

École polytechnique de Louvain

Exercices INGINIOUS destinés à l'enseignement secondaire

Introduction aux langages HTML/CSS

Auteur: **Clément CORDIER**

Promoteur: **Olivier BONAVENTURE**

Lecteurs: **Benoît DUHOUX, Ludovic TAFFIN**

Année académique 2022–2023

Master [60] en sciences informatiques

Remerciements

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce mémoire :

Mon promoteur le Prof. Olivier Bonaventure pour le suivi et l'aide apportée tout au long de l'année à travers les diverses réunions organisées,

M. Ludovic Taffin pour avoir rendu ce sujet d'étude possible, ce qui m'a permis de réaliser ce travail,

M. Benoît Duhoux pour avoir accepté et pris le temps de prendre part à la relecture de ce mémoire en tant que membre du Jury,

M. Nicolas Rybowski pour avoir répondu à nos interrogations relatives à l'utilisation et au fonctionnement d'INGInious ainsi que pour son système d'intégration continue, mais aussi l'ensemble de l'équipe travaillant sur INGInious pour la mise à disposition d'un module d'introduction à la plateforme en tant qu'enseignant,

Et tout particulièrement à Mme Marie-France Belin, enseignante en informatique dans le secondaire au collège Notre-Dame de Bon Secours à Binche, pour son implication et son enthousiasme vis-à-vis d'une collaboration entre l'enseignement secondaire et le supérieur, ainsi qu'à ses élèves pour avoir joué le jeu.

Table des matières

1	Introduction	4
1.1	Contexte	4
1.2	Problématique	5
1.3	Objectif	6
1.4	Approche	6
2	Les langages	7
2.1	HTML & CSS : qu'est-ce que c'est ?	7
2.2	Pertinence pour l'enseignement secondaire	8
3	Ressources existantes	10
3.1	Plateformes	10
3.1.1	W3Schools	11
3.1.2	Codecademy	11
3.1.3	Khan Academy	11
3.1.4	Technofutur TIC	12
3.2	Avantages & Inconvénients	12
3.3	Conclusion	13
4	Organisation	14
4.1	Premier pas avec INGINious	14
4.2	Choix des outils	15
4.3	Première implémentation & prise de contact	16
4.4	Suite du développement	16
5	Contenu du cours	17
5.1	Structure	17
5.2	Formats des exercices	18
5.3	Référence	18
5.4	Matières abordées	19
5.4.1	Bases théoriques	19

5.4.2	Contenu de la section HTML	19
5.4.3	Contenu de la section CSS	22
6	Evaluation des exercices	24
6.1	Validation	25
6.2	Analyse syntaxique	28
6.3	Correction	30
6.4	Conclusion	31
7	Mise en condition réelle	32
7.1	Déroulement	32
7.2	Résultats	33
7.3	Observations	34
7.4	Améliorations	35
7.5	Conclusion	36
8	Pistes d'améliorations	37
A	Exercices HTML servant d'illustration	42
B	Structure de données utilisées pour la correction	45
C	Résultats du sondage	48

Chapitre 1

Introduction

1.1 Contexte

L'enseignement secondaire en Belgique fait face à plusieurs difficultés, et ce depuis plusieurs années maintenant. Parmi celles-ci, nous pouvons notamment citer la pénurie d'enseignants, une infrastructure, des équipements ou encore des référentiels pas toujours adaptés. La pénurie d'enseignants amène certains professeurs à devoir dispenser des matières en dehors de leur domaine de prédilection.

Plusieurs initiatives ont vu le jour, telles que *Le Pacte pour un Enseignement d'excellence*[1] démarré en 2015, ainsi que *La stratégie numérique pour l'éducation*[2] en parallèle depuis 2018, toutes deux adoptées par le gouvernement de la Fédération Wallonie-Bruxelles (FWB). Parmi les objectifs visés, on retrouve notamment :

- L'élaboration d'un nouveau tronc commun interréseaux de la maternelle aux secondaires.
- La définition (et l'investissement dans) de l'équipement numérique minimal pour les établissements.
- La révision des formations aux TIC pour les enseignants en cours de carrière.

Au-delà des TIC, une initiative plus récente a aussi vu le jour avec pour ambition de former les enseignants à l'informatique[3].

Outre le manque de formation, une absence de ressources adéquates est aussi à souligner pour les enseignants d'informatique[4]. Il existe de nombreuses ressources (physique ou en ligne) visant à apprendre des branches spécifiques du domaine de l'informatique. Malgré leur nombre, il est difficile d'en extraire celles qui sont réellement adaptées à l'enseignement secondaire. De plus, contrairement aux matières de l'enseignement général telles que les mathématiques ou les sciences, l'informatique ne dispose pas de supports attitrés servant de fil conducteur. Suite à

une démarche gouvernementale lancée en 2018[5], un tel référentiel a pourtant vu le jour en Angleterre, sous forme d’une plateforme en ligne dédiée à l’enseignement de l’informatique dans le secondaire, destiné aux élèves de 16 à 19 ans.[6]

En plus d’une approche théorique, l’apprentissage d’un langage informatique se fait avant tout par de la pratique. Cet aspect de l’apprentissage est primordial pour qu’un enseignant puisse évaluer la compréhension et les acquis de ses étudiants. Par ailleurs, la majorité des ressources sont pensées et conçues pour un apprentissage en autodidacte, et ne conviennent pas toujours pour faire office de support de cours.

L’élaboration de ce mémoire est une suite logique au travail “Adapter INGINious aux besoins des enseignants du secondaire”[7]. Si ce dernier avait pour but de se pencher sur le contenant, ce mémoire a quant à lui pour objectif de travailler sur le contenu, en proposant un cours en ligne constitué d’une batterie d’exercices en HTML/CSS et hébergé sur INGINious.

1.2 Problématique

Sans même parler des tutoriels, il existe une quantité indénombrable de plateformes en ligne proposant une introduction au développement web à travers les fondamentaux : HTML & CSS. Toutefois, la plupart se contente d’une approche théorique, rarement accompagnée d’exercices pratiques.

De plus, les langages HTML et CSS sont permissifs en terme d’erreurs, et les navigateurs indulgents pour ce qui est d’afficher un rendu, ce qui complique d’autant plus l’enseignement de ces langages et des bonnes pratiques, comme le relève l’auteur d’une étude visant à classer les erreurs commises en HTML et CSS : “*Browsers are tolerant of errors, and often render HTML and CSS code that is riddled with bugs. When a beginner writes code that has many errors but still renders as desired, they receive positive feedback in the form of the properly formatted web page. These errors are latent, remain unresolved, and reinforce faulty understandings that can become difficult to overcome.*” (Thomas H. Park, 2013)[8]

1.3 Objectif

Le but de ce travail est de fournir un recueil d'exercices HTML/CSS adapté à l'enseignement secondaire ou à une première expérience avec ces langages, sans connaissance ni prérequis préalables et qui facilite une évaluation des acquis.

1.4 Approche

La mise en place de ces exercices a été pensée sous plusieurs formes : l'idée initiale étant de fournir un contenu exclusivement axé sur l'apprentissage par la pratique. Une alternative sous la forme d'un syllabus interactif a aussi été envisagée afin de proposer un contenu plus exhaustif alliant des sections théoriques, auxquelles seraient insérés des exercices INGINious. Ces idées n'ont finalement pas été retenues. A la place, une solution visant un entre-deux a été choisie : un cours INGINious classique, mais qui en plus de proposer des exercices pratiques, contient des tâches dédiées à la théorie, ce afin de répondre à cet objectif d'autonomie sans nécessité de support externe, facilitant le déploiement dans le programme existant des enseignants tout en étant plus attractif pour les étudiants qu'un syllabus interactif.

Chapitre 2

Les langages

Ce chapitre a pour objectif de présenter les langages HTML & CSS et justifier la pertinence de leur apprentissage dans le cadre de l'enseignement secondaire.

2.1 HTML & CSS : qu'est-ce que c'est ?

HyperText Markup Language (HTML) est le langage utilisé pour créer la structure d'une page web. Il définit les éléments qui la composent tels que des titres, des paragraphes, des images ou encore des liens. Ces éléments sont représentés sous la forme de balises indiquant au navigateur le contenu à afficher. Ainsi, une balise `<p>` indique la présence d'un paragraphe, tandis que `<img>` indique l'insertion d'une image.

Un élément est constitué de deux balises : une balise ouvrante et une balise fermante. La seconde se distingue de la première avec le caractère `/` qui vient s'ajouter devant le nom de l'élément : la balise ouvrante `<p>` est généralement accompagnée de sa balise fermante `</p>`.

Exception faite pour quelques éléments HTML particuliers, appelés éléments vides. Ceux-ci sont constitués d'une seule et unique balise : `<img>` ou encore `<img/>` sont deux notations possibles pour cette catégorie d'éléments. Ces éléments n'ont ni balise fermante, ni contenu qui leur est associé. Enfin, des paramètres supplémentaires peuvent être appliqués aux éléments via ce que l'on appelle des attributs. Ceux-ci sont renseignés sous la forme de paires `attribut="valeur"` au sein de la balise ouvrante. Selon l'élément considéré, certains attributs sont obligatoires tandis que d'autres sont optionnels. Par exemple, il est nécessaire d'inclure l'attribut `src` à un élément `img` afin de renseigner le chemin vers l'image à afficher : ``.

Cascading Style Sheets (CSS) est le langage utilisé pour mettre en forme, styler une page web. À travers des règles, il définit le visuel à appliquer aux éléments qui la composent telles que des polices d'écriture, des couleurs ou encore des marges. Ainsi, la règle `p {color: blue;}` applique la couleur bleue à l'ensemble des paragraphes d'une page.

Avec Javascript, ils constituent les 3 langages essentiels à la création de pages web modernes. Bien que HTML et CSS soient des langages informatiques, il est important de noter que ce ne sont pas des langages de programmation pour autant. En effet, ni l'un ni l'autre n'est conçu pour créer des programmes capables d'exécuter des instructions logiques. HTML est un langage de *balisage* tandis que CSS est un langage de *feuille de styles*.

2.2 Pertinence pour l'enseignement secondaire

L'apprentissage du développement web présente plusieurs atouts :

Compréhension d'Internet

Tout d'abord, l'élaboration de pages web permet de mieux comprendre comment les sites sont construits et fournit plus globalement des pistes de compréhension sur le fonctionnement d'Internet. Cette compréhension va au-delà de la simple utilisation des sites web, les aidant à devenir des utilisateurs avertis et critiques d'Internet.

Aspect ludique et créatif

En outre, l'attrait du développement web sera sans nulle doute plus prononcé pour les élèves en comparaison à un langage de programmation de part son côté ludique. L'avantage de la réalisation de page web pour des élèves est l'obtention très rapide de résultats concrets, visuellement parlant. Cet aspect du développement stimule et encourage la créativité des élèves. Cela peut les encourager à développer des compétences artistiques tout en travaillant avec la technologie.

Renforcement de la logique

De plus, l'apprentissage de l'HTML/CSS induit inévitablement les élèves à la résolution de problèmes et entraîne leurs compétences analytiques. La programmation web, y compris HTML et CSS, stimule le développement de la pensée logique chez les élèves.

La structure proposée des langages et la nécessité de décomposer les problèmes complexes en éléments plus simples appellent la raison logique et analytique.

Sensibilisation à l'accessibilité et à l'inclusion

L'apprentissage des langages HTML et CSS est l'occasion d'introduire des notions d'accessibilité et d'inclusion. Les élèves peuvent apprendre à créer des sites web qui sont adaptés à tous les utilisateurs, y compris ceux ayant des besoins spécifiques, par exemple pour des personnes atteintes de déficience visuelle. Cela favorise une sensibilisation à la diversité et à l'éthique dans la conception technologique.

Ouverture au monde de l'informatique

Enfin, le développement web donne un avant-goût de ce qu'est la programmation et représente une porte d'entrée vers l'apprentissage d'autres langages. Il pourrait préparer les jeunes à se diriger vers des carrières technologiques. La technologie ne cesse de se développer dans le monde entier. Plus que jamais, un besoin accru en professionnels dans le domaine de l'informatique se fait sentir. L'apprentissage du développement web dès le plus jeune âge permet aux élèves de se préparer à des carrières dans des domaines tels que le développement de logiciels, le design d'interface utilisateur, la gestion de sites web et bien d'autres.

Chapitre 3

Ressources existantes

Ce chapitre a pour objectif d’analyser les différentes plateformes pouvant répondre au sujet de ce mémoire, à savoir l’apprentissage par la pratique des langages HTML & CSS, afin d’évaluer les avantages & inconvénients de chacune ainsi que leur pertinence dans le cadre de l’enseignement secondaire.

3.1 Plateformes

Coursera[9], Udemy[10], freeCodeCamp[11], Udacity[12], et bien d’autres encore, sont autant d’exemples de plateformes en ligne proposant, entre autres, des formations au développement web, en commençant par les fondamentaux : HTML & CSS.

Pour réduire le champ des possibles et se rapprocher autant que faire se peut de ce que propose un cours INGINious, seules des plateformes proposant un apprentissage par la pratique (éventuellement accompagné de théorie) **avec feedback** sont considérées dans ce chapitre.

Ci-dessous sont résumées les différentes plateformes ayant été retenues. Pour chacune d’entre elles, une description générale de la solution proposée par le site est fournie, suivie d’une analyse focalisée sur la mise en place des exercices pratiques. En particulier, les critères suivants seront considérés dans cette analyse :

- Disponibilité du contenu en français
- Pertinence du niveau de difficulté
- Suivi de la progression des élèves
- Efficacité pour évaluer la maîtrise des concepts
- Qualité du feedback fourni aux élèves
- Instantanéité du feedback

3.1.1 W3Schools

Bien que W3Schools[13] n'ait pas très bonne réputation dans le milieu, le contenu proposé sur leur plateforme n'en reste pas moins adapté aux débutants. A défaut d'être utilisé en tant que référence pour l'élaboration du cours INGINious, W3Schools constitue une ressource intéressante à destination des élèves, pour faire ses premiers pas en HTML et CSS.

La popularité de la plateforme est entre autre due à son éditeur de code en ligne : "Try it Yourself". Les exercices n'exploitent toutefois pas cet outil ; ils sont présentés sous la forme de texte à trous. Les réponses attendues se limitent à un élément (HTML)[14]/une déclaration (CSS)[15], voire une sous-partie de ceux-ci.

3.1.2 Codecademy

Codecademy[16] est une plateforme d'apprentissage en ligne axée sur la programmation et le développement web. Elle propose des ressources théoriques pour chaque langage, réparties par concepts. Codecademy met aussi à disposition un éditeur de code en ligne : les Workspaces¹. Toutefois, à l'inverse de W3Schools, elle propose des cours interactifs qui reposent directement sur cet outil d'édition de code intégré.

3.1.3 Khan Academy

Khan Academy[17] est une plateforme d'apprentissage en ligne qui couvre un large éventail de sujets, allant des mathématiques aux sciences. Elle propose des vidéos, des exercices interactifs et des leçons pratiques. En outre, Khan Academy propose une solution dédiées aux enseignants, permettant de gérer leur classe avec la possibilité d'assigner du contenu personnalisé ainsi qu'un suivi de la progression de chaque élève.

En ce qui concerne le domaine de l'informatique, la plateforme dispose d'une section "Apprendre à coder" dont l'un des chapitres est une introduction au HTML/CSS. Les exercices qu'il contient se font via un éditeur de code en ligne dans lequel un template est fourni, template que l'élève doit compléter avec un ou plusieurs éléments tel que décrit dans l'énoncé. La détection d'erreur/correction se fait en temps réel à travers un assistant textuel.

1. <https://www.codecademy.com/pages/workspaces>

3.1.4 Technofutur TIC

Sur base de la section “Qui sommes-nous?” de leur site web, Technofutur TIC[18] se présente comme suit : “Créé à l’initiative du Gouvernement de la Wallonie en 1998, Technofutur TIC est un « centre de compétence² » labellisé, situé sur l’Aéropôle de Gosselies au Nord de Charleroi.”

Parmi les thématiques de formation disponibles, on retrouve une section “Programmation & Développement Web” qui comporte deux formations d’initiation aux langages HTML et CSS. Ces formations sont conçues pour être dispensées à distance et peuvent être aussi bien suivies en classe ou à la maison. Un coach Technofutur en charge du cours est disponible pour assister les étudiants et corriger leurs soumissions.

3.2 Avantages & Inconvénients

Cette section a pour but de comparer les 4 plateformes présentées, évaluées selon les 6 critères suivants : la disponibilité du contenu en français, le niveau de difficulté adapté à une initiation, le suivi de la progression de ses élèves pour l’enseignant, l’efficacité du contenu pour évaluer la maîtrise des concepts, la qualité du feedback fourni aux étudiants, et enfin l’instantanéité du feedback.

	W3Schools	Codecademy	Khan Academy	Technofutur TIC
Contenu disponible en français	X	X	✓	✓✓
Difficulté adaptée à une initiation	✓✓	✓	✓	✓
Suivi de la progression	X	X	✓✓	✓
Évaluation des connaissances	XX	✓	✓	✓✓
Qualité/précision du feedback	✓	X	X	✓✓
Instantanéité du feedback	✓	✓✓	✓	XX

2. Les centres de compétence ont été conçus comme des pôles de formation, de veille et de sensibilisation, s’inscrivant dans une logique de développement, pour répondre aux besoins exprimés par les entreprises, les travailleurs, les demandeurs d’emploi, l’enseignement et les partenaires sectoriels.

3.3 Conclusion

Il existe doré et déjà des solutions diverses et variées en ce qui concerne l'apprentissage des langages HTML/CSS. Comme beaucoup de ressources en ligne, celles-ci sont majoritairement anglophones, ce qui n'est pas forcément le type de contenu recherché par les enseignants du secondaire désirant un cours adapté au plus grand nombre. En effet, la restriction de la langue représente une barrière à l'entrée indéniable pour certains élèves du secondaire.

Chaque plateforme présente des avantages et des inconvénients qui leurs sont propres. Bien que W3Schools propose des exercices abordables à tous, ceux-ci ne suffisent pas à évaluer les connaissances, ni même la maîtrise de la syntaxe du langage.

Technofutur se distingue par son feedback manuel. Une supervision humaine offre un feedback personnalisé pour chaque étudiant, le rendant forcément plus précis. L'inconvénient se trouve au niveau du délai du feedback : les étudiants sont contraints d'attendre un certain temps avant d'avoir un retour sur leur travail. De plus, l'apprentissage par essais-erreurs est ici impossible, ce qui empêche les étudiants de réfléchir sur les erreurs qu'ils commettent. Cette étape de réflexion joue un rôle clé et leur sera bénéfique au-delà de l'apprentissage de ces deux langages spécifiques.

Codecademy et Khan Academy sont les deux alternatives se rapprochant le plus de ce qu'INGInious peut proposer et ne souffrent pas des problèmes mentionnés au-dessus. Leur principal inconvénient se situe au niveau du manque de précision du code de l'étudiant, considéré comme correct sans forcément répondre à l'énoncé.

Chapitre 4

Organisation

La première étape de ce travail a été la prise en main de l’outil servant de support de cours, à savoir INGINious. La seconde étape a été de se focaliser sur la réflexion de la méthodologie de correction ainsi que les outils nécessaires à cette correction, ainsi que la sélection de cesdits outils. Une fois les outils sélectionnés, il a fallu les mettre à l’épreuve en réalisant quelques exercices basiques. Dès lors qu’une première version fonctionnelle du cours fut en place, une prise de contact a été engagée avec l’enseignement secondaire afin de proposer les exercices INGINious en tant que support de cours. Ci-dessous vont être détaillées chacune de ces étapes.

4.1 Premier pas avec INGINious

Les premières semaines ont été consacrées à l’apprentissage d’INGInious. En tant qu’étudiants, nous autres mémorants avons eu l’occasion d’utiliser la plateforme à de nombreuses reprises pour la réalisation d’exercices de programmation, via divers cours de notre cursus. Cependant, il n’a jamais été question de créer des exercices, uniquement d’y répondre. L’élaboration de ce mémoire a donc été l’occasion de découvrir les coulisses de cet outil et occuper le rôle de nos professeurs à travers la réalisation de nos cours respectifs.

A cette fin, plusieurs ressources se sont avérées utiles : la première étant évidemment la documentation INGINious[19]. Celle-ci dispose d’une section dédiée au rôle “enseignant” qui présente les différentes options permettant d’administrer son cours, ainsi que les différentes commandes disponibles via les librairies INGINious, utilisées pour traiter l’input d’un étudiant ou encore lui retourner du feedback. Outre la documentation officielle, le code source de certains cours INGINious, réalisés à la fois par des professeurs mais aussi par d’anciens mémorants, est accessible publiquement[20]. Cela aura permis d’avoir un aperçu du workflow complet du système de correction et du rendu final que peut avoir un cours type.

4.2 Choix des outils

Bien que la plateforme INGINious propose une API prête à l’emploi[21] pour l’élaboration et le déploiement d’exercices, le système de correction nécessaire aux langages de balisage et feuilles de style que sont HTML et CSS ne peut se faire de façon similaire à un exercice type d’un langage de programmation. La première solution a été de se reposer sur des expressions régulières. Cette option est envisageable pour corriger des cas simples tels qu’un unique élément HTML ou une unique règle CSS. Elle montre toutefois rapidement ses limitations pour des cas de figure plus avancés. HTML et CSS sont des langages plus complexes qu’il n’y paraît à première vue : leur syntaxe est variée et permet des structures imbriquées. Par conséquent, les expressions régulières envisagées tiraient en longueur, rendant le code difficilement lisible et maintenable en plus de devenir une source de bugs difficiles à identifier et corriger. Pour toutes ces raisons, les expressions régulières ne sont définitivement pas suffisantes pour subvenir à nos besoins : des outils spécialisés s’avèrent nécessaires. Après quelques recherches, j’ai découvert l’existence de validateur en ligne, mentionné à juste titre par Prof. Olivier Bonaventure après lui avoir fait part de cette nécessité. En particulier, le “Nu Html Checker”[22] et le “CSS Validator”[23], tous deux développés par le World Wide Web Consortium (W3C). Chacun des deux outils dispose d’une version portable en ligne de commande, alternative idéale pour les inclure directement au workflow propre à INGINious, à travers le script de correction.

Bien que ces outils permettent de distinguer une soumission syntaxiquement correcte d’une soumission qui ne l’est pas, cela ne suffit pas à déterminer si la soumission d’un étudiant répond effectivement à l’énoncé de la question qui lui est posée. Pour nous aider à vérifier cela, d’autres outils sont nécessaires : des analyseurs syntaxiques. Ces analyseurs vont nous permettre de déconstruire la structure d’une soumission de manière atomique : chaque composante d’un élément HTML ou d’une règle CSS est récupérée et isolée¹.

Contrairement aux validateurs, le choix de ces outils a été sujet à divers changements en cours de route, aussi bien pour HTML que pour CSS. Les premiers choix de bibliothèques se sont d’abord portés sur BeautifulSoup4[24] (HTML) et cssutils[25] (CSS). Le premier a été remplacé par lxml[26] pour des raisons de performance, bien qu’en pratique cette différence soit peu notable dans le contexte d’une correction d’un exercice INGINious. Le second a été remplacé par tinycss2[27] pour des raisons de maintenabilité : après avoir fait le constat de certaines notations plus récentes non prises en charge par cssutils[28] et correctement interprétées par tinycss2.

1. L’aspect technique concernant l’utilisation de ces outils est détaillée à la section 6.2.

4.3 Première implémentation & prise de contact

Après avoir mis la main sur tous les outils nécessaires à la correction, il ne restait plus qu'à les mettre à l'épreuve en élaborant des exercices HTML basiques. Ces premiers exercices se rapprochaient du format proposé sur W3Schools : réponse attendue concise, avec un template de l'élément fourni dans la zone de réponse, auquel il faut ajouter une composante telle qu'une balise ou encore un attribut.

Une fois qu'une première suite d'exercices a été déployée, une prise de contact auprès de différents établissements du secondaire a été entreprise. M^{me} Belin, enseignante en charge des cours d'informatique du 2^e degré au collège Notre-Dame de Bon Secours à Binche, m'a rapidement fait part de son intérêt pour le cours et sa volonté d'entreprendre une collaboration avec l'enseignement supérieur. Le développement des exercices s'est donc naturellement poursuivi dans l'optique de répondre aux différents besoins relevés par M^{me} Belin afin d'assister ses élèves dans leur apprentissage et constituer une ressource tierce adaptée à leur programme scolaire.

4.4 Suite du développement

Avant d'envisager d'effectuer des tests en situation réelle avec les élèves, le contenu du cours INGINious se devait d'être plus étoffé.

Après plusieurs échanges, M^{me} Belin m'a informé que son programme de cours prévoyait essentiellement un apprentissage du HTML, et une brève introduction du CSS. Par conséquent, le développement de nouveaux exercices s'est dans un premier temps orienté sur le HTML.

Au cours d'une réunion organisée par le Prof. Olivier Bonaventure, M. Nicolas Rybowski, qui travaille activement sur le développement d'INGInious, nous a fait part de son outil d'intégration continue permettant de tester les résultats obtenus avec des soumissions prédéfinies et s'assurer qu'ils soient identiques après chaque mise à jour apportée. Cet outil assure une stabilité sur le long terme de façon automatisée, sans avoir à vérifier chaque exercice manuellement, à condition de générer des soumissions de test pour chaque exercice au préalable.

L'évolution du cours s'est poursuivie tout au long de l'année, avec le suivi continu de M^{me} Belin me partageant un retour constructif à chaque mise à jour dont je lui faisais part : les nouveautés, modifications, corrections qu'elle aimerait y voir ainsi que les difficultés types rencontrées par ses élèves sur lesquelles le cours devait accentuer le trait.

Le résultat final de cette collaboration est détaillé dans le chapitre suivant.

Chapitre 5

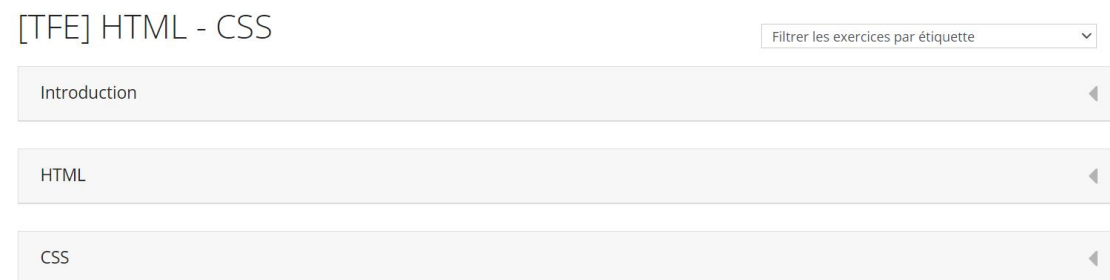
Contenu du cours

Ce chapitre a pour but de présenter la solution apportée^{1 2} afin de fournir une introduction aux langages HTML/CSS aux élèves de l’enseignement secondaire. Tout d’abord, la structure du cours est décrite. Les décisions quant au format des questions sont ensuite discutées. Le choix du référentiel ainsi que les raisons de ce choix sont présentés, suivi d’une description des points de matière abordés à travers le cours.

5.1 Structure

Comme indiqué en introduction, il a été décidé de déployer les exercices via le système de cours “classique”. Suite au travail effectué par mes prédécesseurs dans le cadre du mémoire “Adapter INGINious aux besoins des enseignants du secondaire”[7], la possibilité de diviser un cours en sections permet de distinguer aisément exercices HTML et CSS au sein d’un seul et même cours INGINious.

Voici comment ce dernier se présente :



-
1. Cours INGINious : https://inginiuous.org/course/html_css_clement
 2. Dépôt GitHub : https://github.com/ClemCordier/html-css_exercices

La section d'introduction, constituée d'une unique tâche, a pour objectif de contextualiser la philosophie du cours. Elle présente l'intégration des zones bacs à sable qui feront office d'environnement de développement, accompagnée d'une démonstration de leur usage, ainsi que le système de feedback qui a été pensé pour ce cours.

S'en suit les deux sections HTML et CSS. Chacune commence par une tâche vide³ afin d'enseigner les bases théoriques des deux langages. Bien que condensées, ces tâches se veulent exhaustives sans pour autant entrer dans les détails de chaque langage. Elles mettent l'accent sur la raison d'être du langage et sur leur syntaxe de base. La suite des deux sections mélange théorie et pratique sous différents formats, détaillés dans la section suivante.

5.2 Formats des exercices

Naturellement, la difficulté proposée à travers les exercices est progressive. Cela se reflète notamment à travers le type des questions : Le premier exercice de chaque section se présente sous la forme d'un QCM afin d'évaluer les connaissances assimilées par l'étudiant suite à la lecture de la tâche théorique et s'assurer qu'il part sur de bonnes bases.

Après ça, une suite d'exercices requiert des réponses courtes, de l'ordre d'un unique élément HTML ou d'une unique déclaration CSS. Ces exercices ont pour but de permettre à l'étudiant d'appréhender la syntaxe, en insérant les concepts au fur et mesure de son avancée (élément simple, élément accompagné d'un seul puis de plusieurs attributs, élément vide, ...)

Enfin, la complexité des réponses attendues augmente progressivement, jusqu'à demander à l'étudiant de remplir le corps d'un document HTML ou des règles CSS au complet.

5.3 Référence

Le contenu du cours s'inspire en grande partie de la MDN Web Docs[29], ressource connue et reconnue dans le domaine du développement web. En plus d'être constamment mise à jour[30], la MDN représente une référence idéale dans le cadre de ce mémoire : son contenu est toujours très détaillé, le rendant abordable

3. dans ce contexte, "vide" sous-entend un exercice n'attendant aucune réponse de la part de l'étudiant

et compréhensible pour un débutant. Plusieurs instances du site sont maintenues et activement développées dans différentes langues[31], dont une version en français. De plus, la MDN accorde une attention particulière à l’accessibilité. Cet aspect, bien souvent négligé et d’autant plus lorsque l’on est en période d’apprentissage, est un concept fondamental du développement web. L’importance de l’accessibilité dans la création de sites web fait partie des concepts que le cours essaye de transmettre et inculquer aux élèves.

5.4 Matières abordées

5.4.1 Bases théoriques

Comme mentionné plus tôt, aussi bien la section HTML que la section CSS commencent par une tâche théorique afin d’introduire la syntaxe de chaque langage. Les premiers exercices sont de type QCM. Ils ont été conçus pour laisser le temps à l’étudiant d’assimiler les bases.

5.4.2 Contenu de la section HTML

Éléments d’en-tête

Le premier exercice de type “code” aborde les éléments que l’on retrouve généralement au niveau de l’élément d’en-tête `<head>` : l’élément `<title>` et l’élément `<meta>`, permettant respectivement de définir le titre ainsi que le jeu d’encodage d’une page, y sont abordés. Le format de cet exercice est inspiré de ce qui est proposé sur W3Schools : plutôt que d’attendre un élément complet de la part de l’étudiant, un template est déjà inclus dans la zone textuelle INGINIOUS prévue pour accueillir sa réponse. Seule une partie de l’élément, indiquée par les caractères `[?]`, est manquante et doit être renseignée par l’étudiant afin de reformer l’élément complet.

Éléments courants

Ce second exercice fait découvrir les éléments que l’on retrouve le plus souvent au niveau du corps du document, le `<body>`. La première question introduit les éléments titre de section `<h1>`–`<h6>`. Elle se présente sous la même forme que l’exercice précédent. La deuxième question quant à elle présente l’élément paragraphe `<p>`. Un template à compléter est toujours fourni à l’étudiant, cependant il doit cette fois-ci compléter les balises manquantes de deux éléments à la fois.

Insertion de commentaires

Cet exercice introduit le concept de commentaires et attend de l'étudiant qu'il renseigne deux commentaires `<!-- -->` à des endroits précis du squelette HTML fourni. Une explication générale sur l'usage et l'intérêt des commentaires en programmation est donnée à l'étudiant à travers le feedback qui lui est retourné.

Mise en évidence

Ce quatrième exercice contient deux questions ; l'une sur l'élément d'emphase `<em>`, l'autre sur l'élément de haute importance `<strong>`. Il met en avant et rappelle la raison d'être, le rôle principal du langage HTML : structurer une page web. L'exercice insiste donc sur le fait que ces éléments ne doivent pas être utilisés pour le rendu (italique, gras) qu'ils offrent mais bien pour leur sémantique et met l'accent sur ce point en mentionnant les éléments `<i>` et `<b>` dont le rendu visuel est identique aux éléments `<em>` et `<strong>`, respectivement.

Saut de ligne

Précédée d'une brève section théorique décrivant le rendu des espaces dans du code HTML une fois interprété par le navigateur, cet exercice se veut être une suite logique et présente l'une des manières d'espacer du contenu en HTML : l'élément `<br>`. Cet exercice est aussi l'occasion de faire un rappel sur les éléments vides et leur syntaxe particulière.

Éléments de liste

Afin de faire le tour des éléments de base, les éléments `<ol>`, `<ul>` et `<li>` permettant de créer des listes ordonnées et non ordonnées, sont mentionnés. Cet exercice est le premier pour lequel aucun template n'est fourni ; l'étudiant doit créer la liste de A à Z.

Récapitulatifs

Viennent ensuite deux exercices de synthèse. Tout deux font intervenir les différents éléments vus dans le cours jusqu'à maintenant.

Le premier évalue l'étudiant sur sa maîtrise de la syntaxe et des éléments d'en-tête : le squelette type d'un document HTML rempli d'erreurs est fourni à l'étudiant. Il lui est demandé de corriger ce document afin de le rendre valide, sans y ajouter de nouveaux éléments.

Le second exercice évalue l'étudiant sur les éléments de contenu abordés jusqu'à présent dans le cours. Il présente une page web à reproduire : seul le contenu textuel

de cette page est fourni, l'étudiant doit y ajouter l'ensemble des balises manquantes afin de reproduire la page illustrée dans l'énoncé.

Attributs universels

Le concept d'attributs est abordé, plus particulièrement les attributs dits universels comme l'attribut `lang` et son usage commun au niveau de l'élément `<html>` pour déclarer la langue principale d'un document HTML.

Liens & images

Une suite d'exercices permet ensuite aux étudiants d'apprendre l'insertion de liens et d'images à une page web. Cette suite d'exercices débute avec une explication théorique du concept de "chemin" en informatique et fait la différence entre chemins/liens relatifs et absolus.

Après ce passage théorique vient ensuite un exercice de type QCM afin d'évaluer la compréhension des concepts "relatif" et "absolu" à travers des exemples concrets.

Les exercices de code font ensuite leur réapparition à travers l'utilisation de l'élément `<img>`, accompagné de ses attributs `src` et `alt` mais aussi de l'élément `<a>` et ses attributs `href` et `target`.

Un second exercice sur l'élément `<a>` est inclus afin d'illustrer la possibilité d'effectuer des redirections internes et l'intérêt que cela peut avoir dans la réalisation d'une page web.

Audio & vidéo

Pour continuer sur l'inclusion de contenu multimédia, le cours se poursuit avec les éléments `<audio>` et `<video>` et les divers attributs permettant de modifier leur affichage et comportement par défaut : `controls`, `loop`, `autoplay`.

Cadre intégré

L'élément `<iframe>`, qui permet l'intégration d'une page HTML dans une autre, est présenté au travers d'une utilisation concrète : l'intégration d'une carte interactive.

Structuration

Afin d'insister sur l'importance de la sémantique, l'utilisation des éléments `<header>`, `<main>` et `<footer>` pour structurer le contenu d'un document HTML sont abordés.

Tableaux

L'insertion de tableaux est présentée. Cet exercice permet notamment de faire travailler l'élève avec des éléments (`<table>`, `<tr>`, `<th>`, `<td>`) imbriqués sur plusieurs niveaux, semblable aux listes vues précédemment dans le cours, mais en plus complexe et avec l'ajout d'attributs pour personnaliser la structure du tableau.

Formulaire

Finalement, une suite d'exercices visant à introduire les formulaires a été ajoutée en dernier lieu. Les différents types de formulaires y sont présentés à travers les valeurs que peut prendre l'attribut `type` associé à l'élément `<input>`. Bien que nous limitant au page web statique, il me semblait être pertinent de conclure la section HTML sur ça, afin d'éventuellement éveiller la curiosité auprès des élèves désirant aller plus loin.

5.4.3 Contenu de la section CSS

Méthodes d'application

Les premiers exercices de la section CSS consiste à illustrer les différentes méthodes possibles pour appliquer du CSS à notre page web : d'abord avec le style en ligne et ensuite avec le style interne, qui permettent de faire une transition en douceur avec la section HTML, avant de passer au style externe pour le restant du cours.

Couleurs

Après cela, une suite d'exercices présente les différentes façons de renseigner une valeur à la propriété `color`. D'abord, via la façon la plus directe, avec des couleurs nommées. Ensuite, via des méthodes plus complexes mais permettant d'affiner le choix de la couleur, en commençant par la notation hexadécimale. Afin de rendre cette notion compréhensible pour les élèves, une section théorique intermédiaire est incluse dans le cours afin de les aider à appréhender cette notation. Finalement, via la notation fonctionnelle `rgb()` et les différentes valeurs quelle accepte (décimales ou pourcentage).

Polices d'écriture

La suite du cours porte sur la personnalisation de la police d'écriture, d'abord à travers la propriété `font-family` et les multiples valeurs que celle-ci peut prendre afin d'assurer une compatibilité accrue avec les différents navigateurs. La taille de

la police d'écriture est ensuite abordée. L'exercice présente les différentes unités de mesure et fait la distinction entre unité absolue et relative en relevant l'importance pour l'accessibilité d'une page web de bien choisir entre ces deux catégories d'unités.

Modèle de boîte

La section CSS se conclue avec l'un des concepts les plus importants du langage : le modèle de boîte[32]. Cela se traduit à travers une série d'exercices qui présentent tour à tour les propriétés `border`, `width & height`, `padding` et `margin`, avant d'être repris ensemble pour définir le modèle de boîte CSS.

Chapitre 6

Evaluation des exercices

La plateforme INGINious a été développée dans l’optique d’évaluer des exercices de programmation. HTML et CSS n’étant pas des langages de programmation à proprement parler, il était nécessaire d’établir un système d’évaluation qui leur est propre, différent de ce qui se fait pour les cours INGINious existants.

Ce chapitre aborde l’aspect technique du cours et a pour but de décrire la solution mise en place spécifiquement pour les langages HTML et CSS.

Cette solution fait passer la soumission d’un exercice par trois grandes phases : la validation, l’analyse syntaxique et enfin la correction. Ces trois étapes sont détaillées ci-dessous et illustrées à travers un exercice du cours[33]. Cet exercice est composé de deux sous-problèmes ; l’un attendant un élément unique en réponse, l’autre un ensemble de plusieurs éléments. Leur identifiant respectif sont `h1_tag` et `p_tags`. Les énoncés de ces deux exercices sont disponibles en annexe A.

Le workflow général utilisé est le même pour les deux langages mais s’avère légèrement plus complexe en HTML, ce pour quoi l’exercice choisi ici fait donc partie de la section HTML. Cinq types de réponses seront considérés :

1. Une réponse non valide au premier sous-problème
2. Une réponse non valide au second sous-problème
3. Une réponse valide au second-sous-problème, mais contraire aux bonnes pratiques
4. Une réponse valide et correcte au premier sous-problème, mais incorrecte au second
5. Une réponse valide et correcte au second-sous-problème, mais incorrecte au second

6.1 Validation

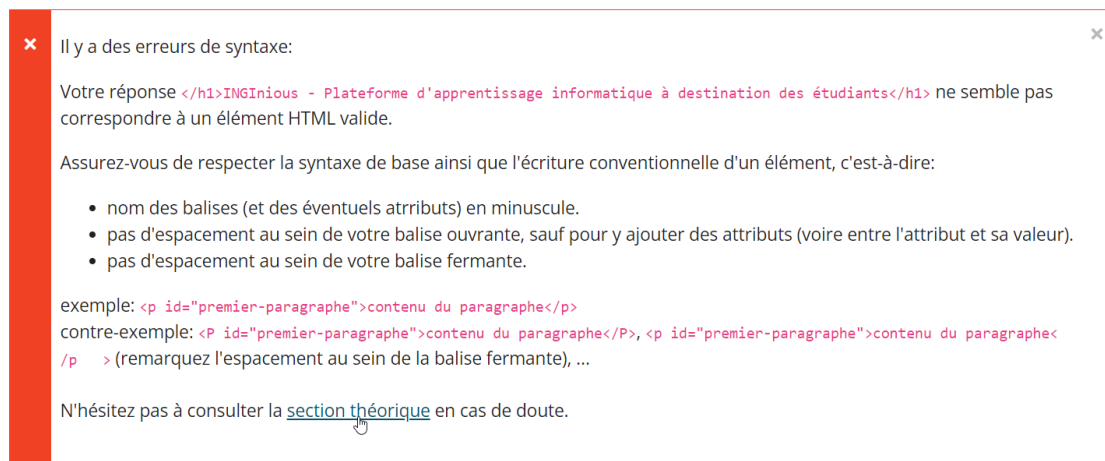
Avant même de procéder à une quelconque correction de la soumission d'un étudiant, il faut au préalable s'assurer que sa réponse soit **valide**, autrement dit qu'elle respecte la syntaxe du langage. Pour ce faire, le cours INGINious repose sur les deux outils fournis par le World Wide Web Consortium mentionnés précédemment. Afin de pouvoir traiter et valider ou invalider du code, les validateurs attendent un fichier en entrée. Pour cette raison, le code de l'étudiant n'est pas passé en brut mais est d'abord récupéré et inclus dans un fichier `template[.html/.css]`.

Le validateur CSS se suffit à lui-même et permet de gérer l'ensemble des cas de figure abordés dans le cours. L'étape de validation du HTML est quant à elle plus complexe. Deux méthodes sont utilisées selon le type de question. Pour être notamment capable de proposer des exercices sur les éléments d'en-tête (`<head>`) en première partie du cours, une méthode basée sur des expressions régulières a été développée. Expressions régulières et HTML ne font pas bon ménage en raison de la complexité syntaxique du langage, comme le fait comprendre cette réponse originale[34], devenue populaire sur le forum Stackoverflow. Cette méthode reste toutefois envisageable lorsque le contexte est restreint à un sous-ensemble du langage ; dans notre cas à un unique élément HTML. Lorsqu'une soumission au sous-problème `h1_tag` est effectuée avec le code erroné suivant :

```
1  </h1>INGInious - Plateforme (...) des étudiants</h1>
```

FIGURE 6.1 – Balise fermante en lieu et place de la balise ouvrante

Le système va vérifier que la réponse de l'étudiant représente une chaîne de caractères conforme à l'expression régulière, conçue pour correspondre à un élément HTML bien formé. Ce n'est pas le cas ici : la première composante de l'élément HTML est une balise fermante et non une balise ouvrante. Le feedback suivant est retourné à l'utilisateur :



Cette solution est donc utilisée pour les premiers exercices de la section HTML et permet de valider un seul élément HTML à la fois.

La seconde méthode est similaire à la validation du CSS : le code de l'étudiant est transféré dans un fichier `template.html`, à la différence toutefois que celui-ci n'est pas vide au préalable. Cette différence s'explique par ce qui constitue un document HTML valide et bien formé. Le minimum requis est le squelette suivant ¹ :

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <title>A simple HTML skeleton handling student response</title>
5   </head>
6   <body>
7     @@problem_id@@
8   </body>
9 </html>
```

FIGURE 6.2 – Squelette d'un document HTML

Sans ce squelette inclus dans le template, les soumissions seraient systématiquement invalidées. Cette solution est utilisée pour les questions plus avancées de la section HTML, où il est demandé à l'étudiant de renseigner un ensemble d'éléments HTML dans le corps (`<body>`) du document, afin de contourner cette problématique sans constamment attendre un document HTML complet en réponse. Lorsqu'une soumission au sous-problème `p_tags` est effectuée avec le code erroné suivant :

1. “@@problem_id@@” est un identifiant utilisé par INGINIOUS et n'est pas lié à la syntaxe HTML

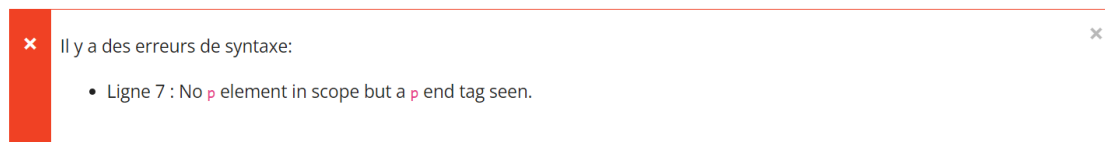
```

1 <p>
2   La plateforme INGIInious est créée et maintenue par (...)
3   Entièrement libre d'accès, son code-source est (...)
4 </p>
5   Quand un étudiant soumet une réponse à un exercice (...)
6   Grâce à ces tests, l'étudiant peut rapidement vérifier si (...)
7 </p>

```

FIGURE 6.3 – Omission de la balise ouvrante du deuxième paragraphe

Le code est inséré dans le fichier template, à l'intérieur du `<body>` à la place du champ réservé, délimité par les caractères “@@” que l'on retrouve à la figure 6.2. Le fichier est ensuite passé au validateur pour analyse. Le résultat de son exécution est traité avant d'être retourné à l'utilisateur :



Si toutefois l'utilisateur avait soumis la réponse suivante à cette même question :

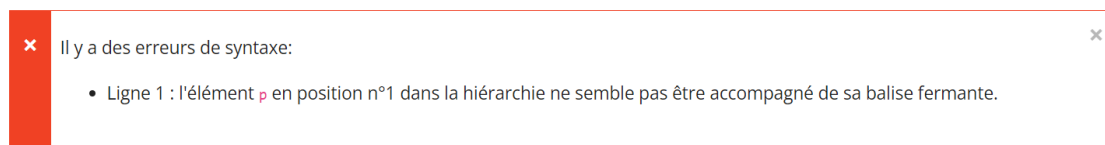
```

1 <p>
2   La plateforme INGIInious est créée et maintenue par (...)
3   Entièrement libre d'accès, son code-source est (...)
4 <p>
5   Quand un étudiant soumet une réponse à un exercice (...)
6   Grâce à ces tests, l'étudiant peut rapidement vérifier si (...)
7 </p>

```

FIGURE 6.4 – Omission de la balise fermante du premier paragraphe

le validateur n'aurait relevé aucune erreur, et la réponse de l'étudiant serait passée avec succès à l'étape suivante. En effet, les validateurs respectent les spécifications HTML, même non normatives, qui sont en vigueur. Celles-ci autorisent pour quelques éléments l'omission de balises sous certaines conditions[35] et l'élément `p` fait partie de ceux-ci. Ce comportement n'étant ici pas désiré, une étape de validation supplémentaire a été ajoutée pour éviter que ce genre de solution soit acceptée. Cette étape consiste simplement à s'assurer que chaque balise ouvrante est bien accompagnée de sa balise fermante. Si ce n'est pas le cas, le feedback suivant est retourné à l'utilisateur :



Pour chacun de ces trois cas de figure type, l'exécution du script de correction se termine après avoir retourné le feedback à l'utilisateur. Autrement, le processus de correction se poursuit et passe à la deuxième étape : l'analyse syntaxique.

6.2 Analyse syntaxique

Une fois que l'on s'est assuré d'avoir une soumission syntaxiquement correcte, nous pouvons traiter le code de manière plus approfondie : analyser ses composantes sous forme de “tokens”.

Pour illustrer cette section ainsi que l'étape de correction, les réponses incorrectes suivantes seront considérées :

```
1 <h2>INGInious - Plateforme (...) des étudiants</h2>
```

FIGURE 6.5 – Élément h2 en lieu et place de l'élément h1

```
1 <p>
2   La plateforme INGInious est créée et maintenue par (...)
3   Entièrement libre d'accès, son code-source est (...)
4
5   Quand un étudiant soumet une réponse à un exercice (...)
6   Grâce à ces tests, l'étudiant peut rapidement vérifier si (...)
7 </p>
```

FIGURE 6.6 – Omission d'un élément paragraphe

L'analyse itère sur chaque élément HTML/règle CSS et récupère les différents tokens qui le composent, à savoir :

- HTML : le nom de l'élément, les noms et les valeurs de ses attributs potentiels, ou encore le contenu qui lui est associé.
- CSS : le sélecteur de la règle ou encore les propriétés et valeurs de ses déclarations.

Ces tokens sont stockés dans des structures de données. L'agencement de ses structures est détaillé en annexe B.

Voici à quoi cela ressemblerait dans notre exemple :

```
{
  "h1_tag": [
    {
      'element_name': 'h2',
      'element_attr': {},
      'element_text': "INGInious - Plateforme (...) des étudiants"
    }
  ],
  "p_tags": [
    {
      'element_name': 'p',
      'element_attr': {},
      'element_text': "La plateforme INGInious est créée et maintenue par (...) \
Entièrement libre d'accès, son code-source est (...) \
Quand un étudiant soumet une réponse à un exercice (...) \
Grâce à ces tests, l'étudiant peut rapidement vérifier si (...)"
    }
  ]
}
```

FIGURE 6.7 – Structure générée à la fin de l'analyse syntaxique

Une fois la structure construite, l'exécution suit son cours avec la troisième étape.

6.3 Correction

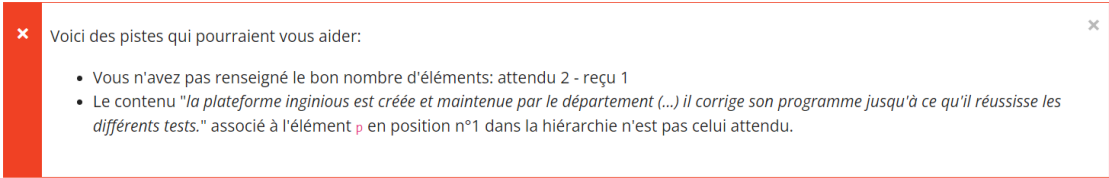
Maintenant que nous avons récupéré et réagencé la réponse de l'étudiant dans une structure prédéfinie, nous pouvons la comparer, élément par élément, avec une structure pré-remplie faisant office de réponse attendue :

```
{
  "h1_tag": [
    {
      'element_name': 'h1',
      'element_attr': {},
      'element_text': "inginius - plateforme (...) des étudiants"
    }
  ],
  "p_tags": [
    {
      'element_name': 'p',
      'element_attr': {},
      'element_text': "la plateforme inginius est créée et maintenue par (...) \
entièrement libre d'accès, son code-source est (...)"
    },
    {
      'element_name': 'p',
      'element_attr': {},
      'element_text': "quand un étudiant soumet une réponse à un exercice (...) \
grâce à ces tests, l'étudiant peut rapidement vérifier si (...)"
    }
  ]
}
```

FIGURE 6.8 – Structure attendue et utilisée pour la comparaison

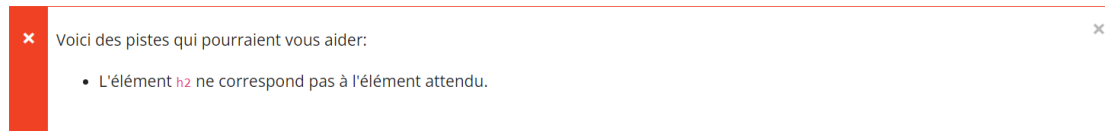
Une première itération est effectuée sur la structure obtenue à l'étape précédente, afin de normaliser et évaluer les éléments correctement renseignés par l'utilisateur, jusqu'à ce que la première erreur soit éventuellement rencontrée. Si tel est le cas, le feedback retourné à l'utilisateur indiquera tout d'abord les X premiers éléments correctement renseignés.

La deuxième indication fournie à l'utilisateur dépendra du nombre d'éléments qui composent sa réponse : si cela ne correspond pas au nombre d'éléments effectivement attendus, l'étudiant est informé de cette différence :

Voici des pistes qui pourraient vous aider:

- Vous n'avez pas renseigné le bon nombre d'éléments: attendu 2 - reçu 1
- Le contenu "la plateforme inginius est créée et maintenue par le département (...) il corrige son programme jusqu'à ce qu'il réussisse les différents tests." associé à l'élément **p** en position n°1 dans la hiérarchie n'est pas celui attendu.

Sinon, les éléments de l'étudiant sont comparés un par un avec ceux de la réponse attendue sur base des différents tokens. Le premier token évalué est le nom de l'élément. Si celui-ci diffère, comme c'est le cas de la réponse illustrée à la figure 6.5, l'utilisateur en est averti :



et l'itération continue avec le prochain élément à comparer. Si toutefois l'élément s'avérait être celui attendu, alors ses attributs ainsi que son contenu seraient évalués, et le feedback généré en conséquence, avant de passer à l'élément suivant.

6.4 Conclusion

Les validateurs permettent d'assurer une correction plus aisée et efficace d'un exercice. En effet, ces derniers nous évitent d'avoir à traiter des réponses (trop) mal formées, ce qui réduirait drastiquement la fiabilité de l'analyse syntaxique, voire la rendrait tout simplement impossible.

En contrepartie, cette décision implique une exigence accrue auprès des étudiants, particulièrement en ce qui concerne HTML : de part la souplesse du langage, le code qu'il soumet pourrait leur sembler correct et fonctionnel en local, sans être valide pour autant.

Aux premiers abords, cette solution stricte pourrait paraître inadaptée dans le cadre d'un cours se voulant être une initiation au développement web. Contrairement à un langage de programmation, lorsqu'un navigateur affiche une page HTML, il va essayer de corriger les erreurs de lui-même, ou simplement ignorer l'élément en erreur. De même avec le CSS, lorsqu'une règle invalide est rencontrée, le navigateur va simplement l'ignorer sans appliquer de style. Comme le relève une étude portée sur les erreurs effectuées en HTML/CSS[8], ce comportement rend d'autant plus difficile la détection de cesdites erreurs et complexifie davantage l'adoption de bonnes pratiques pour un débutant.

Chapitre 7

Mise en condition réelle

Ce chapitre aborde la période d’essai du cours, effectuée en parallèle de son développement et durant laquelle les élèves de M^{me} Belin ont eu accès à une série d’exercices. L’ensemble des résultats du sondage sont inclus en annexe C et synthétisé sous forme de tableau dans ce chapitre. Ci-dessous sont détaillés les résultats ainsi que les conséquences tirées de tout cela.

7.1 Déroulement

Étant donné que les élèves utilisent déjà Teams comme vecteur de communication dans le cadre de leur cursus scolaire, ils nous a semblé pertinent à M^{me} Belin et moi-même de mettre en place une discussion Teams dédiée au cours INGINIOUS afin de regrouper les élèves et faciliter les échanges. Cette discussion était aussi l’occasion de mettre à disposition un espace réservé dans le but de centraliser les éventuelles questions en tout genre des étudiants, afin que ceux-ci puissent interagir aussi bien entre eux qu’avec moi, en partageant les difficultés auxquelles ils font face.

Suite à cette expérience, les élèves ont été invités à répondre à un sondage de façon anonyme, comprenant les 8 questions suivantes, présentées sous la forme d’échelle de notation, chacune allant de 1 à 5 :

1. *Le cours a-t-il répondu à vos attentes en terme d’apprentissage ? (1 : non – 5 : oui)*
2. *Avez-vous trouvé le contenu du cours clair / facile à comprendre ? (1 : non – 5 : oui)*
3. *Les exercices proposés étaient-ils pertinents et adaptés à votre niveau de connaissances ? (1 : non – 5 : oui)*

- 4.1 *Le système d'évaluation automatique vous a-t-il semblé utile / efficace sur l'aspect « détection des erreurs » : les feedbacks en cas de mauvaises réponses vous ont-ils aidés à détecter / corriger vos erreurs ? (1 : non – 5 : oui)*
- 4.2 *Le système d'évaluation automatique vous a-t-il semblé utile / efficace sur l'aspect « informations additionnelles » : les feedbacks en cas de bonnes réponses (les “Tips”) vous ont-ils appris quelque chose ? (1 : non – 5 : oui)*
5. *Y a-t-il des aspects du cours qui mériteraient selon vous d'être retravaillés / repensés / améliorés ? (1 : oui – 5 : non)*
6. *Y a-t-il des thématiques que vous auriez aimé voir abordées à travers le cours mais qui n'y figuraient pas ? (1 : oui – 5 : non)*
7. *Globalement, recommanderiez-vous ce cours à un ami / collègue ? (1 : non – 5 : oui)*

A la suite de cette série de questions se trouvaient deux espaces textuels facultatifs : le premier pour justifier et développer la/les raison(s) des notes attribuées, le second pour partager un retour plus global vis-à-vis du cours, qui ne serait pas directement lié aux questions du sondage :

8. *Utilisez cette zone textuelle pour développer / argumenter vos réponses aux questions précédentes (soit de manière concise et générique en quelques phrases/paragraphes, soit de manière détaillée question par question)*
9. *Finalement, avez-vous des commentaires / suggestions supplémentaires à faire aux personnes en charge de ce cours ?*

7.2 Résultats

Parmi les élèves testeurs, 7 ont répondu au sondage.
Voici un tableau résumant les notes obtenues pour les différentes questions :

	1	2	3	4	5
Q1 : Réponse aux attentes	0	0	1 (14,3%)	0	6 (85,7%)
Q2 : Clareté	0	0	0	5 (71,4%)	2 (28,6%)
Q3 : Pertinence	0	0	1 (14,3%)	0	6 (85,7%)
Q4.1 : Détection des erreurs	0	1 (14,3%)	1 (14,3%)	3 (42,9%)	2 (28,6%)
Q4.2 : Tips	0	0	2 (28,6%)	3 (42,9%)	2 (28,6%)
Q5 : Potentiel d'améliorations	0	1 (14,3%)	5 (71,4%)	0	1 (14,3%)
Q6 : Attentes perso. atteintes	0	0	1 (14,3%)	0	6 (85,7%)
Q7 : Recommandable	0	0	1 (14,3%)	2 (28,6%)	4 (57,1%)

Les notes démontrent une satisfaction globale des élèves vis-à-vis du cours INGINIOUS. On observe toutefois un résultat mitigé en ce qui concerne la qualité du feedback, tendance que l'on retrouve à travers deux des commentaires laissés à l'avant dernière question facultative du sondage :

- Elève A : *“Pour la question 4.1 : je trouve que les feedback ne m’ont pas aidés pour détecter les erreurs. Je ne comprenais pas les explications que ça me donnait quand j’avais une erreur. Je trouve que c’était un peu vague ? Je ne sais pas comment l’expliquer.”*
- Elève B : *“Tout était plutôt clair et facile à comprendre, la seule chose que j’ai pu voir comme un problème est la correction auto qui affichait parfois mauvais alors que le code était correct.”*

Il me semble intéressant d’essayer de comprendre les causes/raisons qui ont menés ces deux élèves à cet avis. Cette démarche est entreprise à la section 7.3.

Finalement, un troisième élève relève une insatisfaction relative à la section du cours abordant l’hexadécimal :

- Elève C : *“Pour la question 5 : ce serait d’améliorer la partie hexadécimal.”*

7.3 Observations

En observant les tentatives de soumissions échouées effectuées par les élèves testeurs, j’ai pu me rendre compte que la source de leurs erreurs provenait avant tout d’une incompréhension de la question initiale, due à une ambiguïté laissée par l’exercice ou simplement un manque de clarté de l’énoncé.

Prenons par exemple la première question de type “code” présent dans la section CSS. Cet exercice aborde le style “en ligne” permettant d’appliquer des déclarations CSS à un élément HTML spécifique avec l’attribut `style`. La zone de réponse est pré-remplie avec le template suivant :

```
<h1 style="[?]">Le titre de votre site est de couleur:</h1>
```

L’objectif de cette question étant de remplacer l’espace réservé ([?]) par la déclaration CSS adéquate, afin d’appliquer au choix la couleur rouge, verte ou bleue au titre de premier niveau `h1`.

L’un des élèves a interprété les crochets comme étant des caractères à part entière de l’élément HTML. Voici l’une de ses premières soumissions :

```
<h1 style="[red]">Le titre de votre site est de couleur:</h1>
```

Et voici sa huitième tentative :

```
<h1 style="[color: blue]">Le titre de votre site est de couleur:</h1>
```

Bien que ses précédents échecs soient effectivement des erreurs liées à la syntaxe CSS, ce n'est pas réellement le cas pour cette soumission. Crochets mis à part, l'élève a bien renseigné une déclaration CSS valide, déclaration attendue en réponse à cet exercice.

En outre, la rigidité du système de correction s'est avéré contre-productif pour un cas de figure spécifique, en étant trop restrictif dans les réponses considérées “correctes” attendues. Ce cas de figure concerne la question abordant la notation `RGB()` en CSS. La question autorise volontairement de répondre avec différente syntaxe de la fonction `RGB()` :

- Notation “obsolète” : `rgb(0, 100, 255[, 0.5])`
- Notation “moderne” : `rgb(0 100 255[ / 0.5])`

Bien qu'obsolète, la notation à virgule est interprétée comme un alias de la notation moderne et reste donc valide. Cependant, voici le résultat d'une soumission effectuée par un étudiant utilisant la notation à virgule :

```
1 h1 {  
2   color: rgb(100%,50%,25%);  
3 }
```

• Ligne 2: La valeur `rgb(100%,50%,25%)` associée à la propriété `color` est incorrecte.

(a) Réponse soumise

(b) Feedback généré

Les valeurs 100%, 50%, 25% étaient bien celles attendues en réponse à cet exercice. Toutefois, seules les valeurs accompagnées d'un espace après la virgule étaient acceptées pour la notation obsolète.

7.4 Améliorations

Plusieurs aspects du cours ont pu être perfectionnés pendant et suite à cette période d'essai à travers des corrections sur le contenu existant mais aussi des ajouts.

Tout d'abord, un bug dans la génération du feedback associé aux exercices CSS m'a été signalé par M^{me} Belin, après en avoir fait le constat avec ses élèves. Dans certains cas de figure, les paires propriétés-valeurs se retrouvaient mélangées entre elles :

<pre> 1 header { 2 width: 50%; 3 background-color: red; 4 } 5 h1 {width: 75%; 6 border: dashed blue; 7 } </pre>	• Ligne 3: La valeur <code>50%</code> associée à la propriété <code>background-color</code> est incorrecte. • Ligne 2: La valeur <code>red</code> associée à la propriété <code>width</code> est incorrecte. • Ligne 6: La valeur <code>75%</code> associée à la propriété <code>border</code> est incorrecte. • Ligne 5: La valeur <code>dashed blue</code> associée à la propriété <code>width</code> est incorrecte.
---	--

(a) Réponse soumise

(b) Feedback généré

En outre, suite aux observations mentionnées à la section précédente, des modifications ont été apportées au niveau des énoncés pour limiter les malentendus. Comme indiqué précédemment, le système de correction a été conçu pour préserver au mieux le format des soumissions, telles que rédigées par l'étudiant. Cela engendre certaines contraintes techniques, notamment l'impossibilité de récupérer les composantes de couleur RGB une à une (autrement, celles-ci seraient automatiquement converties par l'analyseur syntaxique et normalisées en nombres décimaux). Pour cette raison, le système de correction est rigide et a été conservé en l'état. Plutôt que d'élargir les réponses acceptées, cette décision a été prise afin de rester cohérent avec la direction du cours. Étant donné son usage pour représenter les couleurs en CSS, l'hexadécimal semble être un concept important à aborder dans le cadre d'une introduction à ce langage. Cette notation n'est toutefois pas évidente à appréhender pour les élèves, comme me l'a indiqué M^{me} Belin à juste titre. Un exercice théorique a été déployé afin d'introduire le système hexadécimal, accompagné d'un QCM pour entraîner les étudiants à la conversion Base 10 $\leftrightarrow$ Base 16 et évaluer leur maîtrise d'un concept probablement nouveau pour une majorité d'entre eux.

7.5 Conclusion

Au-delà des résultats du sondage, j'ai pris conscience en réalisant cette expérience que la rédaction d'exercices est un réel défi, plus complexe qu'il n'y paraît. Cela demande rigueur et précision ; un exercice clair à mes yeux ne l'est pas forcément pour les élèves. Cette clarté est un élément clé à la bonne réalisation d'un cours adapté aux étudiants, d'autant plus lorsqu'il s'agit d'un cours introductif. De fait, la philosophie volontairement stricte adoptée pour ce cours s'est avérée contre-productive par moments, de part des énoncés ambigus ou des restrictions trop importantes en ce qui concerne les réponses acceptées.

Chapitre 8

Pistes d'améliorations

Dans l'éventualité d'une extension ou simplement d'une reprise du travail né de ce mémoire par de futurs mémorants, des évolutions sont envisageables sur plusieurs aspects.

Tout d'abord, la forme du cours pourrait être repensée de diverses façons : Soit en laissant de côté l'aspect théorique et en se focalisant exclusivement sur des exercices pratiques, ce pour quoi la plateforme INGINIOUS est prévue à la base. Soit à l'inverse, étendre l'aspect théorique en rédigeant un Syllabus en ligne qui viendrait accueillir les exercices INGINIOUS en son sein.

En outre, le temps consacré au développement d'un système de correction et à la génération de feedback représente une part importante dans l'élaboration de ce mémoire, au détriment de la quantité d'exercices reposant dessus. Le contenu du cours pourrait donc être étendu avec une série d'exercices d'un niveau de difficulté plus élevé ou abordant des concepts plus avancés, afin d'élargir le public cible aux élèves du 3^e degré du secondaire voire aux étudiants en première année de l'enseignement supérieur.

Sur un aspect plus technique, une interprétation dynamique du feedback généré par les validateurs HTML/CSS et retourné aux utilisateurs en cas d'erreur syntaxique pourrait être envisagée afin d'en obtenir une traduction en français et/ou plus compréhensible pour l'utilisateur, particulièrement pour des élèves du secondaire.

Enfin, une collaboration avec d'autres plateformes belges existantes, telles que CoderDojo[36] ou encore Technofutur TIC[18], pourrait être envisagée. Par exemple, une intégration du cours aux formations d'initiation au développement web proposé sur la plateforme Technofutur avait été discutée dans le cadre de ce travail mais l'idée est finalement restée sans suite.

Conclusion

L'enseignement secondaire fait face à diverses difficultés, qui se traduisent notamment dans les cours d'informatique par un manque de référentiels à disposition des enseignants, et de façon similaire un manque de contenu pouvant répondre à leurs attentes. La complexité à évaluer les connaissances pratiques en informatique explique en partie ce manque. L'apprentissage d'un langage informatique est toutefois peu envisageable sans passer par de la pratique. Ce mémoire a vu le jour afin de subvenir partiellement à ce besoin, en proposant un cours INGINious constitué d'une suite d'exercices en HTML/CSS.

Afin de pouvoir tirer profit du système d'évaluation automatique proposé par INGINious, le développement d'une solution permettant la prise en charge des langages HTML et CSS était nécessaire. Un système d'évaluation propre à ces langages a donc vu le jour. Ce système fait passer la soumission d'une réponse à un exercice par trois grandes étapes : la validation, l'analyse syntaxique et enfin la correction.

Une première version du cours reposant sur ce système d'évaluation a été proposée à M^{me} Bellin, enseignante en informatique du 2^e degré, et à ses élèves, afin de le mettre à l'épreuve. Suite à cela, aussi bien des améliorations que des ajouts ont pu être apportés afin de compléter le cours. Cette phase de test aura également permis plusieurs observations en ce qui concerne les difficultés rencontrées lors de la réalisation des exercices par les étudiants. Les erreurs qu'ils commettent ne sont pas toujours celles auxquelles on pourrait s'attendre.

Les résultats du sondage soumis aux élèves testeurs révèlent une satisfaction globale du cours, répondant à leurs attentes en terme d'apprentissage, et de façon claire. Bien qu'il ait évolué entre temps, le sondage indique également que le cours reste perfectible.

Ce mémoire s'étant avant tout focalisé sur le système de correction, la matière parcourue et la difficulté proposée ne suffit pas à répondre à tous les cycles du secondaire. Au vu de l'enthousiasme provenant aussi bien de l'enseignante que de ses élèves, une continuité à ce travail pourrait être apportée afin de proposer du contenu répondant aux attentes d'un public plus large.

Bibliographie

- [1] Fédération WALLONIE-BRUXELLES. *Pacte pour un enseignement d'excellence*. URL : <https://pactepourunenseignementdexcellence.cfwb.be>.
- [2] Fédération WALLONIE-BRUXELLES. *Stratégie numérique pour l'éducation*. URL : <http://www.enseignement.be/index.php?page=28101>.
- [3] Marie-Hélène COMTE et al. “*Un espace de formation francophone dédié à l'apprentissage de l'informatique*”. Didapro9 2022. Mai 2022. URL : <https://inria.hal.science/hal-03643914>.
- [4] Julie HENRY et Noémie JORIS. “*Informatics at secondary schools in the French-speaking region of Belgium : myth or reality ?*” English. In : *. The International Conference on Informatics in Schools : Situation, Evolution and Perspectives ; Conference date : 13-10-2016 Through 15-10-2016. Oct. 2016. URL : <http://issep2016.ens-cachan.fr/>.
- [5] Rachel GARDNER. “*How Isaac is helping students learn Computer Science from home*”. In : (2020). URL : <https://www.cst.cam.ac.uk/news/how-isaac-helping-students-learn-computer-science-home>.
- [6] Jane WAITE et al. “*An Online Platform for Teaching Upper Secondary School Computer Science*”. In : *Proceedings of the 2021 Conference on United Kingdom & Ireland Computing Education Research*. UKICER '21. Glasgow, United Kingdom : Association for Computing Machinery, 2021. ISBN : 9781450385688. DOI : 10.1145/3481282.3481287. URL : <https://doi.org/10.1145/3481282.3481287>.
- [7] Benjamin SIMON. “*INGInious High School : adapter INGInious aux besoins des enseignants du secondaire*”. Prom. : Mens, Kim ; Goletti, Olivier. Ecole polytechnique de Louvain, Université catholique de Louvain, 2020. URL : <http://hdl.handle.net/2078.1/thesis:25171>.
- [8] Thomas H. PARK et al. “*Towards a Taxonomy of Errors in HTML and CSS*”. In : *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*. ICER '13. San Diego, San California, USA : Association for Computing Machinery, 2013, p. 75-82.

ISBN : 9781450322430. DOI : 10.1145/2493394.2493405. URL : <https://doi.org/10.1145/2493394.2493405>.

- [9] COURSERA. URL : <https://www.coursera.org>.
- [10] UDEMY. URL : <https://www.udemy.com/fr/>.
- [11] FREECODECAMP. URL : <https://www.freecodecamp.org>.
- [12] UDACITY. URL : <https://www.udacity.com>.
- [13] W3SCHOOLS. URL : <https://www.w3schools.com>.
- [14] W3SCHOOLS. *HTML exercises*. URL : <https://www.w3schools.com/html/exercise.asp>.
- [15] W3SCHOOLS. *CSS exercises*. URL : <https://www.w3schools.com/css/exercise.asp>.
- [16] CODECADEMY. URL : <https://www.codecademy.com>.
- [17] KHAN ACADEMY. URL : <https://fr.khanacademy.org>.
- [18] Technofutur TIC. URL : <http://www.technofuturtic.be>.
- [19] Equipe INGINIOUS. *INGInious' documentation*. URL : <https://docs.inginiuous.org/en/latest/>.
- [20] Equipe INGINIOUS. *Courses made with INGINious*. URL : <https://github.com/UCL-INGI/INGInious-courses>.
- [21] Equipe INGINIOUS. *The inginiuous_container_api package (commands inside containers)*. URL : https://docs.inginiuous.org/en/latest/teacher_doc/inginiuous_container_api.html.
- [22] World Wide Web CONSORTIUM. URL : <https://github.com/validator/validator>.
- [23] World Wide Web CONSORTIUM. URL : <https://github.com/w3c/css-validator>.
- [24] Leonard RICHARDSON. URL : <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.
- [25] Christof HOEKE et Jason R. COOMBS. URL : <https://cssutils.readthedocs.io/en/stable/>.
- [26] Stefan BEHNEL, Martijn FAASSEN et Ian Bicking et AL. URL : <https://lxml.de/lxmlhtml.html>.
- [27] Simon SAPIN. URL : <https://doc.courtbouillon.org/tinycss2/v1.0.0/>.
- [28] Jason R. COOMBS. URL : <https://github.com/jaraco/cssutils/issues/29>.

- [29] MDN Web DOCS. *Resources for Developers, by Developers*. URL : <https://developer.mozilla.org/en-US/>.
- [30] MDN Web DOCS. *MDN Content Repository*. URL : <https://github.com/mdn/content>.
- [31] MDN Web DOCS. *MDN Translated Content Repository*. URL : <https://github.com/mdn/translated-content>.
- [32] MDN Web DOCS. *The box model*. URL : https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model.
- [33] Cordier CLÉMENT. *[TFE] HTML-CSS, éléments courants*. URL : https://inginius.org/course/html_css_clement/common_tags.
- [34] BOBINCE. *RegEx match open tags except XHTML self-contained tags*. URL : <https://stackoverflow.com/a/1732454>.
- [35] WHATWG. *HTML Living Standard – Section 13.1.2.4 Optional tags*. URL : <https://html.spec.whatwg.org/#optional-tags>.
- [36] CoderDojo BELGIUM. URL : <https://coderdojobelgium.be/fr>.

Annexe A

Exercices HTML servant d'illustration

Question 1: Élément « titre de section »

Il existe 6 variantes de cet élément: **h1**, **h2**, ..., **h6**.

Ceux-ci correspondent aux titres de sections, sous-sections, ... de votre document.

Chacun dénote un niveau d'importance décroissant, **h1** ayant la plus haute importance.

Remplacez le **[?]** par la balise adéquate dans le template ci-dessous.

HTML

43

EDIT ON CODEPEN

Result

LIVE

Resources

Rerun

[?]INGInious - Plateforme d'apprentissage informatique à destination des étudiants

[?]INGInious - Plateforme d'apprentissage informatique à destination des étudiants

[?]INGInious - Plateforme d'apprentissage informatique à destination des étudiants</h1>

Question 2: Élément « paragraphe »

Remplacez les [?] par les balises adéquates dans le template ci-dessous.

HTML

Result

EDIT ON CODEPEN

1x

0.5x

0.25x

Re-run

LIVE

<p>
 La plateforme INGINIOUS est créée et maintenue par le département d'ingénierie informatique de l'UCLouvain.
 Entièrement libre d'accès, son code-source est disponible sous licence open-source. [?]
 Quand un étudiant soumet une réponse à un exercice de programmation, INGINIOUS exécute automatiquement le programme écrit par l'étudiant et le soumet à une batterie de tests définis par l'enseignant. Grâce à ces tests, l'étudiant peut rapidement vérifier si sa réponse est correcte et sinon il corrige son programme jusqu'à ce qu'il réussisse les différents tests.

Quand un étudiant soumet une réponse à un exercice de programmation, INGINIOUS exécute automatiquement le programme écrit par l'étudiant et le soumet à une batterie de tests définis par l'enseignant.
 Grâce à ces tests, l'étudiant peut rapidement vérifier si sa réponse est correcte et sinon il corrige son programme jusqu'à ce qu'il réussisse les différents tests.

Resources

1

2

3

4

5

6

7

8

<p>

La plateforme INGINIOUS est créée et maintenue par le département d'ingénierie informatique de l'UCLouvain.

Entièrement libre d'accès, son code-source est disponible sous licence open-source.

[?]

[?]

Quand un étudiant soumet une réponse à un exercice de programmation, INGINIOUS exécute automatiquement le programme écrit par l'étudiant et le soumet à une batterie de tests définis par l'enseignant.

Grâce à ces tests, l'étudiant peut rapidement vérifier si sa réponse est correcte et sinon il corrige son programme jusqu'à ce qu'il réussisse les différents tests.

</p>

1

2

3

4

5

6

7

8

<p>

La plateforme INGINIOUS est créée et maintenue par le département d'ingénierie informatique de l'UCLouvain.

Entièrement libre d'accès, son code-source est disponible sous licence open-source.

[?]

[?]

Quand un étudiant soumet une réponse à un exercice de programmation, INGINIOUS exécute automatiquement le programme écrit par l'étudiant et le soumet à une batterie de tests définis par l'enseignant.

Grâce à ces tests, l'étudiant peut rapidement vérifier si sa réponse est correcte et sinon il corrige son programme jusqu'à ce qu'il réussisse les différents tests.

</p>

Annexe B

Structure de données utilisées pour la correction

HTML

```
{
  "task_id1": [
    {
      'element_name': "elem_name1",
      'element_attr': {},
      'element_text': "some_text"
    },
    {
      'element_name': "elem_name2",
      'element_attr': {"attr_name1": "attr_value1", ...},
      'element_text': "some_text"
    },
  ],
  "task_id2":
    ...
}
```

Chaque élément dans la liste est représenté sous la forme d'un dictionnaire avec trois clés :

1. `element_name` : contient le nom de l'élément HTML.
2. `element_attr` : contient un dictionnaire (éventuellement vide) dont les paires clé-valeur correspondent respectivement aux attributs et aux valeurs qui leur sont associées.
3. `element_text` : contient une chaîne de caractères (éventuellement vide) correspondant au contenu de l'élément HTML.

CSS

```
{
  "task_id1":
    [
      ("selector_name", {
        "decl1_property": "decl1_value",
        "decl2_property": {"decl2_value1", "decl2_value2"}
      })
    ],
  "task_id2":
    ...
}
```

Chaque élément dans la liste est représenté sous la forme d'un tuple avec deux éléments :

1. Le premier élément correspond au sélecteur de la règle CSS.
2. Le deuxième élément est un dictionnaire dont les paires clé-valeur correspondent respectivement aux propriétés et aux valeurs qui leur sont associées. Ces valeurs peuvent être représentées par une simple chaîne de caractères (si la réponse attendue est unique) ou un ensemble de plusieurs chaînes de caractères (si les réponses autorisées sont multiples).

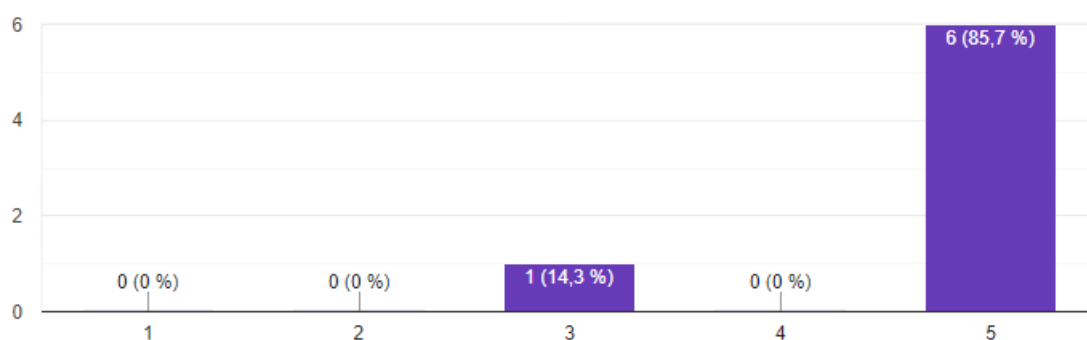
Annexe C

Résultats du sondage

1) Le cours a-t-il répondu à vos attentes en terme d'apprentissage ?

 Copier

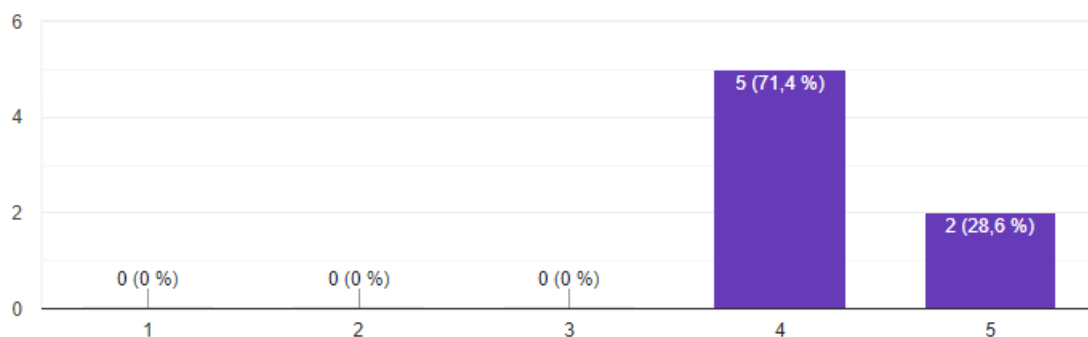
7 réponses



2) Avez-vous trouvé le contenu du cours clair/facile à comprendre ?

 Copier

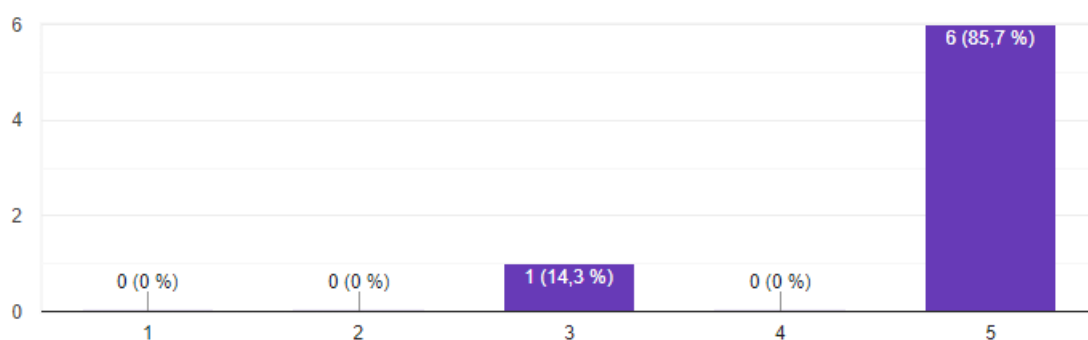
7 réponses



3) Les exercices proposés étaient-ils pertinents et adaptés à votre niveau de connaissances ?

 Copier

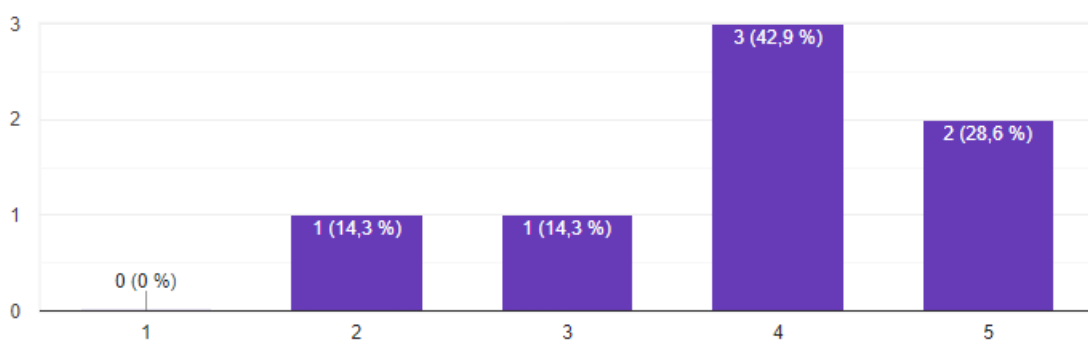
7 réponses



4.1) Le système d'évaluation automatique vous a-t-il semblé utile/efficace sur l'aspect « **détection des erreurs** » : les feedbacks en cas de mauvaises réponses vous ont-ils aidés à détecter/corriger vos erreurs ?

 Copier

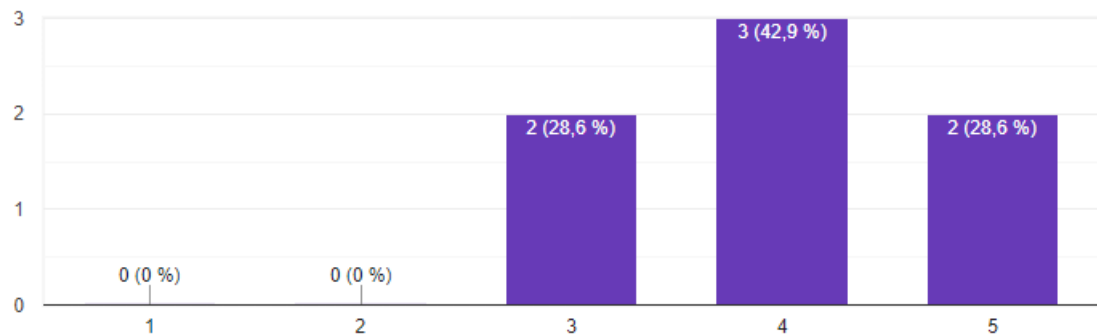
7 réponses



4.2) Le système d'évaluation automatique vous a-t-il semblé utile/efficace sur
l'aspect « informations additionnelles » : les feedbacks en cas de bonnes réponses
 (les "Tips") vous ont-ils appris quelque chose ?

 Copier

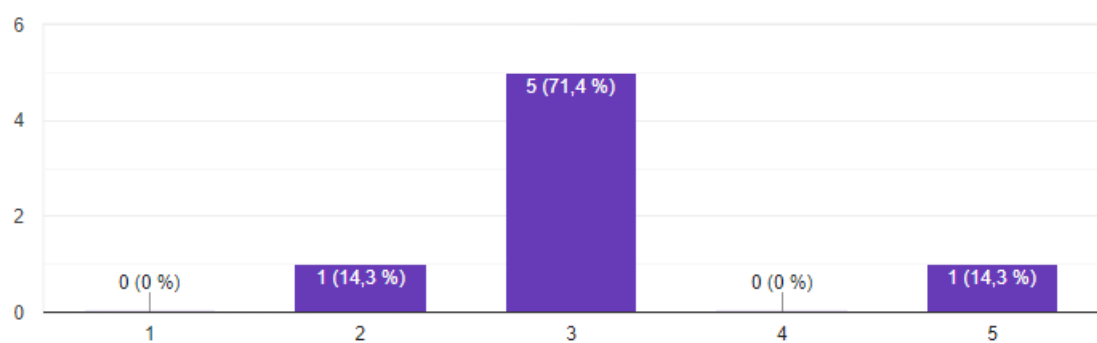
7 réponses



5) Y a-t-il des aspects du cours qui mériteraient selon vous d'être
 retravaillés/repensés/améliorés ?

 Copier

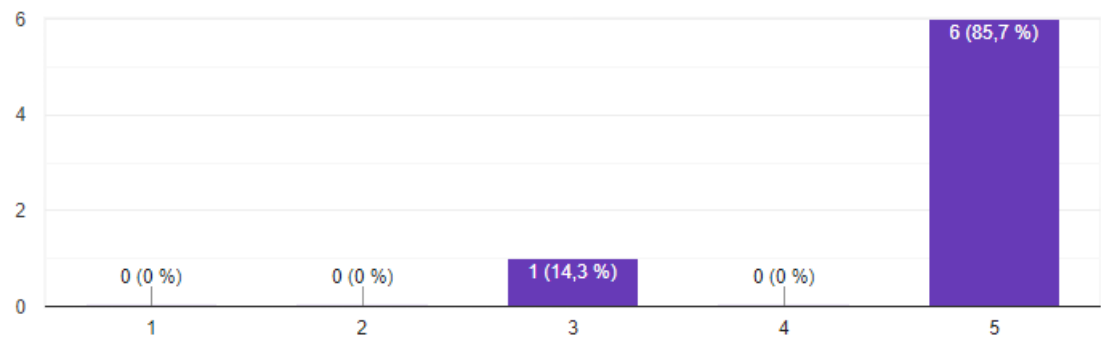
7 réponses



6) Y a-t-il des thématiques que vous auriez aimé voir abordées à travers le cours mais qui n'y figuraient pas ?

 Copier

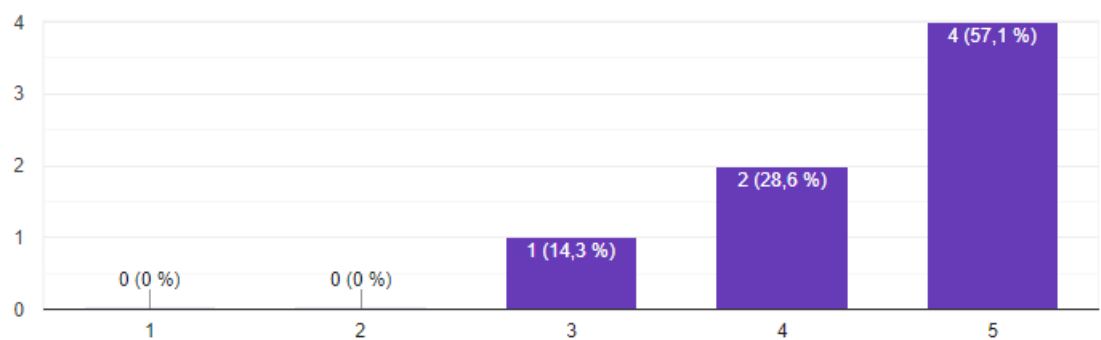
7 réponses



7) Globalement, recommanderiez-vous ce cours à un ami/collègue ?

 Copier

7 réponses



Utilisez cette zone textuelle pour développer/argumenter vos réponses aux questions précédentes (soit de manière concise et générique en quelques phrases/paragraphes, soit de manière détaillée question par question)

7 réponses

Je trouve que les feedback ne m'ont pas aidé pour détecter les erreurs.
Je ne comprenais pas les explications que ça me donnait quand j'avais une erreur. Je trouve que c'était un peu vague? Je ne sais pas comment l'expliquer.

je pense que c'est un bon exercices pour pouvoir s'améliorer en matière de HTML/CSS merci pour votre aide à comprendre ces modules

tout était plutôt claire et facile a comprendre la seul chose que j'ai pu voir comme un problème sont les correction auto qui affichais parfois mauvais alors que le code était correcte .

Je remercie Clément Cordier de nous avoir offert un service plus que suffisant pour nous dans le HTML/CSS. je vous recommanderai a des amis qui ont besoin de cette formation. Merci pour tout et passé de bonne vacances

Pour la question 5: ce serait d'amélioré la partie hexadécimal

/

ie trouve ce cour complet et ie n'hésiterai pas à le recommander

Finalement, avez-vous des commentaires/suggestions supplémentaires à faire aux personnes en charge de ce cours ?

5 réponses

Peut-être, je pense que ce serait bien si, quand on dit qu'il y a une erreur, aussi dire quel matière revoir.
Par exemple: si on a eu faux à une question qui dit d'insérer une vidéo, dire de revoir comment insérer une vidéo en html et peut-être donner un lien où on peut aller voir. Ou donner un exemple de code.
Je ne pense pas avoir bien expliqué mais je ne dit pas que c'était pas bien fait.
Je trouve que tout était bien et ça ne nécessite pas vraiment un changement.
J'ai simplement donné une idée qui n'a pas besoin d'être prise sérieusement.

continuez comme ça vous m'avez appris beaucoup de choses et avez rendus le cours encore plus pertinent ;)

je n'ai aucun commentaires ni aucune suggestions supplémentaires

non

non à part corriger quelque erreurs

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl