Informatique

Publication numérique

Isaac Pante Bureau 2047 très souvent et accessoirement bureau 3037

peut être suivi sur twitter - isaacpante

Mardi 2 octobre 2018

Clé pour Moodle

pubnum

Nouveau collectif d’écriture

Infinite RPG : plateforme sur laquelle des auteurs se rassemblent pour construire une sorte de jeu de rôle ensemble. Plusieurs personnes vont écrire des personnages qui vont collaborer ensemble

Voir comment nos pratiques d’écriture et de lecture évoluent. Le travail a pour objectif de nous faire développer des choses. Citer parti des compétences pour développer nos propres outils.

Possible de le faire sur des Livres jeux, par exemple

Les stratégies de publication qu’on retrouve du côté des développeurs contamine les publications numériques.

Babel 2.0

Quand on est dans une langue naturelle on sait qu’il y a un lexique et une syntaxe. En plus l’élément déterminant est les usages. Ce n’est pas trivial. Il y a eu des switch de paradigmes.

Une langue c’est évolutif, ce sont des communautés, des usages.

Par contre quand on est dans un langage formel on a tendance a oublié tout cela.

HTML

Usages - Lexique W3C - Syntaxe W3C

On a alors l’impression que les langages de programmation serait quelque chose de beaucoup plus fixe. On est dans un domaine plus expérimental : si l’on ne fournit pas des données compréhensibles à l’ordi il va nous dire qu’il y a une « violation de la syntaxe ». Quand on va creuser, on va se rendre compte qu’à ce niveau de langage, il y a une très grande marge de manoeuvre très conditionnée par les usages et les communautés

Des usages, des pratiques ont été détournés par des hackers et le W3C a décidé d’intégrer ces nouvelles pratiques. Beaucoup de choses viennent de la base utilisateur.

On peut aborder la programmation sous la forme d’une langue. On peut s’intéresser aux usages. L’objet nous en devient plus familier. Quant on est face à un langage… En matière de public

On a maintenant des historiens qui travaillent sur ces histoires culturelles.

Le HTML a évolué jusqu’à la version 4.1 (a débuté en 1991) où on avait alors pensé que son évolution était faite.

Puis on a décider d’arrêter avec le langage HTML parce qu’il était beaucoup trop permissif (nous pouvions écrire du texte libre sans balises, et même chose si une balise n’était par fermée il la fermait pour nous,

Au fur et à mesure on s’est rendu compte que le web était de moins en moins destiné à façonner des pages et à savoir quelle serait la résolution finale. Il fallait donc repenser toute la chose.

On s’est dit alors que le langage XML était un méta langage de description.

Quand on a souhaité avoir un langage plus propre qui nous permette d’inventer le langage que l’on veut par exemple le MathML. On s’est dit pourquoi ne pas le réutiliser pour réécrire le XML (vérifier ?).

Vague autour du XML pendant un certain temps.

Automatiquement la spécification va bloquer. Les anciennes pages ne seront pas respectueuses mais les nouvelles devraient l’être.

Il y a eu alors des oppositions de camps entre ceux qui prônaient le XML et ceux qui le rejetaient.

Il aurait dû y avoir un XHTML 2.0 après le XHTML 1.0

Mais pendant ce temps un autre groupe ayant une autre vision du web liée aux usages. Le web de 2005 n’est plus celui de 1991. D’autant plus que l’on sait que les 3/4 des pages sont présentées plus ou moins de la même façon ne pourrait-on pas avoir des balises qui décrivent directement certaines choses (par exemple video qui permettrait de charger directement une video).

Il s’est alors installé un certain flou, en se demandant si on allait vers le XHTML ou vers le HTML 5. Finalement c’est le HTML 5 qui semble l’avoir emporté.

Aujourd’hui on en arrive à la conclusion que passer par des numéros est une mauvaise idée, on va parler du HTML comme d’un standard vivant. On va voir des balises ajoutées, apparaître et au fur et à mesure le HTML va évoluer.

Un des grands enjeux qui agite le web en ce moment c’est que aujourd’hui si on estime que le contenu que l’on met sur notre site est un contenu de qualité, et que l’on veut être rémunéré pour cela, il faudrait une sorte de porte (on paie pour accéder aux ressources). C’est le format qui est utilisé actuellement, une partie ouverte et une partie payante.

Mais des discussions sont en cours. Tiber Messli de W3C y est favorable.

Il faudrait une balise DRM digital right media la section qui serait alors à l’intérieur de la balise vous demanderait alors un code pour accéder à son contenu.

Quel problème cela créée-t-il ? Le HTML prône dès son départ un projet humain et open source, donc le jour ou l’on met une balise qui nécessite un code, cela donne une zone opaque qui n’est plus accessible à tous. Cela semble attaquer la neutralité du web. Il y a des gros enjeux qui se discutent autour de cela. Il y a des enjeux supra nationaux qui impactent ensuite l’entier du web mondial.

On voit que ces technologies ne sont pas neutres.

On a tellement peu la culture qu’il est difficile pour les gens de faire le lien pour expliquer les enjeux de manière claire.

On va aborder dans ce cours le HTML , le XHTML et le XML.

On va s’intéresser à l’ePUB, au HTML 5 et au CSS 3

Autour de cela nous devons prendre un certain nombre d’acteurs dont l

W3C

IDPF

A côté de cela il faut aussi considérer les utilisateurs et la manière dont ils apportent des éléments.

Le hacking et la philosophie qui va avec : l’idée est qu’il y a hacking à partir du moment où l’on détourne quelque chose. Pour ces différents langages on a aussi des effets de hacking.

En développement web on le fait un peu tout le temps. Lorsque l’on veut produire un effet qui n’existe pas, on va le créer.

Autour de cela il y a bien sûr des acteurs commerciaux comme Amazon qui sont les acteurs qui dictent ce qui va se passer.

Nous allons partir dans le codage

Ce que l’on a garder du XHTML c’est que l’on s’est rendu compte que finalement vu qu’on doit faire beaucoup de transformation de contenu pour les différents supports.

D’un côté on a le fond (HTML) et la forme (CSS).

Cela fait d’autant plus sens qu’on quitte de plus en plus le format page.

Inévitablement il faut donc une dissociation du fond et de la forme. Dans quelques temps on sait que le web sera lu, qu’il va disparaître. Il se diffusera via un nombre d’outils colossaux. Il y aura de moins en moins d’interfaces qui se diffuseront par contre sur des objets connectés.

Le web a utilisé dès son départ des technologies open source.

HTML 5 dans lequel on va mettre de la structure et CSS 3 qui va nous aider à mettre la forme.

La question de la source :

Qu’avons-nous comme texte initial ?

Préparer un texte (regExp)

Si l’on part d’un texte papier que l’on veut numériser puis le préparer comme matériaux de première main qui puisse être récupérer par les publications numériques.

Une première solution : scanner le document.

Dans un autre cas, lorsque ce n’est pas notre usage et que l’on veut un texte plus lisible, on va enlever certaines choses - c’est alors là qu’un travail académique commence. On voit qu’ici, il s’agira de gérer certains paramètres.

Si on veut passer du manuscrit au texte on passe par un algorithme

C’est alors OCR (ce n’est pas le nom de l’algorithme mais du système)

Cette page OCR on la fait ensuite passer par un logiciel comme Prizmo . On voit que dans l’exemple il a même réussi à mettre les paragraphes.

Ca c’est la théorie mais en général c’est bien plus compliqué que cela.

Que va t’il se passer par rapport à un ouvrage papier.

Livres papiers : fins de lignes

que le logiciel va considérer automatiquement par un retour à la ligner alors que ce n’est qu’une contrainte physique. De même les tirets qui séparent les mots.

Du côté du livre numérique, il n’y a pas de fin de lignes.

Ce qui est paradoxale c’est que l’on doit les enlever pour devoir ensuite les rajouter. Car il y a des habitudes de lectures. On va d’abord purifier un texte pour rajouter ensuite des effets avec des plug-in.

Un autre avantage c’est la taille des caractères que l’on peut changer mais qui implique alors que des éléments vont passer sur la ligne suivante.

La chose qui reste ce sont les paragraphes.

Notre unité pour la ligne ce sera la typographie et pour le paragraphe ce sera la sémantique. L’exception étant la poésie. La ligne étant alors signifiante. Usage particulier d’un texte porteur de message.

Comment va-t’on faire ?

On va d’abord passer par de l’OCR. Comme certains caractères ne marchent pas (par exemple le é) on va faire des remplacements. On va décider d’enlever tous les tirets de fin de ligne (par exemple) On doit faire appelle à des règles. En programmation on va pouvoir définir des expressions régulières (qui viennent vérifier certaines règles). Expressions régulières : les quantificateurs.

On a des logiciels, notamment des traitements de textes qui travaillent sur ces expression régulières.

Le premier élément de ces règles c’est le quantificateur.

Le problème des expressions régulières c’est qu’elles sont extrêmement critiques

Cela devient encore plus compliqué si on ajoute d’autres règles

Expressions régulières : les groupements

Expressions régulières : les délimiteurs.

Expressions régulières : autres

Pour nous aider, en général on va trouver des plateformes et des sites web on peut construire nos expressions et avoir un contenu qui nous permettent de voir directement ce qui se passe.

par exemple [regex101.com](http://regex101.com)

Aller voir aussi visual studio code ?

Allez télécharger VISUAL STUDIO CODE