

# HTML

felipe@ift3225 Hiver 2020

# Histoire

- 1991: HTML Tags — Tim Berners Lee
- 1995: HTML 2.0
- 1997: HTML 3.2 W3C
- 1997: HTML 4.0 W3C
- 1999: HTML 4.01 W3C Recommendation
- 2000: XHTML 1.0 W3C Recommendation
- 2008: HTML 5 Draft
- 2014: HTML 5 W3C Recommendation
- 2017: HTML 5.2 W3C Recommendation



# Éléments de syntaxe

document  
XML

doctype

## En XHTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">
<html xml:lang="en" lang="en" xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>My first Web page.</p>
  </body>
</html>
```

namespace

un peu lourd ...

# Avec validation

```
<?xml version = "1.0" encoding = "utf-8" ?>
```

```
<!-- file: helloworld2.html -->
```

commentaire

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>Bonjour le monde</title>
```

```
</head>
```

```
<body>
```

```
<p> Bonjour le monde. </p>
```

- balises fermantes
- encodage caractères
- tags en minuscule
- attributs entre ""

```
<p style="text-align: center">
```

```
<a href="http://validator.w3.org/check/referer" title="Valid XHTML 1.0">
```

```

```

```
</a>
```

```
</p>
```

```
</body>
```

```
</html>
```

# HTML 5

```
<!DOCTYPE html>
```

Indiquer la langue de la page

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

```
<head>
```

```
  <title>Bonjour le Monde</title>
```

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

```
</head>
```

```
<body>
```

Bonjour le monde !

```
<div>
```

```
  <a href="http://validator.w3.org/check?uri=referer">
```

```
    </a>
```

```
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
```

```
    </a>
```

```
</div>
```

```
</body>
```

```
</html>
```

attribut alt dans les images

# Simplifications de HTML5

- Les guillemets peuvent être simples, double ou absent

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

- Balises auto-fermantes

```
  

```

préférable

- La casse n'est pas importante

```
<iMg sRC="mon_image" Alt="une image" >
```

éviter

- Les balises `html`, `head` et `body` sont optionnelles

utilisez les !

```
<!doctype html>  
<meta charset="utf-8">  
<title>Bonjour</title>  
<p>bla</p>
```

- Pas nécessaire de spécifier une valeur pour un attribut booléen

```
<input type="text" autofocus />  
<video autoplay loop controls />
```

# Éléments HTML

Une centaine de **tags** environ



HTML-5-tutorial	W3C	W3schools	HTML-5-tutorial	W3C	W3schools
• <!DOCTYPE>	DOCTYPE	DOCTYPE	• <kbd>	kbd	kbd
• <a>	a	a	• <keygen>	keygen	keygen
• <abbr>	abbr	abbr	• <label>	label	label
• <acronym> <sup>2</sup>			• <legend>	legend	legend
• <address>	address	address	• <li>	li	li
• <applet> <sup>5</sup>			• <link>	link	link
• <area>	area	area	• <map>	map	map
• <article>	article	article	• <mark>	mark	mark
• <aside>	aside	aside	• <menu>	menu	menu
• <audio>	audio	audio	• <meta>	meta	meta
• <b>	b	b	• <meter>	meter	meter
• <base>	base	base	• <nav>	nav	nav
• <basefont> <sup>4</sup>			• <noframes>		
• <bdi>	bdi	bdi	• <noscript>	noscript	noscript
• <bdo>	bdo	bdo	• <object>	object	object
• <big> <sup>4</sup>			• <ol>	ol	ol
• <blockquote>	blockquote	blockquote	• <optgroup>	optgroup	optgroup
• <body>	body	body	• <option>	option	option
•  	br	br	• <output>	output	output
• <button>	button	button	• <p>	p	p
• <canvas>	canvas	canvas	• <param>	param	param
• <caption>	caption	caption	• <pre>	pre	pre
• <center> <sup>4</sup>			• <progress>	progress	progress
• <cite>	cite	cite	• <q>	q	q
• <code>	code	code	• <rp>	rp	rp
• <col>	col	col	• <rt>	rt	rt
• <colgroup>	colgroup	colgroup	• <ruby>	ruby	ruby
• <command>	command	command	• <s>	s	s
• <datalist>	datalist	datalist	• <samp>	samp	samp
• <dd>	dd	dd	• <script>	script	script
• <del>	del	del	• <section>	section	section
• <details>	details	details	• <select>	select	select
• <dfn>	dfn	dfn	• <small>	small	small
• <dir>		dir	• <source>	source	source
• <div>	div	div	• <span>	span	span
• <dl>	dl	dl	• <strike> <sup>4</sup>		
• <dt>	dt	dt	• <strong>	strong	strong
• <em>	em	em	• <style>	style	style
• <embed>	embed	embed	• <sub>	sub	sub
• <fieldset>	fieldset	fieldset	• <summary>	summary	summary
• <figcaption>	figcaption	figcaption	• <sup>	sup	sup
• <figure>	figure	figure	• <table>	table	table
• <font>			• <tbody>	tbody	tbody
• <footer>	footer	footer	• <td>	td	td
• <form>	form	form	• <textarea>	textarea	textarea
• <frame>			• <tfoot>	tfoot	tfoot
• <frameset>			• <th>	th	th
• <h1> - <h6>	h1 - h6	h1 - h6	• <thead>	thead	thead
• <head>	head	head	• <time>	time	time
• <header>	header	header	• <title>	title	title
• <hgroup>	hgroup	hgroup	• <tr>	tr	tr
• <hr>	hr	hr	• <track>	track	track
• <html>	html	html	• <tt>		
• <i>	i	i	• <u>	u	u
• <iframe>	iframe	iframe	• <ul>	ul	ul
• <img>	img	img	• <var>	var	var
• <input>	input	input	• <video> <sup>5</sup>	video	video
• <ins>	ins	ins	• <wbr>	wbr	wbr

- Elements in grey are in previous versions of HTML, but are not supported in HTML5.
- Elements in red are new to HTML5 and have broad browser support.
- Elements in pink are new to HTML5, but don't yet have broad browser support.



# Inline versus bloc

- Les éléments **inline** s'affichent dans le flux sans rupture (pas de retour chariot) à contrario des éléments **bloc**
- On peut changer la propriété visuelle des blocs mais par leur modèle de contenu
  - *display:inline* et *display:block*
- Règle: les éléments en ligne ne peuvent contenir que des éléments en ligne

The following elements are inline by default (although block and inline elements are no longer defined in HTML 5, use content categories instead):

<code>&lt;a&gt;</code>	<code>&lt;embed&gt;</code>	<code>&lt;samp&gt;</code>
<code>&lt;abbr&gt;</code>	<code>&lt;i&gt;</code>	<code>&lt;script&gt;</code>
<code>&lt;acronym&gt;</code>	<code>&lt;iframe&gt;</code>	<code>&lt;select&gt;</code>
<code>&lt;audio&gt;</code> (if it has visible controls)	<code>&lt;img&gt;</code>	<code>&lt;slot&gt;</code>
<code>&lt;b&gt;</code>	<code>&lt;input&gt;</code>	<code>&lt;small&gt;</code>
<code>&lt;bdi&gt;</code>	<code>&lt;ins&gt;</code>	<code>&lt;span&gt;</code>
<code>&lt;bdo&gt;</code>	<code>&lt;kbd&gt;</code>	<code>&lt;strong&gt;</code>
<code>&lt;big&gt;</code>	<code>&lt;label&gt;</code>	<code>&lt;sub&gt;</code>
<code>&lt;br&gt;</code>	<code>&lt;map&gt;</code>	<code>&lt;sup&gt;</code>
<code>&lt;button&gt;</code>	<code>&lt;mark&gt;</code>	<code>&lt;svg&gt;</code>
<code>&lt;canvas&gt;</code>	<code>&lt;meter&gt;</code>	<code>&lt;template&gt;</code>
<code>&lt;cite&gt;</code>	<code>&lt;noscript&gt;</code>	<code>&lt;textarea&gt;</code>
<code>&lt;code&gt;</code>	<code>&lt;object&gt;</code>	<code>&lt;tina&gt;</code>
<code>&lt;data&gt;</code>	<code>&lt;output&gt;</code>	<code>&lt;u&gt;</code>
<code>&lt;datalist&gt;</code>	<code>&lt;picture&gt;</code>	<code>&lt;tt&gt;</code>
<code>&lt;del&gt;</code>	<code>&lt;progress&gt;</code>	<code>&lt;var&gt;</code>
<code>&lt;dfn&gt;</code>	<code>&lt;q&gt;</code>	<code>&lt;video&gt;</code>
<code>&lt;em&gt;</code>	<code>&lt;ruby&gt;</code>	<code>&lt;subr&gt;</code>
	<code>&lt;s&gt;</code>	

reste encore vrai (généralement) en HTML5

- Cette distinction n'est pas maintenue en HTML5 au profit des catégories de contenu

# Attributs noyaux

- **class** ajouter une propriété afin par la suite de *scripter* les éléments ayant cette propriété

```
<h2 class="city">London</h2>  
<h2 class="city asian-city">Tokyo</h2>
```

- **id** un identificateur unique (dans la page)

```
<h1 id="myHeader">My Cities</h1>
```

- **style** pour définir des propriétés visuelles de l'élément

```
<body style="background-color:powderblue;">  
<h1>This is a heading</h1>  
</body>
```

- **name** réservé à quelques éléments

```
<input type="text" name="fullname">
```

<button>  
<fieldset>  
<form>  
<iframe>  
<input>  
<map>  
<meta>  
<object>  
<output>  
<param>  
<select>  
<textarea>

Plus: <https://www.w3.org/TR/2010/WD-html-markup-20100624/common-attributes.html>

# Retraits/changements

## Retraits

- Alternatives en CSS: `basefont`, `big`, `center`, `font`, `strike`, `tt`, `u`
- Nuisent à l'accessibilité: `frame`, `frameset`, `noframes`
- Créent de la confusion: `acronym`, `applet`, `isindex`, `dir`
- Attributs retirés (CSS): `border` sur les balises `table` et `object`
- Et `bcp` plus ...

## Changements de sens

- `<b>` indique qu'une partie de texte ressort visuellement
- `<strong>` passe de forte *emphase* à forte *importance*
  - possible de les imbriquer pour renforcer l'importance
  - les lecteurs de texte ne changent pas de ton (contrairement à `<em>`)
- `<i>` pour les mots qui ne sont pas les vôtres
  - Mots étrangers, techniques, ou les citations dont on veut changer le ton

Il faut `<i>piper</i>` la sortie dans la commande.

Et il ose me dire `<i>HTML5 c'est simple</i>` !

- `<small>` pour indiquer les « petites lignes » (ex: conditions légales)
- `<cite>` ne devrait pas baliser un auteur mais les livres, essais, chansons, poèmes, etc.

Chilly Gonzales a composé `<cite>Solo Piano</cite>`

# Nouveautés

## - Balises structurantes

- **header**: un ensemble pouvant inclure un titre (h1-h6), une introduction, des logos, etc.
- **footer**: infos sur la section à laquelle il appartient (ex: auteur)
- **nav**: liens vers d'autres pages ou d'autres parties de la page
- **aside**: contenu non essentiel à une entité (ex: publicité)
- **article**: portion de page qui se suffit sémantiquement
- **section**: regroupement thématique de contenu contenant habituellement un titre

## - Heure et date

`<time datetime="2020-01-14"> 14 Janvier 2020</time>`

article publié `<time datetime="2020-01-14" pubdate> hier</time>`

`<time datetime="2020-01-14T11:30"> 11h30, le 14 Janvier 2020</time>`

Le bug de `<time datetime="2000">l'an 2000</time>`

Je reviens dans `<time datetime="P30M">une demi-heure</time>`

Heures  
Minutes  
Secondes  
Week  
Day

## - Figures

`<figure>`

``

`<figcaption>Figure 1.2 - rapport du ...</figcaption>`

`</figure>`

# Plan de page

```
<body>  
<article>Ceci est mon article</article>  
<nav>mon fil d'ariane</nav>  
<aside>ma publicité</aside>  
<section>ceci est ma section</section>  
</body>
```



1. *Untitled Section*
1. *Untitled Section*
2. *Untitled Section*
3. *Untitled Section*
4. *Untitled Section*

<https://gsnedders.html5.org/outliner/>

```
<body>  
<h1>le titre de body</h1>  
<article>  
  <h2>Titre de mon article</h2>  
  Ceci est mon article  
</article>  
<nav>  
  <h2>navigation</h2>  
  mon fil d'ariane  
</nav>  
<aside><h2>aside title</h2>ma publicité</aside>  
<section><h2>titre de section</h2>ceci est ma section</section>
```



1. le titre de body
1. Titre de mon article
2. navigation
3. aside title
4. titre de section

# Tableaux

## EXAMPLE 2

```
<p id="summary">In the following table, characteristics are given in the second column, with the negative side in the left column and the positive side in the right column.</p>
```

```
<table aria-describedby="summary">
```

```
  <caption>Characteristics with positive and negative sides</caption>
```

```
  <thead>
```

```
    <tr>
```

```
      <th id="n"> Negative
```

```
      <th> Characteristic
```

```
      <th> Positive
```

```
  <tbody>
```

```
    <tr>
```

```
      <td headers="n r1"> Sad
```

```
      <th id="r1"> Mood
```

```
      <td> Happy
```

```
    <tr>
```

```
      <td headers="n r2"> Failing
```

```
      <th id="r2"> Grade
```

```
      <td> Passing
```

```
</table>
```

Notez l'absence de balises fermantes !

Crédit: W3C

# Tableaux

## EXAMPLE 3

```
<figure aria-labelledby="caption">
  <p>Characteristics are given in the second column, with the
  negative side in the left column and the positive side in the right
  column.</p>
  <table>
  <caption id="caption">Characteristics with positive and negative sides</caption>
  <thead>
    <tr>
      <th id="n"> Negative
      <th> Characteristic
      <th> Positive
    </tr>
  </thead>
  <tbody>
    <tr>
      <td headers="n r1"> Sad
      <th id="r1"> Mood
      <td> Happy
    </tr>
    <tr>
      <td headers="n r2"> Failing
      <th id="r2"> Grade
      <td> Passing
    </tr>
  </tbody>
</table>
</figure>
```

crédit: W3C

# Tableaux

```
<table border="1">
  <caption> School auction sign-up sheet </caption>
  <thead>
    <tr>
      <th><label for=e1>Sellers Name</label>
      <th><label for=e2>Product for sale</label>
      <th><label for=e3>Picture of product</label>
      <th><label for=e4>Reserve Price</label>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ms Danus
      <td>Doughnuts
      <td>
      <td>$45
    </tr>
    <tr>
      <td><input id=e1 type=text name=who required form=f>
      <td><input id=e2 type=text name=what required form=f>
      <td><input id=e3 type=url name=pic form=f>
      <td><input id=e4 type=number step=0.01 min=0 value=0 required form=f>
    </tr>
  </tbody>
</table>
```

th

crédit: W3C

Lire le modèle de tableau



# Formulaires

```
<form method="post"
  enctype="application/x-www-form-urlencoded"
  action="https://pizza.example.com/order.cgi">
```

```
<p><label>Customer name: <input name="custname"></label></p>
<p><label>Telephone: <input type=tel name="custtel"></label></p>
<p><label>E-mail address: <input type=email name="custemail"></label></p>
```

```
<fieldset>
<legend> Pizza Size </legend>
<p><label> <input type=radio name=size value="small"> Small </label></p>
<p><label> <input type=radio name=size value="medium"> Medium </label></p>
<p><label> <input type=radio name=size value="large"> Large </label></p>
</fieldset>
```

```
<fieldset>
<legend> Pizza Toppings </legend>
<p><label> <input type=checkbox name="topping" value="bacon"> Bacon </label></p>
<p><label> <input type=checkbox name="topping" value="cheese"> Extra Cheese </label></p>
<p><label> <input type=checkbox name="topping" value="onion"> Onion </label></p>
<p><label> <input type=checkbox name="topping" value="mushroom"> Mushroom </label></p>
</fieldset>
```

```
<p>
  <label>Preferred delivery time:
    <input type=time min="11:00" max="21:00" step="900" name=« delivery »>
  </label>
</p>
```

```
<p><label>Delivery instructions: <textarea name="comments"></textarea></label></p>
<p><button>Submit order</button></p>
</form>
```

# Syntaxe

- Préférable de fermer les balises

```
<p><label> <input type=radio name=size value="small" /> Small </label></p>
```

- Syntaxe alternative

```
<form>
  <table>
    <caption>Example, <label>'s for attribute</caption>
    <tr>
      <th><label for="name">Customer name: </label></th>
      <td><input name="name" id="name"></td>
    </tr>
  </table>
</form>
```

- Ceci:

```
<p><label>Delivery instructions:
  <textarea name="comments"></textarea></label></p>
```

- mais pas:

```
<p><label>Delivery instructions:
  <textarea name="comments" /></label></p>
```

Delivery instructions:

```
</label></p>
<p><button>Submit
```

# Vérification côté client

- **required** impose une valeur dans un input (après submit)

```
<p><label>Customer name: <input name="custname" required /></label></p>
```

- **maxlength** #max de caractères dans le champ input (lors de la saisie)
- **minlength** #min de caractères (après submit)
- **pattern** pour imposer des contraintes lors de la validation

```
<input type="text" pattern="^\d{5,6}(?:[-\s]\d{4})?$" ...>
```

Content type	name attribute	autocomplete attribute
Name	<code>name fname mname lname</code>	<ul style="list-style-type: none"> <li><code>name</code> (full name)</li> <li><code>given-name</code> (first name)</li> <li><code>additional-name</code> (middle name)</li> <li><code>family-name</code> (last name)</li> </ul>
Email	<code>email</code>	<code>email</code>
Address	<code>address city region province state zip zip2 postal country</code>	<ul style="list-style-type: none"> <li>For one address input: <ul style="list-style-type: none"> <li><code>street-address</code></li> </ul> </li> <li>For two address inputs: <ul style="list-style-type: none"> <li><code>address-line1</code></li> <li><code>address-line2</code></li> </ul> </li> <li><code>address-level1</code> (state or province)</li> <li><code>address-level2</code> (city)</li> <li><code>postal-code</code> (zip code)</li> <li><code>country</code></li> </ul>
Phone	<code>phone mobile country-code area-code exchange suffix ext</code>	<code>tel</code>
Credit Card	<code>ccname cardnumber cvc cmonth cyear exp-date card-type</code>	<ul style="list-style-type: none"> <li><code>cc-name</code></li> <li><code>cc-number</code></li> <li><code>cc-csc</code></li> <li><code>cc-exp-month</code></li> <li><code>cc-exp-year</code></li> <li><code>cc-exp</code></li> <li><code>cc-type</code></li> </ul>
Username	<code>username</code>	<ul style="list-style-type: none"> <li><code>username</code></li> </ul>
Password	<code>password</code>	<ul style="list-style-type: none"> <li><code>current-password</code> (for sign-in forms)</li> <li><code>new-password</code> (for sign-up and password-change forms)</li> </ul>

# Quelques points


- Utilisez *autocomplete* et *placeholder* pour faciliter la vie de l'utilisateur

```
<label for="frmNameA">Name</label>  
<input type="text" name="name" id="frmNameA"  
  placeholder="Full name" required autocomplete="name">
```

```
<label for="frmEmailA">Email</label>  
<input type="email" name="email" id="frmEmailA"  
  placeholder="name@example.com" required autocomplete="email">
```

```
<label for="frmEmailC">Confirm Email</label>  
<input type="email" name="emailC" id="frmEmailC"  
  placeholder="name@example.com" required autocomplete="email">
```

```
<label for="frmPhoneNumA">Phone</label>  
<input type="tel" name="phone" id="frmPhoneNumA"  
  placeholder="+1-555-555-1212" required autocomplete="tel">
```



aide  
l'agent

crédit

# Quelques points

- **datalist** pour énumérer des suggestions

```
<label for="frmFavChocolate">Favorite Type of Chocolate</label>  
<input type="text" name="fav-choc" id="frmFavChocolate" list="chocType">  
<datalist id="chocType">  
  <option value="white">  
  <option value="milk">  
  <option value="dark">  
</datalist>
```

crédit

- Indiquer un attribut **autofocus** pour (au moins) un élément input textuel

```
<input type="search" name="search" autofocus />
```

**note:** possibilité de mentionner un **input** en dehors de l'élément **form**

```
<form id="maforme">  
  <input type="submit" />  
</form>  
<input type="text" name="mytext" form="maforme" />
```

Pas rétro-compatible, ne pas en abuser

Regular expression	
Postal address	[a-zA-Z\d\s\-\, \#\.\ \+]+
Zip Code (US)	^\d{5,6}(?:[-\s]\d{4})?S
IP Address (IPv4)	^(?:((?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.)}{3}(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\$
IP Address (IPv6)	^(((?:[0-9a-fA-F]{1,4}:){7,7}[0-9a-fA-F]{1,4} (?:[0-9a-fA-F]{1,4}:){1,7}:  (?:[0-9a-fA-F]{1,4}:){1,6}: [0-9a-fA-F]{1,4} (?:[0-9a-fA-F]{1,4}:){1,5}(: [0-9a-fA-F]{1,4}){1,2}  (?:[0-9a-fA-F]{1,4}:){1,4}(: [0-9a-fA-F]{1,4}){1,3}  (?:[0-9a-fA-F]{1,4}:){1,3}(: [0-9a-fA-F]{1,4}){1,4}  (?:[0-9a-fA-F]{1,4}:){1,2}(: [0-9a-fA-F]{1,4}){1,5}  [0-9a-fA-F]{1,4}: ( (?:[0-9a-fA-F]{1,4}){1,6})  ( (?:[0-9a-fA-F]{1,4}){1,7}  :)  fe80:(?:[0-9a-fA-F]{0,4}){0,4}% [0-9a-zA-Z]{1,}  : (ffff(:0{1,4}){0,1}:){0,1}((25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]).){3,3}(25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]  ([0-9a-fA-F]{1,4}:){1,4}: ( (25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]).){3,3} (25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]))\$
IP Address (both)	^(?:((?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.)}{3}(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)  ( (?:[0-9a-fA-F]{1,4}:){7,7}[0-9a-fA-F]{1,4}  (?:[0-9a-fA-F]{1,4}:){1,7}:  (?:[0-9a-fA-F]{1,4}:){1,6}: [0-9a-fA-F]{1,4}  (?:[0-9a-fA-F]{1,4}:){1,5}(: [0-9a-fA-F]{1,4}){1,2}  (?:[0-9a-fA-F]{1,4}:){1,4}(: [0-9a-fA-F]{1,4}){1,3}  (?:[0-9a-fA-F]{1,4}:){1,3}(: [0-9a-fA-F]{1,4}){1,4}  (?:[0-9a-fA-F]{1,4}:){1,2}(: [0-9a-fA-F]{1,4}){1,5}  [0-9a-fA-F]{1,4}: ( (?:[0-9a-fA-F]{1,4}){1,6})  ( (?:[0-9a-fA-F]{1,4}){1,7}  :)  fe80:(?:[0-9a-fA-F]{0,4}){0,4}% [0-9a-zA-Z]{1,}  : (ffff(:0{1,4}){0,1}:){0,1}((25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]).){3,3}(25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]  ([0-9a-fA-F]{1,4}:){1,4}: ( (25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]).){3,3} (25[0-5] (2[0-4] 1{0,1}[0-9])){0,1}[0-9]))\$
Credit Card Number	^(?:4[0-9]{12}(?:[0-9]{3})? 5[1-5][0-9]{14} 3[47][0-9]{13} 3(?:0[0-5]  [68][0-9])[0-9]{11} 6(?:011 5[0-9]{2}))[0-9]{12} (?:2131 1800  35\d{3})\d{11})\$
Social Security Number	^\d{3}-\d{2}-\d{4}\$
North American Phone Number	^(?:((?:\+?1\s*(?:[.-]\s*)?)?(?:\(\s*([2-9]1[02-9]  [2-9][02-8]1 [2-9][02-8][02-9])\s*\)  ([2-9]1[02-9]  [2-9][02-8]1 [2-9][02-8][02-9]))\s*(?:[.-]\s*)?)?([2-9]1[02-9]  [2-9][02-9]1 [2-9][02-9]{2})\s*(?:[.-]\s*)?([0-9]{4})(?:\s*(?:# x\.? ext\.? extension)\s*(\d+))?\$

# Cools inputs

- Dans HTML5, il existe d'autres types d'input natifs: *password*, *color*, *date*, *datetime-local*, *month*, *week*, *number*, *range*

**Attention:** dans Safari et dans Explorer sur un ordinateur de table (desktop), il n'y a pas forcément d'objet dédié qui est affiché, contrairement aux *devices* mobiles

```
<p> <label>Pwd: <input type="password"
name="pwd" placeholder="12345" required /></
label> </p>
<p> <label>Color: <input type="color"
name="colour" value="#ff0000" /></label> </p>
<p> <label>Date: <input type="date" name="date"
placeholder="YYYY-MM-DD" required/></label></
p>
<p><label>Date&Time: <input type="datetime-
local" name="datetime" placeholder="YYYY-MM-
DD:HH-MM" required/> </label> </p>
<p>
  <label>Mois: <input type="month"
name="month" placeholder="???" required /></
label>
  <label>Semaine: <input type="week"
name="week" placeholder="???" required /></
label>
</p>
<p> <label>Nombre: <input type="number"
name="nombre" placeholder="123" min=1
max=100 step=5 required /> </label> </p>
<p> <label>Intervalle: <input type="range"
name="range" min=1 max=100 required /> </
label> </p>
```

Urls:

Pwd:

Color:

Date:

Date&Time:

Mois:  Semaine:

Nombre:

Intervalle:

Safari 13.0 (desktop)



# Cools inputs

- Dans HTML5, il existe d'autres types d'input natifs: *password*, *color*, *date*, *datetime-local*, *month*, *week*, *number*, *range*

**Attention:** dans Safari et dans Explorer sur un ordinateur de table (desktop), il n'y a pas forcément d'objet dédié qui est affiché, contrairement aux *devices* mobiles

```
<p> <label>Pwd: <input type="password"
name="pwd" placeholder="12345" required /></
label> </p>
<p> <label>Color: <input type="color"
name="colour" value="#ff0000" /></label> </p>
<p> <label>Date: <input type="date" name="date"
placeholder="YYYY-MM-DD" required/></label></
p>
<p><label>Date&Time: <input type="datetime-
local" name="datetime" placeholder="YYYY-MM-
DD:HH-MM" required/> </label> </p>
<p>
  <label>Mois: <input type="month"
name="month" placeholder="???" required /></
label>
  <label>Semaine: <input type="week"
name="week" placeholder="???" required /></
label>
</p>
<p> <label>Nombre: <input type="number"
name="nombre" placeholder="123" min=1
max=100 step=5 required /> </label> </p>
<p> <label>Intervalle: <input type="range"
name="range" min=1 max=100 required /> </
label> </p>
```

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	?

Chrome v79.0.3945.117 (macOS/desktop)

# Cools inputs

- Dans HTML5, il existe d'autres types d'input natifs: *password*, *color*, *date*, *datetime-local*, *month*, *week*, *number*, *range*

**Attention:** dans Safari et dans Explorer sur un ordinateur de table (desktop), il n'y a pas forcément d'objet dédié qui est affiché, contrairement aux *devices* mobiles

```
<p> <label>Pwd: <input type="password"
name="pwd" placeholder="12345" required /></
label> </p>
<p> <label>Color: <input type="color"
name="colour" value="#ff0000" /></label> </p>
<p> <label>Date: <input type="date" name="date"
placeholder="YYYY-MM-DD" required/></label></
p>
<p><label>Date&Time: <input type="datetime-
local" name="datetime" placeholder="YYYY-MM-
DD:HH-MM" required/> </label> </p>
<p>
  <label>Mois: <input type="month"
name="month" placeholder="???" required /></
label>
  <label>Semaine: <input type="week"
name="week" placeholder="???" required /></
label>
</p>
<p> <label>Nombre: <input type="number"
name="nombre" placeholder="123" min=1
max=100 step=5 required /> </label> </p>
<p> <label>Intervalle: <input type="range"
name="range" min=1 max=100 required /> </
label> </p>
```

13:56 Lun. 20 janv. 84%

Non sécurisé — www-labs.iro.umontreal.ca

Page d'accueil IFT 3225 | gdss - Recherche Google | Canadian\_AI\_2020 - Online L... | Formulaires | tes - Recherche Google

Bacon  
 Extra Cheese  
 Onion

Effacer

17	octobre	2017
18	novembre	2018
19	décembre	2019
20	janvier	2020
21	février	2021
22	mars	2022
23	avril	2023

Date: 20 janv. 2020

Date& Time: 20 janv. 2020 13:21

Mois: janvier 2020 | Semaine:

Nombre: 567

Intervalle:

iOS 13.3

<form method="get"

enctype="application/x-www-form-urlencoded"

action="http://www-labs.iro.umontreal.ca/~felipe/IFT3225-Hiver2020/ressources/treat\_form\_get.php">

User:

Customer name:

Telephone:

E-mail address:

Code promotion:

Pizza Size

Small

Medium

Large

Pizza Toppings

Bacon

Extra Cheese

Onion

Mushroom

test de nouveaux inputs

Urls:

Pwd:

Color:

Date:

Date&Time:

Mois:  Semaine:

Nombre:

Intervalle:

## Requête GET

[http://www-labs.iro.umontreal.ca/~felipe/IFT3225-Hiver2020/ressources/treat\\_form\\_get.php?custname=John+Doe&custtel=don%27t+have+one&custemail=toto%40toto&size=medium&topping=bacon&topping=cheese&topping=onion&topping=mushroom&location=https%3A%2F%2Ftools.ietf.org%2Fhtml%2Frfc2045&pwd=&colour=%23201af4&date=2020-01-16&datetime=2020-01-16T00%3A59&month=2020-01&week=2020-W03&nombre=&range=25](http://www-labs.iro.umontreal.ca/~felipe/IFT3225-Hiver2020/ressources/treat_form_get.php?custname=John+Doe&custtel=don%27t+have+one&custemail=toto%40toto&size=medium&topping=bacon&topping=cheese&topping=onion&topping=mushroom&location=https%3A%2F%2Ftools.ietf.org%2Fhtml%2Frfc2045&pwd=&colour=%23201af4&date=2020-01-16&datetime=2020-01-16T00%3A59&month=2020-01&week=2020-W03&nombre=&range=25)

## Résultat

custname -- John Doe  
custtel -- don't have one  
custemail -- toto@toto  
size -- medium  
topping -- bacon, cheese, onion, mushroom  
location -- https://tools.ietf.org/html/rfc2045  
pwd --  
colour -- #201af4  
date -- 2020-01-16  
datetime -- 2020-01-16T00:59  
month -- 2020-01  
week -- 2020-W03  
nombre --  
range -- 25

# Élément *output*

form-w3school.html (adapté de [w3school](#))

```
<form oninput="x.value=a.valueAsNumber + b.valueAsNumber">  
  0 <input type="range" id="a" name="a" value="50" /> 100  
  +  
  <input type="number" id="b" name="b" value="50"/>  
  =  
  <output name="x" for="a b">input please ! </output>  
  <br />  
  <input type="reset" />  
</form>
```

ids impliqués

valeur par défaut

0  100 +  = 107

Réinitialiser

Fait pour faire du calcul sur des données d'un formulaire (ex: calcul des taxes)

**Note:** x n'est pas transmis

# Marquage sémantique

Idée: partage de vocabulaires

## RDFa (XML) W3C

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/"  
  about="http://www.example.com/books/wikinomics">  
  In his latest book  
  <span property="dc:title">Wikinomics</span>,  
  <span property="dc:creator">Don Tapscott</span>  
  explains deep changes in technology,  
  demographics and business.  
  The book is due to be published in  
  <span property="dc:date" content="2006-10-01">October 2006</span>.  
</div>
```

## Microformat directement en HTML

```
<div class="vcard">  
  <p>  
    <span class="fn">Jean Bout</span><br/>  
    <span class="org">Société Exemple</span><br/>  
    <span class="tel">604-555-1234</span><br/>  
    <a class="url" href="http://exemple.com/">http://exemple.com/</a>  
  </p>  
</div>
```

cons: ajout de marquage spécifique non dédié, pas une norme

# microdata

- Norme WHATWG
- Utilisé par les moteurs de recherche, par les navigateurs
- Attributs (les principaux):
  - **itemscope** (signale un objet)
  - **itemtype** (à côté de itemscope) pour indiquer la nature de l'objet
  - **itemprop** (définit une propriété d'un objet)

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
    Director: <span itemprop="name">James Cameron</span>
    (born <time itemprop="birthDate" datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href=" ../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>
</div>
```

- Des vocabulaires partagés: schema.org

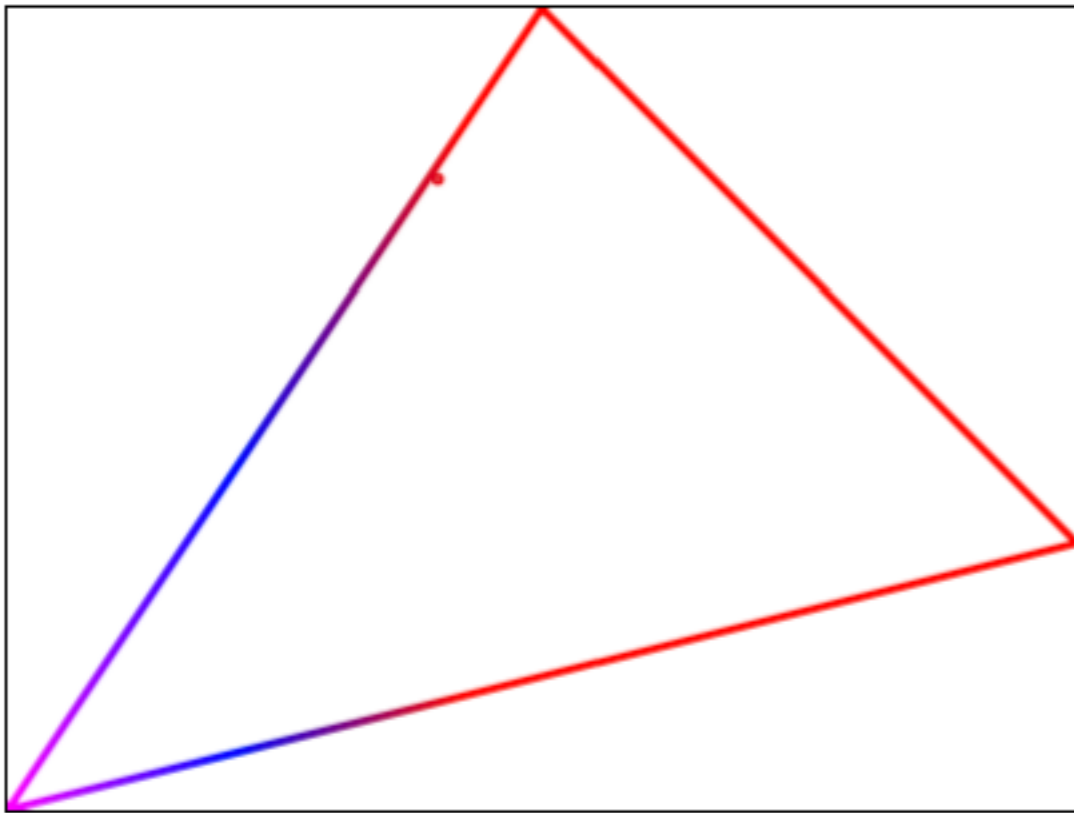
# microdata

Depuis 2007, **Google** recommande d'utiliser plutôt **JSON-LD**

```
<script type="application/ld+json">
{
  "@context": "http://schema.org/",
  "@type": "Movie",
  "name": "Avatar",
  "director":
  {
    "@type": "Person",
    "name": "James Cameron",
    "birthDate": "1954-08-16"
  },
  "genre": "Science fiction",
  "trailer": "../movies/avatar-theatrical-trailer.html"
}
</script>
```

# Canvas

Soit un triangle et un point de départ  $p$ . Lorsque vous allez cliquer sur `jouer`, je vais tirer aléatoirement un sommet  $s$ , afficher un point à mi-chemin entre  $p$  et  $s$ , puis recommencer en prenant ce point comme nouveau départ. Que croyez-vous qu'il va se passer ?



```
<body>
```

```
<p>Soit un triangle et un point de départ p</p>
```

```
    Lorsque vous allez cliquer sur <button id="jouer">jouer</button>, je vais tirer aléatoirement un sommet s, afficher un point à mi-chemin entre p et s, puis recommencer en prenant ce point comme nouveau départ. Que croyez-vous qu'il va se passer ?
```

```
</p>
```

```
<canvas id="myCanvas"
        width="400" height="300"
        style="border:1px solid #000000; >>
```

```
Pas d'élément canvas
```

```
</canvas>
```

```
<br />
```

```
<progress id="progress" max=10000 value=« 1"></progress>
```

```
</body>
```

**Note: pas très élégant de mettre du css dans le HTML**



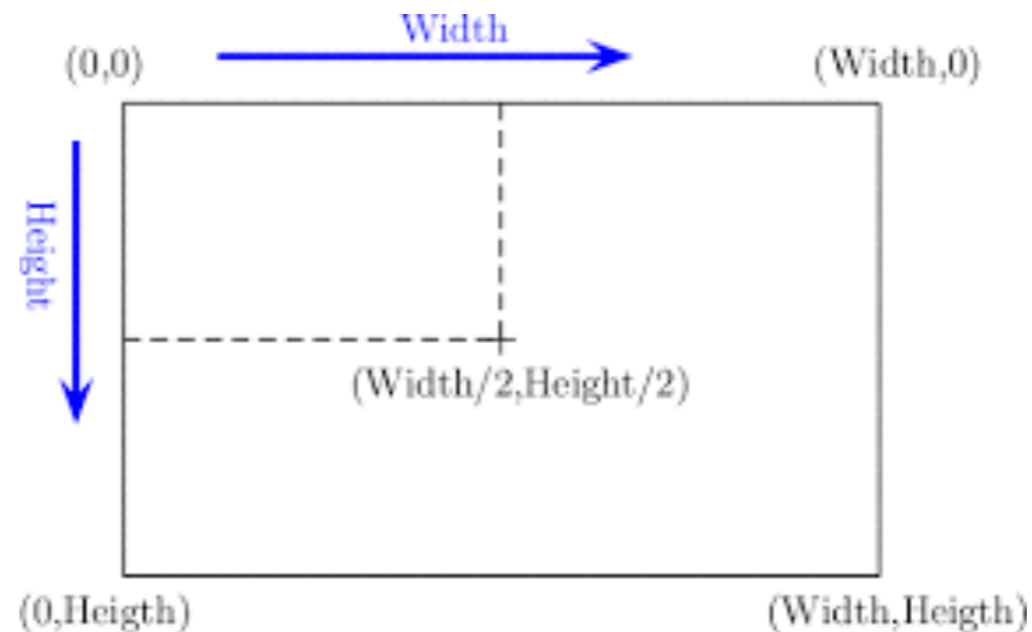
# Canvas

Que l'on peut ensuite *scripter*:

```
var c = document.getElementById("myCanvas");  
ctx = c.getContext("2d");
```

```
ctx.beginPath();  
ctx.arc(10, 100, 2, 0, 2 * Math.PI);  
ctx.closePath();  
ctx.stroke();
```

plus: [w3schools](http://www.w3schools.com)



# HTML ... une expertise...

## Accessibility concerns

The presence of the `del` element is not announced by most screen reading technology in its default configuration. It can be made to be announced by using the CSS `content` property, along with the `::before` and `::after` pseudo-elements.

```
1 del::before,  
2 del::after {  
3   clip-path: inset(100%);  
4   clip: rect(1px, 1px, 1px, 1px);  
5   height: 1px;  
6   overflow: hidden;  
7   position: absolute;  
8   white-space: nowrap;  
9   width: 1px;  
10 }  
11  
12 del::before {  
13   content: " [deletion start] ";  
14 }  
15  
16 del::after {  
17   content: " [deletion end] ";  
18 }
```

```
<p><del>This text has been  
deleted</del>,  
here is the rest of the  
paragraph.</p>  
<del><p>This paragraph has been  
deleted.</p></del>
```

Some people who use screen readers deliberately disable announcing content that creates extra verbosity. Because of this, it is important to not abuse this technique and only apply it in situations where not knowing content has been deleted would adversely affect understanding.

- [Short note on making your mark \(more accessible\) | The Paciello Group](#)
- [Tweaking Text Level Styles | Adrian Roselli](#)

# Compatibilité

```
<head>
```

```
<title>Bonjour le Monde</title>
```

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

```
<!--[if lt IE 9]>
```

```
<script src="bower_components/html5shiv/dist/html5shiv.js"></script>
```

```
<![endif]->
```

```
</head>
```

## html5Shiv

pour injecter les nouveaux éléments dans la DOM  
pour les navigateurs qui ne les reconnaissent pas