Langages de script et langages dynamiques

Groupe 20
Mardi, de 13h30 à 16h30 Voir local au: https://portailetudiant.uqam.ca/ (cours)

Jeudi, de 13h30 à 15h30 Voir local au: https://portailetudiant.uqam.ca/ (atelier)

Responsable(s) du cours

Nom du coordonnateur : TREMBLAY, Guy

Nom de l'enseignant : TREMBLAY, Guy
Local : PK-4435
Téléphone : (514) 987-3000 #8213
Courriel : tremblay.guy@uqam.ca
Site Web : http://www.labunix.uqam.ca/~tremblay

Description du cours

Ce cours à contenu variable vise à permettre d'aborder de nouvelles approches prometteuses en informatique et génie logiciel non couvertes par les autres activités de la banque de cours.

Objectifs du cours

Général :
Initier les étudiant-e-s à la programmation à l'aide de langages de script et de langages dynamiques.

Spécifiques :
À la fin du cours, l'étudiant-e devrait être capable . . .
- d'expliquer ce qui différencie les langages de script et langages dynamiques des langages «ordinaires» et d'en comprendre les avantages vs. limites ;
- de lire des scripts écrits dans divers langages : scripts bash, Ruby, Python (?) ;
- d'écrire des scripts bash, notamment pour automatiser des tâches ;
- d'écrire des programmes objets en Ruby, tout en sachant utiliser le style fonctionnel lorsqu'approprié ;

Contenu du cours

- Introduction : Que sont les langages de script? Les langages dynamiques?
- Notions de base Unix:
  - Utilisation du shell
  - Contrôle du code source
  - Assemblage de logiciels
  - Commandes et utilitaires
- Scripts bash
  - Variables
  - Structures de contrôle
  - Fonctions
  - Expressions régulières et pattern-matching

- Manipulation de texte
  - grep
  - sed
  - awk
- Ruby
  - Types de base (Fixnum, String, Symbol, Range)
  - Structures de données (Array, Hash)
  - Définition et appel de méthodes
  - Structures de contrôle
  - Définition de classes
  - Expressions régulières et pattern-matching
  - Lambda-expressions et blocs
  - Itérateurs
  - Modules et mixins
  - Exceptions
  - Interactions avec l'environnement
  - Tests unitaires (MiniTest): tests, stubs vs. mocks
  - Règles de style
- Autres sujets (en Ruby, selon le temps)
  - Automatisation des tâches avec Rake
  - Développement de gems
  - Applications en ligne de commandes "à la git"
  - Métaprogrammation
  - Langages spécifiques au domaine (DSL)
  - Approche "Convention over Configuration"
- Autres langages (selon le temps)
  - Python
  - Perl
  - JavaScript (?)

Modalités d'évaluation

Descpriction	Date	Pondération
Examen intra	16 octobre	25 %
Examen final	11 décembre	30 %
Trois (3) travaux pratiques		45 %

- L'utilisation de documentation personnelle est permise aux examens.
- Une moyenne d'au moins 50 % aux examens est exigée pour réussir le cours. Idem pour les travaux pratiques.
- Les travaux pratiques peuvent être réalisés seul ou en équipe de deux (2) personnes.
- Une pénalité de 10 % par jour de retard (partiel ou complet) sera appliquée pour la remise des travaux.
- Le devoir #3 sera un projet en Ruby sur un sujet «au choix» -- avec certaines contraintes qui seront spécifiées ultérieurement.
- La qualité du français sera prise en considération, tant dans les examens que dans les travaux pratiques (jusqu'à 10 % de pénalité).

Politique d'absence aux examens

L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : **http://info.uqam.ca/politiques/**

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : **http://info.uqam.ca/politiques**

Intégrité académique

PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- **la substitution de personnes;**
- **l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;**
- **la transmission d'un travail pour fins d'évaluatiion alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;**
- **l'obtention par vol, manoeuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;**
- **la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;**
- **l'utilisation pendant un examen de la copie d'examen d'une autre personne;**
- **l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;**
- **la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce persone, quelles que aoient les circonstances;**
- **la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-création, un rapport de stage ou un rapport de recherche;**
- **Les sanctions reliées à ces infrations sont précisées à l'article 3 du Règlement no 18.**

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : **http://www.sciences.uqam.ca/etudiants/integrite-academique.html** et **http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html**

Médiagraphie

- Divers éléments (notes de cours, exemples) seront mis à la disposition des étudiants sur le site suivant:
  - http://www.labunix.uqam.ca/~tremblay_gu/INF600A/Materiel
  -
- Le manuel suivant est *suggéré*, **mais non obligatoire**:
  - D. Thomas, A. Hunt, and C. Fowler. **Programming Ruby 1.9 & 2.0: The Pragmatic Programmer's Guid**e. Addison-Wesley, 2013.
    (En vente à la COOP : prix membre ≈ 62.55 $ + taxes)

Bibliographie plus détaillée:

[Bar00] D. Barron. The World of Scripting Languages. John Wiley & Sons, Ltd, 2000.
[Bla04] C. Blaess. Scripts sous Linux--Shell Bash, Sed, Awk, Perl, Tcl, Tk, Python, Ruby. Eyrolles, 2004.

[Bro09] G.T. Brown. Ruby Best Practices. O'Reilly, 2009.
[CGMG09] J. Campbell, P. Gries, J. Montojo, and Wilson G. Practical Programming--An introduction to Computer Science using Python. The Pragmatic Bookshelf, 2009.
[Cha09] S. Chacon. Pro Git. Apress, 2009.
[Cop12] D.B. Copeland. Build Awesome Command-Line Applications in Ruby: Control Your Computer, Simplify Your Life. The Pragmatic Bookshelf, 2012.
[Dou90] D. Dougherty. Sed & Awk. O'Reilly, 1990.
[Fow11] M. Fowler. Domain-Specific Languages. Addison-Wesley, 2011.
[Ful07] H. Fulton. The Ruby Way, Second Edition. Addison-Wesley, 2007.
[Gho11] D. Ghosh. DSLs in Action. Manning, 2011.
[Joh09] C.J. Johnson. Pro Bash Programming: Scripting the Linux Shell. Apress, 2009.
[Jon14] P.J. Jones. Effective Ruby: 48 Specific Ways to Write Better Ruby. Addison-Wesley Professional, 2014.
[Kol14] A. Koleshko. Rake Task Management Essentials. Packt Publishing, 2014.
[Mar06] B. Marick. Everyday Scripting with Ruby--For Teams, Testers, and You. The Pragmatic Bookshelf, 2006.
[Mil15] R. Miller. Text Processing with Ruby--Extract Value from the Data That Surrounds You. The Pragmatic Bookshelf, 2015.
[NR05] C. Newham and B. Rosenblatt. Learning the bash shell. O'Reilly, 2005.
[Ols08] R. Olsen. Design Patterns in Ruby. Addison-Wesley, 2008.
[Ols11] R. Olsen. Eloquent Ruby. Addison-Wesley Professional Ruby Series, 2011.
[Per10] P. Perrotta. Metaprogramming Ruby. The Pragmatic Bookshelf, 2010.
[RB05] A. Robbins and N.H.F. Beebe. Classic Shell Scripting. O'Reilly, 2005.
[Swi08] T. Swicegood. Pragmatic Version Control Using Git. The Pragmatic Bookshelf, 2008.
[Tat10] B.A. Tate. Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages. Addison-Wesley, 2010.
[ZK05] A. Zeller and J. Krinke. Essential Open Source Toolset. John Wiley & Sons, Ltd., Chichester, UK, 2005.

A : article - C : comptes rendus - L : logiciel S: Standard - U : uri - V : volume	C : complémentaire - O : Obligatoire - R : recommandé