

CSI 3540 - SÉANCE 02

---

# INTRODUCTION À L'HTML

## OBJECTIVES

- ▶ Chapitre 2 et 3 du manuel
- ▶ Introduction au HTML
- ▶ Création d'une page web
- ▶ Ajout des images,
- ▶ hyperliens,
- ▶ listes,
- ▶ Tableaux
- ▶ Forms / entrées



### Language markup

- \* Pour le traitement, la définition et la présentation du texte
- \* Le langage spécifie la forme, et le style, dans un fichier texte
- \* Les éléments utilisés pour spécifier la forme sont appelés des “tags”.

Abbr	Nom
LaTeX	Lamport Tex
XML	Extensible Markup Language
SGML	Standard Generalized Markup Language
HTML	Hypertext Markup Language
HAML	HTML Abstraction Markup Language
EEx	Embedded Elixir (HTML + Elixir)
Erb	Embedded Ruby (HTML + Ruby)

Exemples des languages markup

```
        .
        .
        .
<h3>
    Prerequisites:
</h3>
<p>CSI2110, CSI2132</p>

<h3>Professeur</h3>
<table>
    <tr>
        <th>Professor</th>
        <td>Andrew Forward</td>
        <t
</1><table      Tag
<t>tbody      Tag
    <td      Tag
    <template      Tag
    <textarea      Tag
</1><tfoot      Tag
<t>th      Tag
    <thead      Tag
        <a href="https://uottawa.brightspace.com/d2l/le/cont
    </td>
</tr>
<tr>
```

## Éditeurs de texte

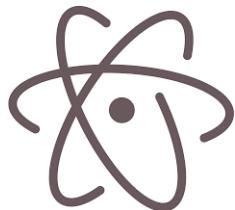
Vous n'avez pas besoin d'un IDE complet,

Ce qui aide vraiment

Éclaircissement de syntaxe

Suggestions de code

## EXAMPLE TEXT EDITORS



Exemples d'éditeurs de texte

Des éditeurs tels que sublime text, atom, notepad++ et VIM offrent une excellente expérience d'édition pour l'écriture et la lecture pour les technologies du web.

```
01_boilerplate.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Welcome</title>
6   </head>
7   <body>
8     <p>Welcome to HTML5!</p>
9   </body>
10 </html>
```

Déclaration "Document Type" requise

Méta-données sur le document

Les éléments vides qui n'ont pas un tag de fin

Double, pas de guillemets simples

Le contenu du document

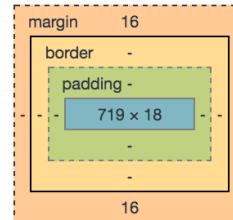
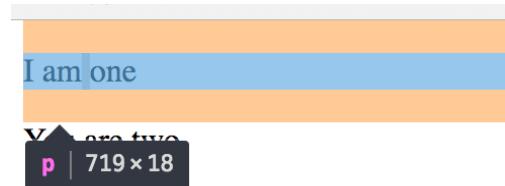
Comme pour XML, les tags de début et de fin sont habituellement requises

Les tags minuscules

Code HTML gabarit

## <P>PARAGRAPH</P>

```
, need  
<body>  
    <p>I am one</p>  
    <p>You are two</p>  
</body>  
</html>
```



## VALIDATION DE HTML

<http://validator.w3.org/#validate-by-upload>

Validate by URI    Validate by File Upload    Validate by Direct Input

Validate by File Upload

Upload a document for validation:

File:

More Options

Check

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 2 milliseconds.

<H1>HEADINGS</H1> ... <H6></H6>

```
,head.  
<body>  
  <h1>Heading 1</h1>  
  <h2>Heading 2</h2>  
  <h3>Heading 3</h3>  
  <h4>Heading 4</h4>  
  <h5>Heading 5</h5>  
  <h6>Heading 6</h6>  
  <h7>Heading 7</h7> <!-- Not a valid tag -->  
</body>
```



**Heading 1**

**Heading 2**

**Heading 3**

**Heading 4**

**Heading 5**

**Heading 6**

**Heading 7**

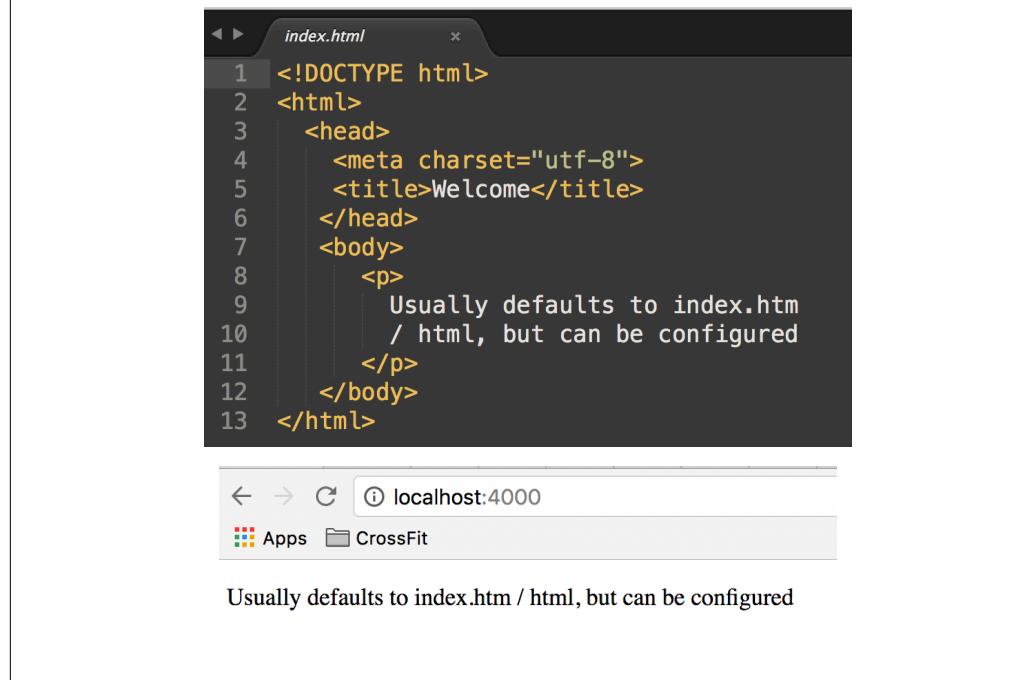
## <A>HYPERLINKS</A> (OU ANCRES)

```
<body>
  <a href="http://a4word.com">a4word</a><br>
  <a href="https://google.ca">Google Search</a><br>
  <a href="mailto:aforward@gmail.com">Email me</a><br>
  <a href="01_boilerplate.html">Boilerplate</a><br>
  <a href="/01_boilerplate.html">Boilerplate (Absolute)</a> <!-- invalid -->
```



Hypertext reference

[a4word](#)  
[Google Search](#)  
[Email me](#)  
[Boilerplate](#)  
[Boilerplate \(Absolute\)](#)



The screenshot shows a web browser window with the title "index.html". The code editor at the top displays the following HTML content:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Welcome</title>
6   </head>
7   <body>
8     <p>
9       Usually defaults to index.htm
10      / html, but can be configured
11    </p>
12  </body>
13 </html>
```

Below the code editor, the browser's address bar shows "localhost:4000". The main content area of the browser displays the text "Usually defaults to index.htm / html, but can be configured".

Document par défaut (index.htm)

## <IMG SRC="LOGO.PNG" /> IMAGES

The diagram shows a snippet of HTML code with annotations explaining different styling methods:

- "source" (src) pas href: Points to the first image tag.
- La tag alt est utilisée pour l'accessibilité: Points to the alt attribute of the fourth image tag.
- Style nommé: Points to the .skinny class definition in the CSS block.
- Style en ligne: Points to the style="width: 200px;" attribute in the third image tag.

Below the code, four images are displayed:

- A landscape image of Half Dome.
- An identical landscape image of Half Dome.
- A photograph of Michael Scott from the TV show "The Office".
- A small image of a cat with the caption "A cute cat".

Accessibilité "A cute cat"

Un navigateur peut ne pas être capable de rendre l'image

La tag alt peut également être lue à l'utilisateur

ARIA (Accessible Rich Internet Applications Suite) comme alternative au HTML pur

## TEXT + STYLE > IMAGES

```
<body>
  <a href="http://a4word.com" class="button">My Fancy Button</a>
</body>
```

My Fancy Button

!Tab;	&NewLine;	!	"	#	\$	%	&	'
(	)	*	+	,	.	/	:	;
<	=	>	?	@	[	\	]	^
—	‘	{		}	&nbsp;	¡	¢	£
é	à	ç	à	ç		í	ó	ö

## Caractères spéciaux

\* <https://ascii.cl/htmlcodes.htm>

\* [https://www.w3schools.com/html/html\\_symbols.asp](https://www.w3schools.com/html/html_symbols.asp)

```
<style>
  .divider {
    border-top: 1px solid black;
    padding-top: 10px;
  }
</style>
</head>
<body>
  <p>Above</p>
  <hr>
  <p>Below</p>
  <br />
  <p>Better Above</p>
  <p class="divider">Better Below</p>
</body>
```

Above

---

Below

---

Better Above

---

Better Below

<hr>Horizontal Rule

```
<ul>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Carrots</li>
</ul>

<ol>
  <li>Study</li>
  <li>Write Exam</li>
  <li>Graduate</li>
</ol>
<br/>
```

- Apples
- Bananas
- Carrots

1. Study
2. Write Exam
3. Graduate

<li>List Items</li>

```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Grade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Andrew</td>
      <td>92%</td>
    </tr>
    <tr>
      <td>James</td>
      <td>78%</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Average</th>
      <th>85%</th>
    </tr>
  </tfoot>
</table>
```

Information d'entête

Le corps de la table

Rangée (table-row)

Cellule (table-data)

Information sur le pied  
de la table

Entête (table-header)

### Name Grade

Andrew 92%

James 78%

Average 85%

<table></table>

```
<table>
  <thead>
    <tr>
      <th colspan="2">Name</th>
      <th>&nbsp;</th>
    </tr>
    <tr>
      <th>First</th>
      <th>Last</th>
      <th>Grade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Andrew</td>
      <td rowspan="2">Forward</td>
      <td>92%</td>
    </tr>
    <tr>
      <td>78%</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th colspan="2">Average</th>
      <th>85%</th>
    </tr>
  </tfoot>
</table>
```

```
<style>
  table { border-collapse: collapse; }
  table td, table th { border: 1px solid black; padding: 10px; }
  table thead tr { background-color: #bdc3c7; }
  table tfoot tr { background-color: #ecf0f1; }
</style>
```

Name		
First	Last	Grade
Andrew	Forward	92%
		78%
Average		85%

## Rowspan / colspan

Où envoyer les données,  
par défaut à lui-même

le "id" pour le client (JavaScript)

"get" et "post" seulement

"Submit" le form

N'utilise pas!

le "name" pour le sever

```
<body>
  <form method="get" action="process.php">
    <input type="text" name="first_name" id="firstName" />
    <input type="submit" value="Go" />
    <input type="reset" value="Clear" /> <!-- rarely used -->
  </form>
</body>
```

localhost:4000/11\_form.php?first\_name=Andrew+Forward

Array  
(  
  [first\_name] => Andrew Forward  
)

&lt;form&gt;

```
<body>
<?php
if (!empty($_POST)) {
    ?>
    <h1>Hello <?php echo $_POST["first_name"] ?></h1>
    <a href="11b_form.php">Try Again</a>
    ?>
} else {
    ?>
    <form method="post">
        <input type="text" name="first_name" id="firstName" />
        <input type="submit" value="Go" />
    </form>
    ?>
}
?>
```

## REQUEST Variables

Habituellement encapsulé par le framework Web que vous utilisez.

Precedence est POST en premier, GET second

## <INPUT TYPE="X">

```
<input type="text" name="my_name" placeholder="Enter name">
<input type="email" name="my_email" placeholder="Enter email">
<input type="password" name="my_password" placeholder="Password">
```

Un bel attribut d'utilisabilité lorsque le champ est vide

Enter name

Masque l'entrée sur l'écran

aforward@gmail.com

.....

<input type="x">

## <INPUT TYPE="HIDDEN">

```
<input type = "hidden" name = "_ method" value = "put">
```

Un truc par les frameworks web  
pour supporter tous les verbes HTTP

type="hidden"

Peut stocker des données sans être affiché

Envoyé en clair au client

Pas de sécurité, plus de commodité

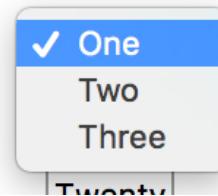
## <SELECT>

```
<select name="my_select">
    <option value="1">One</option>
    <option value="2">Two</option>
    <option>Three</option>
</select>
```

Seule la valeur est envoyée au serveur

Le texte entre les tags est appelé le nom

Si aucune valeur n'est fournie, alors la "valeur" est le nom de l'option



## Select

## <SELECT MULTIPLE="MULTIPLE">

```
<select multiple="multiple" name="my_many_select[]>  
    <option>Ten</option>  
    <option>Twenty</option>  
    <option>Thirty</option>  
</select>
```

Notez les parenthèse carré  
dans le nom



Ten  
Twenty  
Thirty

Select multiple

## <LABEL>

```
<label for="firstName">Enter your first name</label>
```

```
<input type="text" name="first_name" id="firstName" placeholder="Enter your first name" />
```

Associer les "label" aux entrées et améliorer l'usabilité

Les labels augmentent la surface à cliquer sur les buttons radio et les checkboxes

Your name

Option 1

Option 2

Check 1

Check 2

Label

```
<div class="block" id="myA" >A  
  <a href="#myB">to B</a>  
</div>  
  
<div class="block" id="myB" >B  
  <a href="#myA">to A</a>  
</div>
```

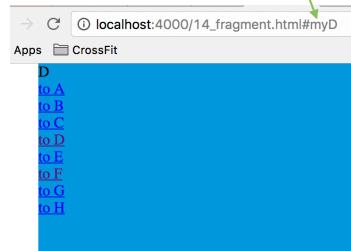
Les fragments ne sont pas envoyés au serveur

Les ancrées peuvent également être utilisées pour relier une page

Appelé un fragment

Les fragments permettent un lien "deep" entre les pages ou la même page

Attention, les fragments sont fortement utilisés dans les frameworks clients\



## #Fragments

## <META>

```
<head>
  <meta charset="utf-8">
  <title>Examples of relevant metatags</title>
  <meta name="description" content="An introduction to meta tags">
  <meta name="author" content="Andrew Forward">
  <!-- obsolete for SEO -->
  <meta name="keywords" content="HTML,How-To,MetaTags">
  <link rel="author" href="[YOUR PERSONAL PAGE]">
</head>
<body>
  <meta property="og:title" content="describe the block of text"/>
  <meta property="og:type" content="article"/>
  <p>Welcome to Meta</p>
</body>
</html>
```

## <Meta>

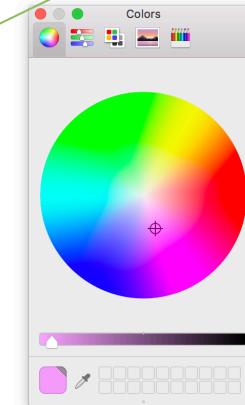
- \* Fournit des conseils utiles aux machines de recherche sur la nature de votre page
- \* Search Engine Optimization (SEO) pourrait être un cours entier
- \* Les règles sont en constante évolution car une fois que vous savez comment vous êtes évalué certains vont jeu du système
  - \* Exemple, les mots clés ne sont plus pertinents
  - \* La taille de page et la vitesse de chargement sont maintenant pertinentes
- \* Avant Google, les moteurs de recherche n'étaient rien de plus que des répertoires Web
  - \* C'est là que Yahoo a brillé (avec Altavista), mais sont maintenant disparus ou hors de propos

## COLOR

```
<input type="color" name="my_colour" id="myColour"  
value="#f39efb">
```

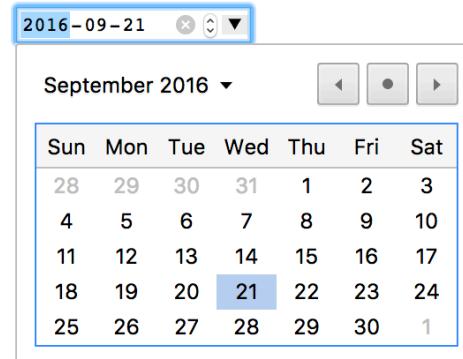
Code hexadécimal  
comme entrée valide

Ce n'est pas une convention de dénomination  
réaliste, mais utile pour montrer la différence  
entre l'accès côté client et côté serveur



## DATE

```
<input type="date" name="my_date" id="myDate"  
value="2016-09-21">
```



## DATETIME

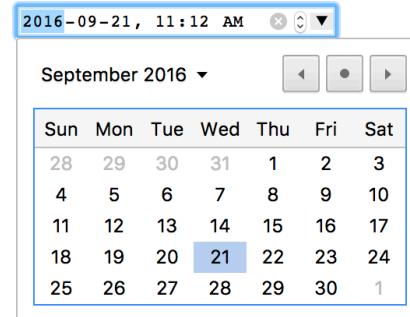
```
<input type="datetime"  
name="my_date_time" id="myDateTime"  
value="2016-09-21T10:11">
```

2016-09-21T10:11

- Pas tous les rendus d'une manière fonctionnelle
- Décisions de conception concernant les navigateurs pris en charge
- Et surtout, essayer de dégrader doucement

## DATETIME-LOCAL

```
<input type="datetime-local" name="my_date_time_local"  
id="myDateTimeLocal" value="2016-09-21T11:12">
```



## TIME

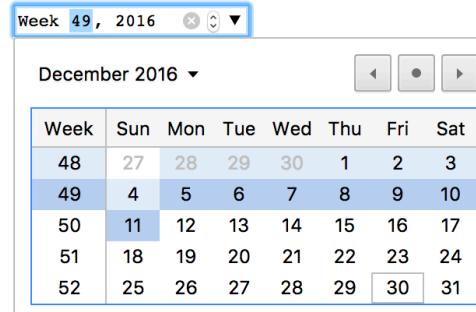
```
<input type="time" name="my_time" id="myTime"  
value="10:11:12">
```



10 : 11 : 12 AM

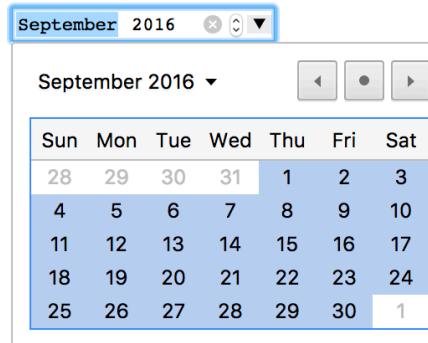
## WEEK

```
<input type="week" name="my_week" id="myWeek" >
```



## MONTH

```
<input type="month" name="my_month" id="myMonth"  
value="2016-09">
```



## NUMBER

```
<input type="number" name="my_number"  
id="myNumber" min="40" max="50" step="2" value="48">
```

46 

## RANGE

```
<input type="range" name="my_range" id="myRange"  
min="70" max="80" step="2" value="78">
```



## SEARCH

```
<input type="search" name="my_search" id="mySearch"  
value="ponies">
```

## TEL

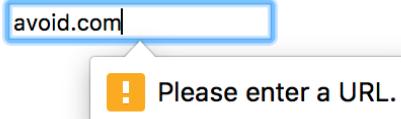
```
<input type="tel" name="my_tel" id="myTel" value="(613)  
555-5555">
```

(613) 555-5555

## URL

```
<input type="url" name="my_url" id="myUrl" value="http://  
a4word.com">
```

http://a4word.com



## EMAIL

- ▶ Éviter SVP
- ▶ <https://www.w3.org/Protocols/rfc822/>
- ▶ bob."@".[smith@mywebsite.com](mailto:smith@mywebsite.com)

.smith@mywebsite.com



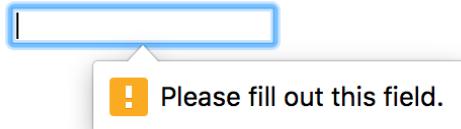
A part followed by '@' should not contain the symbol "'".

## AUTOFOCUS

```
<input type="text" name="my_text" id="myText" autofocus>
```

## REQUIRED

```
<input type="text" name="my_text" id="myText" required>
```



## PATTERN

```
<input type="text" name="my_text" id="myText"  
pattern="[a-zA-Z]{4}" placeholder="4 letters">
```



## VALIDATION

- ▶ Auto-validation dans le client
- ▶ Et alors! Toujours besoin de validation côté serveur comme le client peut mentir, voler et tricher

## FORMATS DE VALIDATION

Type	Format	Type	Format
color	Hex code	month	yyyy-mm
date	yyyy-mm-dd	number	\d*
datetime*	yyyy-mm-ddThh:mm:ssZ	email	good luck!
datetime-local	yyyy-mm-ddThh:mm	url	nice try!
time	hh:mm	week	yyyy-Wnn

## AUTOCOMPLETE

```
<form method="post" autocomplete="on">  
    <input type="text" name="my_text" id="myText" placeholder="people">  
    <input type="text" name="your_text" id="yourText" placeholder="shapes">  
    <button type="submit">Go</button>  
</form>
```



## DATALIST

Autocomplete peut être

réglé sur des éléments individuels

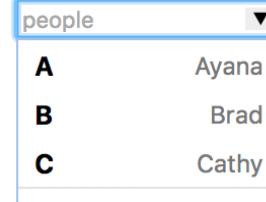
```
<form method="post" >
    <input type="text" name="my_text" id="myText"
        placeholder="people" list="myPeople"
        autocomplete="off">

    <datalist id="myPeople">
        <option value="A">Ayana</option>
        <option value="B">Brad</option>
        <option value="C">Cathy</option>
    </datalist>

    <input type="text" name="your_text"
        placeholder="random thoughts"
        autocomplete="on" >

    <button type="submit">Go</button>
</form>
```

Le nom est  
supplémentaire



La valeur est placé dans l'entré et  
envoyée au serveur

## HEADER, SECTION, FOOTER

- ▶ Ajout de plus de sémantique au document

Ancienne façon de spécifier les sections

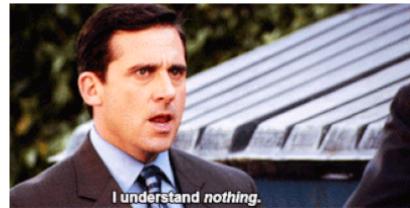
```
<div class="header">The Top</div>
<div class="section">Section 1</div>
<div class="section">Section 2</div>
<div class="footer">The Bottom</div>
```

```
<body>
  <header>The Top</header>
  <section>Section 1</section>
  <section>Section 2</section>
  <footer>The Bottom</footer>
</body>
```

Nouvelle voie sémantique

## FIGURE, FIGCAPTION

```
body>
  <figure>
    
    <figcaption>I understand nothing</figcaption>
  </figure>
</body>
```



I understand nothing

## MARK

```
<p>I <mark>understand</mark> nothing</p>
```

I understand nothing

## DETAILS, SUMMARY

```
<body>
  <details>
    <summary>Lorem Ipsum is simply dummy text</summary>
    <p>
      Lorem Ipsum is simply dummy text of the printing an
      been the industry's standard dummy text ever since
      a galley of type and scrambled it to make a type sp
      five centuries, but also the leap into electronic t
      unchanged. It was popularised in the 1960s with the
      Lorem Ipsum passages, and more recently with desko
      PageMaker including versions of Lorem Ipsum.
    </p>
  </details>
</body>
```

### ▼ Lorem Ipsum is simply dummy text

Lorem Ipsum is simply dummy text of the printin  
to make a type specimen book. It has survived no  
containing Lorem Ipsum passages, and more rece

## ARTICLE

```
<article>
  <h1>Lorem Ipsum is simply dummy text</h1>
  <p>
    Lorem Ipsum is simply dummy text of the p
    been the industry's standard dummy text e
    a galley of type and scrambled it to make
    five centuries, but also the leap into el
    unchanged. It was popularised in the 1960
    Lorem Ipsum passages, and more recently w
    PageMaker including versions of Lorem Ips
  </p>
</article>
```

### **Lorem Ipsum is simply dummy text**

Pas de changement de l'UX,  
plus pour améliorer la  
sémantique de la page

→ Lorem Ipsum is simply dummy text of the printing and typesetting indust  
dummy text ever since the 1500s, when an unknown printer took a galley  
book. It has survived not only five centuries, but also the leap into electro  
It was popularised in the 1960s with the release of Letraset sheets contain  
with desktop publishing software like Aldus PageMaker including versio

## ASIDE

```
<p>
    "There is no one who loves pain itself,
    who seeks after it and wants to have it,
    simply because it is pain..."
</p>
<aside>For more great quotes checkout myquotes.com</aside>
</body>
```

Le "aside" est lié, mais pas directement  
au contenu ci-dessus

Pas de changement de l'UX,  
plus pour améliorer la sémantique de la page

"There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain..."  
For more great quotes checkout myquotes.com

## METER

```
<body>
  <div id="myStatus"></div>
  <meter min="0" max="20" value="0" id="myMeter">
</body>
<script>
  var m = document.getElementById("myMeter");
  var s = document.getElementById("myStatus");
  var tick = function() {
    if (m.value < m.max) {
      m.value = m.value + 1; ←
      s.innerHTML = "Done " + m.value + " out of " + m.max; ←
      setTimeout(tick, 100); ←
    } else {
      s.innerHTML = "Finished!";
    }
  }
  tick(); ←
</script>
```

Charger les éléments  
HTML DOM

Augmente la valeur  
Chaque 100ms

Démarrer le compteur

Done 19 out of 20



Finished!



## WBR

```
<span>How do you say  
Supercalifragilistic<wbr>expialidocious?</span>
```

Dites au navigateur un bon  
endroit pour diviser un long mot  
(word break)

How do you say Supercalifragilistic  
expialidocious?

Please upgrade your browser to use Basecamp.

We built Basecamp using the latest techniques and technologies. This makes Basecamp faster and easier to use. Unfortunately, your browser doesn't support those technologies.

Download one of these great browsers and you'll be on your way:

Internet Explorer Version 9+ Mozilla Firefox Version 4+ Google Chrome Version 7+ Apple Safari Version 4+

Already upgraded but still having problems? Check out our troubleshooting guide.

Improve Your Experience

You're using a web browser we don't support. Try one of these options to have a better experience on Facebook.

Internet Explorer Mozilla Firefox Opera

This website needs Internet Explorer

This website uses technology that will work best in Internet Explorer.

Open with Internet Explorer

Keep going in Microsoft Edge

**Internet Explorer 6, you cannot be serious?**

We don't support old technologies and neither should you!

Please upgrade to the latest version of Internet Explorer or use a different browser to visit our site.

Il n'est pas acceptable de forcer vos utilisateurs à mettre à jour leur navigateur ; peut-être qu'ils ne peuvent pas, ou ne savent pas comment. Offrir un service dégradé mais toujours utilisable, c'est sans doute mieux.

## REFERENCES

- ▶ <http://www.creativebloq.com/netmag/10-great-text-editors-web-designers-71412411>
- ▶ <http://www.clarissapeterson.com/2012/11/html5-accessibility/>
- ▶ <https://css-tricks.com/examples/ButtonMaker/#>
- ▶ <https://dev.w3.org/html5/html-author/charref>
- ▶ <http://www.iacquire.com/blog/18-meta-tags-every-webpage-should-have-in-2013>
- ▶ <https://developers.facebook.com/docs/sharing/opengraph>

## RÉFÉRENCES

- ▶ <https://elliot.land/post/its-impossible-to-validate-an-email-address>
- ▶ <https://mathiasbynens.be/demo/url-regex>
- ▶ <https://code.tutsplus.com/articles/10-rare-html-tags-you-really-should-know--net-3908>