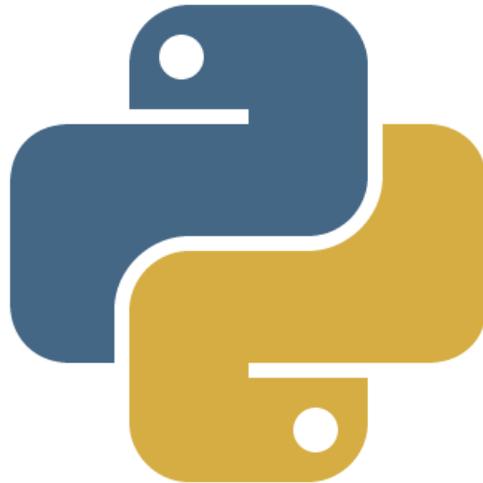


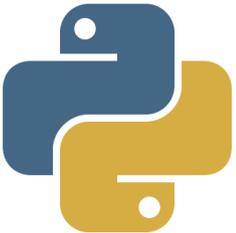
L'informatique du tronc commun en TSI



Séminaire nouveaux programmes TSI

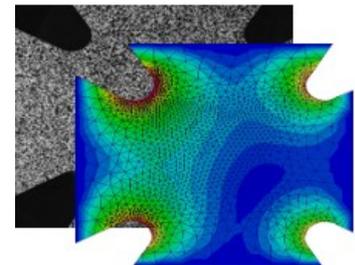
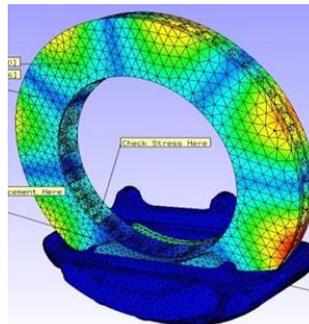
L'informatique en TSI

- Pourquoi s'engager dans les enseignements d'informatique
- Le nouveau programme informatique de TSI
- Les outils : Python
- Les ressources pédagogiques disponibles
- L'informatique en sciences de l'ingénieur



L'informatique dans les Sciences Industrielles de l'Ingénieur (S2I)

- Calcul numérique (Calcul EF, simulation dynamique, CAO, etc);
- Commande numérique;
- Traitement des signaux;
- Communication par bus industriels et réseaux, objets communicants;
- Intelligence Artificielle.



Apporter notre point de vue sur *l'informatique pour l'ingénieur*

- Une informatique au service de l'ingénierie (résolution numérique de problèmes contextualisés)
- Une informatique interfacée avec l'extérieur (mesure, commande, traitement de données, réseaux)
- Des compétences théoriques ET opérationnelles (autonomie des étudiants en TIPE)
- Des épreuves de modélisation au concours TSI
- *L'Inspection Générale recommande officiellement que les étudiants soient confrontés à des professeurs des trois disciplines Mathématiques, Physique et SI (BO 2015 n27).*

Programme d'informatique en TSI

Esprit du programme :

- L'informatique pour des ingénieurs généralistes

Modalités d'enseignement :

En première année:

- 1h de cours / semaine
- 2h de TP / quinzaine

En deuxième année, uniquement au semestre 3:

- 1h de cours / semaine
- 2h de TD / quinzaine

Programme d'informatique de CPGE en TSI

- S1 : 7 thèmes de Cours/TP;
- S2 : méthodes de programmation et analyse des algorithmes, représentation des nombres en machine, bases des graphes;
- S3 : bases de données, dictionnaires et algorithmique pour l'intelligence artificielle et les jeux, ingénierie numérique.

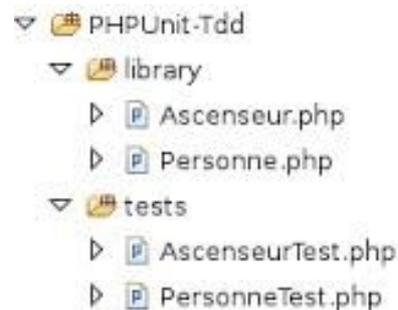
“L’organisation de la progression au sein des deux premiers semestres relève de la responsabilité pédagogique de la professeure ou du professeur.”

Programme d'informatique en TSI

Semestre 1: Programmation et algorithmique

Les thèmes de travaux pratiques:

- *mise en place de l'environnement de travail;*
- mise en place d'une discipline de programmation;
- introduction à la validation et l'étude de complexité des algorithmes sur un ensemble de thèmes varié;

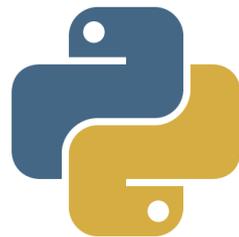


Programme d'informatique en TSI

Semestre 1: Programmation et algorithmique

Les thèmes de travaux pratiques:

- structures de programmation de base (boucles, conditions, fonctions, etc.);
- recherche séquentiel dans un tableau, dictionnaire;
- utilisation de modules et bibliothèques Python;



Programme d'informatique en TSI

Semestre 1: Programmation et algorithmique ***Les thèmes de travaux pratiques:***

- algorithmes dichotomiques, recherche de zéro;
- fonction récursive;
- matrices de pixel et images;
- algorithme de tris.



Programme d'informatique en TSI

Semestre 2: Codage de l'information en mémoire

- Représentation des nombres entiers, flottants.
- Comprendre comment une information est codée sous forme binaire en mémoire.
- Comprendre le problème de précision d'un calcul numérique.

Programme d'informatique en TSI

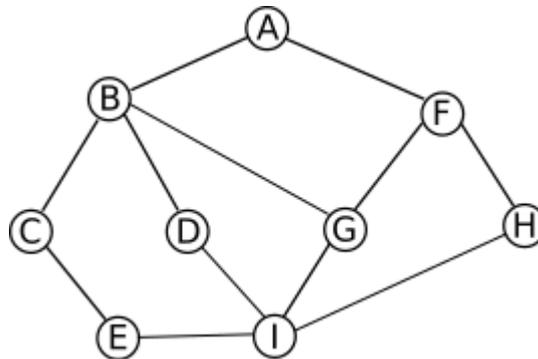
Semestre 2: Méthodes de programmation et analyse des algorithmes

- terminaison et correction
- annotation et commentaires d'un programme;
- assertion;
- jeu de test.

Programme d'informatique en TSI

Semestre 2: les graphes

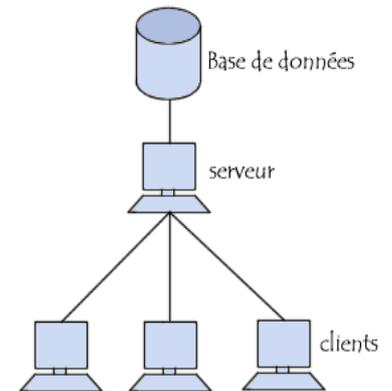
- bases des graphes, notations;
- parcours d'un graphe;
- pondération d'un graphe.



Programme d'informatique en TSI

Semestre 3: Bases de données

- Quelques éléments théoriques sur l'organisation d'une BDD
- Langage de requête SQL



```
SELECT col1, col2 FROM my_table WHERE col3 > 50
```

Programme d'informatique en TSI

Semestre 3: Dictionnaires et algorithme pour l'Intelligence Artificielle

- dictionnaires, clés et valeurs;
- algorithme des k plus proches voisins;
- algorithme des k-moyennes;
- jeux d'accessibilité à deux joueurs sur un graphe.

Programme d'informatique en TSI

Semestre 3: Algorithmique numérique

- méthode d'Euler;
- méthode de Gauss avec recherche partielle du pivot;
- interpolation polynomiale de Lagrange.

Comparaison informatique du tronc commun (MP, PSI, PT) / TSI

- Suppression des points très théoriques
 - réorganisation des TPs;
 - quelques items sur les BDD;
 - bases des dictionnaires;
 - programmation dynamique.

- Remplacement par de l'ingénierie numérique
 - dichotomie;
 - intégration numérique;
 - euler;
 - interpolation polynomiale.

Les outils au programme : Python

Pourquoi Python ?

Langage et interfaces logicielles libres

Langage informatique impératif, généraliste

Langage interactif

Pourquoi pas le C ?

Trop complexe pour une initiation bac+2, pas interactif.

Les outils au programme : Python

Attention !!

L'objectif du programme n'est pas d'apprendre Python !!

L'objectif est d'apprendre à programmer dans un langage impératif, de façon à pouvoir s'adapter à n'importe quel langage par la suite...

A éviter...

Éviter les formes de programmation exclusives à Python :

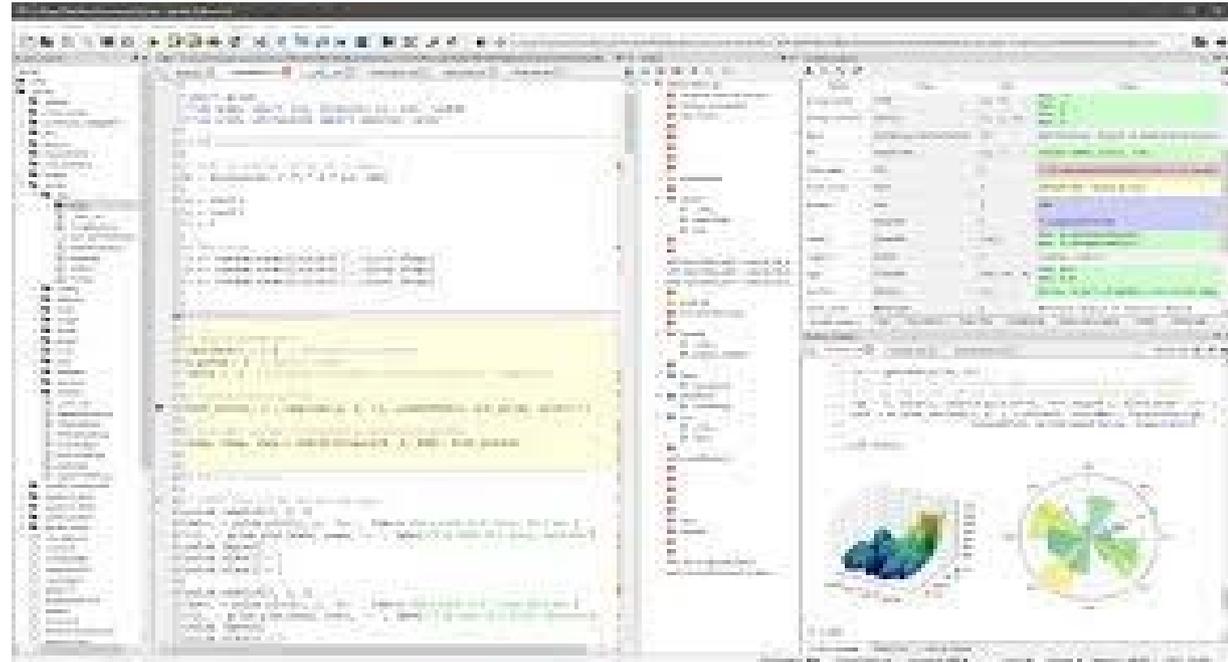
```
y=[sin(x) for x in range(12)]  
g = lambda x: x*2
```

Python & Spyder

Choix de version...

Python 3.x

Installation



Linux / Mac / Windows, installation portable.

Ressources:

Site de l'UPSTI



teaching sciences
for innovation

Ressources en SI, mais aussi en informatique :

Espace adhérent > Documents pédagogique > Informatique

Une centaine de ressources sur des thèmes divers (cours / TP / DS) sur S1, S2 et S3.

TAG: Informatique

L'informatique et les Sciences de l'Ingénieur

- **faire des liens entre les disciplines**
- **évaluation aux oraux de SI des concours**
- **TIPE**

Exemples de sujets de concours

CCP

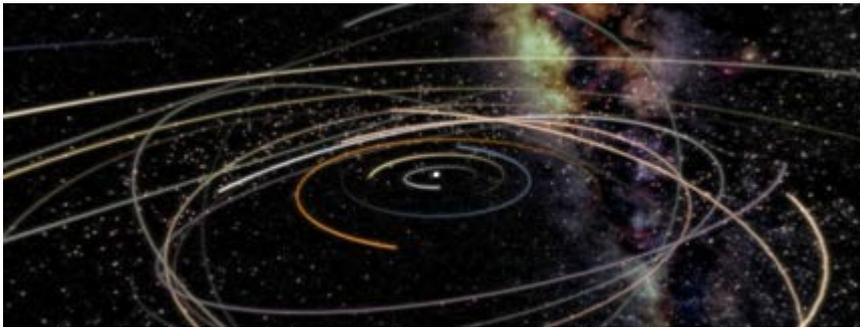
Robot Evolap



Cassini-Huygens



Centrale Problème à N corps



Niveaux de vol / TCAS



Épreuve de 4h pouvant comporter 30 min d'informatique

TP0 : support Maxpid

Activité 7:

- A l'aide de Python ou Scilab, définir et programmer une fonction « loi_geometrique » permettant d'obtenir β en fonction de θ . Tracer l'évolution de β en fonction de θ pour un angle de rotation du bras variant de 0° à 90° .
- Vérifier la courbe obtenue à l'aide du logiciel du bras Maxpid si nécessaire.

```
from pylab import *
```

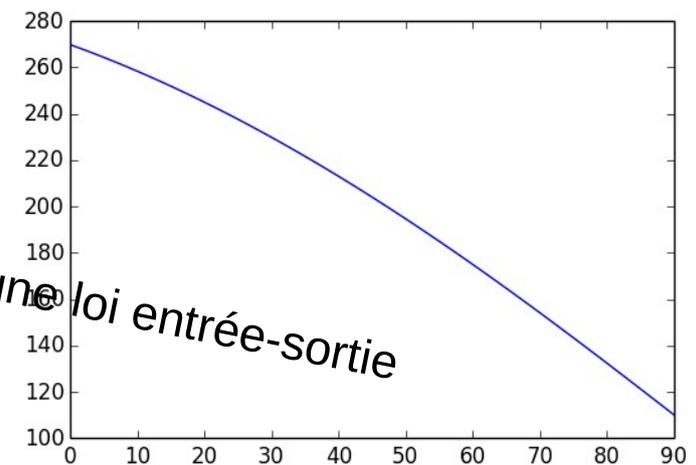
```
a = 70  
b = 80  
l = 82  
p = 4
```

```
### Construction de la loi entree sortie
```

```
def loies(theta):  
    return 2*pi/p*sqrt((a+l*cos(theta))**2+(l*sin(theta)-b)**2)
```

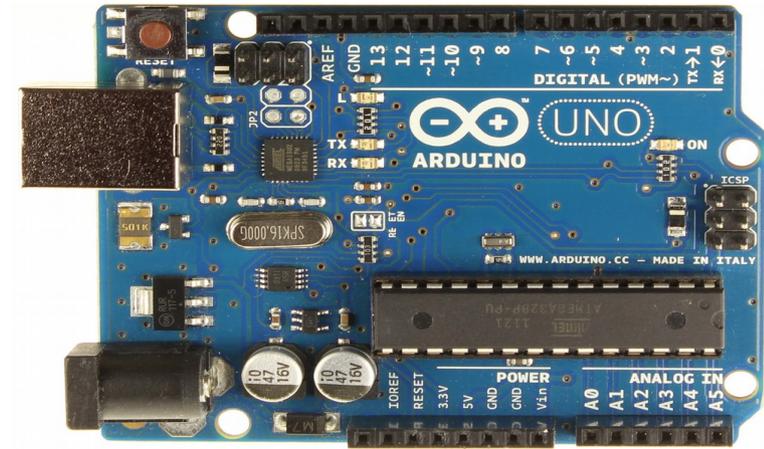
```
theta = linspace(0, pi/2,100)  
beta = loies(theta)
```

```
plot(theta*180/pi,beta)  
show()
```



TIPE

De plus en plus de projets à base d'Arduino



Arduino permet :

- réaliser des mesures : analogiques, numériques, capteurs ;
- piloter des actionneurs : contacteur, led, moteur

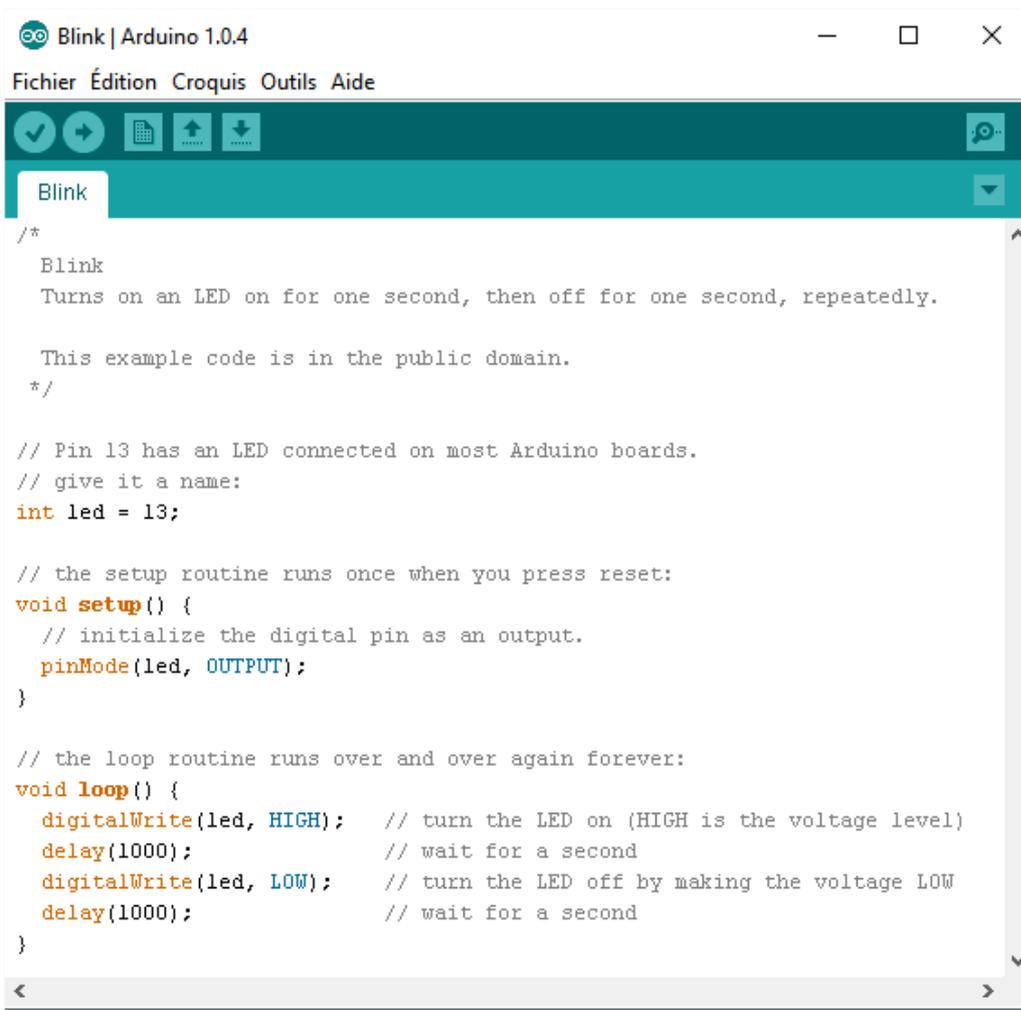
Programmation en langage C de base

- pas toujours simple pour les étudiants
- besoin d'afficher de tracer des graphiques, récupérer les informations obtenues par Arduino

Librairie py2duino : démosciences

Idée : dialoguer avec la carte Arduino depuis Python

Exemple : blink



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The code is written in C++ and includes comments explaining the setup and loop functions. The IDE window title is 'Blink | Arduino 1.0.4' and the menu bar includes 'Fichier', 'Édition', 'Croquis', 'Outils', and 'Aide'. The toolbar shows icons for saving, opening, and running. The code is as follows:

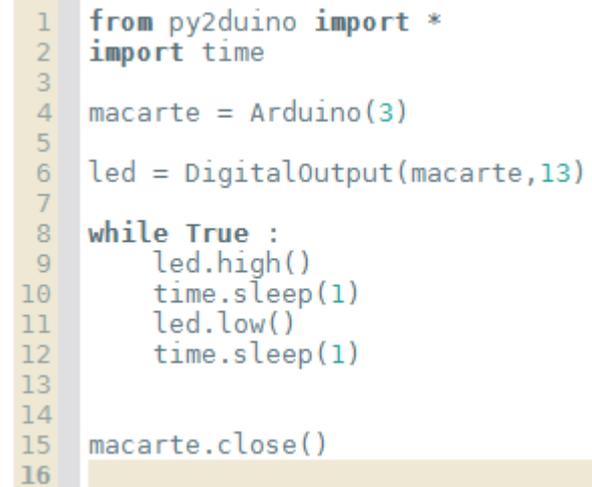
```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Supprime la difficulté syntaxique du C



The code snippet shows the Python equivalent of the Blink example using the py2duino library. The code is as follows:

```
1 from py2duino import *
2 import time
3
4 macarte = Arduino(3)
5
6 led = DigitalOutput(macarte,13)
7
8 while True :
9     led.high()
10    time.sleep(1)
11    led.low()
12    time.sleep(1)
13
14
15 macarte.close()
16
```

Merci pour votre attention

Thibault.gorris@upsti.fr