourd

Application cliente graphique exécutée sur le système d'exploitation de l'utilisateur

- possède généralement des capacités de traitement évoluées
- peut posséder une interface graphique sophistiquée
- demande un effort de développement
- tend à mêler la logique de présentation (l'interface graphique) avec la logique applicative (les traitements)

Client lourd

Application cliente graphique exécutée sur le système d'exploitation de l'utilisateur

- possède généralement des capacités de traitement évoluées
- peut posséder une interface graphique sophistiquée
- demande un effort de développement
- tend à mêler la logique de présentation (l'interface graphique) avec la logique applicative (les traitements)
- problèmes de mise à jour

Client léger

application accessible via une interface web HTML

- consultable à l'aide d'un navigateur web

Client léger

application accessible via une interface web HTML

- consultable à l'aide d'un navigateur web
- la totalité de la logique métier est traitée du côté du serveur

Client léger

application accessible via une interface web HTML

- consultable à l'aide d'un navigateur web
- la totalité de la logique métier est traitée du côté du serveur
- interfaces relativement pauvres en interactivité en HTML (sauf à utiliser javascript)

Client léger

application accessible via une interface web HTML

- consultable à l'aide d'un navigateur web
- la totalité de la logique métier est traitée du côté du serveur
- interfaces relativement pauvres en interactivité en HTML (sauf à utiliser javascript)
- grande souplesse de mise à jour

Client léger

application accessible via une interface web HTML

- consultable à l'aide d'un navigateur web
- la totalité de la logique métier est traitée du côté du serveur
- interfaces relativement pauvres en interactivité en HTML (sauf à utiliser javascript)
- grande souplesse de mise à jour
- problèmes de compatibilité des différents navigateurs...

Client riche

un compromis entre le client léger et le client lourd

- interface graphique décrite avec une grammaire de description basée sur la syntaxe XML
- permet d'obtenir des fonctionnalités similaires à celles d'un client lourd (glisser déposer, onglets, multi fenêtrage, menus déroulants)
- essentiel des traitements du côté du serveur.
- les données sont transmises dans un format d'échange standard utilisant la syntaxe XML (SOAP, XML-RPC), puis interprétées par le client riche

Client riche

un compromis entre le client léger et le client lourd

- interface graphique décrite avec une grammaire de description basée sur la syntaxe XML
- permet d'obtenir des fonctionnalités similaires à celles d'un client lourd (glisser déposer, onglets, multi fenêtrage, menus déroulants)
- essentiel des traitements du côté du serveur.
- les données sont transmises dans un format d'échange standard utilisant la syntaxe XML (SOAP, XML-RPC), puis interprétées par le client riche

Les principaux standards permettant de définir une application riche sont les suivants

- XAML (eXtensible Application Markup Language), un standard XML proposé par Microsoft, utilisé notamment dans les applications utilisant le framework .NET
- XUL, un standard XML proposé par la fondation Mozilla, utilisé par exemple dans le client de messagerie Mozilla Thunderbird ou dans le navigateur Mozilla Firefox
- Flex, un standard XML proposé par la société Macromedia

Architecture d'égal à égal

Pas de serveur dédié

- chaque ordinateur dans un tel réseau est un peu serveur et un peu client
- chacun des ordinateurs du réseau est libre de partager ses ressources

Architecture d'égal à égal

Pas de serveur dédié

- chaque ordinateur dans un tel réseau est un peu serveur et un peu client
- chacun des ordinateurs du réseau est libre de partager ses ressources

Inconvénients des réseaux d'égal à égal

- pas du tout centralisé, ce qui le rend très difficile à administrer
- la sécurité est très peu présente
- aucun maillon du système n'est fiable

Architecture d'égal à égal

Pas de serveur dédié

- chaque ordinateur dans un tel réseau est un peu serveur et un peu client
- chacun des ordinateurs du réseau est libre de partager ses ressources

Inconvénients des réseaux d'égal à égal

- pas du tout centralisé, ce qui le rend très difficile à administrer
- la sécurité est très peu présente
- aucun maillon du système n'est fiable

Avantages de l'architecture d'égal à égal

- coût réduit
- simplicité

Webmastering - Introduction à la création de pages web

Notion de site web

Un site web (site internet par abus de langage) est

- un ensemble de fichiers HTML
- liés par des liens hypertextes
- stockés sur un serveur web

Webmastering - Introduction à la création de pages web

Notion de site web

Un site web (site internet par abus de langage) est

- un ensemble de fichiers HTML
- liés par des liens hypertextes
- stockés sur un serveur web

Intérêts d'un site web

- visibilité
- amélioration de la notoriété
- collecte de données
- vente en ligne
- mise en place d'un support aux utilisateurs

Webmastering - Introduction à la création de pages web

Plusieurs catégories de sites web

- sites vitrine
- sites catalogue
- sites d'information
- sites marchands
- sites institutionnels
- sites personnels (parfois pages perso)
- sites communautaires
- sites intranet

Qu'est-ce que le webmastering ?

Le webmestre est en charge d'un site web

Phases de vie du site web:

- création : concrétisation d'une idée en un site en ligne, référencé et visité
- exploitation : gestion quotidienne du site, son évolution et sa mise à jour

Qu'est-ce que le webmastering ?

Le webmestre est en charge d'un site web

Phases de vie du site web:

- création : concrétisation d'une idée en un site en ligne, référencé et visité
- exploitation : gestion quotidienne du site, son évolution et sa mise à jour

La création

- conception : formalisation de l'idée
- réalisation : développement du site web
- hébergement : mise en ligne du site, de manière permanente

Qu'est-ce que le webmastering ?

Le webmestre est en charge d'un site web

Phases de vie du site web:

- création : concrétisation d'une idée en un site en ligne, référencé et visité
- exploitation : gestion quotidienne du site, son évolution et sa mise à jour

La création

- conception : formalisation de l'idée
- réalisation : développement du site web
- hébergement : mise en ligne du site, de manière permanente

Exploitation

- veille : suivi des technologies, du positionnement du site et de celui des concurrents
- promotion et référencement : développer son audience
- maintenance et mise à jour : animation quotidienne du site et maintien de son bon fonctionnement

HTML

HTML est la « langue maternelle » du navigateur

HTML a été inventé en 1990 par un scientifique nommé Tim Berners-Lee.
L'objectif était de faciliter l'accès par des scientifiques d'universités différentes aux documents de recherche de chacun.
En inventant HTML, il posa les fondations du Web tel que nous le connaissons aujourd'hui.
Ce que vous voyez quand vous regardez une page sur Internet est l'interprétation par votre navigateur du code HTML.

HTML

« HyperText Mark-up Language »

- « Hyper » s'oppose à linéaire. Au départ, les programmes informatiques s'exécutaient de façon linéaire : lorsque le programme avait terminé une action, il allait à la ligne suivante, puis encore à la suivante, et ainsi de suite. Mais HTML est différent : vous pouvez aller n'importe où et quand vous le voulez.
- « Text » s'explique tout seul.
- « Mark-up » (balisage) est ce que vous faites avec le texte. Vous marquez le texte de la même façon que vous le feriez dans un logiciel de traitement de texte avec des titres, des puces et des caractères gras, etc.
- « Language » (langage) est ce qu'est HTML.

XHTML (Extensible HyperText Mark-up Language)

En bref : une nouvelle façon mieux structurée d'écrire du HTML.

Les éléments et les balises

Les éléments

- donnent la structure d'un document HTML
- indiquent comment le navigateur doit présenter le site Web
- en général, les éléments se composent d'une balise ouvrante, d'un contenu et d'une balise fermante.

Les balises

- étiquettes utilisées pour marquer le début et la fin d'un élément
- format : < texte > pour les balises ouvrantes
- format : < / texte > pour les balises fermantes
- format : < texte / > pour les balises sans contenu

Structure d'une page

Balises de structure

```
<html>
     <head>
          contenu de l'entête de la page
     </head>
     <body>
           contenu du corps de la page
     </body>
</html>
```

Entête

Informations concernant la page (n'apparaissent pas dans la page affichée par le navigateur)

Corps

Informations qui constituent la page

Balises de l'entête

Informations sur la page

- <title> ... < /title>
- <meta name="keywords" lang="fr" content="mots clefs"/ >
- <meta name="Description" content="Description du site et/ou de la page..."/ >
- <meta name="author" lang="fr" content="Prénom Nom"/ >
- <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
- ...

Balises du corps

Structure

- <h1>...< /h1> titre de section
- <h2>...< /h2> titre de sous-section
- <h3>...< /h3> titre de sous-sous-section
- <p>...< /p> paragraphe
- ...

Mise en valeur du texte

- <b>...< /b> gras
- <i>...< /i> italique
- ...

Balises du corps

Séparation de texte

- <br/ > saut de ligne
- <hr/ > ligne horizontale
- ...

Liste

```
<ul>
     <li>Item 1</li>
     <li>Item 2</li>
</ul>
```

Liste numérotée : <ol> remplace

Les attributs

Personnaliser ou préciser les balises

L'attribut se spécifie dans la balise ouvrante (comme pour les meta):
<body style="background-color: red;"> mettra tous le corps de la page sur fond rouge.
Presque tous les éléments peuvent avoir des attributs.

On préfère attribuer des style via les CSS (voir prochains cours...).
Les attributs sont utilisés dans certaines balises essentielles : les liens.

Les commentaires

Syntaxe du commentaire

```
<!-- Commentaire -->
```

Caractère spéciaux

Caractères interprétés par HTML

- < : <
- > : >
- & : &

Caractères spéciaux

- é : é
- è : è
- à : à
- ...

Les tableaux

```
<table  border=1>
  <caption>Légende du tableau</caption>
  <thead>
    <tr>
      <th>Titre de la première colonne</th>
      <th>Titre de la deuxième colonne</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Bas colonne 1 titre</th>
      <td>Bas colonne 2</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Cellule ligne 2,  colonne 1</td>
      <td>Cellule ligne 2,  colonne 2</td>
    </tr>
    <tr>(...)</tr>
    <tr>(...)</tr>
  </tbody>
</table>
```

Légende du tableau

Titre de la première colonne	Titre de la deuxième colonne
Cellule ligne 2, colonne 1	Cellule ligne 2, colonne 2
Bas colonne 1 titre	Bas colonne 2

Liens hypertexte

Il existe deux types de liens dans une page

- liens classiques : attribut `href` qui pointe vers l'adresse de la ressource à visiter.

Exemples

```
<a href="./section.html">Lien vers la section</a>
<a href="mailto:gaelle@loosli.fr?cc=toto@tata.fr& subject=coursHtml"</a>
<a href="ftp://ftp.loosli.fr"</a>
```

Liens hypertexte

Il existe deux types de liens dans une page

- liens classiques : attribut `href` qui pointe vers l'adresse de la ressource à visiter.
- liens qui n'apparaissent pas dans le corps de la page: élément `link`.

Exemples

```
<a href="./section.html">Lien vers la section</a>
<a href="mailto:gaelle@loosli.fr?cc=toto@tata.fr& subject=coursHtml"</a>
<a href="ftp://ftp.loosli.fr"</a>
<link rel="next" type="text/html" href="fichiersuivant.html">
<link rel="alternate" type="application/postscript" href="document.ps">
```

Les images

Afficher une image

```
<img src="images/flecheshaut.gif" alt="Retour" height="21" width="21">
```

Taille des image

- Toujours utiliser des images à taille réelle (redimensionner avant!)
- Toujours remplir le texte alternatif
- Prendre garde au poids des images pour ne pas ralentir la page (utiliser des formats compressés)

Validité du code

Le W3C

"Le World Wide Web Consortium est un organisme de standardisation à but non-lucratif, fondé en octobre 1994 comme un consortium chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, CSS, PNG, SVG et SOAP."[a]
Insérer avant la balise <html>:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

[a] source : wikipedia

Validateur

`http://validator.w3.org`

CSS - Cascading Style Sheets

Pourquoi le CSS ?

- langage de style qui définit la présentation des documents HTML
- offre plus d'options et se montre plus précis et sophistiqué d'HTML
- est pris en charge par tous les navigateurs actuels

Principe : séparer le fond et la forme

- HTML sert à structurer le contenu, CSS sert à formater un contenu structuré.
- contrôle de la présentation de plusieurs documents par une seule feuille de style
- présentations différentes appliquées à des types de médias différents (à l'écran, à l'impression, etc.)

CSS - Mise en place

Utiliser le CSS

- le code CSS est placé dans un/plusieurs fichiers CSS à part
- le lien vers vers ses fichiers est fait dans l'entête du fichier HTML
- une modification du CSS affecte le rendu visuel de toutes les pages l'utilisant

Concepts de base

Modifier l'apparence d'une balise

L'apparence de chaque balise du langage HTML peut-être personnalisée dans le CSS.

Mettre le titre principal en gros et rouge, centré

```
h1 {
  color: red;
  font-size: 24px;
  text-align: center;
}
```

Concepts de base

Balise muette

Sert à identifier des éléments que l'on souhaite afficher de manière particulière, sans utiliser une balise HTML pré-définie

Identifier visuellement les mots clefs

Dans le fichier HTML

```
<p> Mon joli <span class=motclef> texte </span> d'exemple</p>
```

Dans le fichier CSS

```
.motclef {
   font-size: 110%;
}
```

Concepts de base

Les blocs et divisions

La structuration du contenu se fait de manière globale à l'aide de blocs (balises div), dans lesquels sont regroupées des choses qui vont ensemble d'un point de vue sens.

Exemple pour un blog

- le bloc **article** est composé d'un titre, d'un auteur, d'un *contenu*, d'une date...
- le bloc **contenu** est composé d'un ou plusieurs paragraphes
- le bloc **publications** est composé d'un ensemble d' *articles*
- le bloc **menu** est composé d'une série de liens sur les différentes pages du site
- le bloc **pied de page** est composé du copyright, le la date de dernière mise à jour...
- le bloc **page** est composé d'un *menu*, de *publications* et d'un *pied de page*

```
<div id=page>
  <div id=menu>...</div>
  <div id=publications>
    <div class=article>
      <div class=contenu>...</div>
    </div>
    ...
  </div>
</div>
```

Concepts de base

La mise en forme des bloc

Dans le fichier CSS, on positionne et décore les bloc à l'aide de leur identifiant (id) ou classe (class).

Reprise de l'exemple

style_blog.css

```
#page{
  background-color: #7A95C7;
}
#menu{ ... }
#publication{ ... }
.article{ ... }
.contenu{ ... }
```

```
<div id=page>
  <div id=menu>...</div>
  <div id=publications>
    <div class=article>
      <div class=contenu>...</div>
    </div>
    ...
  </div>
</div>
```

class ou id?

On utilise le mot clef *id* pour un bloc unique dans la page et il est associé au symbole # dans le CSS Le mot clef *class* est utilisé pour les blocs qui peuvent être utilisés plusieurs fois, et est associé au symbole . dans le CSS.

Modifier l'apparence des blocs

Polices

- font-family ("Arial Black", Arial, Verdana, serif, ...)

Tailles

- font-size (12px, 80%, small, medium...)
- font-weight (bold, bolder, lighter, normal)
- font-style (italic, oblique)

Décorations

- text-decoration (underline, overline, line-trough, blink, none)
- font-variant (small-caps, normal)
- text-transform (uppercase,lowercase,capitalize,none)

Modifier l'apparence des blocs

Alignement

- text-align (left,right,center, justify)
- vertical-align (top, middle, bottom)
- line-height (% ou px)
- text-indent (px)
- white-space (césure : normal, nowrap, pre)

Couleurs et fonds

- color (#000000, rgb(20,20,0))
- background-color
- background-image (url)
- background-attachment (fixed, scroll)
- background-repeat (repeat, repeat-x, repeat-y, no-repeat)
- background-position (% ou px par rapport au coin haut-gauche, ou position : top, center, bottom, left, right)

Modifier l'apparence des blocs

Taille

- width (%, px ou auto)
- height
- min-height, max-height
- min-width, max-width
- margin
- margin-top, margin-bottom, margin-left, margin-right,
- padding
- padding-top, padding-bottom, padding-left, padding-right

Modifier l'agencement des blocs

Affichage

- display (none, block, inline)
- visibility (none, hidden, visible)
- clip : rect()
- overflow (visible, hidden, scroll, auto)

Modifier l'agencement des blocs

Positionnement

- float (left, right, non)
- clear (left, right, both, none)
- position (absolute, fixed, relative, static)
- top (px, %)
- bottom
- left
- right
- z-index

Et ce qui n'est pas un bloc

Bordures

- border-width
- border-color
- border-style (none, hidden, solid, double, dashed, dotted, inset, outset, ridge)
- border-left...

Listes

- list-style-type (disc, circle, square, none, decimal, upper-roman, lower-roman,...)
- list-style-position (inside, outside)
- list-style-image (url)

Propriété display

Valeurs possibles

- bock : l'élément génère une boîte type *block*, avec un saut de ligne avant et après
- inline : l'élément génère une boîte type *inline*, sans saut de ligne ni avant ni après
- none : l'élément de génère aucune boîte

Example de code

```
.monbloc
{
  display:block;
}
```

Propriété position

Valeurs possibles

- static : comportement par défaut, affiche les élément dans l'ordre du fichier source
- absolute : l'élément est positionné relativement à son premier ancêtre positionné (non static), à défaut, le navigateur
- relative : l'élément est positionné relativement à sa position normale
- fixed : l'élément est positionné par rapport à la fenêtre du navigateur
- inherit : valeur hérité de l'élément parent

Example de code

```
h2
{
  position:absolute;
  left:100px;
  top:150px;
}
```

Positionner

Propriétés utiles

- top, left, right, bottom : donne une distance à respecter entre l'élément et son voisinage, selon le type de positionnement et la direction donnée
- z-index : permet d'ordonner les éléments en cas de superposition
- overflow (auto, hidden, scroll, visible) précise le comportement de l'élément si le contenu dépasse la taille allouée dans la mise en page.

Groupes

Grouper des sélecteurs aux propriétés identiques

```
h1
{
  color:green;
}
h2
{
  color:green;
}
```

Peut se résumer

```
h1, h2
{
  color:green;
}
```

Imbrication

Spécifier le comportement d'un sélecteur selon son contexte

```
p
{
  color:blue;
  text-align:center;
}
.marked
{
  background-color:red;
}
.marked p
{
  color:white;
}
```

Pseudo-classes

Syntaxe

```
selecteur.classe:pseudo-classe
{
  propriete:valeur;
}
```

Exemples

- a:link : lien non visité
- a:visited : lien visité
- a:hover : lien survolé
- a:active : lien sélectionné

Pseudo-élements

Syntaxe

```
selecteur.classe:pseudo-element
{
  propriete:valeur;
}
```

Exemples

- :first-line : s'applique à la première ligne de l'élément, si l'élément est de type block
- :first-letter : s'applique à la première lettre de l'élément, si l'élément est de type block
- :before : ajoute un contenu avant l'élément
- :after : ajoute un contenu après l'élément

Contenu

Contenu

La propriété content sert à insérer un contenu généré avant ou après un élément, via les pseudo-élements *:before* ou *:after*.

- counter : le contenu ajouté est un compteur
- attr(attribute) : le contenu ajouté est la valeur d'un attribut du sélecteur
- string : ajoute un texte
- open-quote et close-quote : ajoute des guillemets ouvrant ou fermant
- url(str) : ajoute un média (image, son, vidéo, ...)

Ajouter l'adresse du lien entre parenthèses après le lien

```
a:after
{
content: " (" attr(href) ")";
}
```

Les formulaires

HTML : les formulaires

```
<form  method="post" action="script.php">
</form>
```

Cette balise prend 2 arguments, l'un pour définir la façon de communiquer avec le serveur (method="post" ou method="get"), l'autre pour indiquer l'adresse à suivre lors de l'envoie du formulaire (action="envoye.php", adresse du script qui va interpréter et utiliser le contenu du formulaire).

GET et POST

- POST est la valeur qui correspond à un envoi de données stockées dans le corps de la requête,
- GET correspond à un envoi des données codées dans l'URL, et séparées de l'adresse du script par un point d'interrogation

Les formulaires

Balises de formulaire

- texte sur une ligne : `<input type="text"/>`
- texte sur plusieurs lignes : `<textarea rows="5" cols="30"> </textarea>`
- texte étoilé (mot de passe) : `<input type="password"/>`
- téléchargement : `<input type="file"/>`
- case à cocher : `<input type="checkbox"/>`
- boutons radio : `<input type="radio"/>`
- sélection dans une liste : `<select> </select>`
- gestion des option au sein de la sélection : `<optgroup>` et `<option>`
- soumission du formulaire : `<input type="submit"/>`
- remise à zéro du formulaire : `<input type="reset"/>`

Les formulaires

Identification des champs

Chaque champ, pour pouvoir être traité ensuite, doit être identifié

```
<input name="montexte" id="montexte" type="text"/>
```

- L'attribut `name` sert à retrouver un objet afin de l'exploiter en JavaScript. Il est également utilisé lors de l'envoi du formulaire vers un serveur afin d'extraire les données saisies par l'utilisateur.
- L'attribut `id` sert à nommer un objet dans une page web. Cependant, contrairement à l'attribut `name`, tous les `id` doivent être uniques dans toute la page.

Les formulaires

Etiquettes

- Il est recommandé d'utiliser un `label` c'est à dire la balise `<label>` permettant d'associer une légende à un champ d'entrée afin que tous les utilisateurs puissent utiliser votre formulaire sans problème.
- L'attribut `for` sert à spécifier le nom du champ d'entrée dont le `label` est la légende.

```
<label for="montexte">Votre nom </label>
<input name="montexte" id="montexte" type="text" value="Nom"/>
```

Ordre

L'attribut `tabindex` permet de fixer l'ordre de passage d'un champ à l'autre quand on utilise la tabulation.

Les formulaires

Groupement de champs

Pour rendre plus lisible les formulaires, on peut regrouper les champs qui vont ensemble à l'aide de la balise `<fieldset>`. Dans chaque groupement, la balise `<legend>` (obligatoire) permet de donner un titre qui apparaît dans l'encadrement du groupe.

```
<form method="post" action="envoye.html">
  <fieldset class="principal">
    <legend>Vous</legend>
    <p><label for="montexte">Votre nom </label> :
      <input name="montexte" id="montexte" type="text"
        value="Nom" tabindex="10"/></p>
    <p><label for="monpass">Votre mot de passe </label> :
      <input name="monpass" id="monpass" type="password"
        value="nom" tabindex="20" /></p>
  </fieldset>
```

```
<fieldset class="secondaire">
  <legend>Votre demande</legend>
  <p><label for="upload">Choisissez votre fichier </label>
    <input name="upload" id="upload" type="file"
      value="telechargez" tabindex="40"/></p>
  <p><label for="cocher">Cochez si vous voulez! </label>
    <input id="cocher" name="cocher" type="checkbox"
      value="cochez" tabindex="50"/></p>
  <p><label for="grandtexte">Description</label> :
  <textarea id="grandtexte" name="grandtexte" tabindex="30"
      rows="5" cols="30">
      quelque de chose de plutôt long pour voir...</textarea></p>
</fieldset>
```

```
<fieldset class="tertiaire">
  <legend>Compléments</legend>
  <p>
    <label for="radio1">Fille </label><input name="radio" id="radio1"
      type="radio" value="choisissez" tabindex="60"/>
    <label for="radio2">Garçon </label><input name="radio" id="radio2"
      type="radio" value="ou choisissez" tabindex="70"/>
  </p>
```

```
    <p>Choisissez dans la liste : <select name="listeD" tabindex="80">
      <optgroup label="facile">
        <option id="cours1" value="cours1">Cours 1</option>
        <option id="cours2" value="cours2">Cours 2</option>
      </optgroup>
      <optgroup label="pas facile">
        <option id="cours3" value="cours3">Cours 3</option>
      </optgroup>
    </select></p>
    <p>
      <input type="submit" value="Appuyez pour valider" tabindex="90"/>
    </p>
  </fieldset>
</form>
```