

# HTML

Balise	Attributs
Toutes les balises	class, data-*, draggable, id, lang
<article>, <aside>, <datalist>, <div>, <figure>, <figcaption>, <footer>, <head>, <header>, <nav>, <section>, <p>, <pre>, <span>, <table>, <u>, <ul>	<i>Aucun attribut spécifique</i>
<a>	href, media, rel, target, type
<audio>, <video>	autoplay, controls, height, muted, played, preload, src, width
<button>	autofocus, disabled, form, formnovalidate, name, type, value
<form>	action, autocomplete, enctype, method, name, novalidate, target
<html>	manifest
<iframe>	height, name, src, width
<img>	alt, height, src, width
<input>	type, accept, autocomplete, autofocus, autosave, checked, disabled, form, formnovalidate, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, value, width
<label>	for, form
<link>	href, rel, type
<meta>	charset, content, http-equiv, name
<meter>	form, high, low, max, min, value
<option>	disabled, label, selected, value
<output>	for, form, name
<progress>	max, value
<script>	src, type, defer
<select>	autofocus, disabled, form, multiple, name, required, size
<source>	src, type, media
<textarea>	autofocus, cols, disabled, form, maxlength, name, placeholder, required, rows

## CSS

### Propriétés de boîte :

Propriété	Valeurs	Héritée
height, width	<i>longueur, pourcentage, auto</i>	non
min-height, max-height, min-width, max-width	<i>longueur, pourcentage</i>	non
margin-top, margin-right, margin-bottom, margin-left	<i>longueur, pourcentage, auto</i>	non
margin	<i>Combinaison des valeurs de margin</i>	non
padding-top, padding-right, padding-bottom, padding- left	<i>longueur, pourcentage</i>	non
padding	<i>Combinaison des valeurs de padding</i>	non
border-top-color, border- right-color, border-bottom- color, border-left-color	<i>couleur, transparent</i>	non
border-top-style, border- right-style, border-bottom- style, border-left-style	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	non
border-top-width, border- right-width, border-bottom- width, border-left-width	thin, medium, thick, <i>longueur</i>	non
border-top, border-left, border-bottom, border-right, border	<i>Combinaison des valeurs de border</i>	non
outline-color	<i>couleur, invert</i>	non
outline-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset	non
outline-width	thin, medium, thick, <i>longueur</i>	non
outline	<i>Combinaison des valeurs de outline</i>	non

**Propriétés de mise en page :**

Propriété	Valeurs	Héritée
display	block, inline, inline-block, inline-table, list-item, run-in, table-caption, table-cell, table-column, table-column-group, table-footer-group, table-header-group, table-row, table-row-group, none	non
float	left, right, none	non
position	absolute, fixed, relative, static	non
clear	none, left, right, both	non
visibility	visible, hidden, collapse	oui
clear	none, left, right, both	non
top, right, bottom, left	<i>longueur, pourcentage, auto</i>	non
z-index	<i>entier, auto</i>	non
overflow	auto, hidden, scroll, visible	non

**Propriétés de listes**

Propriété	Valeurs	Héritée
list-style-type	circle, disc, square, armenian, decimal, decimal-leading-zero, georgian, lower-alpha, lower-greek, lower-latin, lower-roman, upper-alpha, upper-latin, upper-roman, none	oui
list-style-position	inside, outside	oui
list-style-image	<i>url, none</i>	oui
list-style	<i>Combinaison des trois propriétés</i>	oui

**Propriétés typographiques**

Propriété	Valeurs	Héritée
font-family	<i>famille</i>	oui
font-size	<i>taille absolue, taille relative, longueur, pourcentage</i>	oui
font-weight	100, 200, 300, 400, 500, 600, 700, 800, 900, bold, bolder, lighter, normal	oui
font-style	italic, normal, oblique	oui
font-variant	normal, small-caps	oui
font	<i>Combinaison des 5 propriétés de font</i>	oui
text-align	center, justify, left, right	oui
text-decoration	blink, line-through, overline, underline, none	non
text-indent	<i>longueur, pourcentage</i>	oui
text-transform	capitalize, lowercase, none, uppercase	oui

**Pseudo-classes**

:active	L'élément a été activé par l'utilisateur (utilisé généralement pour un bouton ou un hyperlien).
:checked	L'élément est sélectionné (checkbox, bouton radio ou item d'un menu déroulant).
:disabled, :enabled	L'élément est désactivé, activé, respectivement.
:empty	Désigne un élément qui n'a aucun enfant.
:first-child, :nth-child(n), :last-child	L'élément est le premier, nième ou dernier fils.
:first-of-type, :nth-of-type(n), :last-of-type	Désigne le premier, nième ou dernier élément d'un même type.
:focus	L'élément est celui qui a le focus.
:hover	La souris est au dessus de l'élément.
:in-range, :out-of-range	La valeur associée à l'élément est (n'est pas) dans l'étendue spécifiée .
:invalid, :valid	La valeur fournie à l'item est invalide (valide).
:link	Tout hyperlien qui n'a pas été visité.
:not(sélecteur)	Négation d'un sélecteur
:only-child	L'élément n'a pas de frères.
:only-of-type	L'élément n'a pas de frères du même type que lui.
:read-only, :read-write	Identifie un élément qui peut ou ne peut pas être édité.
:required, :optional	L'élément est un item de formulaire requis (non requis).
:visited	L'élément est un hyperlien sur lequel on a cliqué.

**Pseudo-éléments**

: :after, : :before	Élément virtuel ajouté avant ou après l'élément concerné. Utilisé en général avec la propriété <code>content</code> .
: :first-letter	Désigne la première lettre d'un paragraphe.
: :first-line	Désigne la première ligne d'un paragraphe.
: :selection	Désigne le segment de la page qui a été surligné (highlight) par l'utilisateur.

## DOM

Propriétés et méthodes de l'objet *document* :

activeElement	Retourne l'élément qui a le focus actuellement.
title, head, body	Retourne l'élément correspondant à la balise.
forms	Retourne une liste de tous les formulaires.
images	Retourne une liste de toutes les images.
links	Retourne une liste de tous les hyperliens.
getElementById()	Retourne l'élément possédant le ID passé en paramètre.
getElementsByTagName()	Retourne une collection contenant tous les éléments dont la balise correspond à celle passée en paramètre.
getElementsByClassName()	Retourne une collection contenant tous les éléments dont la classe correspond à celle passée en paramètre.
querySelector()	Retourne le premier élément correspondant au sélecteur CSS passé en paramètre.
querySelectorAll()	Retourne une collection contenant tous les éléments correspondant au sélecteur CSS passé en paramètre.
createElement()	Crée un élément correspondant à la balise passée en paramètre.
createTextNode()	Crée un nouveau noeud texte.

Propriétés et méthodes associées à un noeud  $N$  du DOM :

childNodes	Collection contenant les noeuds fils, incluant les noeuds texte.
firstChild	Retourne le premier noeud fils
lastChild	Retourne le dernier noeud fils
nextSibling	Retourne le noeud qui a le même parent que $N$ et qui est situé immédiatement après.
previousSibling	Retourne le noeud qui a le même parent que $N$ et qui est situé immédiatement avant.
parentNode	Retourne le noeud parent.
parentElement	Retourne l'élément parent.
textContent	Retourne tout le contenu textuel d'un noeud (incluant ses descendants).
nodeType	Retourne le type d'un noeud. En particulier : 1 = noeud élément, 2 = noeud attribut, 3 = noeud texte.
appendChild()	Ajoute le noeud passé en paramètre après tous les noeuds fils de $N$ .
removeChild()	Retire le noeud fils passé en paramètre.
replaceChild(nouv,ancien)	Remplace un noeud fils.
insertBefore(nouv,ref)	Ajoute le noeud <i>nouv</i> juste avant le noeud <i>ref</i> , qui est un noeud fils de $N$ .

Propriétés et méthodes associées à un élément  $E$  du DOM (Rappel : Element est une sous-classe de Node) :

innerHTML	Contient tout le contenu HTML inclus dans un noeud.
children	Collection contenant les noeuds fils, excluant les noeuds texte.
firstElementChild	Retourne le premier fils, en ne considérant pas les noeuds texte.
lastElementChild	Retourne le dernier fils, en ne considérant pas les noeuds texte.
nextElementSibling	Retourne le noeud qui a le même parent que $N$ et qui est situé immédiatement après, en ne considérant pas les noeuds texte.
previousElementSibling	Retourne le noeud qui a le même parent que $N$ et qui est situé immédiatement avant, en ne considérant pas les noeuds texte.
style	Permet d'obtenir et de changer le style d'un élément. Exemple : <code>monElement.style.color = blue.</code>
className	Permet d'obtenir et de changer la classe d'un élément.
addEventListener(ev,gest,cap)	Associe un gestionnaire d'événement à un événement. Si le troisième argument est <i>true</i> le traitement se fait aussi au cours de la capture (pas seulement lors de la propagation vers le haut (bubbling)).
getAttribute()	Retourne la valeur de l'attribut passé en paramètre.
getElementByClassName(), getElementsByTagName(), querySelector(), querySelectorAll()	Mêmes méthodes que celles de l'objet Document, sauf qu'elles sont appliquées uniquement sur l'élément.
remove()	Retire l'élément
removeAttribute()	Retire l'attribut dont le nom est passé en paramètre.
setAttribute()	Fixe la valeur d'un attribut.

## jQuery

.add(balise)	Ajoute un élément correspondant à la balise passée en paramètre.
.addClass(classe)	Ajoute une classe à l'élément.
.after(sélecteur), before(sélecteur)	Ajoute du contenu après/avant l'élément.
.append(contenu), prepend()	Ajoute le contenu à la fin/au début du contenu actuel de l'élément.
.attr(nomAttribut)	Obtient la valeur d'un attribut.
.attr(nom,valeur)	Fixe la valeur d'un attribut.
.blur(gest)	Exécute la fonction <i>gest</i> chaque fois que l'élément perd le focus.
.children(sélecteur)	Retourne tous les noeuds fils correspondant au sélecteur (ce dernier est facultatif).
.click(gest)	Exécute la fonction <i>gest</i> chaque fois qu'on clique sur l'élément.
.clone()	Fait une copie profonde de l'élément.
.css(prop)	Obtient la valeur d'une propriété de style.
.css(prop,valeur)	Fixe la valeur d'une propriété de style.
.each(fct)	Exécute la fonction pour tous les éléments.
.end()	Ignore la cascade d'appels qui le précède et reprend l'élément original.
.filter(sélecteur)	Réduit l'ensemble d'éléments à ceux qui correspondent au sélecteur passé en paramètre.
.find(selecteur)	Recherche tous les éléments correspondant au sélecteur.
.first(), .last()	Retourne le premier/dernier élément de l'ensemble.
.focus()	Exécute la fonction <i>gest</i> chaque fois que l'élément devient le focus.
.hide(), .show()	Rend l'élément invisible/visible.
.hover(gest)	Exécute la fonction <i>gest</i> chaque fois que le curseur de la souris se trouve au dessus de l'élément.
.html()	Obtient le contenu HTML.
.html(contenu)	Change le contenu HTML par celui passé en paramètre.
.hover(gestIn,gestOut)	Associe des gestionnaires exécutés quand le curseur de la souris entre et quitte l'espace occupé par l'élément, respectivement.
.off(ev,sélecteur,gest)	Élimine l'association entre un gestionnaire et l'événement, pour tous les éléments représentés par le sélecteur (ce dernier est facultatif).
.on(ev,sélecteur,gest)	Associe un gestionnaire à un événement pour tous les éléments représentés par le sélecteur (ce dernier est facultatif).
.parent(sélecteur)	Retourne le parent, s'il correspond au sélecteur (ce dernier est facultatif).
.ready(gest)	Exécute la fonction <i>gest</i> quand le DOM est complètement chargé en mémoire.
.remove(sélecteur)	Retire l'élément.
.removeClass(classe)	Retire une classe de l'élément.
.replaceWith()	Remplace un élément par le contenu passé en paramètre.
.text()	Obtient tout le texte contenu dans l'élément.
.text(texte)	Remplace le contenu de l'élément par le texte passé en paramètre.
.toggle()	Affiche ou cache un élément, selon son état actuel.
.toggleClass(classe)	Retire ou ajoute une classe, selon qu'elle est déjà présente ou non.
.val()	Fixe ou obtient la valeur d'un élément <code>input</code> , <code>select</code> ou <code>textarea</code> .
\$.ajax(url,params)	Effectue une requête Ajax.
\$.get(url,data,callback)	Effectue une requête Ajax avec la méthode GET.
\$.post(url,data,callback)	Effectue une requête Ajax avec la méthode POST.

## Jade

Code sans output	-
Code avec output	=
Code avec output non filtré	!=
Commentaires traduits en HTML	//
Commentaires non traduits en HTML	//-
Conditionnels	if ... else ... unless case ... when ... when... default...
itérations	each <var> in <collection>
Texte non formaté	 On peut aussi le mettre directement après la balise qui doit contenir le texte On peut mettre un point "." après une balise pour indiquer un bloc de texte
Interpolation	#{expression} L'expression peut être une simple variable ou n'importe quelle expression Javascript
Attributs	a(class='button', href='google.com') Google La valeur de l'attribut peut être fournie par n'importe quelle expression Javascript qui retourne un string La valeur de l'attribut class peut être une liste
Attributs de classe	balise.classe
Attribut id	balise#id
Inclusion	include nom_fichier
Extensions	Dans le fichier de référence, on identifie des points d'extension avec la directive <code>block</code> . Dans le fichier d'extension, on reprend les blocs avec les mêmes noms et y associe des directives en Jade pour chaque bloc. Dans le fichier d'extension, les blocs du fichier de référence seront remplacés par le contenu spécifié. Pour ne pas remplacer un bloc, on peut utiliser <code>block prepend</code> ou <code>block append</code> pour ajouter le nouveau contenu avant ou après le contenu du bloc de référence.
Blocs réutilisables	<code>mixin</code> pour définir le bloc (peut prendre des paramètres). + pour utiliser le bloc. ...arg pour représenter un argument de taille variable (sera traitée comme une liste).



## Principaux codes de status de réponse HTTP

100	Continue
200	OK
201	Created
202	Accepted
204	No content
300	Multiple Choice
301	Moved Permanently
302	Found (Moved Temporarily)
303	See Other
304	Not Modified
307	Temporary Redirect
308	Permanent Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Payload Too Large
414	URI Too Long
415	Unsupported Media Type
416	Request Range Not Satisfiable
417	Expectation Failed
431	Request Header Fields Too Large
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
502	Service Unavailable

## Angular2

Fonctions de base	
@NgModule() @Component()	Définition d'un module. Ses attributs principaux sont : <i>imports, declarations, bootstrap</i> Définition d'une composante. Ses attributs principaux sont : <i>selector, template, styles, templateUrl, styleUrls, providers</i> .
@Injectable() @Pipe()	Utiliser pour définir un service. Pour définir un filtre. Son attribut principal est <i>name</i> . Le filtre doit implémenter la classe Pipe-Transform.
Services	
Http	Service pour communiquer avec un serveur HTTP. Fonctions offertes : <i>get(), head(), post(), put(), delete(), jsonp(), patch()</i> .
{{ expression }}	Interpolation dans le document HTML
Filtres	
currency	Transforme en valeur monétaire. Par défaut \$ est utilisé. Paramètres : le symbole de monnaie, nombre de décimales après le point.
date	Transforme en date. Paramètres : format de date, le fuseau horaire.
filter	Filtre générique qui transforme un tableau. Paramètres : expression prédicative, comparateur.
json	Convertit un objet Javascript en JSON.
lowercase, uppercase	Pour transformer en minuscules et majuscules.
limitTo	Pour ne sélectionner qu'un certain nombre d'items. Paramètres : nombre d'items (on compte à partir de la fin si le nombre est négatif), position de départ.
async	Permet de récupérer la valeur d'une promesse.
decimal	Permet de choisir le nombre de chiffre avant et après la virgule d'un chiffre décimal.
percent	Transforme en pourcentage. On décide le nombre de chiffre avant et après la virgule avec son paramètre.
Directives	
ngClass	Pour ajouter dynamiquement une ou plusieurs classes. Si l'expression associée à cette directive correspond à un string, ce string représente la classe ajoutée. On peut aussi utiliser une expression qui correspond à un tableau de strings pour ajouter plusieurs classes.
click	Évalue l'expression lors d'un clic.
ngIf	L'élément est inséré si l'expression est vraie.
ngModel	Associe la valeur de l'input à une variable du contrôleur.
ngModelChange	Associe une expression à l'événement de changement d'une valeur d'un input.
ngFor	Répète un élément pour chaque item d'une liste. On peut utiliser les valeurs suivantes à l'intérieur de l'élément sur lequel on boucle : <i>index, first, middle, last, even, odd</i> .
ngSubmit	Associe une expression à l'événement submit.
ngSwitch	Pour choisir entre plusieurs éléments alternatifs définis avec <i>ngSwitchCase</i> .

## MongoDB

Méthodes de base de données	
createCollection()	Pour créer une nouvelle collection.
Méthodes de collection	
count(critère)	Pour compter le nombre de documents qui correspondent au critère.
drop()	Pour éliminer toute la collection.
find(critère,projection)	Pour extraire les documents qui correspondent au critère spécifié.
findOne(critère,projection)	Pour extraire un document qui correspond au critère spécifié.
findAndModify(requête)	Pour rechercher un document et le modifier.
insert(document), insert(liste de documents)	Pour insérer un ou plusieurs documents.
remove(critère)	Élimine les documents qui correspondent au critère spécifié.
update(critère, modifications)	Met à jour les documents qui correspondent au critère spécifié.
Méthodes de curseur	
count()	Compte le nombre de documents.
forEach(function)	Applique une fonction à tous les documents.
hasNext()	Vérifie s'il y a un prochain document.
limit()	Limite le nombre de documents que peut contenir le curseur.
map(funciton)	Applique une fonction à tous les documents et retourne les résultats dans un tableau.
next()	Retourne le prochain document.
size()	Retourne le nombre de documents.
sort(critère)	Trie les documents selon le critère spécifié.
toArray()	Retourne un tableau conteant tous les documents.
Opérateurs de requête	
\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin	Opérateurs de comparaison
\$or, \$and, &not, \$nor	Opérateurs logiques
\$exists	Le document doit avoir le champ en question
Opérateurs de mise à jour	
\$inc	Pour incrémenter la valeur d'un champ.
\$mul	Pour multiplier par une certaine valeur la valeur d'un champ.
\$rename	Pour renommer un champ.
\$set	Pour fixer la valeur d'un champ.
\$unset	Pour éliminer un champ.
\$addToSet	Ajoute des éléments à un tableau, s'ils n'existent pas déjà.
\$pop	Retire le premier item d'un tableau.
\$push	Ajoute un item dans un tableau.