

UNIVERSITÉ DE
SHERBROOKE**Département d'informatique****IFT 211/IFT 776 – Programmation scientifique en Python****Plan d'activité pédagogique
Hiver 2018**

Enseignant**Gabriel Girard**

Courriel : gabriel.girard@usherbrooke.ca
Local : D4-1010-1
Téléphone : (819) 821-8000 poste 63033
Site : <http://info.usherbrooke.ca/GabrielGirard/cours>
Disponibilité : sur rendez-vous

Responsable(s) : Gabriel Girard

Horaire

Exposé magistral : Mardi 13h30 à 15h20 salle D3-2039
Vendredi 13h30 à 15h20 salle D4-1023 (Laboratoire) ou D3-2040

Les semaines pour le cours IFT211 sont les 8, 15 et 29 janvier ainsi que le 12 février.

Horaire spécial semaine 1 (8 janvier) : mercredi 13h30 (D3-2039) et
vendredi 08h30 (D4-1023).

Description officielle de l'activité pédagogique¹

Cibles de formation : Pouvoir développer des programmes de bonne qualité à l'aide du langage de programmation Python.

Contenu : Introduction aux ordinateurs. La syntaxe générale et les types de bases de Python. Les structures de contrôle : séquence, sélection, itération, récursivité. Concept de fonctions et d'abstraction procédurale. Concept de base de l'abstraction de données. Les entrées/sorties. Utilisation de bibliothèques pour la programmation scientifique.

Crédit 1

Organisation 2 heures d'exposé magistral par semaine
2 heure d'exercices par semaine
0 heures de travail personnel par semaine

Particularités Aucune

1. <http://www.usherbrooke.ca/fiches-cours/IFT/ift211/ift776.htm>

1 Présentation

Cette section présente les cibles de formation spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation d'un comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

Le cours IFT 211, Programmation scientifique en Python, est un premier cours de programmation qui s'adresse aux étudiants de tous les cycles de physique, de chimie et de biologie. Il ne présuppose pas de connaissance en programmation. Il ne requiert que la connaissance de l'utilisation usuelle d'un ordinateur. Comme son nom l'indique c'est un cours de programmation en Python. Il vise l'apprentissage des notions de base de ce langage avec pour objectif de l'utiliser dans des applications spécifiques au domaine d'étude des étudiants.

Ce cours sera complété par un cours de laboratoire spécialisé dans une des domaines d'application soit la physique, la biologie ou la chimie.

1.2 Cibles de formation spécifiques

L'objectif du cours est d'apprendre à programmer en Python et à utiliser ce langage dans un domaine scientifique.

De façon plus précise, à la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable :

1. de comprendre le fonctionnement d'un ordinateur dans le contexte de l'utilisation d'un outil d'élaboration d'une solution programmée ;
2. de manipuler les environnements de développement associés à Python;
3. de développer des outils informatique à l'aide du langage Python;
4. de respecter des normes et standards de programmation ;
5. d'utiliser les mécanismes élémentaires de contrôle et d'organisation des données;
6. de documenter la solution.

1.3 Contenu détaillé

Thème	Contenu	Nbr d'heures	Cibles de formation	Travaux
1	Brève introduction à l'environnement : présentation des environnements matériel et logiciel, étapes de la mise en oeuvre et de l'exécution d'un programme.	0.5	1	
2	Développement de programmes séquentiels : éléments de base du Python, notions de programmes, notions de compilation. Organisation des données et types, Énoncés d'affectation, de lecture et d'écriture.	3.5	2,3,4,5,6	TP
3	Les fonctions : concept de modularité; concept de fonctions, utilisations des fonctions ; paramètres d'entrée et de sortie; bibliothèques.	4	2,3,4,5,6	TP
4	Structures sélectives des langages : expressions logiques; énoncé « if », énoncés composés, énoncé « if » emboîté.	4	2,3,4,5,6	TP
5	Structures itératives des langages : concept d'itération, boucle conditionnelle et de comptage, énoncés « while », « for » ; boucles emboîtées.	3	2,3,4,5,6	TP
6	Conclusion.	1		

1. Les heures associées à un thème particulier inclues les heures d'exercices/laboratoires pour un total de 16 heures.
2. Le cours doit comprendre au moins cinq travaux pratiques couvrant tous les sujets marqués « TP » dans le tableau.

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Le cours est d'une durée d'environ cinq semaines. Une semaine comprend quatre heures de présence en classe: deux heures de cours magistral et deux heures de laboratoire. Chaque semaine, il y aura donc environ deux heures d'exposés magistraux décrivant la théorie ainsi que des exemples développés au tableau. Tous les thèmes du cours, sauf le premier, seront abordés de la même manière: au moins une étude de cas sera étudiée; l'analyse du problème sera effectuée et une fois correcte, la solution sera programmée; enfin on reviendra sur les éléments nouveaux du langage vus dans la programmation de la solution.

Chaque semaine il y aura un laboratoire portant sur le thème abordé.

2.2 Calendrier

Semaine	Date	Thèmes	Laboratoire
1	8 janvier 2018	1-2	Présentation de l'environnement, programme simple
2	15 janvier 2018	3	Fonctions + Bibliothèques
-	22 janvier 2018		<i>Cours PHQ 202</i>
3	29 janvier 2018	4-5	Sélection/itération
-	5 février 2018		<i>Cours PHQ 202</i>
4	12 février 2018	4-5-6	Itération et conclusion
-	19 février 2018		<i>Cours PHQ 202</i>
5	26 février 2018		Examen IFT 211

2.3 Évaluation

Devoirs	40 %
Laboratoires	20 %
Final	40 %

Conformément aux articles 36, 37 et 38 du règlement facultaire d'évaluation des apprentissages² l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignant. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 8.1.2 du *Règlement des études*³ de l'Université de Sherbrooke. L'étudiant ou l'étudiante peut s'exposer à de graves sanctions dont automatiquement un zéro (0) au devoir ou à l'examen en question.

Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes tant que la rédaction finale des documents et la création du programme reste le fait de votre équipe. En cas de doute de plagia, l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. En cas de doute, ne pas hésiter à demander conseil et assistance à l'enseignant afin d'éviter toute situation délicate par la suite.

2. <http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-devaluation/>

3. <http://www.usherbrooke.ca/programmes/references/reglement/>

2.4 Échéancier des travaux

TP #	Réception de l'énoncé	Date de remise	Thème
1	12/01/18	19/01/18	Séquence
2	19/01/18	02/01/18	Fonctions et bibliothèques
3	02/02/18	16/02/17	Sélection
4	16/02/17	23/02/17	Itération

Directives particulières :

Examen

L'examen portera sur la programmation. À moins d'avis contraire, la consultation du manuel de cours et des notes de cours est autorisé à l'examen. Aucun ordinateur, téléphone cellulaire, calculatrice ou autre appareils électroniques n'est autorisé lors des examens.

Devoirs

1. Il y aura de 4 laboratoires. Les laboratoires doivent être fait individuellement. Ils sont tous corrigés et comptabilisés.
2. Les travaux se feront en équipe de 2 personnes. Un travail effectué par une équipe de trois se verra imposé une pénalité de 25%. Cette pénalité sera de 100% pour des équipes de plus que trois.
3. La qualité du français et de la présentation est considérée dans le résultat du travail.
4. Les sujets des travaux seront disponibles sur la page WEB du cours (page web du DMI sous la rubrique « Le personnel/Professeurs d'informatique/Gabriel Girard ») au jour spécifié dans le plan de cours pour la réception du devoir. La remise du travail s'effectue le jour et à l'heure exigés. Le non respect de la date de remise entraîne une pénalité de 25% de la note par jour de retard, à moins d'un cas exceptionnel. Il est à noter qu'un oubli ou un emploi du temps chargé n'est pas un cas exceptionnel. Il en est de même en ce qui concerne une panne électrique, d'ordinateur, d'imprimante, ou du réseau. Cela signifie qu'il faut toujours viser à terminer son travail de programmation au moins 24 heures avant la date de remise pour tenir compte des pannes possibles et de la surcharge quasi-inévitable. Ceci est un conseil qui vaut son pesant de points. Les modalités de remise de chaque travail vous seront fournies avec le sujet de chaque travail.
5. Le respect des normes départementales est impératif. Le document « Norme de documentation des programmes » [6] contient les normes du département en matière de programmation. Ces normes pourront être adaptées au langage Python.

Cette contrainte permet de vérifier que l'étudiant ou l'étudiante sait s'astreindre à une discipline de programmation. Elle permet de plus de mieux insister sur les concepts importants du cours.

6. La programmation en Python est testée dans l'environnement UNIX, qui est la base du système Linux. Elle peut être réalisée sous tout autre environnement (Windows, Linux ou Mac OS X par exemple), mais devra se trouver et fonctionner sous UNIX au moment de la remise du travail. Il est de votre responsabilité de vous assurer que votre programme est fonctionnel dans l'environnement de correction (Linux).

De l'aide technique est disponible sur les ordinateurs des laboratoires, que ceux-ci tournent sous Windows 10 et Linux. Sur le site, les étudiants et étudiantes pourront trouver des liens et de la documentation pour travailler à partir de chez eux. Cependant, aucune autre aide ne sera fournie par manque de temps et de ressources. Il est conseillé, pour un travail hors des laboratoires, d'utiliser l'environnement de développement PyCharm ou Spyder qui est disponible sur la plupart des plates-formes principalement sous Linux (Ubuntu).

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permise dans le plan de cours.

Dans ce cours le règlement 4.2.3 s'applique à moins d'avoir obtenu personnellement l'autorisation du professeur. Cette permission peut être retirée en tout temps, si l'appareil n'est pas utilisé uniquement à des fins d'apprentissage.

Tel qu'indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission du professeur.

Note : L'utilisation du courriel électronique est recommandée pour poser vos questions.

3 Matériel nécessaire pour l'activité pédagogique

Les acétates du cours sont disponibles sur le WEB.

Les normes de programmation du Département d'informatique sont décrites dans [6]. Vous devez absolument vous procurer ce document et le lire.

Il n'y a pas de manuel de référence spécifique pour le cours. Les manuels [4, 9, 13, 14, 16, 19] sont tous de très bonnes références. Certains sont disponibles en ligne ou en PDF. Certains sont gratuits d'autres sont disponibles à faible coût.

Les formations en lignes [1, 7, 12, 20] sont extrêmement pertinentes et utiles.

Le site PythonTutor [8] offre aussi une façon de comprendre en permettant de visualiser son code.

4 Références

- [1] ANONYME : Python tutorial. <http://pythonprogramminglanguage.com/>, 2015.
- [2] Joey BERNARD : Doing astronomy with python. www.linuxjournal.com/, Août 2015.
- [3] Joey BERNARD : General relativity in python. www.linuxjournal.com/, Juillet 2015.
- [4] Jamie CHAN : *Python : Learn Python in One Day and Learn It Well*. LCF Publishing, 2014.
- [5] Allen B. DOWNEY : *Think Python 2nd Edition*. O'Reilly Media, 2015. Disponible à l'adresse : <http://www.greenteapress.com/thinkpython2/index.html>.
- [6] Marc FRAPPIER : Normes de rédaction et de programmation du département. <http://www.usherbrooke.ca/informatique/intranet/ressources-et-documentation/logiciels-services-outils/normes-de-programmation/>, 2005. Aussi disponible sur le site Web du cours.
- [7] Python Software FOUNDATION : Python tutorial. <https://docs.python.org/3/tutorial/>, 2015.
- [8] Philip GUO : Python tutor. <http://www.pythontutor.com/>.
- [9] Cody JACKSON : *Learning to Program Using Python*. 2013. Une version pdf est disponible à : <http://www.onlineprogrammingbooks.com/learning-program-using-python/>.
- [10] Robert JOHANSSON : *Introduction to Scientific Computing in Python*. 2014.
- [11] Hans Petter LANGTANGEN : *A Primer on Scientific Programming with Python*. Springer, 2014.
- [12] LEARNPYTHON : Learn python. <http://www.learnpython.org/>.
- [13] Mark LUTZ : *Learning Python, 5th Edition*. O'Reilly Media, 2013.
- [14] Mark LUTZ : *Python Pocket Reference, 5th Edition*. O'Reilly Media, 2014.
- [15] MATPLOTLIB : Matplotlib crash course python tutorial. <https://pythonprogramming.net/matplotlib-graphing-series/>, 2015.

4. <https://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-des-etudes/>

5. <http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-devaluation/>

- [16] Mark PILGRIM : *Dive Into Python 3 2nd Edition*. APRESS, 2009. Disponible en ligne ou en pdf à l'adresse : <http://www.diveintopython3.net/>.
- [17] Lennart REGBRO : Porting to python 3 : An in-depth guide. www.onlineprogrammingbooks.com/porting-to-python-3-an-in-depth-guide/, Novembre 2012. voir aussi <http://python3porting.com/>.
- [18] Nicolas P. ROUGIER : Matplotlib tutorial. <http://www.labri.fr/perso/nrougier/teaching/matplotlib/>, 2015.
- [19] C. H. SWAROOP : A byte of python. <http://www.swaroopch.com/notes/python/>. Une version pdf est disponible à : files.swaroopch.com/python/byte_of_python.pdf.
- [20] TUTORIALSPPOINT : Python - tutorial. <http://www.tutorialspoint.com/python/>, 2015.

L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille!

Extrait du Règlement des études

8.1.2 Relativement aux activités pédagogiques

L'expression délit désigne d'abord tout acte ou toute manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique.

Sans restreindre la portée générale de ce qui précède, est considéré comme un délit :

- a) la substitution de personnes ou l'usurpation d'identité lors d'une activité évaluée ou obligatoire;
- b) le plagiat, soit le fait, dans une activité évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui;
- c) l'obtention par vol ou par toute autre manœuvre frauduleuse de document ou de matériel, la possession ou l'utilisation de tout matériel non autorisé avant ou pendant un examen ou un travail faisant l'objet d'une évaluation;
- d) le fait de fournir ou d'obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour un examen ou un travail faisant l'objet d'une évaluation;
- e) le fait de soumettre, sans autorisation préalable, une même production comme travail à une deuxième activité pédagogique;
- f) la falsification d'un document aux fins d'obtenir une évaluation supérieure dans une activité ou pour l'admission à un programme.

Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets
- Reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire
- Utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources
- Résumer ou paraphraser l'idée d'un auteur sans en indiquer la source
- Traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets
- Utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord)
- Acheter un travail sur le Web ou ailleurs et le faire passer pour sien
- Utiliser sans autorisation le même travail pour deux activités différentes (autoplégat)

Autrement dit : mentionnez vos sources.
