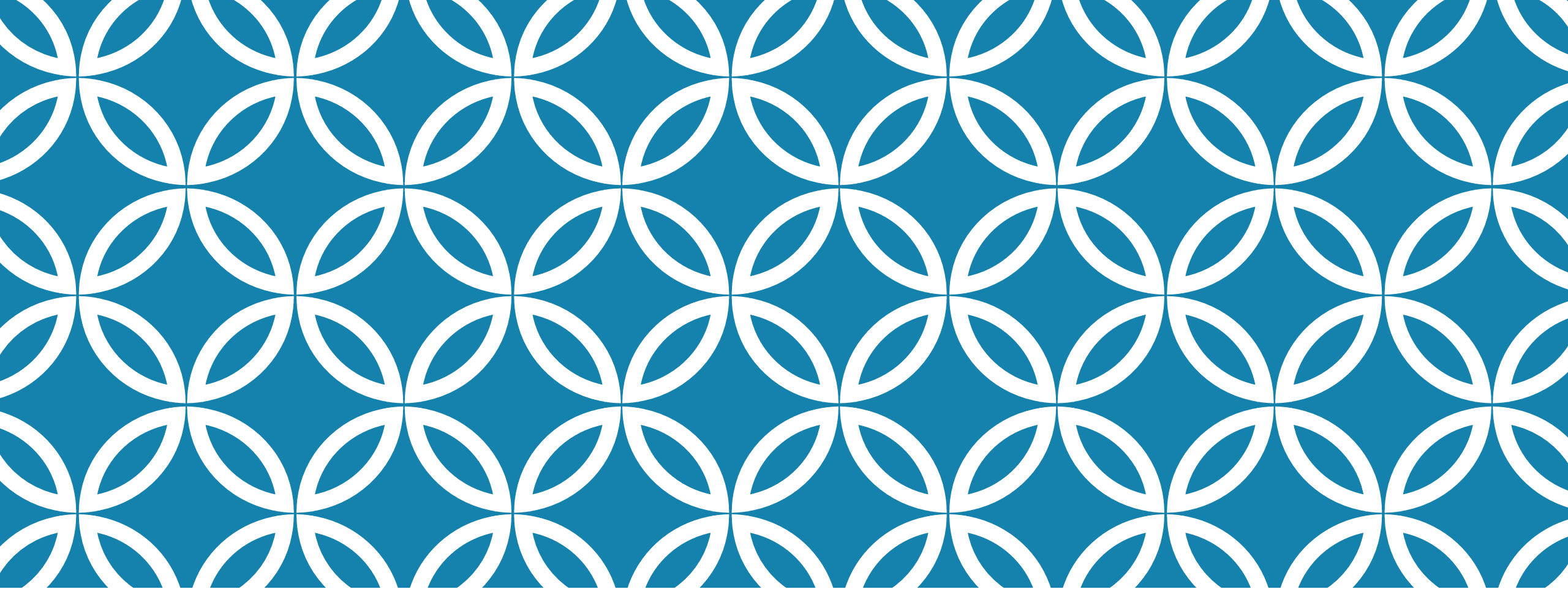


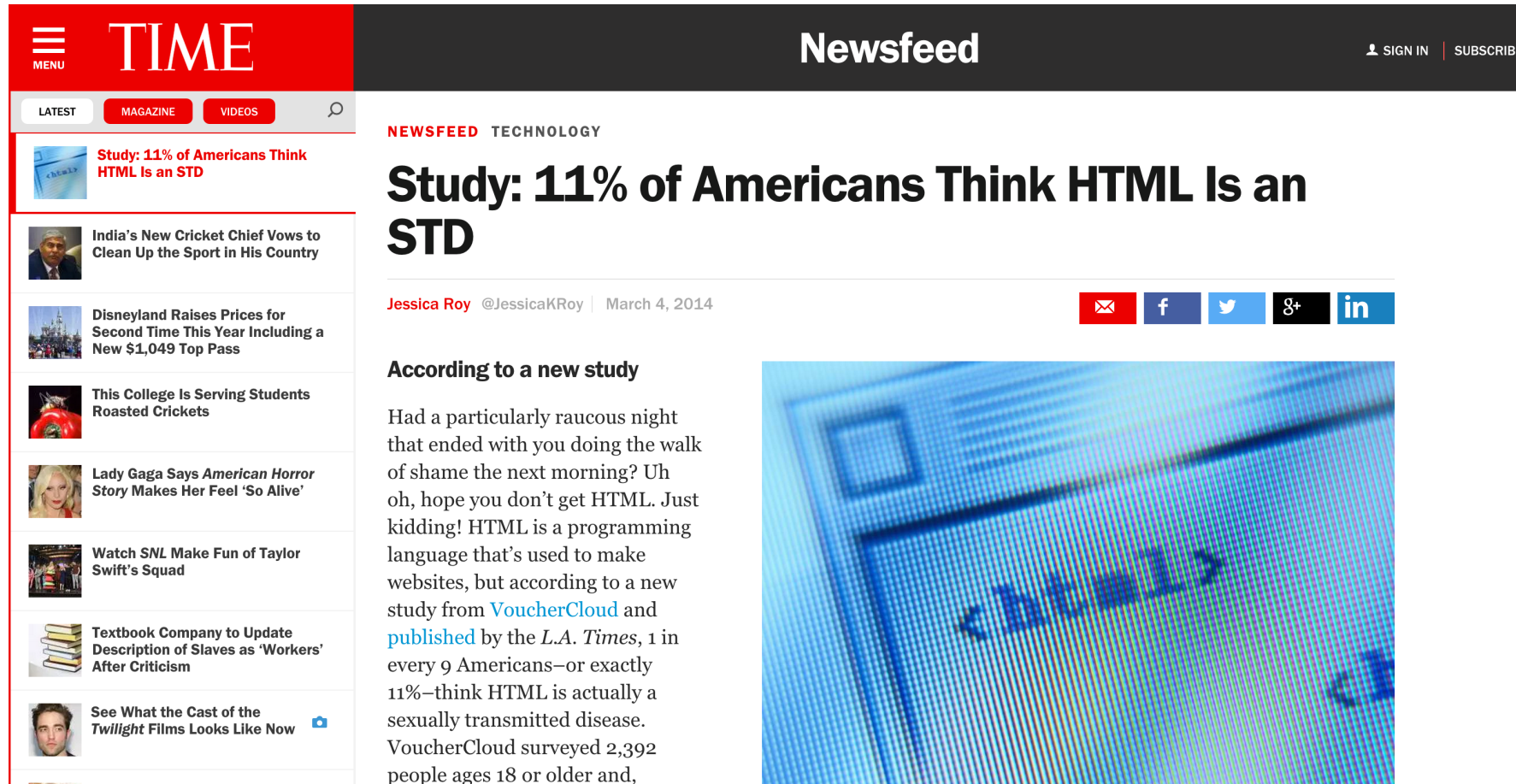


INTRODUCTION AU WEB

LE LANGAGE HTML

Par Jean-Pierre Lozi
Basé sur les cours d'Andrea
Tettamanzi et Philippe Renevier

QU'EST-CE QUE HTML ?



The screenshot shows the TIME Newsfeed interface. The header includes the TIME logo, a menu icon, and the word "Newsfeed". Navigation tabs for "LATEST", "MAGAZINE", and "VIDEOS" are visible. The main article is titled "Study: 11% of Americans Think HTML Is an STD" by Jessica Roy, dated March 4, 2014. The article text discusses a survey by VoucherCloud and the L.A. Times, stating that 11% of Americans consider HTML a sexually transmitted disease. A sidebar on the left lists other news items, and a social media sharing bar is located below the article title. A large image of a computer screen displaying HTML code is positioned to the right of the article text.

TIME Newsfeed

SIGN IN | SUBSCRIBE

LATEST MAGAZINE VIDEOS

NEWSFEED TECHNOLOGY

Study: 11% of Americans Think HTML Is an STD

Jessica Roy @JessicaKRoy | March 4, 2014

According to a new study

Had a particularly raucous night that ended with you doing the walk of shame the next morning? Uh oh, hope you don't get HTML. Just kidding! HTML is a programming language that's used to make websites, but according to a new study from [VoucherCloud](#) and published by the *L.A. Times*, 1 in every 9 Americans—or exactly 11%—think HTML is actually a sexually transmitted disease. VoucherCloud surveyed 2,392 people ages 18 or older and,

India's New Cricket Chief Vows to Clean Up the Sport in His Country

Disneyland Raises Prices for Second Time This Year Including a New \$1,049 Top Pass

This College Is Serving Students Roasted Crickets

Lady Gaga Says *American Horror Story* Makes Her Feel 'So Alive'

Watch *SNL* Make Fun of Taylor Swift's Squad

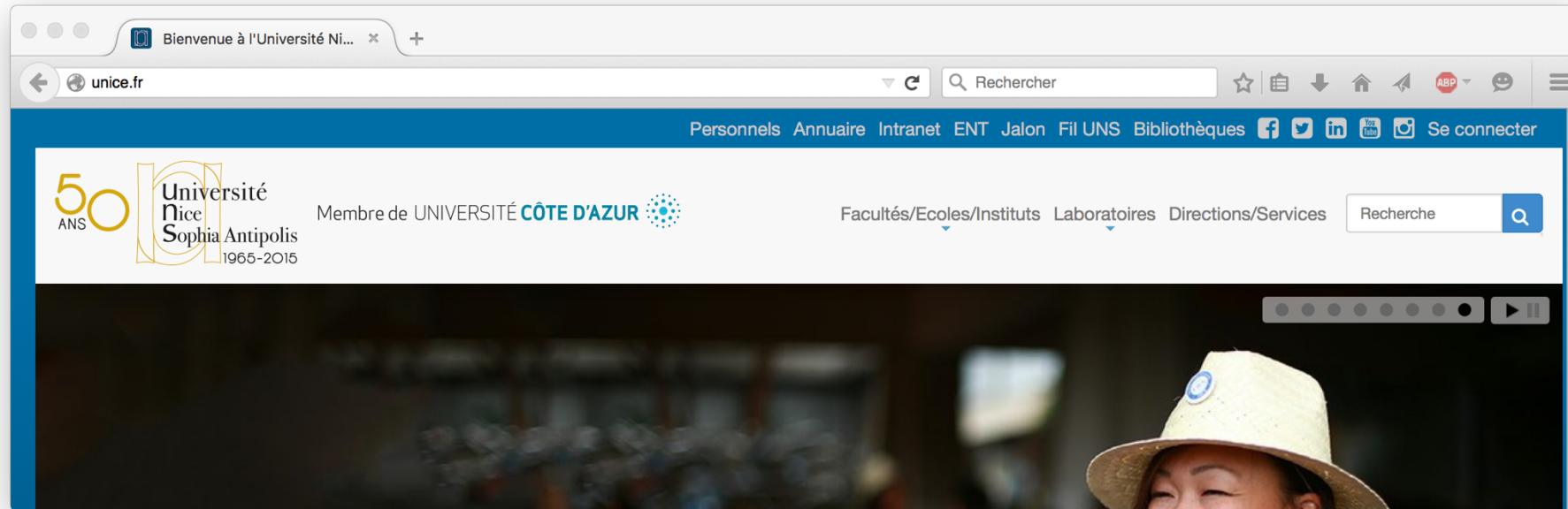
Textbook Company to Update Description of Slaves as 'Workers' After Criticism

See What the Cast of the *Twilight* Films Looks Like Now

`<html>`

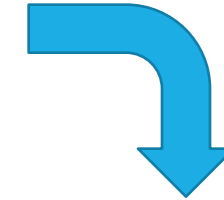
LE LANGAGE HTML : PRÉSENTATION

- Les pages web sont écrites en HTML (.htm ou .html) : HyperText Markup Language
- **Ce qu'est HTML :**
 - Un langage descriptif : contenu de la page, description des paragraphes, images, liens, tableaux...
 - Pas de compilation : le navigateur reçoit le HTML, l'interprète, et produit la page telle qu'affichée au client. Toujours possible pour le client de consulter le code HTML !
- **Ce qu'HTML n'est pas :**
 - Un langage de programmation :
 - Pas de structures conditionnelles (« if ... then ... else »)
 - Pas de boucles (for, while, ...)
 - Pour programmer, il faudra utiliser un langage de programmation côté client (javascript, flash...) ou côté serveur (PHP, perl...)



```
Source de : http://unice.fr/

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr"><head><base href="http://unice.fr/front-page" /><base href="http://unice.fr/front-
3   jQuery(function($){
4     $.datepicker.setDefaults(
5       jQuery.extend($.datepicker.regional['fr'],
6         {dateFormat: 'dd/mm/yy'}));
7   });
8   </script><link rel="alternate" href="http://unice.fr/RSS" title="Université Nice Sophia Antipolis - RSS 1.0" type="application/rss+xml" /><li
9     <link rel="stylesheet" href="/++theme++ThemeUNS/assets/css/font-awesome-ie7.min.css">
10  <![endif]--><!--[if lt IE 9]>
11  <script src="/++theme++ThemeUNS/assets/js/html5shiv.js"></script>
12  <script src="/++theme++ThemeUNS/assets/js/respond.min.js"></script>
13  <![endif]--><meta property="og:title" content="" /><meta property="og:description" content="" /><meta property="og:image" content="" /><meta
14
15  <div class="container">
16
17    <nav class="navbar navbar-static-top navbar-inverse visible-xs" role="navigation" id="navigation_globale"><div class="navbar-header">
18      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbars1-collapse">
19        <span class="icon-bar"></span>
20        <span class="icon-bar"></span>
21        <span class="icon-bar"></span>
22      </button>
23      <a class="navbar-brand" href="http://unice.fr">UNS</a>
24    </div>
25
26    <div class="collapse navbar-collapse navbars1-collapse">
```



←
→
↺
☆

Partager cette page
 Enregistrer sous...

Afficher l'image de fond
 Tout sélectionner

Code source de la page
 Informations sur la page

Examiner l'élément



LES LANGAGES DE BALISAGE

- Des langages spécialisés dans l'enrichissement (markup) d'information textuelle
- Balise = unité syntaxique délimitant une séquence de caractères à l'intérieur d'un flux de caractères (par exemple un fichier texte). Hiérarchique : des balises dans des balises...
- L'inclusion de balises permet de spécifier, à la fois :
 - La structure (logique, graphique...) du document
 - Son contenu
- Les balises sont faciles à analyser par un programme (navigateur par exemple), ce qui permet un traitement automatisé de son contenu
- Historiquement, les langages à base de balise servent surtout à structurer ou formater des documents

SGML → HTML → XML

- Historiquement, HTML est une application dérivée de SGML
- SGML = Standard Generalized Markup Language
- Inventé par Charles Goldfarb pour rationaliser le système documentaire d'IBM, devient un standard ISO en 1986
- XML : une extension de SGML pour structurer des données hiérarchiquement
 - Pas seulement pour la mise en page
 - Très à la mode depuis quelques années
 - Utilisé par exemple dans les flux RSS
- XHTML est une variante de HTML qui est aussi du XML
 - Plus facile à traiter automatiquement

LES BALISES

- Une balise est définie par un identifiant/étiquette, par exemple h1
 - En-tête (titre) de plus haut niveau (h = header, 1 = premier niveau)
- Balise d'ouverture : `<h1>`
- Balise de fermeture : `</h1>`
- Un couple de balises (d'ouverture et de fermeture) délimite un bloc de texte
- Balises vides : `<br />` (br = line break, retour à la ligne) juste `<br>` possible si pas XHTML
- Les balises sont imbriquées hiérarchiquement :

```
<body>
```

```
    <h1>Titre</h1>
```

```
    <p>Paragraphe</p>
```

```
</body>
```

INTERPRÉTATION DE HTML

- Éléments reconnus lors de l'interprétation par le navigateur :
 - `<NOM_DE_BALISE> ... </BALISE>`
 - `<BALISE />` ou juste `<BALISE>` si pas XHTML
 - Texte (espaces, tabulations, retours à la ligne consécutifs considérés comme un seul espace)
 - Certains caractères non autorisés (lettres accentuées en ASCII...), peut dépendre de l'encodage
 - Façon la plus sûre : utiliser des codes, évite problèmes d'encodage (ex : «J'aime l'√©at√©a» au lieu de «J'aime l'été»)
 - Format : `&code;`

é	é
è	è
ê	ê
É	É
ñ	ñ

ï	ï
espace	
<	<
>	>
&	&

Liste complète :

<http://dev.w3.org/html5/html-author/charref>

EXEMPLE DE STRUCTURATION

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Texte organisé visuellement

<h1>Le dormeur du val**</h1>**

<p>C'est un trou de verdure où chante une rivière**
**
Accrochant follement aux herbes des haillons**
**
D'argent ; où le soleil, de la montagne fière, **
**
Luit : c'est un petit val qui mousse de rayons.**</p>**

<p>Un soldat jeune, bouche ouverte, tête nue, **
**
Et la nuque baignant dans le frais cresson bleu, **
**
Dort ; il est étendu dans l'herbe, sous la nue, **
**
Pâle dans son lit vert où la lumière pleut.**</p>**

<p>Les pieds dans les glaïeuls, il dort. Souriant comme**
**
Sourirait un enfant malade, il fait un somme :**
**
Nature, berce-le chaudement : il a froid.**</p>**

<p>Les parfums ne font pas frissonner sa narine ;
**
**Il dort dans le soleil,

la main sur sa poitrine, **
**

Tranquille. Il a deux trous rouges au côté droit.**</p>**

Texte structuré pour l'ordinateur

HTML : STANDARDISATION

- HTML n'appartient pas à une seule entreprise : plusieurs navigateurs ! (Microsoft, Google...)
- Doit être standardisé, pour que chacun sache précisément comment interpréter le langage
 - Si approximatif, pages web affichées différemment d'un navigateur à l'autre !
 - C'était le cas il y a longtemps, de moins en moins le cas, standards de mieux en mieux implémentés
- Standardisé par le World Wide Web Consortium, ou W3C
 - Créé par Tim Berners-Lee
 - Standardise de nombreuses technologies web (HTML, XHTML, XML, CSS, PNG, SVG, SOAP, RDF...)

<http://www.w3.org/>

DOCUMENT HTML MINIMAL

- En 1997, HTML 4.01, resté très courant pendant très longtemps
- En 2000, XHTML 1.0, très similaire mais peut-être analysé par des parseurs XML
- Depuis 2014 : HTML 5.0 ! Nous utiliserons cette version (<http://www.w3.org/TR/html5/>)
- Document HTML 5.0 minimal :

<code><!doctype html></code>	<code><- Le type de document</code>
<code><html></code>	
<code><head></code>	<code><- En-tête : contient informations sur la page</code>
<code> <meta charset="utf-8"/></code>	<code><- Encodage utilisé : UTF-8 par défaut</code>
<code> <title>...</title></code>	<code><- Titre de la page</code>
<code></head></code>	
<code><body></code>	<code><- "Corps" du document</code>
<code>...</code>	<code><- Mettre ici le contenu du document</code>
<code></body></code>	
<code></html></code>	

LES BALISES : PLUS EN DÉTAIL

- Les balises conteneurs contiennent :
 - Du texte
 - D'autres balises... parmi celles autorisées !
 - Par exemple, balises « bloc » (<div>, <h1>, <p>, <form>, souvent les éléments qui commencent sur une nouvelle ligne) et « en ligne » (, , , se suivent comme du texte)
 - Une balise bloc peut contenir d'autres balises bloc, des éléments en ligne, du texte
 - Un élément en ligne ne peut pas contenir de balise bloc
 - Balises peuvent contenir seulement certains types de contenu, on verra ces types plus loin
 - Utilisées pour présenter ou structurer le document
 - Les balises vides : sont destinées à insérer un élément hors du texte dans le document : séparateur (<hr />), saut de ligne, saut de paragraphe...
 - Commentaires : ignorés par l'interpréteur : <!-- Commentaire -->

LES BALISES : PLUS EN DÉTAIL

- Les balises peuvent prendre des attributs
- Précise la nature ou le fonctionnement de la balise
 - Certains attributs sont obligatoires
 - Propres à chaque balise : voir par exemple <https://www.w3.org/TR/html5/>
 - Exemple : ``
 - Autre exemple : `<a href="http://www.unice.fr">Le site de l'université</a>`
- Conseil : valeur entre guillemets
 - Pour attributs « boolean », i.e., juste la présence, utiliser par exemple `reversed="reversed"`
 - Juste `reversed` possible en HTML 5 (mais pas en XHTML, pas recommandé XML invalide)

LES BALISES : TOUR D'HORIZON

- `<html>` : la balise racine (root)
- Les métadonnées
 - `<head>`, `<title>`, `<link />` : pour ajouter une feuille de style CSS
 - `<meta>` : description, mots-clé, auteur, `<style>` : CSS directement dans le document
 - `<base />` : racine par défaut des URLs sur la page
- Les sections (nombreux tags nouveaux dans HTML5 !)
 - `<body>`, `<h1>...<h6>`
 - Nouvelles balises HTML 5 : `<main>`, `<article>`, `<section>`, `<nav>` (liens de navigation), `<aside>` (infos supplémentaires), `<header>`, `<footer>`
- Les regroupements
 - `<p>`, `<pre>` (police chasse fixe, préserve espaces et retours à la ligne)

LES BALISES : TOUR D'HORIZON

■ Les regroupements :

- `<ol>` : liste numérotée, `<ul>` : liste avec puces, `<li>` : élément de liste

Exemple :

```
<ol reversed="reversed">  
  <li>Abricots</li>  
  <li>Pommes</li>  
  <li>Bananes</li>  
</ol>
```



3. Abricots
2. Pommes
1. Bananes

- `<dl>`, `<dt>`, `<dd>` : description list (termes et définitions), peut-être utilisé pour un glossaire
- `<figure>`, `<figurecaption>` : pour des figures dans un texte
- `<div>` : conteneur générique, très utile pour appliquer un style à un bout du document (par exemple, `<div class="message">`, puis style CSS sur la classe "message")
- `<hr />` : ligne horizontale, `<blockquote>` : citation d'une autre source, spécifiée avec `source="..."`

LES BALISES : TOUR D'HORIZON

- Les éléments de phrase (« text-level semantics ») :
 - `<a href="URI">` : lien vers une autre page
 - `<a href="#paragraphe1">` : lien vers l'endroit dans la page qui a une balise avec comme paramètre `id="paragraphe1"`
 - `<br />` : retour à la ligne, déjà vu
 - `<em>` : emphasized text, par défaut en italique
 - `<strong>` : important text, par défaut en gras
 - `<cite>` : titre d'un ouvrage cité, `<q>` : citation, `<dfn>` : définition d'un terme, `<abbr>` : abbréviation, `<time>` : date, heure, `<code>` : du code, chasse fixe par défaut, `<var>` : variable, `<samp>` : sample output...

LES BALISES : TOUR D'HORIZON

- Les éléments de phrase (« text-level semantics »)
 - `<sup>` : exposant, `<sub>` : indice
 - `<mark>` : élément mis en valeur (surligné) dans le texte
 - `<span>` : élément générique, utilisé pour donner un style à un bout de texte. Même chose que `<div>`, mais pas un block tag, élément en ligne (inline)
 - `<s>` : texte incorrect (barré par défaut)
 - `<wbr />` : « Word Break Opportunity », pour dire où revenir à la ligne
- Édition :
 - `<del>` : élément effacé
 - `<ins>` : élément inséré

LES BALISES : TOUR D'HORIZON

- Pour inclure des éléments extérieurs (embedded contents) :
 - `<img />` : image, `<map />`, `<area />` (image à zones)
 - `<video>`, `<audio>`
 - `<iframe>` : page dans une page, `<object>` : générique, pour par exemple une application Flash
- Les tableaux :

- Par exemple :

```
<table>  
<tr><td>A</td><td>B</td></tr>  
<tr><td>C</td><td>D</td></tr>  
</table>
```



A	B
C	D

LES BALISES : TOUR D'HORIZON

- Les formulaires (forms) :
 - `<form>`, `<input>` : champ de texte, case à cocher, bouton, etc.
- Les éléments liés à javascript (programmation côté client) :
 - `<script>`, `<canvas>` : élément graphique
- Balises liées à d'autres langages (et les balises induites) :
 - `<svg>` : <http://www.w3.org/TR/SVG/>
 - `<math>` : <http://www.w3.org/TR/MathML/>

MISE EN FORME : AVEC CSS UNIQUEMENT !

- Beaucoup de balises ont une mise en forme par défaut !
 - `<em>` (texte mis en valeur) = en italique, généralement (mais pas assuré !)
 - `<strong>` (texte important) = en gras, généralement
 - `<del>` (texte effacé) = généralement barré
 - `<h1>`, ..., `<hN>` (titres) = texte gras de moins en moins grand
 - `<a href="...">lien</a>` = texte bleu souligné...
- Avant CSS (programmation web des années 90...) mise en forme avec balises !
 - `<i>` = italique, `<b>` = gras, `<table border="1">`
 - **À éviter complètement de nos jours !**
 - **HTML spécifie uniquement la structure, pas le style des pages !**

MISE EN FORME : AVEC CSS UNIQUEMENT !

- Comment mettre en page des pages sans balises de mise en page ?
- On décrit bien la structure en HTML. En découpant le texte, éléments etc. dans des `<p>`, `<section>`, `<cite>`...
- Pour des types d'éléments customisés, on peut utiliser des classes : `<div class="commentaire">`, `<span class="question_tp">`...
- Pour des types d'éléments customisés qu'on n'aura une seule fois dans la page, utiliser des identifiants : `<div id="menu_principal">`...
- Puis, avec du CSS (feuilles de style, « Cascading Style Sheets ») :
 - Spécifier la forme (couleur, police, position dans la page...)
 - De tous les éléments `<section>`, de tous les éléments de la classe "commentaire"...
 - De l'élément ayant pour identifiant "menu_principal"...

APERÇU DE CSS : EXEMPLES

- Citations `<cite>` en italique

```
cite {  
    font-style: italic;  
}
```

- Marges de 10 pixels autour des images :

```
img {  
    margin: 10px;  
}
```

- Tous les éléments de la classe « commentaire » encadrés en pointillés avec un fond gris :

```
.commentaire {  
    border: 1px dashed black;  
    background-color: gray;  
}
```

APERÇU DE CSS : EXEMPLES

- Élément id « menu_principal » fixe sur la page à la position « 200px, 0px » :

```
#menu_principal {  
    position: absolute;  
    top: 200px;  
    left: 0px;  
}
```

- CSS soit directement dans la page (dans des balises <style></style>, ou pour un élément, style="" (pas recommandé))
- Soit (mieux), dans un fichier .css, inclus avec une balise <link />.
- On verra CSS en détail au cours suivant !
- **HTML = structure, CSS = style : Stylé décorrélié de la structure !**
- **Possible de changer style de toutes les pages en changeant seulement le fichier CSS !**

RÉSUMÉ : COMMENT FAIRE UN DOCUMENT HTML ?

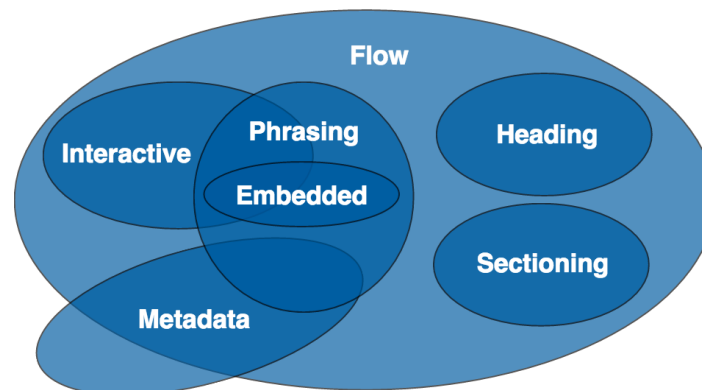
- On commence par la balise `<html>` : définition du document
 - Elle nécessite un `<head>`, un `<body>`
- `<head>` : partie comprenant les en-têtes et métadonnées du document
 - Un `<title>...</title>` obligatoire
 - Balises `<meta>` conseillées (pour les moteurs de recherche : mots-clé, description, auteur...)
 - Balises `<script>` (javascript), `<link>` (inclusion de document tel que feuille de style) optionnelles...
- `<title>` : titre du document
 - Titre de la page (en haut du navigateur, pas dans la page)
 - Contient du texte (non-vide)
- `<body>` : contenu du document
 - **Contient du « Flow Content »**

TYPES DE CONTENU

- **Différents types de contenu : documentation dit quel type de contenu dans une balise**
- **« Metadata Content »** = base link meta noscript script style template title
- **« Flow Content »** = a abbr address area (if descendant of a map) article aside audio b bdi bdo blockquote br button canvas cite code data datalist del dfn div dl em embed fieldset figure footer form h1 h2 h3 h4 h5 h6 header hr i iframe img input ins kbd keygen label main map mark math meter nav noscript object ol output p pre progress q ruby s samp script section select small span strong sub sup svg table template textarea time u ul var video wbr Text
- **« Sectioning Content »** = article aside nav section
- **« Heading Content »** = h1 à h6
- **« Phrasing Content »** = a abbr area (if descendant of a map) audio b bdi bdo br button canvas cite code data datalist del dfn em embed i iframe img input ins kbd keygen label map mark math meter noscript object output progress q ruby s samp script select small span strong sub sup svg template textarea time u var video wbr Text
- **« Embedded Content »** = audio canvas embed iframe img math object svg video

TYPES DE CONTENU

- « **Interactive Content** » = a audio (if controls present) button embed iframe img (if usemap attribute present) input (if type not hidden) keygen label object (if usemap attribute present) select textarea video (if controls present)
- « **Palpable Content** » = Éléments visibles (avec du contenu non vide, non caché)
- « **Transparent Content** » = Le contenu possible est le même que celui pour l'élément parent
- Tous les contenus : <https://www.w3.org/TR/html5/images/content-venn.svg>



IMPORTANT : PRODUIRE DU HTML VALIDE !

- Le HTML est un standard devant être reconnu par tous les navigateurs
- Si quelques erreurs dans la page : navigateurs les ignorent
- Mais pas forcément de la même manière !
- **Important de créer des documents valides !**
- **Dans ce cours : obligatoire !**
- Balises fermées (dans le bon ordre, pas de `<p><a href="...">...</p></a>`), balises imbriquées de manière correcte (pas d'élément bloc dans un élément inline, **bon type de contenu...**), propriétés correctes pour les balises, valeurs des propriétés entre guillemets...
- Pour valider vos pages, validateur du W3C : <https://validator.w3.org/>
- Dans « More options », choisissez « HTML5 (experimental) »

ZOOM SUR QUELQUES BALISES

- Liens (`<a>`)
- Sections et structures (`<header>`, `<section>`, etc.)
- Regroupements (`<p>`, `<li>`, etc.)
- Images (`<img>`)
- Tableaux (`<table>`)
- Date et heure (`<time>`)

BALISE `<A>` : HYPERLIEN

- Pour créer un lien vers un autre document, ou local (position sur la page)
- Faire la différence entre le texte du lien (ce qu'on lit) et la cible du lien (là où on va)
- Cible : attribut « href », pour référencer une page par son URI
 - Lien absolu `<a href="http://i3s.unice.fr/~jplozi/intro_web">Page de l'UE</a>`
 - Lien relatif `<a href="../index.html">Page d'accueil</a>` (cf. `<base>`)
- Lien local : position sur la page
 - Référence : `href="URI#id_balise"` (page courante si URI vide)
 - On aura par exemple `<h1 id="id_balise">` (n'importe quelle balise peut avoir un id)
 - http://www.i3s.unice.fr/~jplozi/intro_web/tp1/#exercice3
- Contenu d'une balise `<a>` : transparent except interactive content
 - Peut contenir mêmes éléments que son parent sauf éventuels éléments interactifs

ATTRIBUTS DE LA BALISE <A>

- « href » nécessaire (si absent, pas de lien)
- « target » : indique où sera ouvert le document
 - _blank : nouvel onglet/fenêtre
 - Autres paramètres : _self, _parent, _top, utile seulement avec des frames. Plus trop à la mode.
- « download » : indique s'il s'agit d'un téléchargement (pas d'ouverture directe). Paramètre booléen, mais attention, valeur doit contenir nom du fichier qui sera enregistré (mettre le même s'il ne change pas)
- « type » : type de contenu MIME (<ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>), généralement pas nécessaire
- « rel » : relation par rapport à la page, par exemple « author » (lien vers la page de l'auteur), « next » (suite), voir : <http://www.w3.org/TR/html5/links.html#linkTypes>
- « hreflang » : langue du document
- Attributs globaux : id, class, style...

STRUCTURATIONS

- `<header>` : notamment pour les menus / barres de navigation
- `<footer>` : notamment pour les pieds de page
- `<nav>` : notamment pour les barres de navigation
- `<article>` : un contenu
- `<section>` : sous-partie d'un article ou regroupement d'articles
- `<aside>` : un encart
- `<root>` : balise unique, pour le contenu principal du document



RELATIONS ENTRE BALISES ET STRUCTURATION

- `<article>`, `<section>`, `<aside>`, `<nav>` : contiennent du Flow Content
 - Pas de `<main>`, direct ou indirect, pour `<aside>` ou `<nav>`
- `<header>` et `<footer>` contiennent du Flow Content
 - Sauf `<header>`, `<footer>` et `<main>`
- Une `<section>` peut avoir plusieurs `<article>` (ou `<nav>`, `<aside>`, etc.)
- Un `<article>` peut avoir plusieurs `<section>` (ou `<nav>`, `<aside>`, etc.)
- Le document (`<body>`) peut avoir des `<header>` et des `<footer>`
- `<article>`, `<section>`, `<aside>`, `<nav>` peuvent avoir des `<header>` et des `<footer>`
 - Il est conseillé de mettre un titre (`<hN>`) à chaque article, dans un `<header>` ou non

LES TITRES

- `<h1>` : titre(s) de plus haut niveau
 - Différent de `<title>`
 - Conseillé d'en mettre un seul (souvent le même que `<title>`, pas toujours lisible)
- Sous-titre : `<h2>`
- Jusqu'à 6 niveaux : `<h1>` à `<h6>`
- Contenu : Phrasing Content (texte annoté)
 - Conseil ergonomique : éviter de mettre des liens dans le titre

LES TITRES

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

REGROUPEMENT : LES LISTES

- `<ul>` (« unordered ») ou `<ol>` (« ordered »)
 - Contiennent uniquement (0 ou +) des `<li>` (ou `<script>`)
 - Attributs :
 - `start (<ol>)` : une valeur de début (un entier)
 - `reversed (<ol>, reversed="reversed")` : pour un décompte
 - `type (<ol>)` : type de numérotation (decimal, lower-alpha, upper-alpha, lower-roman, upper-roman)
- `<li>` (« list item »)
 - Contient du Flow Content
 - Attribut (si enfant de `<ol>`) : « value » (un numéro)

REGROUPEMENT : LES LISTES

```
<ol>  
  <li> Item one </li>  
  <li> Item two </li>  
  <li> Item three </li>  
</ol>
```

```
<ul>  
  <li> Item one </li>  
  <li> Item two </li>  
  <li> Item three </li>  
</ul>
```

1. Item one
2. Item two
3. Item three

- Item one
- Item two
- Item three

REGROUPEMENT : LISTES DE DÉFINITION

- `<dl>` (« definition list »)
- Contient des éléments `<dt>` (terme) et `<dd>` (définition)
- Contient un ou plus éléments `<dt>` ou `<dd>` (pas forcément autant de l'un que de l'autre, un terme peut avoir plusieurs définitions, la liste peut ne contenir que des définitions sans termes... pas très strict)
- `<dt>` contient du Flow Content sauf `<header>`, `<footer>`, du Section Content ou du Heading Content.
- `<dd>` contient uniquement du Flow Content

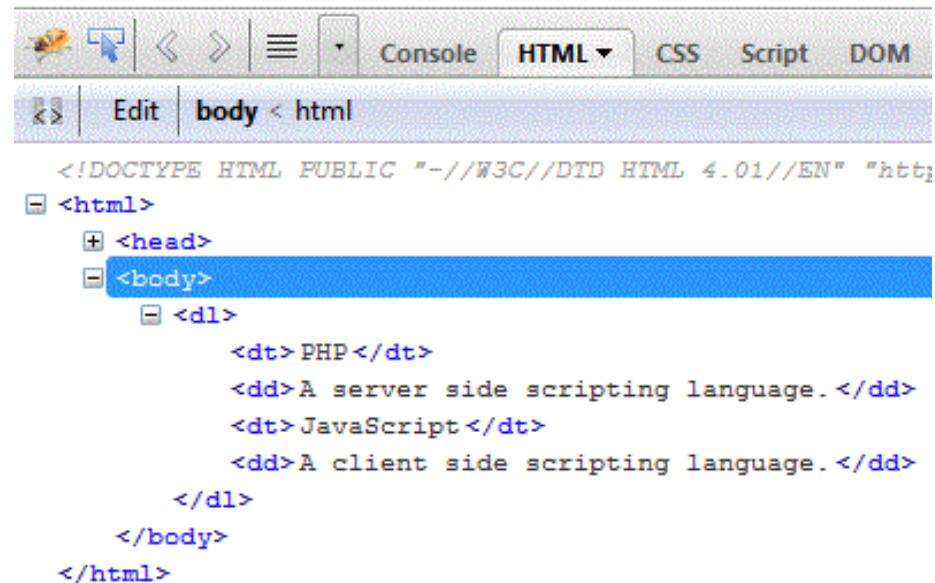
REGROUPEMENT : LISTES DE DÉFINITION

PHP

A server side scripting language.

JavaScript

A client side scripting language.



The screenshot shows a web browser's developer console with the 'HTML' tab selected. The document structure is displayed as a tree view. The root element is <html>, which contains <head> and <body>. The <body> element contains a definition list (<dl>). The list has two items: one for 'PHP' (server side scripting language) and one for 'JavaScript' (client side scripting language). The <body> element is highlighted in blue.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html401/dtd/xhtml1-strict.dtd">
<html>
  <head>
  <body>
    <dl>
      <dt> PHP</dt>
      <dd> A server side scripting language.</dd>
      <dt> JavaScript</dt>
      <dd> A client side scripting language.</dd>
    </dl>
  </body>
</html>
```

REGROUPEMENT : PARAGRAPHES

- `<p>` : « paragraph »
 - Paragraphe de texte
 - Contenu : Phrasing Content
- `<div>` : déjà vu, élément de regroupement pour CSS (ou Javascript)
 - Contient du Flow Content
- `<pre>` (texte pré-formaté)
 - Espaces et retours à la ligne conservés, en police à chasse fixe (Courier...)
 - Pratique pour du code par exemple (pour préserver l'indentation)

REGROUPEMENT : PARAGRAPHES

Imagine That You
Wanted To Display
Some Text In
A Very Specific

Way

This is easy to do the the pre tag!

This is a bird in a tree using nothing but text!

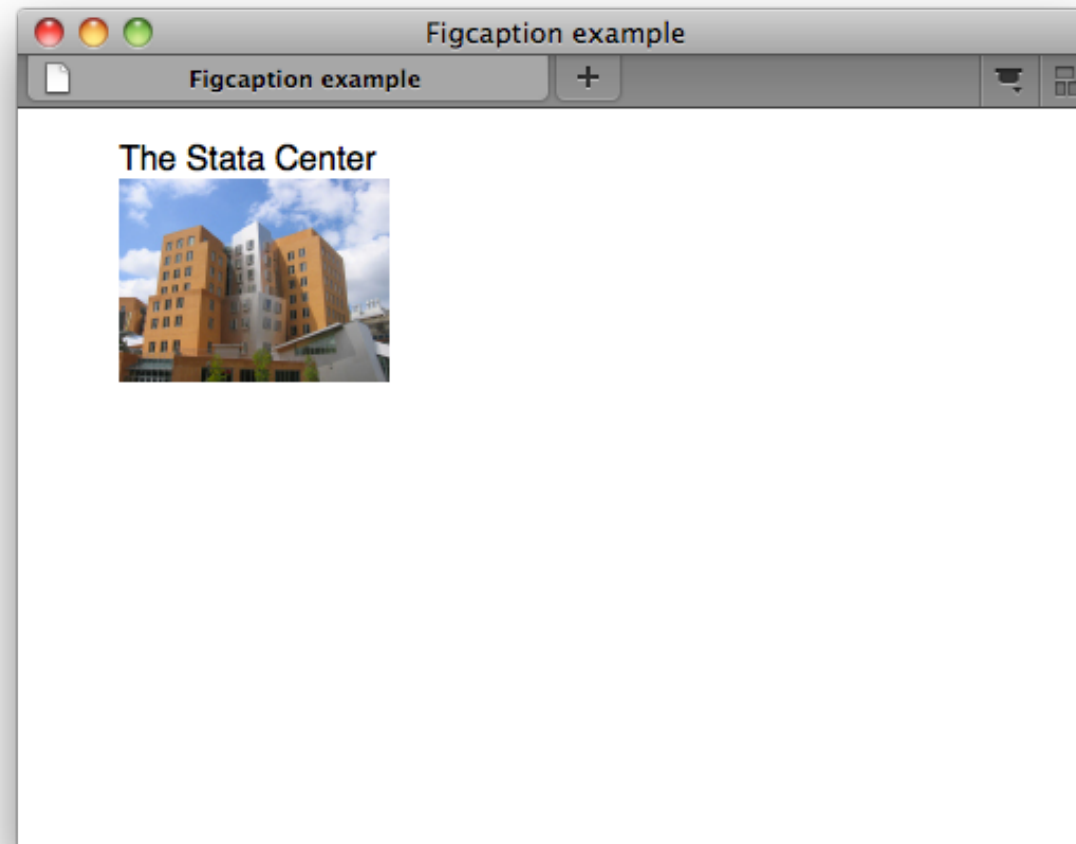
```

      ('>
      /) ) eeeee.
      /e" eeeee ( ) e
      . ee ( ) ee ( ) eeee
      eeeoe eeee ( ) eee
      e ( ) ee\ eee ( ) ee
      e ( ) e | | eeeee'
      ' ee | | eee'
          | |
          | |
          | |
          | |
  
```

REGROUPEMENT : FIGURES

- `<figure>` : pour insérer des figures
 - Pour regrouper illustration et légende
 - Contient : du Flow Content suivi de `<figcaption>` ou `<figcaption>` suivi de Flow Content
- `<figcaption>` : pour définir une légende à la figure
 - Contient : du Flow Content (mais généralement du texte)

REGROUPEMENT : FIGURES



BALISE `<IMG>` : IMAGE

- Balise pour insérer des images
 - Différents formats (png, jpg, gif, svg, etc.) selon les navigateurs
- Balise vide, toutes les infos passées en attribut
- ``
 - « src » : source de l'image (peut être absolue, ou relative, peut-être juste « images/logo_unice.png » si dans le bon répertoire (ou `<base>`)
 - « alt » : description de l'image. En remplacement si impossible de l'afficher, mais aussi souvent lorsqu'on passe la souris dessus.
« The alt attribute on images is a very important accessibility attribute. Authoring useful alt attribute content requires the author to carefully consider the context in which the image appears and the function that image may have in that context. »
<http://www.w3.org/TR/html5/embedded-content-0.html#alt>

BALISE `<IMG>` : L'ATTRIBUT ALT

TAGS

`|<` `< PREV` `RANDOM` `NEXT >` `>|`

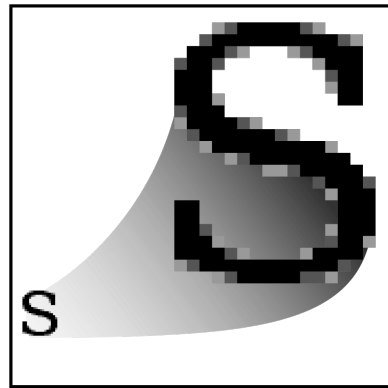
`<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?</SPAN>`

`|<` `< PREV` `RANDOM` `NEXT >` `>|`

`<A>: Like </a>this. `

INSÉRER DU SVG DANS DU HTML

- Scalable Vector Graphics, seulement la balise `<svg>`
- Format qui définit des formes (cercles, rectangles, courbes de béziers) en XML
- <http://www.w3.org/TR/SVG/>
- Avantage : s'agrandit autant que l'on veut sans pixellisation



Raster
.jpeg .gif .png



Vector
.svg

INSÉRER DU SVG DANS DU HTML

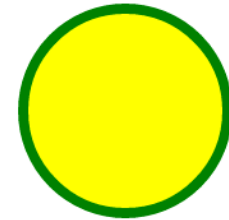
```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

My first SVG



DES LIENS DANS DES IMAGES

- `<map>` : attribut « name » (= « id »)
 - Pour utiliser ce map, une image utilise l'attribut « usemap » avec le nom précédé de #
 - Contient des `<area>`
- `<area>` : une sorte de lien, mais défini par une forme géométrique
 - Pour la partie lien : href (avec download, rel, target, type, hreflang...)
 - Pour la partie image : alt (obligatoire !)
 - Pour la géométrie :
 - « shape » : rect, circle, poly, default (= rect)

DES LIENS DANS DES IMAGES

- `<area>` : une sorte de lien, mais défini par une forme géométrique
 - « coords » :
 - Si « rect », 4 entiers, x_0 , y_0 , x_1 , y_1 où (x_0, y_0) est le coin en haut à gauche de la zone et (x_1, y_1) est son coin en bas à droite
 - Si « circle » : 3 entiers x_0 , y_0 , r où (x_0, y_0) est le centre du cercle et r son rayon
 - Si « poly » : $2n$ entiers pour n points ($n > 2$), en une suite de (x_i, y_i)
 - Origine : coin en haut à gauche de l'image, dimension après affichage (donc déformation)

DES LIENS DANS DES IMAGES

```
<!DOCTYPE html>
<html>
<body>

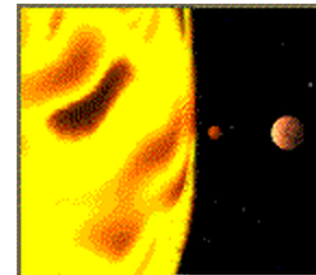
<p>Click on the sun or on one of the planets to watch it closer:</p>



<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>

</body>
</html>
```

Click on the sun or on one of the planets to watch it closer:



BALISE `<TABLE>` : TABLEAUX

- Dans la balise `<table>`
 - `<tr>` : ligne, `<td>` : colonne
 - `<td rowspan="2">` : colonne qui s'étend sur deux lignes
 - `<tr colspan="3">` : ligne qui s'étend sur trois colonnes
 - `<th>` : « table heading, pour la première ligne d'en-tête (à la place de `<td>`)
 - Exemple avec `<th>` :

```
<table>
<tr><th>Month</th><th>Savings</th></tr>
<tr><td>January</td><td>$100</td></tr>
</table>
```

BALISE `<TABLE>` : TABLEAUX

```
<table>
<tr><th>Month</th><th>Savings</th></tr>
<tr><td>January</td><td>$100</td></tr>
<tr><td>February</td><td>$100</td></tr>
<tr><td colspan="2">Sum: $180</td></tr>
</table>
```

Month	Savings
January	\$100
February	\$100
Sum: \$180	

```
<table>
<tr>
<th>Month</th>
<th>Savings</th>
<th>Savings for holiday!</th>
</tr>
<tr><td>January</td><td>$100</td>
    <td rowspan="2">$50</td></tr>
<tr><td>February</td><td>$80</td></tr>
</table>
```

Month	Savings	Savings for holiday!
January	\$100	\$50
February	\$80	

BALISE `<TIME>` : DATE ET HEURE

- Pour les dates et heures : attribut `<time>`
- Une partie lisible : entre `<time>` et `</time>`
- Une partie pour les ordinateurs :
 - Attribut « `datetime` »
 - Utilisé par Google lorsqu'il affiche la date d'un article par exemple
 - Spécification officielle :
<http://www.w3.org/TR/html5/text-level-semantic.html#the-time-element>

BALISE `<TIME>` : FORMAT DE DATETIME

datetime attribute values	Interpretation
<code><time datetime="1905"></code>	The year 1905
<code><time datetime="1905-11"></code>	November 1905
<code><time datetime="11-13"></code>	November 13th (any year)
<code><time datetime="1905-W21"></code>	Week 21 from year 1905
<code><time datetime="1905-11-13 09:00"></code>	November 13th year 1905, time = 9:00
<code><time datetime="1905-11-13T09:00"></code>	Same as previous example, both syntaxes are supported, with and without the "T" between date and time.
<code><time datetime="09:00Z"></code>	9:00 in the morning, GMT
<code><time datetime="09:00-05"></code>	9:00 in the morning, GMT minus 5 hours
<code><time datetime="09:00+05:45"></code>	9:00 in the morning, GMT plus 5 hours 45 minutes, (like in Nepal that has 5:45 of difference with GMT)

Source : Michel Buffa

BONUS : AJOUTER UNE « FAVICON »

- `<link rel="icon" type="image/png" href="logo.png" />` (dans l'en-tête)



AUTRES BALISES

- Pour toute autre balise, voir la documentation officielle !

<http://www.w3.org/TR/html5/>