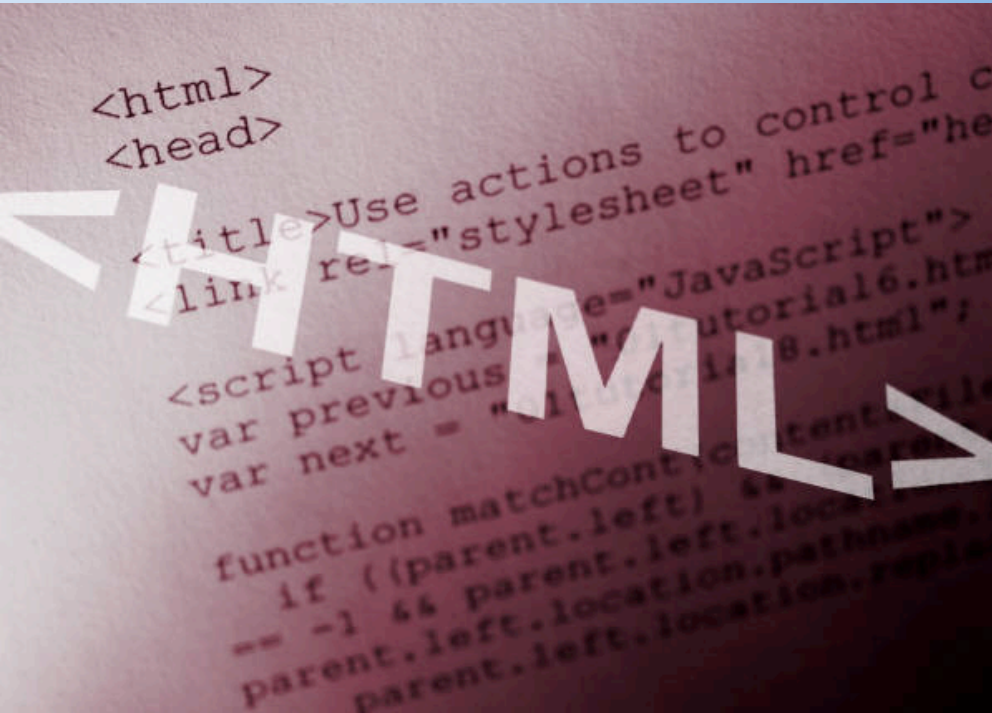


MODULE CONCEPTION DE DOCUMENTS WEB



1. **Langages pour l'IHM**
2. **HTML**
3. CSS
4. Javascript
5. JQuery

C'est un module... HTML + CSS + Javascript + JQuery

**TOUT AUTRE langage, bibliothèque, API, utilitaire...
... est interdit !**

PAS DE PHP : trouvez les solutions en HTML ou en JS !

**PAS d'AVANCE DE PHASE : si en cours on n'a vu que
HTML, la réponse au TP n'est PAS en Javascript : il
faut la trouver en HTML**

HTML permet de définir une IHM.....

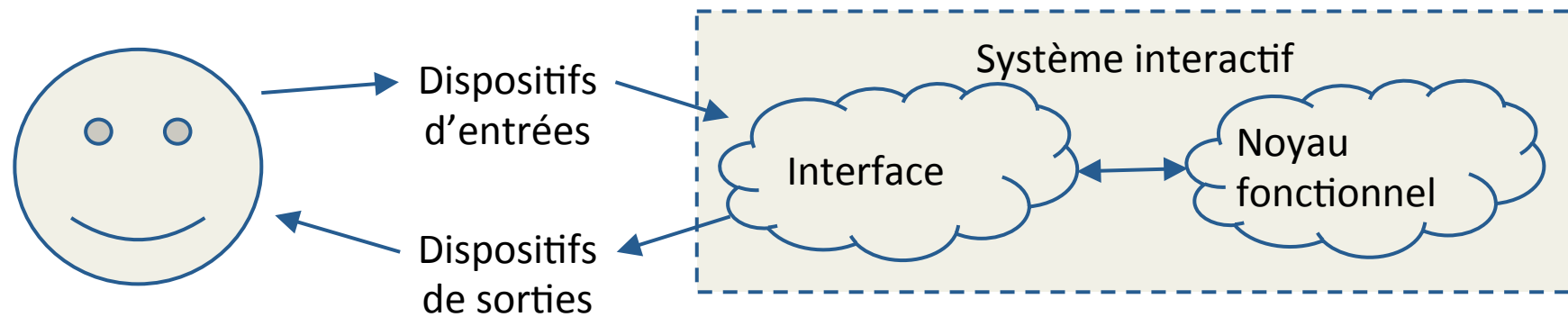
IHM?

Interaction **H**umain - **M**achine

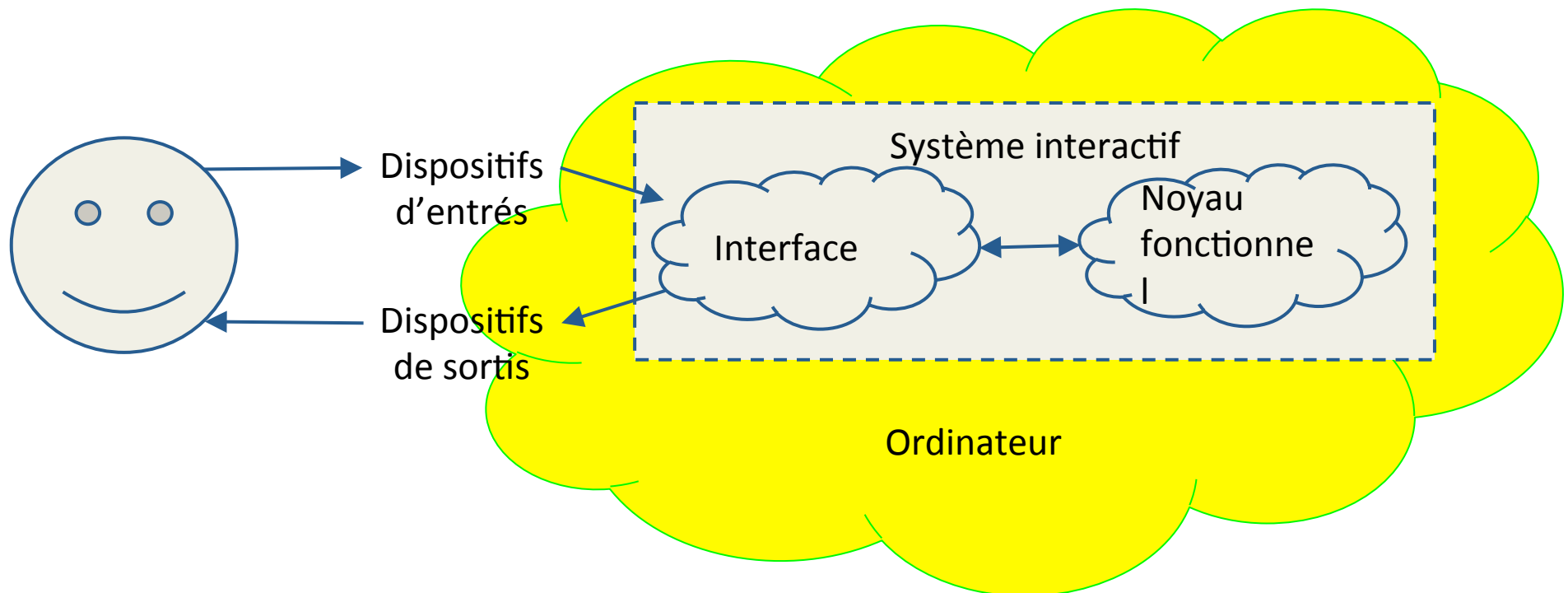
Décrire l'interaction entre une machine et ses utilisateurs :

- Conducteur - Bus
- Ouvrier - Machine
- ...
- Personnes - Ordinateur

Modifier / Agir sur l'état du système



Percevoir / Comprendre l'état du système



Une interface est une zone, réelle ou virtuelle qui sépare deux éléments.

L'interface désigne ainsi ce que chaque élément a besoin de **connaître** de l'autre **pour** pouvoir **fonctionner correctement**.

Types d'interactions

CLI : Command Line Interface

- + Flexible
- + adapté aux experts
- + très peu coûteux (réseaux...)
- Mémorisation
- Apprentissage
- Gestion des erreurs
- Rebutte les non experts

```
FCS transmogriifier
Version: 1.0b Build: 2011042001
Framework Version: 1.0b Build: 2011042001

Copyright (C) 2009-2011 Beau Hunter, 318 Inc.

Usage:
transmogriifier.py [options] [target]
transmogriifier.py [-f configfile] [-d supportdir] [-a action] [-t mediatitle]

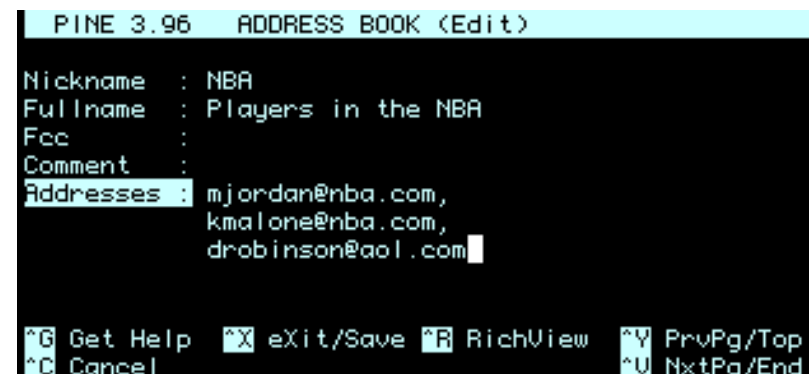
Working with assets:
transmogriifier.py --setField="Keywords" --value="MyAsset" -t MyAsset
transmogriifier.py --setField="Keywords" --value="MyAsset" -i /fcxmlinfile.xml
transmogriifier.py --appendField="Status" --value="Update!" -i /fcxmlinfile.xml

transmogriifier.py --getAssetID --assetPath="/FCS/Media/myfile.mpg"
transmogriifier.py --getEntityPath --assetID=1
transmogriifier.py --getFilePath --assetID=1
transmogriifier.py --getArchiveFilePath --assetID=1
transmogriifier.py --getThumbnailPath --assetID=1
transmogriifier.py --getProxyPath --assetID=1
transmogriifier.py --getPosterFramePath --assetID=1
transmogriifier.py --archive --assetID=1
transmogriifier.py --restore --assetID=1
transmogriifier.py --analyze --assetID=1
```

Types d'interactions

Formulaires

- + Simplifie la saisie
- + Guide
- Occupe de la place
- Rigide



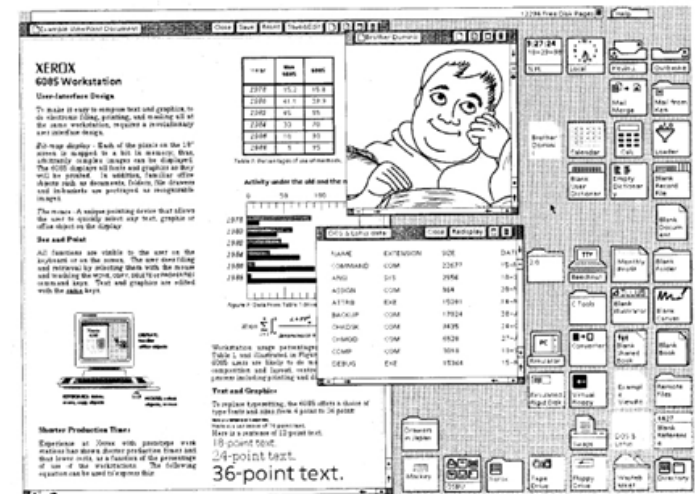
```
PINE 3.96 ADDRESS BOOK (Edit)
Nickname : NBA
Fullname : Players in the NBA
Fcc :
Comment :
Addresses : mjordan@nba.com,
            kmalone@nba.com,
            drobinson@aol.com
^G Get Help ^X eXit/Save ^R RichView ^Y PrvPg/Top
^C Cancel ^U NxtPg/End
```

Notion de widget, d'interacteur...

Types d'interactions

WIMP : Window Icon Menu Pointer

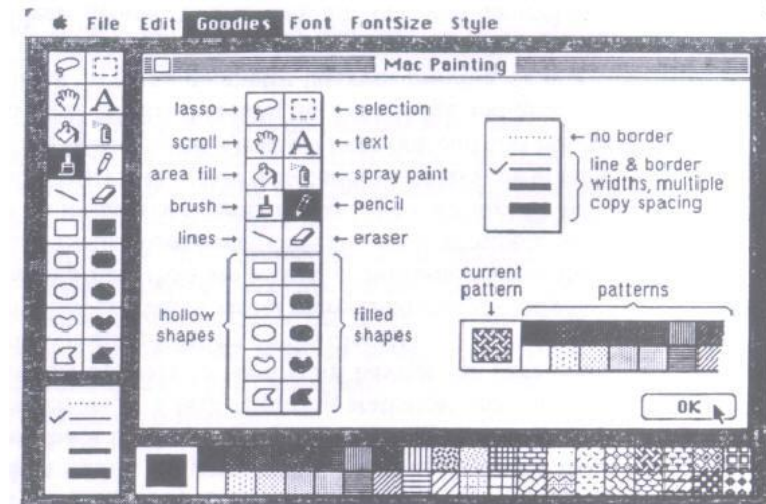
- + Fenêtrage, plusieurs écrans en un
- + Icones comme raccourcis
- + Menu, structuration
- + Pointeur
 - * précision
 - * indique où se porte l'attention
- + Standardisation
- Trop spécialisé au contexte PC
- Indirections (souris, manip...)



Types d'interactions

Manipulation directe

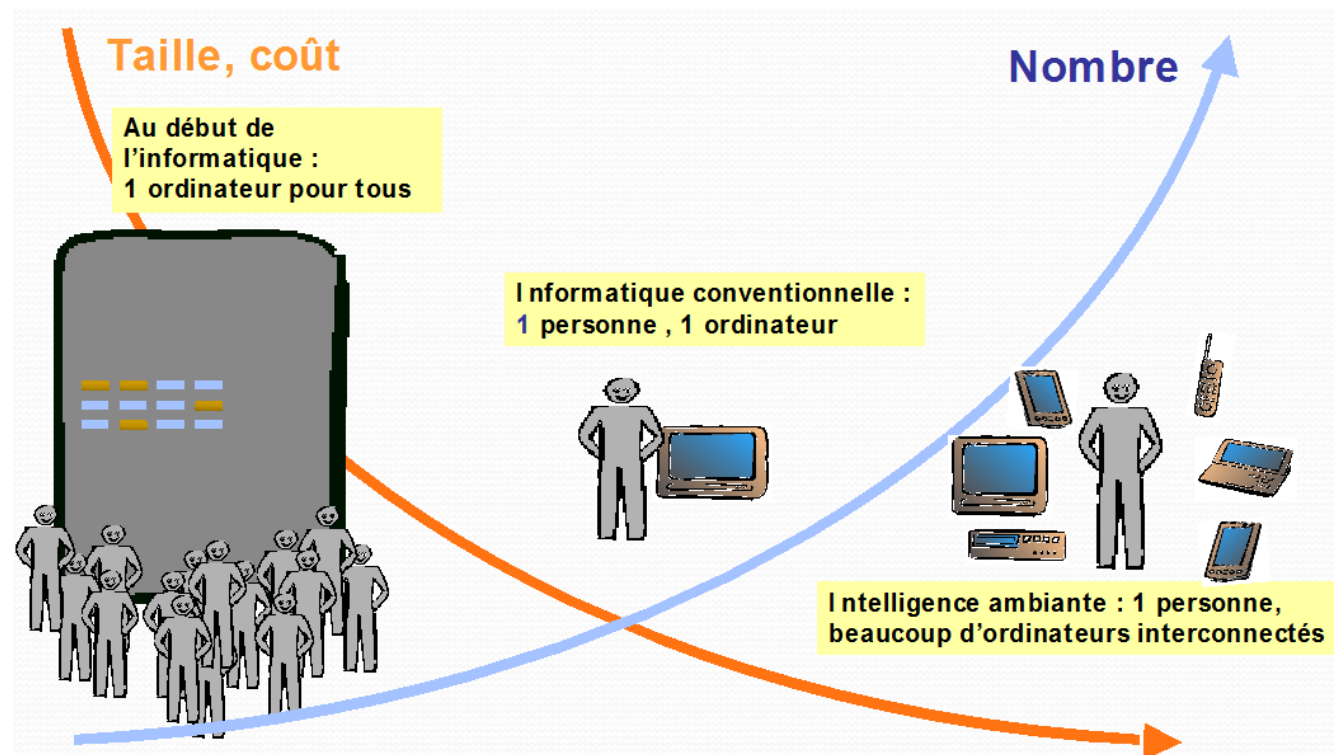
- + Représentation des concepts
- + Utilisation de métaphores
- + Apprentissage, exploration
- + Réduction des erreurs
- Représentation parfois non pertinente
- Métaphores parfois trompeuses



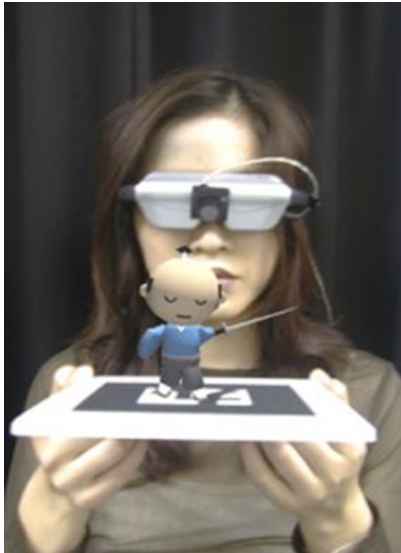
Types d'interactions

Post-WIMP

- Mobile
- Gestuelle
- Ambiante
- ...



Types d'interactions



Réalité augmentée

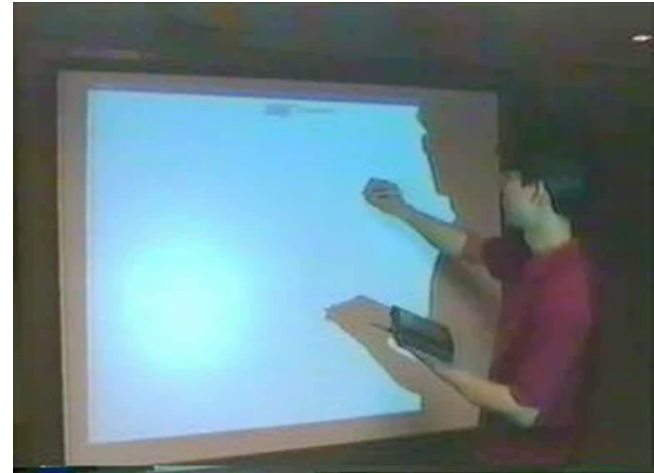
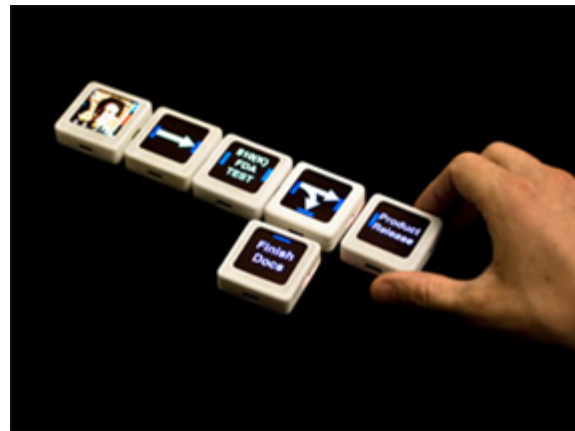


Tableau blanc interactif



Interface tangible

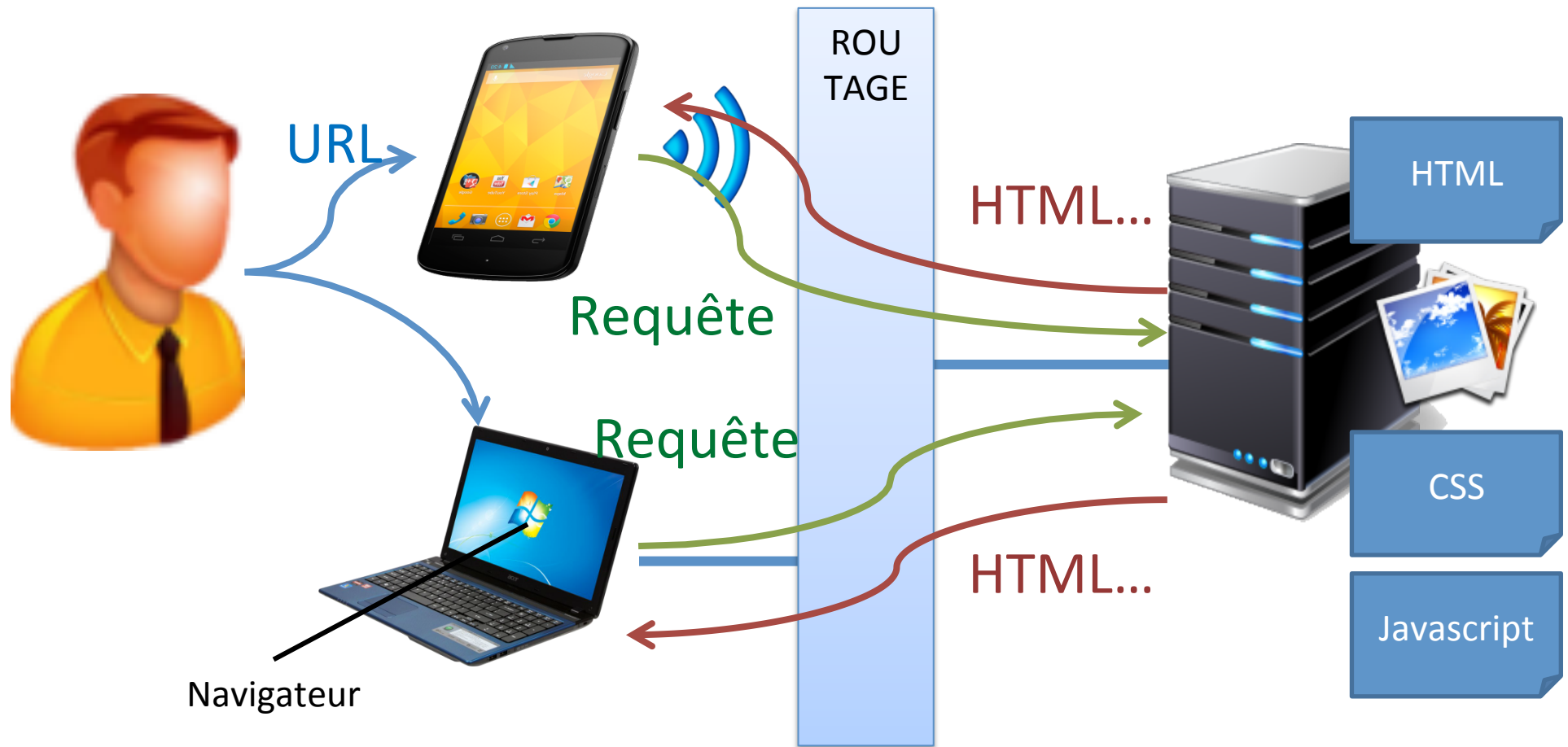
Quel que soit le type d'interaction...

Réaliser des logiciels utiles et utilisables

- Adéquation entre fonctionnalités proposées et besoins des utilisateurs
- Adéquation entre l'interface et les utilisateurs

... EN HTML/CSS/Javascript/Jquery : AUSSI !

HTML : schéma de principe



HTML

- Présentation
- Les balises, les attributs
- Structure d'un document
- Typologie des éléments
- Les chemins
- Bonnes pratiques

Lectures recommandées

- *Premiers pas en CSS3 et HTML5*, de F. Draillard
- *Introduction à HTML5*, de B. Lawson, R. Sharp
- *HTML 5 pour les Web Designers*, de J. Keith
- *Responsive Web Design avec HTML5 et CSS3*, de B. Frain

Une page web c'est...



Le contenu : texte, images, liens, tableaux... que le visiteur va voir concrètement. Le créateur de la page indique librement le contenu qu'il souhaite.

La mise en page : ce qui indique au navigateur comment présenter le contenu. Pour cela on utilise des langages de présentation : HTML, XHTML, HTML5,...

Une page, c'est donc un (des) fichier(s) qui décrivent la structure du document Web et le contenu à afficher.

Extension **.html** ou htm

HyperText Markup Language

Fichiers au **format texte**
permettant la description de **documents Web**

Sous l'autorité du **W3C**

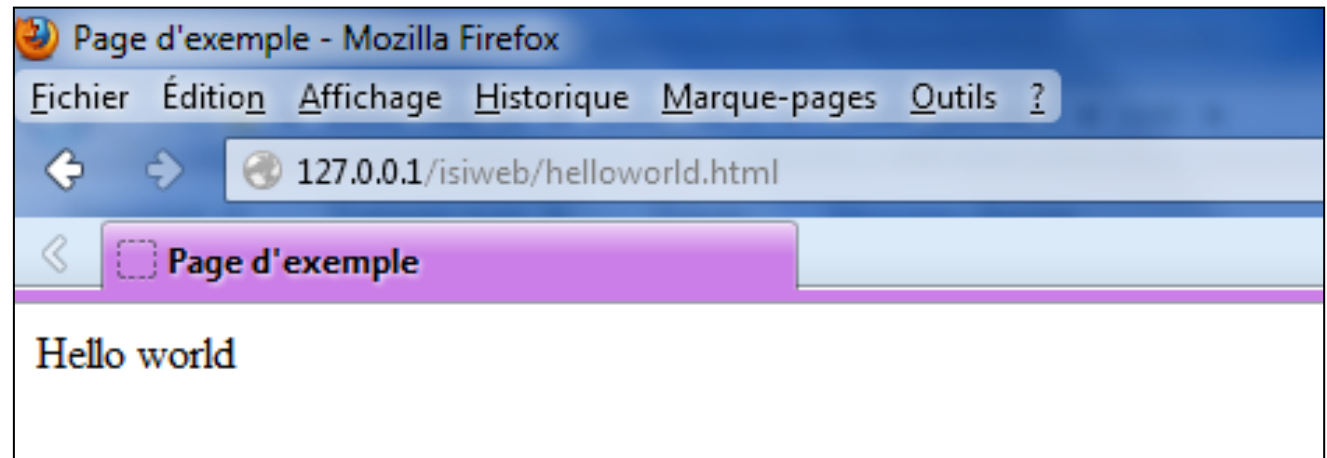
World Wide Web Consortium

<http://www.w3c.org>

Langages non propriétaires

HTML : exemple

```
<html>  
  <head>  
    <meta charset="ISO-8859-1">  
    <title>Page d'exemple</title>  
  </head>  
  <body>  
    <p>Hello world</p>  
  </body>  
</html>
```



HTML : structure générale

```
<html>  
  <head>  
    <meta charset="ISO-8859-1"/>  
    <title>Page d'exemple</title>  
  </head>  
  <body>  
    <p>Hello world</p>  
  </body>  
</html>
```

1- Des codes entre
< et > : les **balises** !

HTML : structure générale

```
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Page d'exemple</title>  
  </head>  
  <body>  
    <p>Hello world</p>  
  </body>  
</html>
```

2- Du contenu

HTML : langage de balisage composé de **balises (ou marqueurs ou tags)**

- Utilité : décrire la **structure** de la page (= définir les zones et les widgets de la page)
- Syntaxe :

`<mot_cle>texte</mot_cle>`
Ou `<mot_cle/>`
- Les balises sont prédéfinies par le langage

Exemples de balises :

<h1>...</h1> : titre de niveau 1 (hheader1)

<h2>...</h2> : titre de niveau 2 (!)

<p>...</p> : paragraphe de texte (par défaut, avec un petit espace avant et après)

**
** : saut de ligne (break) – devrait s'écrire
 pour fermer la balise, mais
 est compris de tous les navigateurs

<title>...</title> : titre de la page (affiché dans l'onglet du navigateur)

Emplacement des balises

- Le document commence par `<html>` et se termine par `</html>`
- Il contient `<head>...</head>` (définition du contexte de la page) et `<body>...</body>` (définition du contenu de la page)
- Entre `<head>...</head>`, on trouve des balises comme `<meta>`, `<title>...</title>` qui permettent de donner des indications au navigateur sur la façon d'interpréter la page. Par exemple `<title>` définit le titre de la page, `<meta>` permet de définir l'auteur, l'encodage,....
- Entre `<body>...</body>`, on trouve les balises de structuration de la présentation (tableaux, découpage en zones,...) et le contenu

Le chevauchement des balises est illégal

Valide :



<h1>Texte<i>sans chevauchement</i></h1>

The diagram illustrates valid HTML tag nesting. Two green arcs are drawn above the text. The first arc starts at the opening <h1> tag and ends at the closing </h1> tag, encompassing the entire content. The second arc starts at the opening <i> tag and ends at the closing </i> tag, encompassing only the italicized text. This shows that the tags are properly nested without overlapping.

Invalide :



~~**<h1>Texte<i>avec chevauchement</h1></i>**~~

The diagram illustrates invalid HTML tag nesting. Two red arcs are drawn above the text. The first arc starts at the opening <h1> tag and ends at the closing </h1> tag. The second arc starts at the opening <i> tag and ends at the closing </i> tag. Because the </h1> tag is closed before the </i> tag, the arcs overlap, indicating that the tags are not properly nested. A horizontal line is drawn through the entire code snippet to indicate it is invalid.

Les attributs : permettent de préciser une valeur dans les balises ouvrantes ou uniques, structure en paires : **nom="valeur"**

<meta charset="UTF-8"/> définit que l'encodage à utiliser est UTF-8 (le fichier doit donc être encodé en UTF8 !)

<p id="unique">Paragraphe unique**</p>**

<p class="special">Paragraphe spécial**</p>**

Apostrophes : doubles (le plus courant) ou simples optionnelles mais (fortement) recommandées

Les attributs peuvent être définis par le langage (ex : id [identifiant de l'objet], class [classe de l'objet], href [URL référencée par un hyperlien],...) ou définis par le programmeur qui peut ajouter tout ce qu'il veut, dans les limites de la syntaxe...

Ex : **id**="id432" **ou**
 class="typeUrgent"
 monCours="les bases de HTML"

Attention :

- **XHTML : case sensitive ; HTML5 : non**
- **Recommandation : noms d'attributs en minuscules**

Certains attributs sont obligatoires sur certaines balises:

~~****~~

Les attributs standards peuvent avoir des valeurs contraintes :

```
<input type="checkbox" name="rouge" value="ok"  
checked="checked"/>
```

```
<input type="map" name="grenoble" value="grenoble"  
checked="known"/>
```

Un attribut standard ne peut être utilisé que sur les balises correspondantes :

``

~~`<p src="url_de_l_image"/>`~~

Il est possible d'ajouter des attributs non prévus dans le langage, en faisant précéder leur nom de "data-"

`<p data-couleurDuCiel="bleu">texte</p>`

est valide.

<p data-couleurDuCiel="bleu">texte</p>

Nota : "data-" non requis en HTML4, son absence est supportée par les navigateurs, mais non conforme au langage HTML5.

Rappel : couleurDuCiel = couleurduciel
couleurDuCiel = non recommandé, mais + facile

<!-- Un commentaire -->

*Non affiché par les navigateurs
Mais comme tout le reste du document
lisible en affichant la source*

On ne met pas de tiret double
à l'intérieur d'un commentaire

<!-- Un commentaire -->

<!--=====-->

sont valides.

~~**<!-- Un autre - commentaire -->**~~

~~**<!--=====-->**~~

ne sont pas valides.

HTML – Structure des documents

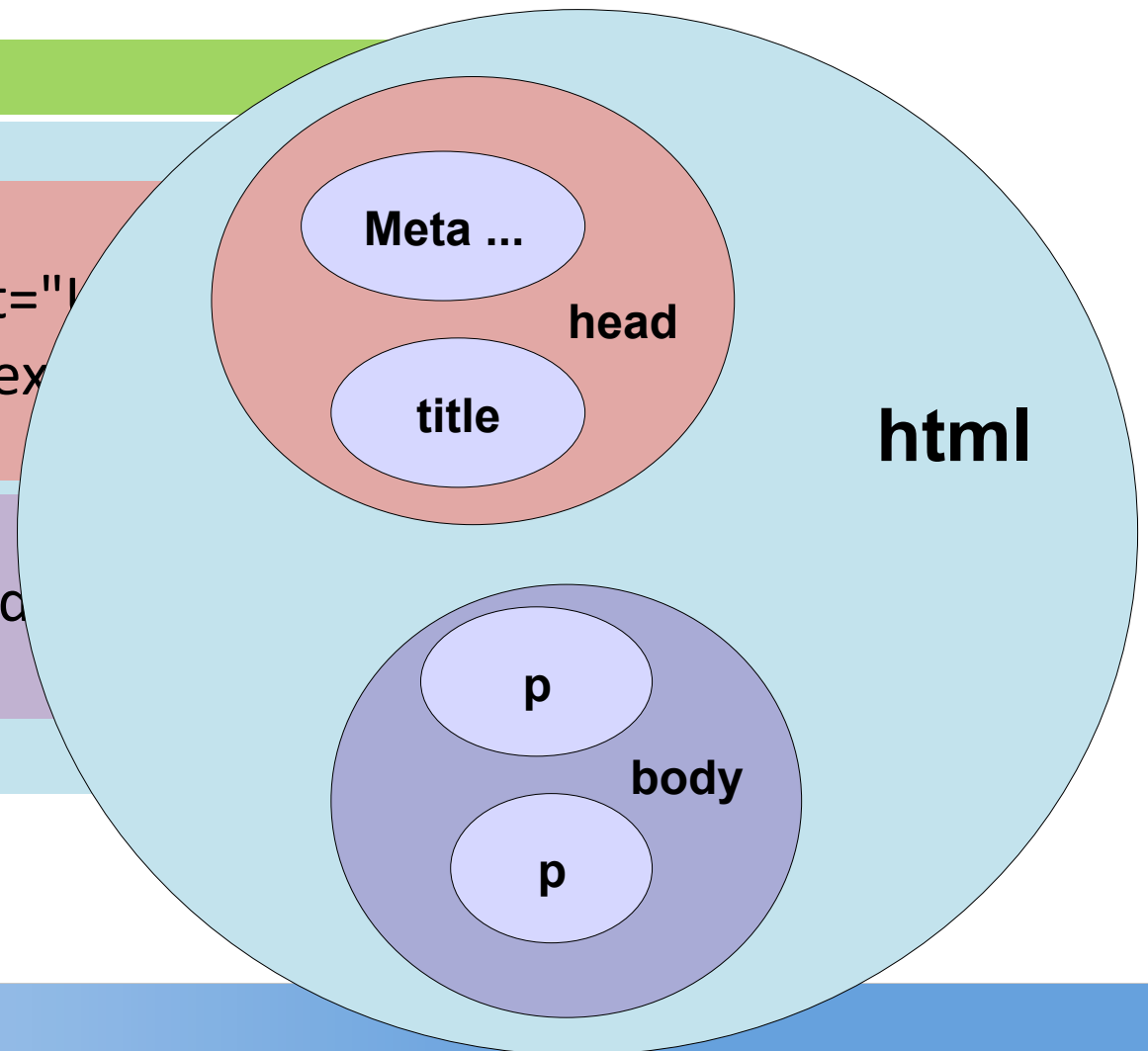
```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Page d'exemple</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

2- Des blocs : le document **Html** qui contient un **en-tête** et un **corps**

HTML – Structure des documents

L'imbrications des éléments

```
<!doctype html>  
<html lang="fr">  
  <head>  
    <meta charset="UTF-8">  
    <title>Page d'exemple</title>  
  </head>  
  <body>  
    <p>Hello world</p>  
  </body>  
</html>
```



Un document HTML comprend plusieurs parties

- Un doctype
- Un élément racine html
 - Un en-tête
 - Un corps

HTML – Structure des documents

- doctype
- html
 - Un en-tête
 - Un corps

```
<!doctype html >  
<html>  
    <head lang="fr">  
    </head>  
    <body>  
    </body>  
</html>
```

HTML – Structure des documents

Déclaration du type de document

Le tag doctype indique au navigateur dans quel type de HTML la page a été écrite (HTML-3.2 «classique», HTML-4 de transition ou strict, XHTML, etc...).

`<!doctype html>` signifie : document en HTML5.

Utile pour les navigateurs, les lecteurs de votre code ou les validateurs de code

HTML – Structure des documents

Langue

La déclaration de la langue se fait dans la balise html.

```
<html lang="fr">
```

Nous déclarons ici que la langue est le français, ce qui sert au référencement (ex : Google vous propose les pages en français ou une traduction) et aux synthétiseurs vocaux.

- Délimité par les balises **<head>** et **</head>**
- Contient des informations non affichées par les navigateurs courants
 - Le titre
 - Les méta-données
 - Les références à d'autres ressources
 - Le type d'encodage des caractères

L'en-tête : le titre

`<title>`Mon document`</title>`

Le titre n'est pas directement affiché dans le document, mais souvent par le navigateur, dans l'interface (nom de l'onglet, par exemple)

L'en-tête : les méta-données

Ce sont :

- **mots clés et description** : décrire le contenu du document et faciliter le référencement
- **informations / instructions aux moteurs de recherche** : indiquent s'il faut suivre les liens lors de l'indexation d'un site, la fréquence de relecture souhaitée
- D'autres méta données permettent de préciser **le nom de l'auteur, la version...**

HTML – Structure des documents

L'en-tête : les méta-données

Exemples :

- `<meta name="description" content="Cours SIL-2, IUT2 de Grenoble"/>`
- `<meta name="author" content="Pierre Dupont">`
- `<meta name="keywords" content="sil2, html, css">`
- `<meta name="robots" content="[[no]index][no[flollow]]"/>`

L'en-tête : encodage

Le type d'encodage des caractères de la page

<meta charset="encodage"/>

ATTENTION : doit être cohérent avec l'encodage physique de la page.

Pour le français on dispose des encodages suivants :

iso-8859-1 : encodage classique pour les langues de l'Europe occidentale (aussi appelé Latin-1)

iso-8859-15 : même encodage comportant quelques caractères supplémentaires comme le signe €...

utf-8 : encodage pour les caractères de la majorité des langues mondiales

A noter : **L'en-tête : encodage**

En utilisant l'encodage iso-8859-1 ou iso-8859-15 les caractères ASCII 7-BIT (codes 32 à 127) sont valides, avec 4 exceptions car ce sont des caractères du langage XHTML.

Ces exceptions sont codées avec des entités :

" (codé **"**;))

& (codé **&**;))

< (codé **<**;))

> (codé **>**;))

L'en-tête : encodage

En utilisant l'encodage iso-8859-1 ou iso-8859-15 :

Les caractères, en dehors de la classification ASCII 7-BIT (donc les codes de 128 à 255), sont codés par **des entités ou des références numériques** :

Par exemple :

é (codé **é** ou **é**;)

è (codé **è** ou **è**;)

à (codé **à** ou **à**;)

ô (codé **ô** ou **,**;)

etc.

Remarque :
le blanc
insécable s'écrit
** **;

L'en-tête : les références

Les références permettent de... référencer des ressources utilisées par le document : feuilles de style CSS, icône, fichiers de scripts externes Javascript...

Exemples :

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

```
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />
```

Le corps du document

Le corps d'un document est délimité par les balises `<body>` et `</body>`. Il peut contenir divers éléments :

- du texte (titres, paragraphes, listes, etc.)
- des images
- des hyperliens
- des tableaux
- des formulaires
- des cadres
- des objets externes (applets Java, Flash ...)
- etc.

HTML – Typologie des éléments

Structure et rendu

Chaque élément a une double identité :

- Sa **structure** (mot clé, attributs standards, ...)
- Son **rendu** (ou apparence), qui est défini par défaut pour chaque navigateur (et qui peut être modifié par CSS).

HTML – Typologie des éléments

Block et inline

2 grands mode de rendu des éléments

Les rendus de type **block**

Les rendus de type **inline**

Cette typologie dicte le comportement
en terme de **positionnement** et d'**affichage**

HTML – Typologie des éléments

Les éléments de type Block

Ce sont :

- Des blocs dans les documents – Exemples : paragraphes, listes...
 - Apparaissent les **uns en dessous des autres**
 - Ont des **dimensions et des marges externes ou internes fixées** par défaut, à l'exception des blocs **<div>**
 - Sont **positionnables** (avec les feuilles de style CSS)

HTML – Typologie des éléments

Les éléments de type Block

Ils peuvent :

- **contenir d'autres blocs**
sauf les blocs de paragraphes (<p>) et de titres (<h1>, <h2>, ... <h6>) qui ne peuvent contenir d'autres blocs
- **Contenir des éléments inline**

HTML – Typologie des éléments

Les éléments de type Block

Exemples

- `<h1></h1>`
- `<h2></h2>`
- ...
- `<h6></h6>`
- `<p></p>`
- `<table></table>`
- ``
- ``
- `<blockquote></blockquote>`
- `<dl></dl>`
- `<div></div>`
- etc.

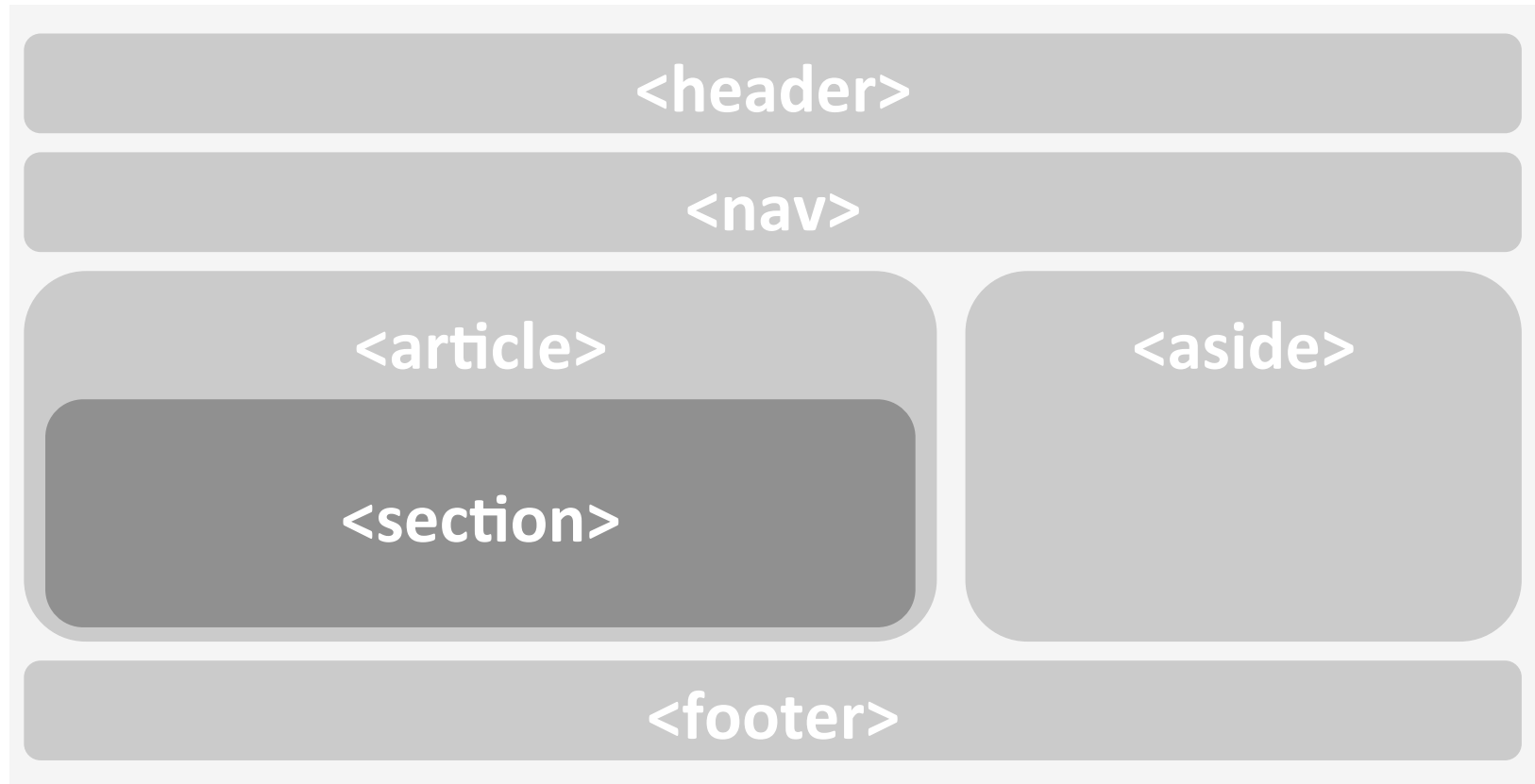
Éléments de type block pour balisage sémantique

N.B. :
proximité
des notions
d'article et de
section.

Section : regrouper
du contenu
apparenté

Article : contenu
apparenté
autonome (?)

Aside : contenu indirectement apparenté

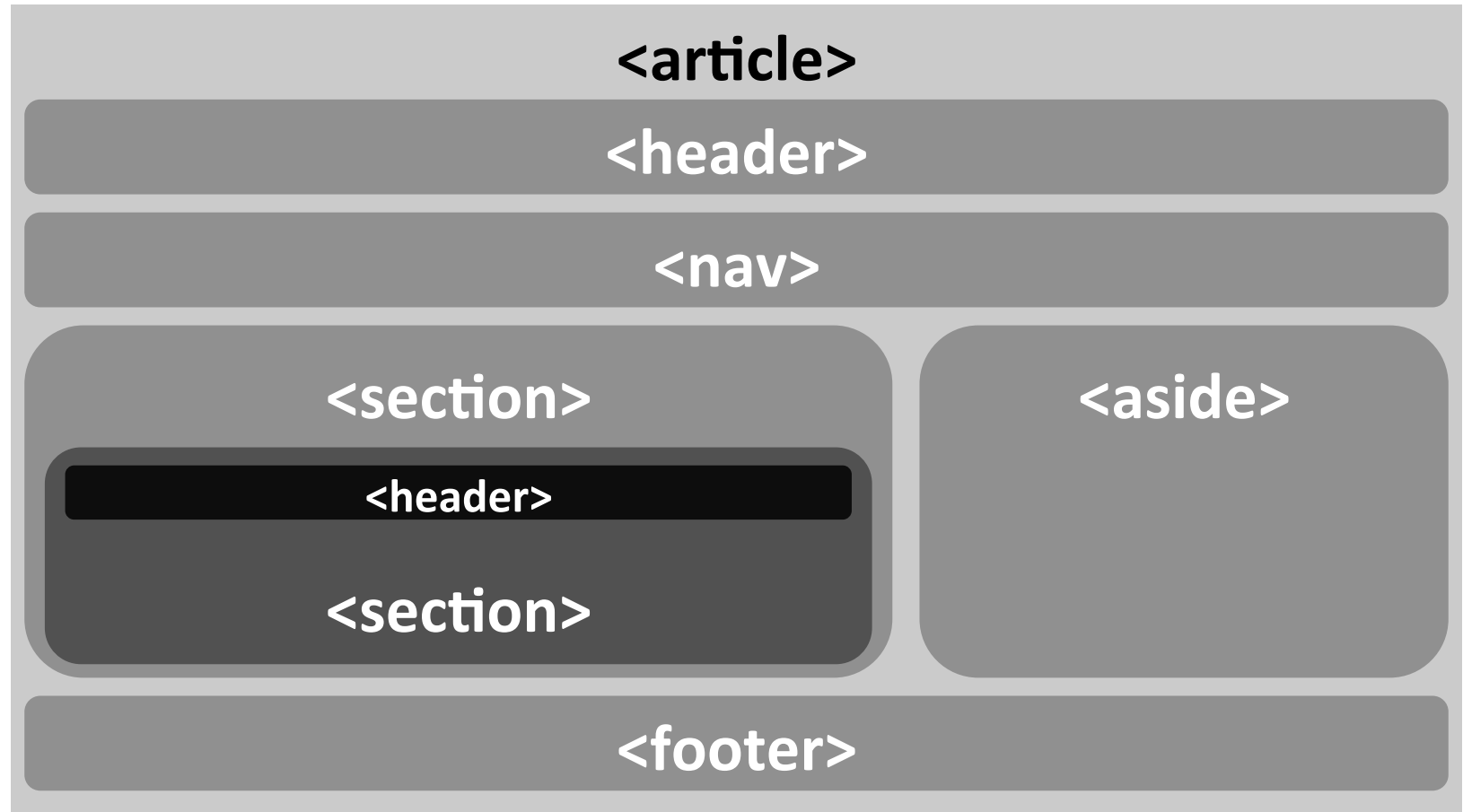


HTML – Structure des documents

Imbrication des éléments

Sections et articles peuvent contenir les autres éléments, y compris des sous-sections et des sous-articles

==> définir la sémantique de ces éléments au regard du contenu du site.



HTML – Structure des documents

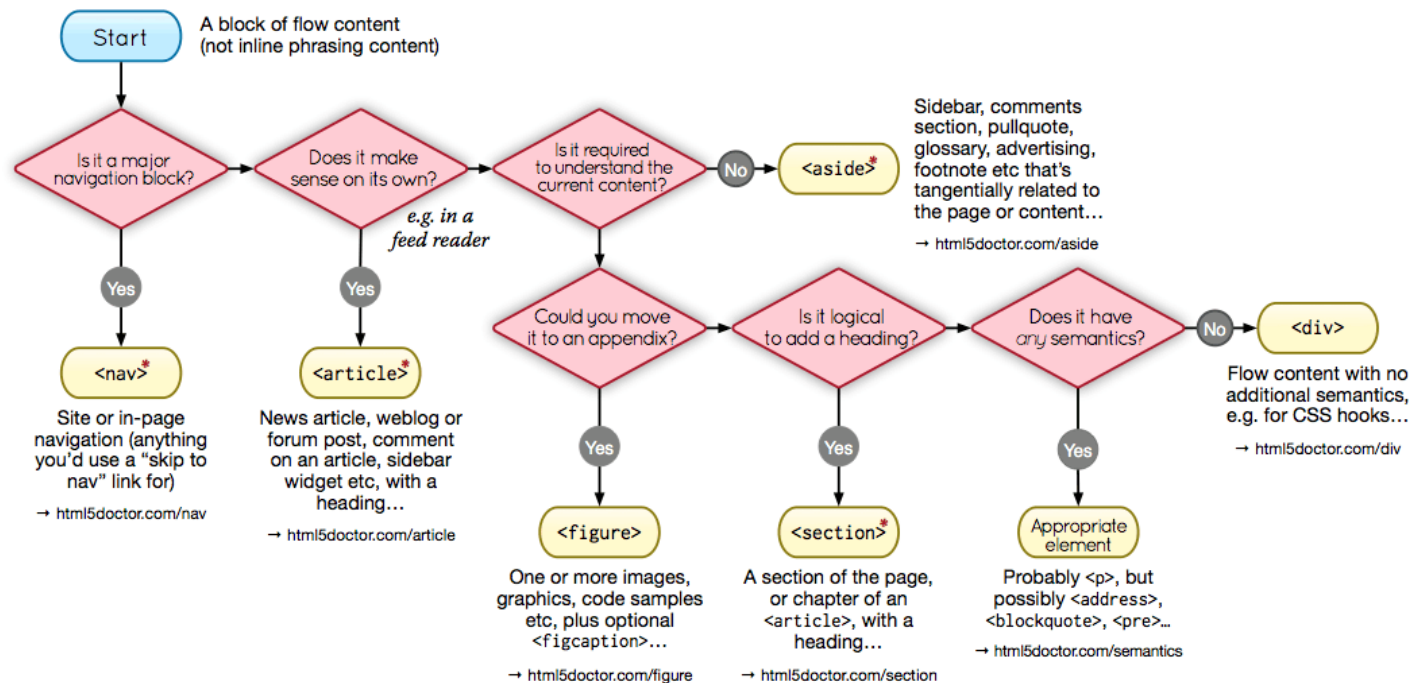
Éléments de type block pour balisage sémantique



HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet
www.html5doctor.com



* Sectioning content element
These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

2011-07-22 v1.5
For more information:
www.html5doctor.com/semantics

Les éléments inline

- Apparaissent **au fil du texte**, ils ne sont pas placés les uns au dessus des autres (ils restent à l'emplacement défini).
- N'ont **pas de marges internes ou externes** par défaut
- Ne sont **pas dédiés à un positionnement précis** (même si cela est possible avec les CSS)
- Servent à modifier, enrichir des portions de textes, apporter du sens

HTML & XHTML – Typologie des éléments

Les éléments inline

Les éléments inline :

- Ne peuvent **contenir que des éléments inline**
(→ **pas de block**)
- Un **élément inline doit être contenu dans un élément de type block**



Les balises apportées par HTML 5

- **time** : date et/ou heure
 - Accompagnée (ou non) de datetime au format ISO AAAA-MM-JJThh:mm:ss (complètement ou non)
 - S'il n'y a pas d'attribut datetime, le contenu doit être au format ISO

```
<time datetime="18:00">18 h</time>  
<time datetime="2010-11-15">15 novembre</time>  
<time>2010-11-15</time>
```

HTML – Typologie des éléments

Les éléments inline

Exemples :

- `<a>`
- ``
- ``
- ``
- `<i></i>`
- ``
- ``

Les éléments inline

Note : en HTML4, certaines balises définissaient un rendu visuel (ex : `` = texte en gras. Ce n'est plus le cas en HTML5.

- **b** (avant : bold) ne signifie pas « en gras », mais stylistiquement décalé sans avoir une importance supplémentaire (sinon, utiliser strong)
- **i** (avant : italic) ne signifie pas « en italique », mais « dans une voix ou une humeur alternative » sans emphase ni importance particulière (sinon, utiliser em)
- **hr** (avant : ligne horizontale) prend un sens sémantique de séparation entre paragraphes (potentiellement rendu comme une ligne horizontale)
- **small** ne signale plus un texte en petite taille, mais caractérise des éléments « en petits caractères » comme un copyright ou une référence écrites « en petits caractères »

Les structures universelles

div et span = éléments pour structurer des documents
Web (en association avec les CSS)

div = élément de type bloc

span = élément en ligne.

n'apportent aucune contrainte de présentation, ils sont
« neutres » à cet égard. Ils servent à « ajouter » de la
structure.

Attention, ces éléments n'ont pas de sens particulier, ils sont neutres également sur le plan de la sémantique. Par conséquent, ils ne doivent pas remplacer systématiquement les autres éléments.

Un document est positionné dans une **arborescence de répertoires et de fichiers**.

Le répertoire de plus haut niveau (contenant tous les autres documents ou fichiers), est appelé la « **racine** ».

Les chemins absolus

On peut exprimer la destination d'un lien de manière absolue (depuis la racine) :

/rep/sousrep/index.html

*Problème : si on déplace l'ensemble des documents, par exemple pour les inclure dans un répertoire placé différemment par rapport à la racine, **les chemins ne seront plus valides.***

Les chemins relatifs

- Descendre dans l'arborescence, vers un sous répertoire :
nom_sous_repertoire/ ou **./nom_sous_repertoire/**
./ représentant le positionnement courant
- Remonter dans l'arborescence :
../
- Remonter de plusieurs niveaux :
../../../
- Remonter puis de redescendre dans l'arborescence :
../../autre_repertoire/autre_sous_repertoire

- **Pas destiné à remplacer le positionnement !**
- **Balises `<table>` `</table>`**
- Il est souhaitable d'ajouter à la balise `<table>` l'attribut **summary** = indiquer un résumé du tableau (`<table summary="ce que contient le tableau">`).

- Chaque **ligne** est encadrée par `<tr> </tr>`
- Les cellules d'en-tête sont encadrées par `<th> </th>`
- Les cellules de valeur sont encadrées par `<td> </td>`

- Les balises **<thead></thead>** **<tfoot></tfoot>** **<tbody></tbody>** permettent de structurer les tableaux
- La balise **<caption></caption>** permet d'indiquer la légende du tableau.

Exemple

```
<table>
  <caption>L'acronyme</caption>
  <thead>
    <tr>
      <th>Cellule d'en t te A</th>
      <th>Cellule d'en t te B</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Cellule de pied de tableau A</td>
      <td>Cellule de pied de tableau B</td>
    </tr>
  </tfoot>
</table>
```

...suite

```
<tbody>
  <tr>
    <td>Valeur A ligne 1</td>
    <td>Valeur B ligne 1</td>
  </tr>
  <tr>
    <td>Valeur A ligne 2</td>
    <td>Valeur B ligne 2</td>
  </tr>
</tbody>
</table>
```

L�gente du tableau	
Cellule d'en t�te A	Cellule d'en t�te B
Valeur A ligne 1	Valeur B ligne 1
Valeur A ligne 2	Valeur B ligne 2
Cellule de pied de tableau A	Cellule de pied de tableau B

Colonnes étendues

```
<table>
  <tr>
    <th colspan="2">Cellule d'en t&ecirc;te &eacute;tendue en largeur</th>
  </tr>
  <tr>
    <td>Valeur A ligne 1</td>
    <td>Valeur B ligne 1</td>
  </tr>
  <tr>
    <td>Valeur A ligne 2</td>
    <td>Valeur B ligne 2</td>
  </tr>
</table>
```

L'attribut **colspan** crée des cellules qui s'étendent sur plusieurs cellules d'un tableau, en ligne

Cellule d'en tête étendue en largeur	
Valeur A ligne 1	Valeur B ligne 1
Valeur A ligne 2	Valeur B ligne 2

Lignes étendues

```
<table>
```

```
<tr>
```

```
<th>Cellule d'en t&ecirc;te A</th>
```

```
<th>Cellule d'en t&ecirc;te B</th>
```

```
</tr>
```

```
<tr>
```

```
<td rowspan="2">Valeur A ligne 1 et 2 (&eacute;tendu)</td>
```

```
<td>Valeur B ligne 1</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Valeur B ligne 2</td>
```

```
</tr>
```

```
</table>
```

L'attribut **rowspan** crée des cellules qui s'étendent sur plusieurs lignes d'un tableau

Cellule d'en tête A	Cellule d'en tête B
Valeur A ligne 1 et 2 (étendu)	Valeur B ligne 1
	Valeur B ligne 2

Balise ****. Ses principaux attributs sont :

- src : emplacement du fichier source de l'image
- width : largeur
- height : hauteur
- alt : texte qui apparaît lorsque l'image ne s'affiche pas et comme info bulle de l'image

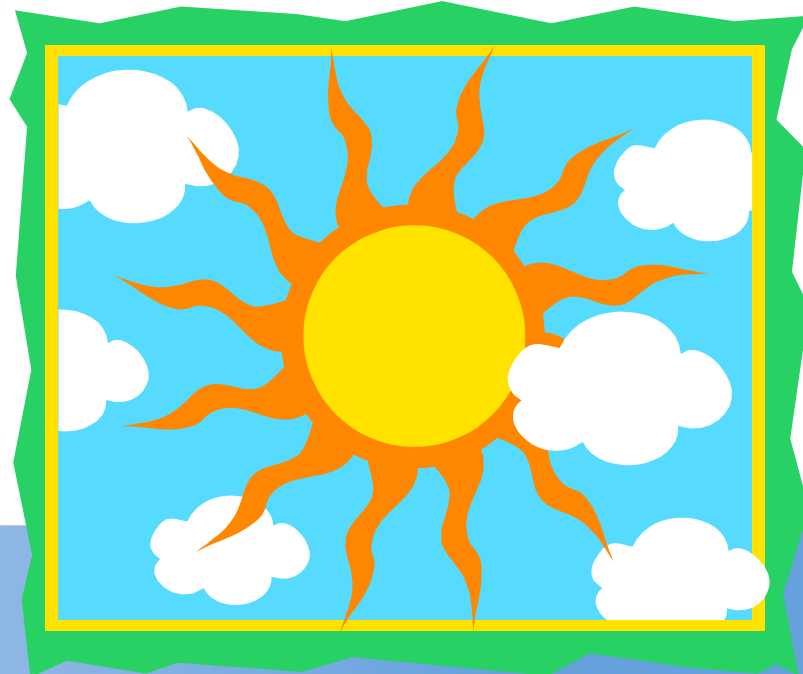
```

```

Les principaux formats d'images

GIF (Graphic Interchange Format)

- Pour les graphismes aux tracés simples, sans dégradé de couleurs
- Limité à une palette de 256 couleurs (choisies parmi des millions).
- Peut être animé.
- Gère la transparence.



JPG OU JPEG (Joint Picture Expert Group)

- Adapté aux photos ou aux images présentant de nombreux dégradés
- Gère la compression (avec perte)



PNG (Portable Network Graphic)

- A l'origine : format alternatif au GIF
- Peut remplacer le GIF (éventuellement le JPG)
- Méthode de compression améliorée
- Attention, il n'est pas supporté par les anciennes versions des navigateurs.
- La transparence peut poser quelques soucis avec certaines versions de navigateurs

- Lien vers une autre page - même site - même répertoire
Titre du lien
- Lien vers une autre page - même site - autre répertoire
Titre
- Lien vers une autre page - nouvelle fenêtre du navigateur
Lien

- Liens vers une page du même site
Suite
- Liens vers une page d'un autre site
Un site
- Lien vers un fichier
Document PDF
Document PDF

- Lien vers un fragment de document

- Définition d'un fragment :

`Sommet`

- Accès au fragment :

`Lien`

`Lien`

- Lien sous forme d'image vers une autre page
``
- Lien spécifique pour téléchargement de fichier (tous formats)
``
le nom de fichier par défaut n'est pas requis : on peut écrire
``

Valider ses documents ?

La validation des documents, même si elle est **nécessaire**, n'est **pas suffisante** : seule la validité de la syntaxe est vérifiée, en aucun cas la qualité structurelle ou sémantique du document.

validator.w3.org