Introduction à la programmation : cours 1

A la découverte d'un nouveau monde !

Benoît Frénay - Faculté d'Informatique











- apprinte a conserver the programma poor acoustic disproblement
- e appropria jurisectione bosci transmission cours in Python

- south in questions port in terminal (if accurant est stoppes)
 - interrompte-moi die que cous avez une curetian
 - w Si sony detrochez, demandez moi de [14] ecolopée
 - propriete dire some attachisers in biodiminini

- apprendre à concevoir des programmes pour résoudre des problèmes
- approche universitaire : ceci n'est pas un cours de Python

- outs in quantum with its imments (it was n'est implie)
- interrompte-moi des que your and une question
- with accordance of the contract of the second of the secon
- p par err din som stituiners a birennilling

- apprendre à concevoir des programmes pour résoudre des problèmes
- approche universitaire : ceci n'est pas un cours de Python

toutes les questions sont les bienvenues (et aucune n'est stupide)

- interrompez-moi dès que vous avez une question
- si vous décrochez, demandez-moi de (ré)expliquer
- le but est que vous appreniez à programmer



Plan du cours

- un peu d'histoire
- 4 la programmation est un jeu d'enfant
- exemple de système programmé
- o notions de base en programmation
- outils utilisés par le cours
- organisation du cours
- quelques références

Un peu d'histoire

L'algorithmique : une science vieille comme le monde

- vient du nom du mathématicien perse Al-Khwarizmi (780-850).
- premiers algorithmes remontent à Babylone (+ de 4000 ans).
- algorithme PGCD d'Euclide dans ses Éléments (300 av. J.-C.)





Avant l'ordinateur électronique





Alan Turing (1912–54) : les bases théoriques

- pose les bases mathématiques de l'informatique en 1936
- conçoit une "bombe" capable de cracker le code d'Enigma (84.000 messages/mois) et permet de forcer le blocus des U-boats allemands (raccourci la guerre de 2 à 4 ans et sauve de 14 à 21 millions de vies)
- propose le test de Turing (= "imitation game") en 1950





John von Neumann (1903–57) ; le touche-à-tout

- mathématicien et physicien ayant participé à la création de l'EDVAC (5,5 ko, 1.160 additions/seconde, 6.000 tubes à vides, 12.000 diodes, 56 kW, 45,5 m², 7.850 kg, \$500.000, équipes de trente personnes)
- a proposé l'architecture Von Neumann à la base des ordinateurs actuels.



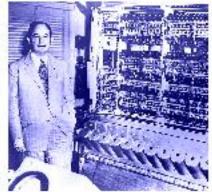


ENIAC, EDVAC, Mark I: les premiers ordinateurs

"I think there is a world market for maybe five computers."

— Thomas Watson, chairman of IBM, 1943.





IBM 704 computer at NASA in 1957



Dual IBM 7090s at NASA Mission Control in 1962



Apollo Guidance Computer (64 ko memory, 0.043MHz)





Deep Blue (1996) vs. Samsung Galaxy S5 (2014)





De 1960 à nos jours : l'essor de l'informatique

- → 1960's : premier compilateur et département d'informatique, ARPAnet
- 4 1970's: premier microprocesseur et micro-ordinateurs, Unix, CRAY-I
- 1980's: explosion des ordinateurs personnels (5 millions en 1982)
- 1990's: Internet, Windows 3.1, e-mails, HTML 4.0, MP3, JPEG
- 2000's: eBay, Facebook, YouTube, Google, Mozilla, Linux (Android)
- 2010's: smartphones, internet of things, big data, machine learning



La programmation est

un jeu d'enfant

Programmation : la "cuisine" des ordinateurs





Programmation : la "cuisine" des ordinateurs

Définition

un programme informatique est une suite d'instructions qui décrivent de façon claire et non-ambigüe à un ordinateur comment effectuer une tâche

rough a partition as alternated rough transferring from Asia,

Programmation : la "cuisine" des ordinateurs

Définition

un programme informatique est une suite d'instructions qui décrivent de façon claire et non-ambigüe à un ordinateur comment effectuer une tâche

- écrit dans un langage spécifique (= langage de programmation)
- constitué de variables, d'instructions, de boucles, de fonctions, etc.
- programmer = écrire un programme pour résoudre un problème

Exercution d'un programme parmet de seculule le problème pour lequel il est cores — exécution — effectuel la taire d'exercitore (peut veres)

Programmation: la "cuisine" des ordinateurs

Définition

un programme informatique est une suite d'instructions qui décrivent de façon claire et non-ambigüe à un ordinateur comment effectuer une tâche

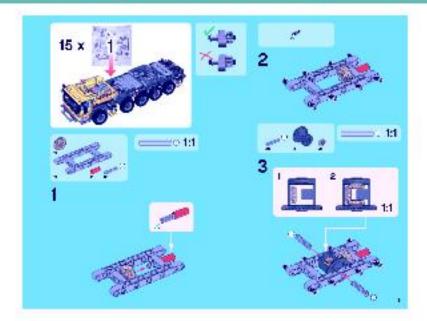
- écrit dans un langage spécifique (= langage de programmation)
- constitué de variables, d'instructions, de boucles, de fonctions, etc.
- programmer = écrire un programme pour résoudre un problème

l'exécution d'un programme permet de résoudre le problème pour lequel il est conçu ⇒ exécution = effectuer la suite d'instructions (peut varier)

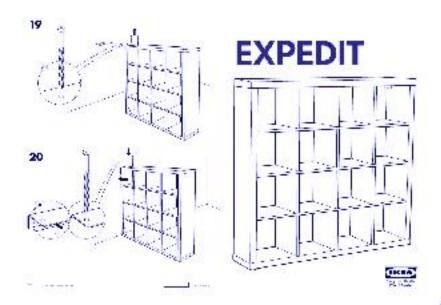
Exemples de "programme" que vous avez un jour exécuté



Exemples de "programme" que vous avez un jour exécuté.



Exemples de "programme" que vous avez un jour exécuté



Exemples de "programme" que vous avez un jour exécuté



- premier de l'atendat une voleur pour la manéronne
 - · Halfilms de distabilité litritées bour le replie
 - "An Baldir diri gağılındahir bini Hir günü volta xwez achind

months is comportanted on programmer paid failures are beside.

- decrived the oraginamina = "gite fairs 4 still modifient ?"
 - Applying a lability disktuple fixed the most
 - Acites liane un biligges parallaller (est le français la
- A programme interpretation take of literarchies

Variable

permet de "stocker" une valeur pour la manipuler

- nombre de personnes invitées pour le repas
- poids du paquet de beurre que vous avez acheté

modifie le comportement du programme pour l'adapter aux besoins

- decivals in gragramms = "gite fains a qual melition 2"
- Application of the state of the state of the sector.
- notes tians un bijezza meditalen (auf le fignesis ti.)
- a programme attenuations use diluterations

Variable

permet de "stocker" une valeur pour la manipuler

- nombre de personnes invitées pour le repas
- poids du paquet de beurre que vous avez acheté

modifie le comportement du programme pour l'adapter aux besoins

Instructions

décrivent le programme = "que faire à quel moment?"

- équivalent à la suite des étapes dans une recette
- écrites dans un langage particulier (pas le français!)
- programme informatique = suite d'instructions

Пистанствонь сонфинента (с

- promot do chose (= textuccione en fracciona (ana cambon)
- Of example I fully vend all spoke point, after oilly 87 sites allie 5.
- Destruit one attoiture Hook-lintaine de pragnitivitue ("embraschements")

- regressive are said and interpretation property full labor, or would be
 - A person of farmer than bloom that includes a "botto a mattle"
- A complete control of the control of
- - mod me programme desucoup plus amplies a scale. Not, configur
- appellication contrat d'adilication ("anni lili fava-il et que fait alle 7") :

Instructions conditionnelles

permet de choisir les instructions en fonction d'une condition

- ⇒ exemple : "si je veux un steak à point, alors cuire 8', sinon cuire 5'"
- permet une structure non-linéaire de programmes ("embranchements")

22

Instructions conditionnelles

permet de choisir les instructions en fonction d'une condition

- ⇒ exemple : "si je veux un steak à point, alors cuire 8', sinon cuire 5'
- permet une structure non-linéaire de programmes ("embranchements")

Fonction

regroupe une suite d'instructions souvent utilisées ensemble

- exemples:peler(...), cuire(...), mélanger(..., ...)
- rend les programmes beaucoup plus simples à écrire, lire, corriger

spécification = contrat d'utilisation ("que lui faut-il et que fait-elle?")

```
permet d'estrese des autroctions de manuels répetition

à comple l'oppose et four autroction de manuels répetition

a permet de chémic à l'execution le nomine de répetitions

a permet de chémic à l'execution le nomine de répetitions

a sommée l'institut qu'il y a de la parte en verset et suite mon entire.
```

Stantines de Sordie es et lichtes

- or cutting les resistes sont main dans on chimosi (type, non, etc.)

a decision of the algorithms, manifestate through publishing on the

Boucle et invariant

permet d'exécuter des instructions de manière répétitive

- exemple : "répéter n fois : verser de la pâte et cuire une crêpe"
- permet de choisir à l'exécution le nombre de répétitions
- une boucle peut aussi dépendre d'une condition d'arrêt
- exemple : "tant qu'il y a de la pâte, en verser et cuire une crêpe"

Struttling the Scatters of Higher

- parinet de regetainter l'information acias forme afficace.
 - or cutsile. Its resitter and miseritors on classes (type, size, etc.
 - A married to the spiritual and spiritual solution by the l

Boucle et invariant

permet d'exécuter des instructions de manière répétitive

- exemple : "répéter n fois : verser de la pâte et cuire une crêpe"
- permet de choisir à l'exécution le nombre de répétitions
- une boucle peut aussi dépendre d'une condition d'arrêt
- exemple : "tant qu'il y a de la pâte, en verser et cuire une crêpe"

Structures de données et fichiers

permet de représenter l'information sous forme efficace

- en cuisine, les recettes sont mises dans un classeur (type, nom, etc.)
- associées à des algorithmes spécifiques (manipulation, tri, etc.)

Résumé

Définition

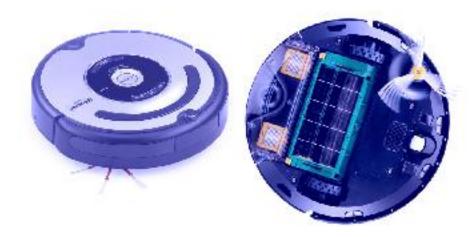
un programme informatique est une suite d'instructions qui décrivent de façon claire et non-ambigüe à un ordinateur comment effectuer une tâche

Quelques-unes des notions vues au cours

- instructions
- variables
- instructions conditionnelles
- fonctions
- · boucle et invariant
- structures de données et fichiers

Exemple de système

programmé

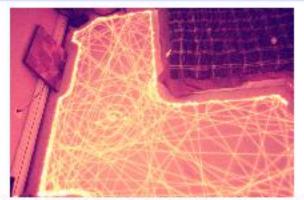


The Roomba uses a variety of cleaning behaviors to cover a room, using input from its sensors to decide where to go next.



source : http://spectrum.ieee.org/windmaton/robotics/home-robots/trobot-roomba-860-vs-mesto-swill

The Roomba uses a variety of cleaning behaviors to cover a room, using input from its sensors to decide where to go next.

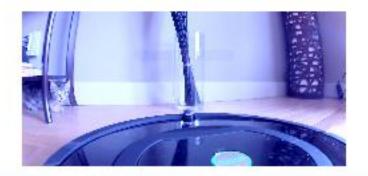


The Roomba uses a variety of cleaning behaviors to cover a room, using input from its sensors to decide where to go next.



The Roomba uses a variety of cleaning behaviors to cover a room, using input from its sensors to decide where to go next.





Dans la peau d'un Roomba avec une GoPro

- https://www.youtube.com/watch?v=xX9RkWsX1KE
- https://www.youtube.com/watch?v=pcW79-опqRw

en savoir plus sur le iRobot Roomba : http://electronics. howstuffworks.com/gadgets/home/robotic-vacuum.htm

Informatique mobile et ambiante









THE PERSON NAMED IN

- INFOM450 : Internet des objets
- INFOM451 : conception d'applications mobiles
- INFOM453 : laboratoire en informatique ambiante et mobile

Notions de base en

programmation

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Simens due to sur

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Éléments d'un langage

- Examinate (continue) in between a michanol both (contex min byta)
- seminimité : aignification de titure (byvase, s'autaxidhament cottoble
- programme is plor screent by differential concepts at construction;
 authoritization in the stiller programme resource are table complete.
- attention. Fordinateur est profesionent stoppe et psychologide l
 - https://grassboulle.net/les-ardinatears-sour-cons

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Éléments d'un langage

 syntaxe : vocabulaire (mots disponibles pour "parler" à l'ordinateur) et grammaire (comment ils peuvent s'enchaîner pour former une phrase)

programmes is quer servent les differentes concepts et construction

estatelan. Confinatair est profesiónions sequide et prechangida l

- https://grassboulle.net/lec-ardinateurs-con-cons-

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Éléments d'un langage

- syntaxe : vocabulaire (mots disponibles pour "parler" à l'ordinateur) et grammaire (comment ils peuvent s'enchaîner pour former une phrase)
- sémantique : signification de toute "phrase" syntaxiquement correcte
- progrestique a quoi servent les differentes concepts et constructions et autout comment les atilier proc résoulre une têche complete
- ettettian. Fordinaleur est profesiónsomt supplie et psychologida l
- https://graseboulle.net/lec-ardinatemps-mont-cons

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Éléments d'un langage

- syntaxe : vocabulaire (mots disponibles pour "parler" à l'ordinateur) et grammaire (comment ils peuvent s'enchaîner pour former une phrase)
- sémantique : signification de toute "phrase" syntaxiquement correcte
- pragmatique : à quoi servent les différentes concepts et constructions et surtout comment les utiliser pour résoudre une tâche complexe

- https://graphoulle.pet/lec-ardinateurs-con-con

Définition

un langage de programmation est un langage formel et exécutable par un ordinateur permettant d'expliquer comment effectuer une tâche

Éléments d'un langage

- syntaxe : vocabulaire (mots disponibles pour "parler" à l'ordinateur) et grammaire (comment ils peuvent s'enchaîner pour former une phrase)
- sémantique : signification de toute "phrase" syntaxiquement correcte
- pragmatique : à quoi servent les différentes concepts et constructions et surtout comment les utiliser pour résoudre une tâche complexe

attention : l'ordinateur est profondément stupide et psychorigide!

https://grisebouille.net/les-ordinateurs-sont-cons

falle d'archagrable ou la parmaire

- thur are it edited use on mal architerability
- the property of the first and place and property of

Enternal Personal

emisse corrects, mais impossible à executer dans certaines essertions

- are complete presents in conference to frigor, along up it act vides
- the situat if assistance a non-client paint on bodishing

Exercise il artemun 7 de l'assure 7 de communicación

- a'escore normalement, mala inspredollates la limitare sicomple
- p refer difficile a detector pubblic in pergraffing se terrote
- a assemble. Take on cake and facility of another

Erreurs syntaxiques

fautes d'orthographe ou de grammaire

- mot qui n'existe pas ou mal orthographié
- mots combinés d'une façon qui n'est pas autorisée

32

Erreurs syntaxiques

fautes d'orthographe ou de grammaire

- o mot qui n'existe pas ou mal orthographié
- mots combinés d'une façon qui n'est pas autorisée

Erreurs d'exécution

syntaxe correcte, mais impossible à exécuter dans certaines situations

- exemple : prendre n oeufs dans le frigo, alors qu'il est vide
- une erreur d'exécution a pour effet l'arrêt du programme

Erreurs syntaxiques

fautes d'orthographe ou de grammaire

- mot qui n'existe pas ou mal orthographié
- mots combinés d'une façon qui n'est pas autorisée

Erreurs d'exécution

syntaxe correcte, mais impossible à exécuter dans certaines situations

- exemple : prendre n oeufs dans le frigo, alors qu'il est vide
- une erreur d'exécution a pour effet l'arrêt du programme

Erreurs d'intention / de logique / de sémantique

- s'exécute normalement, mais ne produit pas le résultat escompté
 - très difficile à détecter, puisque le programme se termine
 - exemple : faire un cake sans farine ni oeufs

Introduction à la programmation

But du cours

apprendre à concevoir des programmes pour résoudre des problèmes

Approache unecessaria

Apprende a programmar, c'est sontout apprendre la pragmatique

a meche la sumurite a enfere de façon rigocompe et non ampaglie

a motoristamment du langage utilité (ex grande hartis)

REACHTERNESS OF THE ARTEST OF THE RECORD OF THE CONTROL POSSESSAN VALUE OF THE PROPERTY OF THE

Introduction à la programmation

But du cours

apprendre à concevoir des programmes pour résoudre des problèmes

Approche universitaire

apprendre à programmer, c'est surtout apprendre la pragmatique

- décrire la démarche à suivre de façon rigoureuse et non ambigüe
- indépendamment du langage utilisé (en grande partie)

la syntaxe et la sémantique seront découverts progressivement...

- province-code = planticle militant francals/anglain.or.symbols.
- code Histolicasina (Hon alliblightes) dans an linteger informatique

Alexander in the second

- # algorithms solution are problems are provide code
- применения по пр

Call Sales of a because in

- gradie de la carba er éclition el un algorithme la readyant.
- a tradiction the l'algorithme en la tipage de plug monates
- a pullballor of the dispersions allow

Code vs. pseudo-code

- pseudo-code = ébauche mêlant français/anglais et symboles
- code = instructions (non ambigües) dans un langage informatique

Algorithms to engineer

- a ligarithms solution for problems an possion code
- применения постав (порядиния постав постав

Call-Called Plant Backstin in

- product de la criba et écition d'un algirithme la resdyant.
- water the language of the programmation
- as well best on an east the agreement already

Code vs. pseudo-code

- pseudo-code = ébauche mêlant français/anglais et symboles
- code = instructions (non ambigües) dans un langage informatique

Algorithme vs. programme

- algorithme = solution au problème en pseudo-code
- programme(s) = implémentation(s) dans un langage spécifique

Collection Fig. Beautiful.

- yearline do la carba ar écultura d'un algunithme la résalyant.
- Confuction de l'algorithms de langage de programmation
- a politication at test dispersions alterna-

Code vs. pseudo-code

- pseudo-code = ébauche mêlant français/anglais et symboles
- code = instructions (non ambigües) dans un langage informatique

Algorithme vs. programme

- algorithme = solution au problème en pseudo-code
- programme(s) = implémentation(s) dans un langage spécifique

Conception d'un programme

- analyse de la tâche et écriture d'un algorithme la résolvant
- * traduction de l'algorithme en langage de programmation
- vérification et test du programme obtenu

Pseudo-code

afficher le message "Hello, World!" à l'écran

```
Implémentation (code) en langage machine
10111010 00010000
00000001 10110100
00001001 11001101
00100001 00110000
11100100 11001101
00010110 10111000
00000000 01001100
11001101 00100001
01001000 01100101
                   (he)
                  (11)
01101100 01101100
01101111 00100000 (o )
01010111 01101111 (wo)
01110010 01101100 (r1)
01100100 00100001
                   (d!)
00100100
                   (3)
```

Implémentation (code) en assembleur

```
section data
      helloMsg: db 'Hello world!',10
      helloSize: equ $-helloMag
section text
      global _start
_start:
      mov eax.4
                            ; Appel système "write" (sys_write)
      mov ebx.1
                            ; File descriptor, 1 pour STDOUT
      mov ecx, helloMsg
                            : Adresse de la chaîne a afficher
      mov edx, helloSize
                            : Taille de la chaîne
      int 80h
                            : Exécution de l'appel système
                            ; Sortie du programme
      mov eax.1
                            ; Appel système 'exit'
      mov ebx.0
                            : Code de retour
      int 80h
```

Implémentation (code) en C

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

source: http://fr.wikipedia.org/wiki/Liste_de_programmes_Hello_sprid

Implémentation (code) en Python

print ("Hello world!")

source : http://fr.eikipedia.org/eiki/Liste_de_progresses_Wello_strld

Outils utilisés par le cours



- site with du cours INFOB130 "Impoduction a la programmation".
- a tout fichier our WebCampus paus être considére comme définité
 - transparents PDF (="slides") disposibles explosion is formats
 - imprimes units prog.X-handout, some pdf. (vi scoon, if les-in)
 - comes TRullers with documents supplementaines, et
- * communication amores forum (+ 4 mails
- outilis standard poor les cours vous affez l'utiliser produit 5 anni.



- site web du cours INFOB131 "Introduction à la programmation"
- e tous dichier sur WebCampus paus être considére comme définit
 - transparents PDF (="sides") disposibles en plusieurs formats.
 - imprimes artes prog.X. Landout, so the pdf. (a soicia, it lessis)
- a communication minosci toline (+ + mint)
- outils standard poor its coins woos alies l'utiliser perdant 5 arms.



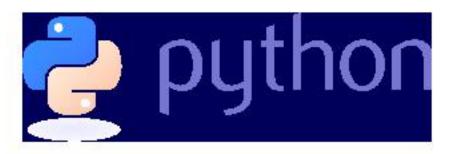
- site web du cours INFOB131 "Introduction à la programmation"
- tout fichier sur WebCampus peut être considéré comme définitif!
 - transparents PDF (="slides") disponibles en plusieurs formats
 - imprimez intro_prog_X_handout_notes.pdf (si soucis, dites-le!)
 - énoncés TPs, liens web, documents supplémentaires, etc.
- a outils standard poor les come vous affez l'utiliser pendant 5 ans



- site web du cours INFOB131 "Introduction à la programmation"
- tout fichier sur WebCampus peut être considéré comme définitif!
 - transparents PDF (="slides") disponibles en plusieurs formats
 - imprimez intro_prog_X_handout_notes.pdf (si soucis, dites-le!)
 - énoncés TPs, liens web, documents supplémentaires, etc.
- communication : annonces, forums (+ e-mails).

Web@mpus

- site web du cours INFOB131 "Introduction à la programmation"
- tout fichier sur WebCampus peut être considéré comme définitif!
 - transparents PDF (="slides") disponibles en plusieurs formats
 - imprimez intro_prog_X_handout_notes.pdf (si soucis, dites-le!)
 - énoncés TPs, liens web, documents supplémentaires, etc.
- communication : annonces, forums (+ e-mails).
- outils standard pour les cours : vous allez l'utiliser pendant 5 ans!

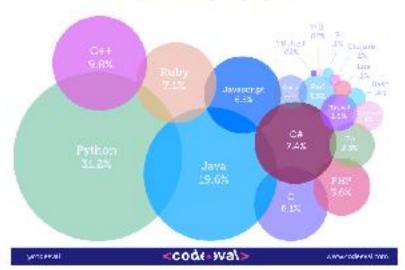


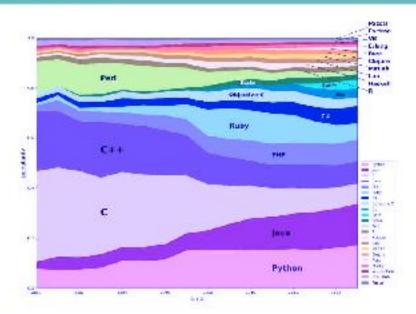


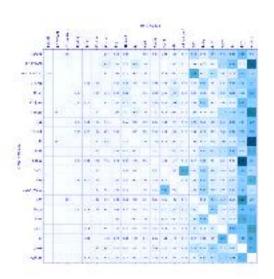


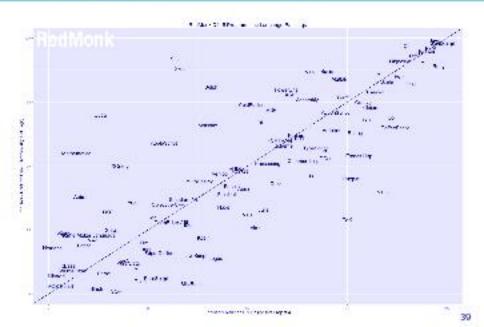
Python, DF mr Professional Developers

Most Popular Coding Languages of 2015









Avantanges

- (premier) langage simple à prendre en main
- lisibilité du code (par ex. indentation obligatoire)
- √ langage généraliste et de haut niveau (peu de code % à C, Java etc.)
- √ possibilité d'exécuter des bouts de code (+ notebooks) + scalable
- libre, gratuit, open source, disponible sur Windows, Mac OS, Linux
- communauté active (librairies, documentation, tutoriaux, etc.)
- √ structure de données disponibles immédiatement.
- système de modules hiérarchique et lisible

lead to the state of

- bills limb = Concutton (resis Cython point side)
- typege flynamiglin (errelini difficiler a distrectin)

Avantanges

- (premier) langage simple à prendre en main
- lisibilité du code (par ex. indentation obligatoire)
- √ langage généraliste et de haut niveau (peu de code % à C, Java etc.)
- possibilité d'exécuter des bouts de code (+ notebooks) + scalable
- libre, gratuit, open source, disponible sur Windows, Mac OS, Linux
- √ communauté active (librairies, documentation, tutoriaux, etc.)
- √ structure de données disponibles immédiatement.
- √ système de modules hiĕrarchique et lisible.

Inconvénients

- x plus lent à l'exécution (mais Cython peut aider)
- x typage dynamique (erreurs difficiles à détecter)

Pourquoi Anaconda + PyCharm?

Christian Palice "de antimed" of 200 ac

- A tour les certies en un sont Machangement
- box and feel identifying ser took in Co.
- centages de nodule dinocibile providoramen
- и деполития III ст постообы путарти.
- a ninponinte pour Python 2.7 et 3.x

A proposed to alternatives

- 4 Dyelin and (at motion) library graduly
- a America Politica and side a given time by cutilly
- A three to cause the sacrownia shirthway one PyCham

Pourquoi Anaconda + PyCharm?

Distribution Python "self-contained" et gratuite

- tous les outils en un seul téléchargement
- look and feel identique sur tous les OS
- centaines de modules disponibles immédiatement
- · gestionnaire, IDE et notebooks intégrés
- disponible pour Python 2.7 et 3.x

A property described and a property of the second s

- 4 Python and (at restory) library graduly
- Amenda PyChom aces alde a giver time lay certile
- A dame to cause the sacross minimize one PyCham

Pourquoi Anaconda + PyCharm?

Distribution Python "self-contained" et gratuite

- tous les outils en un seul téléchargement
- look and feel identique sur tous les OS
- centaines de modules disponibles immédiatement
- gestionnaire, IDE et notebooks intégrés
- disponible pour Python 2.7 et 3.x

A propos des alternatives

- Python est (et restera) libre et gratuit
- Anaconda + PyCharm vous aide à gérer tous les outils
- dans le cadre de ce cours, n'utilisez que PyCharm

Installation de Anaconda + PyCharm





or Professions, Developers

instructions disponible sur le site WebCampus

- n'hésitez pas à demander de l'aide aux assistants
- 🔹 une install party est prévue très bientôt (détails sur WebCampus)

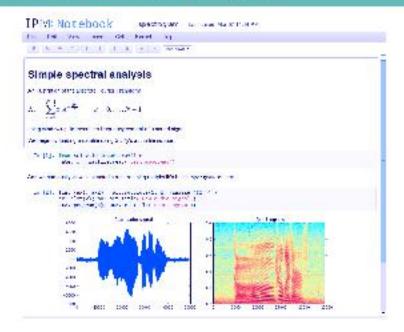
disponible sur les machines du pool informatique

AlgoBot - http://www.algobot.be

- réalisé par Fishing Cactus (Mons, Belgique 1,8 millions DL Shift)
- présenté à la conférence EDEN à Oslo (Norvège) en 2013.
- "Meilleur Serious Game" au Serious Game Expo de Lyon 2013.
- "Best Learning Game, 2nd" aux 2013 European Serious Game awards

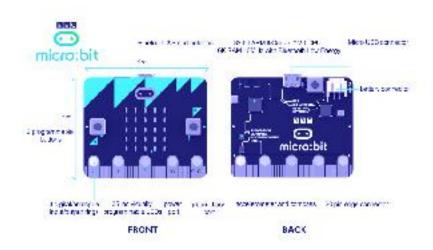


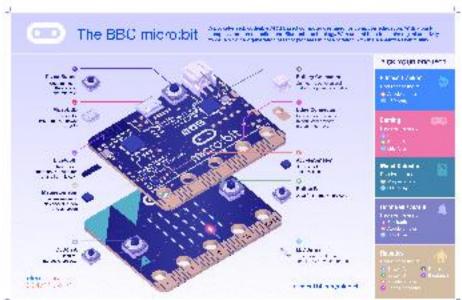
Présentation du système de notebooks



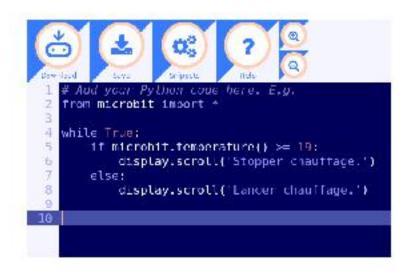














Ressources

micro:bit est un projet de la BBC qui offre plein de ressources

démarrage rapide : http://microbit.org/guide/quick

chaque étudiant recevra son kit avec un BBC micro:bit (contre caution)

Utilisation + encadrement

- le BBC micro:bit sera utilisé pour introduire les concepts (cours + TP)
- à chaque TP, vous recevrez des problèmes à résoudre chez vous
- un étudiant-tuteur assure des permanences au local du CSLabs (2ême étage, local 226) chaque jeudi de 13h30 à 15h30 (accès libre)
- vous serez amené à remplir des questionnaires, merci de le faire sérieusement et à chaque fois ⇒ évaluation du projet micro:PUNCH

Disclaimer: the rest *BBC' and 'microbit' are trademarks of the BBC, micro PUNCH is not endorsed, spontored nor associated with the Poundation. See http://bicodit.cog for more info on the microbit ecosystem.

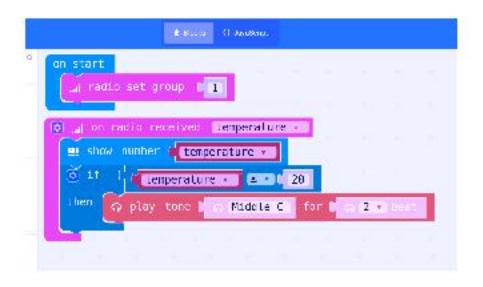
Le BBC micro:bit en action ; alarme antigel sonore

```
& Birres
                          () Javakorter
on start
   al radio set group 1 1
## forever

    temperature (°C)

  ## pause (ms) ( 10000
```

Le BBC micro:bit en action ; alarme antigel sonore



Le BBC micro:bit en action : coach sportif / podomètre



Le BBC micro:bit en action : coach sportif / podomètre

```
d litterat
                         [] .brotkete
on start
  al radio set group b 1
  set no steps . to C 0
  on radio received inc.
  change nb steps + by f inc +
  show number w nb_steps •
            nb steps v ≥ v 8
        set nb steps + to ( 0
  then
        o play tone | Middle C for |
```

Le BBC micro:bit en action ; versions pour un seul micro:bit.



Le BBC micro:bit en action ; versions pour un seul micro:bit.



Organisation du cours

Modalités pratiques

INFOB131 = 11 crédits - 60h+45h

- * EC15 = "Enterposis Credita Transfer System"
- L ECTS = 30h da travail toirt type confimile.
- 10/11 ECTS 100 leaves de traval
 - a 60h de entre 4 69h de TPa (= primaries fabilitations en entre
 - the content of 2001; the former's presented (marketed as more one analysis of
 - a étude program contra l'I le protique C liurne, point des granteses als

Camment present on Council's

Modalités pratiques

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu

Modalités pratiques

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 4 60h de cours + 45h de TPs (= présence "obligatoire" en salle)

 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 4 60h de cours + 45h de TPs (= présence "obligatoire" en salle)

 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

Comment préparer un cours/TP

51

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 4 60h de cours + 45h de TPs (= présence "obligatoire" en salle)

 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

- → mettre en ordre/relire ses notes ⇒ ai-je tout compris?
- officire les illustrations (notabook) ana-je canable de coder ada?
- a little le medion de l'aire dans de com, "noedot time"

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 4 60h de cours + 45h de TPs (= présence "obligatoire" en salle).
 - reste ± 200h de travail personnel (tout seul ou avec vos collégues)
 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

- → mettre en ordre/relire ses notes ⇒ ai-je tout compris?
- refaire les illustrations (notebook) ⇒ suis-je capable de coder cela?
- Hitter le allierions que l'aise diffunctiv coint, "no este clinc"

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 60h de cours + 45h de TPs (= présence "obligatoire" en salle).
 - reste ± 200h de travail personnel (tout seul ou avec vos collégues)
 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

- → mettre en ordre/relire ses notes ⇒ ai-je tout compris?
- refaire les illustrations (notebook) ⇒ suis-je capable de coder cela?
- → exercices complémentaires ⇒ permet une vraie interaction au cours!

INFOB131 = 11 crédits - 60h+45h

- ECTS = "European Credits Transfer System"
- 1 ECTS = 30h de travail tout type confondu
- → 10/11 ECTS = ± 300 heures de travail
 - 60h de cours + 45h de TPs (= présence "obligatoire" en salle)

 - étude, préparer cours/TPs, pratique @ home, poser des questions, etc.

- → mettre en ordre/relire ses notes ⇒ ai-je tout compris?
- refaire les illustrations (notebook) ⇒ suis-je capable de coder cela?
- → exercices complémentaires ⇒ permet une vraie interaction au cours!
- noter les questions que j'ai ⇒ début du cours, "question time".

- bloc n°1: bases de la programmation
 - à la découverte d'un nouveau monde.
 - valeurs, expressions et variables
 - instructions conditionnelles et exceptions
 - fonctions et spécifications
- bloc il 2 c iteration et striichmes de données
 - attraction of community
 - or alterediscipline the thorough of first level
- State of a complete of the definition.
- - THE RESIDENCE OF THE PARTY OF T
- traval obligators or core on puth groups a un ren-projet of group
- . ecceledii qe merris eli bizridhe et qir "liret, zoni jer concebte qir pi
- a diagos nun-propo compres pour 10% (alamos par-junifile 11/20

- bloc n°1: bases de la programmation
 - à la découverte d'un nouveau monde.
 - valeurs, expressions et variables
 - instructions conditionnelles et exceptions
 - fonctions et spécifications
- bloc n°2 : itération et structures de données
 - · boucles et invariants
 - structures de données et fichiers
- president of the color

- bloc n°1: bases de la programmation
 - à la découverte d'un nouveau monde.
 - valeurs, expressions et variables
 - instructions conditionnelles et exceptions
 - fonctions et spécifications
- bloc n°2 : itération et structures de données
 - boucles et invariants
 - structures de données et fichiers
- bloc n°3: notions d'algorithmique
 - introduction à l'algorithmique
 - programmation récursive

- bloc n°1: bases de la programmation
 - á la découverte d'un nouveau monde.
 - valeurs, expressions et variables
 - instructions conditionnelles et exceptions
 - fonctions et spécifications
- bloc n°2 : itération et structures de données
 - boucles et invariants
 - structures de données et fichiers
- bloc n°3: notions d'algorithmique
 - introduction à l'algorithmique
 - programmation récursive

après chaque bloc, une semaine est consacrée à un mini-projet en groupe

- travail obligatoire et coté en petit groupe (groupes imposés)
- occasion de mettre en pratique et de "fixer" tous les concepts du bloc
- chaque mini-projet comptera pour 10% (absence non-justifiée = 0/20)

Travaux pratiques (chaque semaine entre les cours)

- The popular
- Tits machine

Politiki i - - i ali -- ile stille 12

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- m Title machine

Paumini - Silan Since the Pa

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

Panning - an an arrest the 12

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

- Included the particle of the part
- he moutation our begains (placed est de fixos
- premise di profi dei militante material disposible.
- Intersections (natabook an like spreadons guitz: etc.)
- ca "force" à avancer dans la tentière (et au se sont réalis-sont.
- 4 participation active and retire-papers decembe de constitue la marie-
- a parmini serit (nin papire)) portant un la thione CL les concides

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- TPs machine

- lire les slides n'est pas (du tout) suffisant
- ar mouthness must be paint (place), est de fixe.
- presence du profi des andistants: Historial disposible...
- a Interactions (notabook an line greations, quitz, etc.,
- co "force" à anymore daile la tentière (et année sont giolne soil!
- a participation active and retrasposate decimals de constitue la marie-
- a second serie (sur papier)) portage un factitione CL ha concide

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

- lire les slides n'est pas (du tout) suffisant
- émulation par les pairs (plus on est de fous...)
- presence dil profi dei andistanta: Historial disposibile...
- Interactions (notabook an line, questions, quite: etc.)
- de filoroni à avenuer dans la tendière des prise sont réclie soliti
- a participation action and restrictions of the constraints in marks
- a second serif (air papiel) portage on to them. CL he execute

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

- lire les slides n'est pas (du tout) suffisant
- émulation par les pairs (plus on est de fous...)
- présence du prof, des assistants, matériel disponible...
- interactions (notebook en live, questions, quizz, etc.)

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

- lire les slides n'est pas (du tout) suffisant
- émulation par les pairs (plus on est de fous...)
- présence du prof, des assistants, matériel disponible...
- interactions (notebook en live, questions, quizz, etc.)
- ça "force" à avancer dans la matière (et on se sent moins seul)

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- ♦ TPs machine

Pourquoi venir au cours et au TP?

- lire les slides n'est pas (du tout) suffisant
- émulation par les pairs (plus on est de fous...)
- présence du prof, des assistants, matériel disponible...
- interactions (notebook en live, questions, quizz, etc.)
- ça "force" à avancer dans la matière (et on se sent moins seul)
- participation active aux mini-projets nécessite de connaître la matière

a second serie (surpapire)) portage sur la this de CL ha concide

Travaux pratiques (chaque semaine entre les cours)

- TPs papier
- TPs machine

- lire les slides n'est pas (du tout) suffisant
- émulation par les pairs (plus on est de fous...)
- présence du prof, des assistants, matériel disponible...
- interactions (notebook en live, questions, quizz, etc.)
- ça "force" à avancer dans la matière (et on se sent moins seul)
- participation active aux mini-projets nécessite de connaître la matière
- examen : écrit (sur papier!) portant sur la théorie et les exercices

Et après ? Le laboratoire de développement de programme !

Le projet en quelques mots (en 2015-16)

par groupes (libres!) de 3 étudiants, implémenter

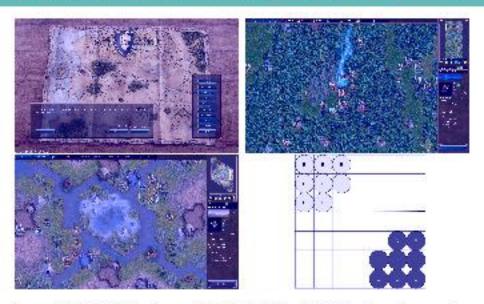
- un programme permettant de jouer à un jeu raisonnablement complexe
- une intelligence artificielle (IA) contre laquelle jouer à ce jeu

le jeu est imposé ⇒ découverte de l'énoncé en groupe en février

Organisation du quadrimestre

- le laboratoire est l'aboutissement d'INFOB131 et des mini-projets
- · uniquement des séances en groupe encadrées par les assistants
- pas assez ⇒ cours de 3 ECTS = 90 heures de travail/étudiant
- les étapes du laboratoire sont soigneusement balisées

Et après ? Le laboratoire de développement de programme!



librament impiré de l'excellent jeu libre et granuk "The Battle for Wenoth" disposible sur brops://www.ecoorb.org

Et après ? Le laboratoire de développement de programme !

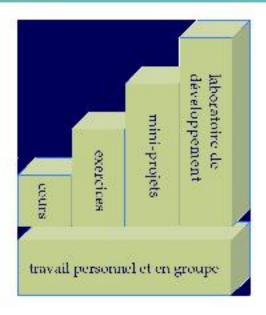




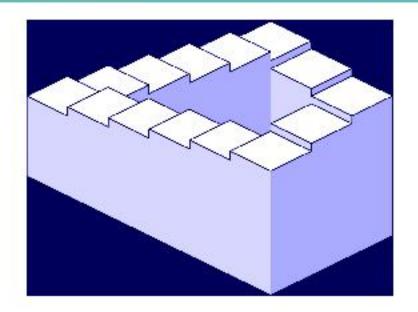




Philosophie du cours



Philosophie du cours



Acquis d'apprentissage

Être capable de résoudre des problèmes simples à l'aide d'algorithmes (pseudo-code) et de les traduire en langage de programmation (Python).

- manipuler des variables et des valeurs
- concevoir des structures conditionnelles et les conditions associées
- abstraire du code à l'aide de fonctions
- écrire une spécification de fonction et documenter son code
- utiliser des boucles for/while et écrire l'invariant correspondant
- choisir et manipuler une structure de données adaptée au problème
- créer, utiliser et manipuler des fichiers, notamment avec Pickle
- écrire un algorithme simple et caractériser sa complexité
- concevoir un algorithme récursif

Acquis d'apprentissage

Faire preuve d'une compréhension des concepts, c-à-d exprimer avec ses propres mots la théorie et les outils vus au cours et expliquer dans quel contexte ceux-ci sont utiles. Une maîtrise du langage Python est attendue.

L'étudiant sera capable de comprendre du code, de le critiquer et d'y apporter les corrections nécessaires.

Références





"Apprendre à programmer avec Python 3", G. Swinnen, éd. 2012 https://inforef.be/swi/download/apprendre_python3_5.pdf

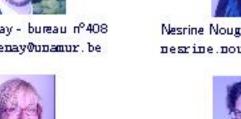
+ ressources sur le site webcampus du cours

certaines images de ce cours proviennent de houps://wikipedis.org

L'équipe pédagogique



Benoît Frênay - bureau nº408 benoit.fremay@unamur.be



Julie Henry - bureau nº426 julie.henry@umamur.be



Nesrine Noughi - bureau nº435 nesrine.noughi@unamur.be



Adrien Bibal - bureau nº435 adrien.bibal@unamur.be

Feedback et questions

Vous remarquez une erreur? Vous avez des questions?

- contactez un membre de l'équipe pédagogique
- en particulier, vous pouvez écrire à benoit frenay@unamur.be
- l'équipe pédagogique est là pour vous aider!

ce cours est "nouveau", tout feedback/suggestion est plus que le bienvenu!



The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...' – Isaac Asimov (1920–1992)

The beginning of knowledge is the discovery of something we do not understand. – Frank Herbert (1920–1986)