CÉGEP
DE L'OUTAOUAIS

Plan de cours

Techniques d'intégration multimédia 582.A1

Programmation multiplateformes

582-323-HU

1-2-1

Enseignant : Pierre Roy
Bureau : Pavillon Félix-Leclerc : local F2537D
Courriel : pierre.roy@cegepoutaouais.qc.ca

Disponibilités : voir horaire sur la porte de bureau

Campus Félix-Leclerc
Automne 2018

1. INTRODUCTION

Intention éducative

Le cours « 582-323-HU - Programmation multiplateformes » permettra à l'étudiant de poursuivre son apprentissage des langages balisés et des feuilles de style en cascade afin d'adapter la page Web à différente plateformes, mais aussi d'approfondir ses connaissances d'un langage de programmation Web.

Le cours sera centré sur des méthodes de programmation plus spécifique afin d'adapter l'interactivité d'un même site Web à plusieurs plateformes.

Les notions vues dans le cours « 582-323-HU - Programmation multiplateformes » seront intégrées dans le cadre du projet final du cours porteur « 582-325-HU - Intégration multiplateformes ».

Préalable absolu

582-223-HU Programmation Web 2

Pondération

La pondération du cours (1-2-1) correspond à une heure d'activité théorique en classe, à deux heures de laboratoire et à une heure de travail personnel en dehors des périodes de cours.

Relation entre le cours et d'autres cours du programme

Le cours « 582-323-HU Programmation multiplateformes » fait partie d'une suite de cinq cours : « 582-123-HU Programmation Web 1 », « 582-223-HU Programmation Web 2 », « 582-423-HU Programmation et base de données » et « 582-523-HU Programmation appliquée ».

2. COMPÉTENCES À DÉVELOPPER

2.1. Énoncé de la compétence

(015P) Programmer des produits multimédias

2.2. Éléments de la compétence

1. Rechercher l'information nécessaire à la programmation.
2. Produire des algorithmes pour la programmation des interfaces.
3. Programmer l'interactivité des interfaces.
4. Exploiter un système de gestion de bases de données dans Internet.
5. Programmer les fonctions particulières des interfaces
6. Adapter la programmation aux différentes plates-formes.
7. Optimiser la codification
8. Résoudre les erreurs de programmation.

2.3. Les critères de performance

1.1 Détermination juste des besoins d'information liés à la programmation.
1.2 Recherche complète de l'information auprès des différentes sources disponibles.
1.3 Organisation méthodique de l'information.
2.1 Analyse juste de la problématique à résoudre.
2.2 Détermination exacte de la structure de programmation.
2.3 Choix approprié des structures logiques du langage de programmation événementielle.
2.4 Élaboration judicieuse des algorithmes nécessaires à la programmation.
2.5 Vérification minutieuse des algorithmes.
3.1 Programmation fonctionnelle des éléments dynamiques des interfaces multimédias.
3.2 Codification précise et structurée des algorithmes.
3.3 Exploitation judicieuse des bibliothèques.
3.4 Création précise de routines générales et de gabarits.
3.5 Contrôle fonctionnel de l'affichage des médias dans la page-écran.
3.6 Orchestration judicieuse des médias selon le scénarimage.
3.7 Exploitation pertinente des fonctions propres aux langages de programmation événementielle.
3.8 Validation juste des données.
3.9 Solutions appropriées aux problèmes liés à la programmation des interfaces.
3.10 Aménagement ergonomique du poste de travail.
4. 1 Connexion précise à la base de données.
4.2 Établissement précis des requêtes.
4.3 Affichage des résultats harmonieux et adapté au scénario.
4.4 Vérification rigoureuse des résultats des requêtes.
4.5 Contrôle strict des accès à la base de données en contextes multitâches et multi-usagers.
4.6 Application rigoureuse des consignes de sécurité.
5.1 Programmation rigoureuse de l'accès à des bases de données.
5.2 Programmation rigoureuse de l'accès à des périphériques.
5.3 Création et gestion correctes de fichiers.

5.4 Modifications judicieuses de codes existants.
6.1 Adaptation correcte des routines de programmation aux plates-formes visées.
6.2 Programmation correcte de l'affichage des médias en fonction des différentes plates-formes.
7.1 Programmation modulaire du code.
7.2 Lisibilité des sources.
7.3 Documentation claire et précise des scripts.
7.4 Rapidité d'exécution du fonctionnement du produit.
8.1 Test rigoureux de validation.
8.2 Repérage précis des erreurs de syntaxe, de logique et d'implantation.
8.3 Emploi efficace des fonctions de déminage (débogage) des logiciels de programmation.
8.4 Analyse exacte des problèmes rencontrés.
8.5 Application rigoureuse de méthodes de résolution de problèmes.
8.6 Solutions appropriées aux problèmes de codification.

2.4. Le contexte de réalisation

- Par une contribution à un projet multimédia
- À partir des scripts utilisés dans des projets Web
- À l'aide d'algorithme
- À partir de structure de code orienté objet
- À l'aide de documentation

3. LE CONTENU DU COURS

Les savoirs

- Syntaxes du langage
- Types de variables
- Structures conditionnelles
- Structures répétitives
- Tableaux
- Propriétés des objets
- Objet *userAgent*
- Objet document
- Objet cookie
- Balise html pour formulaire
- Bibliothèque *jQuery*
- Gestionnaires d'évènements

Les savoir-faire

- Identification des besoins
- Déclaration de variable
- Commenter le code
- Créer des fonctions personnalisées
- Techniques de déminage

Les savoirs être

- Persévérance durant les périodes de déminage
- Débrouillardise pour l'adaptation de script
- Concentration et écoute active lors des explications théoriques

4. LES MÉTHODES D'ENSEIGNEMENT ET LES ACTIVITÉS D'APPRENTISSAGE

L'enseignant aidera l'étudiant dans l'apprentissage en lui fournissant des exemples précis sur l'application de méthodes de travail prescrites par le cours. Des exercices pratiques suivront l'apprentissage théorique afin de permettre à l'étudiant d'intégrer les notions apprises.

5. L'ÉVALUATION DES APPRENTISSAGES

5.1. Les modalités des évaluations formatives

Tout au long de la session, des exercices d'acquisition et d'enrichissement seront offerts afin d'optimiser la réussite scolaire des étudiants.
On nomme ces exercices des « exercices formatifs ». Ils seront corrigés sur demande seulement. Ces exercices augmentent considérablement le rendement scolaire des étudiants.

5.2. Les modalités des évaluations sommatives

Évaluation	Pondération	Semaine
Examen 1	30 %	5
Examen 2	30 %	10
Projet synthèse	40 %	15

5.3 Critère de l'évaluation finale

Critères	Pondération
Modifications judicieuses de codes existants	30 %
Programmation fonctionnelle des éléments d'interfaces	25 %
Adaptation correcte des routines de programmation aux plates-formes visées.	25 %
Programmation correcte de l'affichage des médias en fonction de différentes plates-formes.	20 %

6. LE CALENDRIER DES ACTIVITÉS

Semaines 1, 2, 3 et 4		
Module 1	**Révision JavaScript**	
Section 1	Types de variables	12 heures
Section 2	Opérateurs, ordre de priorité	
Section 3	Instructions conditionnelles (Opérateurs logiques, imbriqués)	
Section 4	Les boucles for et while	
Section 5	Les tableaux	
Section 6	Les fonctions (paramètres, variable locale, instruction return)	
Section 7	Dépannage, erreur syntaxique, erreur logique	
Semaine 5		
Examen 1 sur le module 1		3 heures
Semaines 6 et 7		
Module 2	**DOM et événements**	
Section 1	Structure du DOM (objet Navigator)	6 heures
Section 2	Les événements	
Semaines 8 et 9		
Module 3	**Compatibilité Multiplateformes et Optimisation**	
Section 1	Fureteurs et dispositifs	6 heures
Section 2	Chargement des images	
Semaine 10		
Examen 2 sur les modules 2 et 3		3 heures
Semaines 11 et 12		
Module 4	**Les Formulaires**	
Section 1	Créer et valider un formulaire	6 heures
Section 2	Objets String et Date (méthodes)	
Section 3	Expressions régulières	
Semaines 13, 14 et 15		
	Finalisation du travail synthèse	9 heures
	Remise du travail synthèse	

7. RÈGLES DÉPARTEMENTALES

Tous les règlements adoptés par le département de technique d'intégration multimédia s'appliquent en tout temps. Voir le document « Règlements et remarques » remis lors du premier cours de la session d'automne.

8. LE MATÉRIEL REQUIS POUR LE COURS

Espace d'hébergement d'un nom de domaine (des frais d'activation ou de renouvellement sont à prévoir)

9. BIBLIOGRAPHIE

Flanagan, David, *JavaScript : The Definitive Guide, O'Reily, 6th Edition, 2011, 1079p.*
ISBN 978-0-596-80552-4

Powers, Shelley, *JavaScript CookBook*, O'Reilly, Gravenstein, 2010, 530p.
ISBN 978-0-596-80613-2