

Groupe Numérique Disciplinaire (GND)

en Mathématiques

Février 2021

BENRIDA Mustapha

Professeur de mathématiques

Lycée Jean AICARD – HYERES – Var

PONSONNET Luc

Professeur de mathématiques

Lycée BONAPARTE - TOULON - Var

Objectifs pédagogiques :

De la classe de seconde à la classe de terminale, il est très difficile de mener des TP de programmation pour traiter les algorithmes exigibles dans les programmes scolaires.

Bien souvent les élèves ne possèdent pas les bases de programmation Python. Cet article a pour objectif de proposer aux collègues une méthode afin de mettre en place des séquences de programmation hors de la classe avec l'outil Jupyter Notebook.

Outils utilisés :

Conception des notebooks à l'aide de Jupyter Notebook. Récupération éventuelle des travaux des élèves grâce aux espaces de dépôt Moodle ou Pronote.

Utilisation (seulement sur l'ordinateur de l'enseignant) :

- D'Anaconda afin d'installer un environnement de programmation Python et Jupyter Notebook.
- De la plateforme GitHub dans le but de stocker et partager les Notebooks.
- De Binder qui propose de déployer un environnement Jupyter en ligne sans aucune installation de la part des élèves.
- Pour ceux qui le souhaitent, d'un éditeur Latex (Texmaker ou MikTex) afin de créer des pdf à partir des Jupyter notebooks.

NB : L'élève n'a besoin que d'un accès internet, et il commence le TP de programmation Python par un simple clic sur un lien hypertexte Binder.

Voie : générale ou technologique

Niveau(x) de classe : tous niveaux du lycée

Thématique(s) du programme : algorithmique et programmation Python

SOMMAIRE

- 1) Des exemples de TP Jupyter Notebooks et des mises en œuvre possibles
- 2) Installer Anaconda ou Miniconda. Comment créer un Jupyter Notebook ?
- 3) Créer un compte GitHub et un environnement en ligne avec Binder
- 4) Comment créer un *pdf* à partir d'un Jupyter Notebook ?
- 5) Annexes
 - a) Annexe 1 : consignes élèves concernant les TP Jupyter Notebooks
 - b) Annexe 2 : installation de Jupyter Notebook avec les distributions Anaconda ou Miniconda sous Windows
 - c) Annexe 3 : exemple de pdf créé en passant par un fichier .tex

1) Des exemples de TP Jupyter Notebooks et des mises en œuvre possibles

Une fois le TP Jupyter Notebook créé, il est très facile de le mettre à disposition des élèves par l'intermédiaire d'un lien Binder (exemple : <u>https://mybinder.org/v2/gh/lucponsonnet/TP1 Monte Carlo.git/HEAD</u>). Après avoir cliqué sur ce lien, l'élève lancera le TP Jupyter Notebook en cliquant sur le fichier d'extension *.ipynb* (ici *tp_monte_carlo.ipynb*) qui s'ouvrira à l'aide du navigateur par défaut (Microsoft Edge, Chrome ou Mozilla).

	Visit repo	Copy Binder link	Quit
Files Running Clusters			
Select items to perform actions on them.		Upload	New - 2
	Name 🕹	Last Modified	File size
🗆 🛢 tp_monte_carlo.ipynb		il y a 11 jours	10.3 kB
🗅 🗅 image1.png		il y a 11 jours	4.67 kB
Image2.png		il y a 11 jours	7.95 kB
🗆 🗅 image3.png		il y a 11 jours	15 kB
bimage4.png		il y a 11 jours	6.98 kB
Image5.png		il y a 11 jours	58.9 kB
🗋 🗅 image6.png		il y a 11 jours	83.6 kB
the requirements tot		il y a 11 jours	12 B



B

Il est préférable de traiter un premier TP Jupyter Notebook avec vos élèves. Ils apprendront ainsi à :

- lancer le TP en cliquant sur le fichier d'extension .ipynb.
- enregistrer de temps en temps leur travail à l'aide de l'icône disquette
- Actualiser la page si la connexion avec le serveur s'est interrompue (touche F5 ou l'icône C du navigateur).

- A bien comprendre la nature des différentes cellules qui composent le TP (*cellule Markdown, cellule Code* et *Cellule Texte brut*) pour :
 - Exécuter une *cellule Code* par l'icône Exécuter ► Exécuter ou le réaccourci clavier *CTRL* + *Entrée*.
 - Apprendre à écrire leurs réponses dans les *cellules Texte brut*.
- A enregistrer leur travail au format *.ipynb, .html* ou *.pdf* dans le but de le rendre sur un espace de dépôt Moodle ou Pronote (voir *annexe 1 : consignes élèves concernant le premier TP Jupyter Notebook*).

Une fois cet « aspect technique dépassé », les élèves pourront travailler de chez eux les différents TP Jupyter Notebook, et apprendre à consolider seuls les bases de la programmation Python. Une correction pourra leur être proposée soit au format *.ipynb* soit au format *.pdf* (voir paragraphe **4)** Comment créer des pdf à partir de Jupyter Notebooks).

Voici 7 TP Jupyter Notebook dont le but est de faire retravailler l'ensemble des bases de la programmation Python :

- 1. TP sur les booléens : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_booleen/HEAD</u>
- 2. TP sur les instructions conditionnelles : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_if_else/HEAD</u>
- 3. TP sur la boucle for : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_boucle_for/HEAD</u>
- 4. TP sur la boucle while : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_boucle_while/HEAD</u>
- 5. TP sur les fonctions : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_fonctions/HEAD</u>
- 6. TP sur les listes : <u>https://mybinder.org/v2/gh/lucponsonnet/tp_listes/HEAD</u>
- 7. TP sur les chaînes de caractères : https://mybinder.org/v2/gh/lucponsonnet/tp_chaines/HEAD

Ces TP conviendront tout particulièrement aux élèves des classes de première et terminale. Une fois les contenus compris, ils pourront refaire ces TP plusieurs fois dans l'année pour assimiler véritablement les structures algorithmiques de base voire même développer certains automatismes liés à la programmation Python.

Certains TP pourront être commencés en classe (dans une salle informatique ou une salle normale avec des tablettes), puis les élèves devront les terminer chez eux avec remise éventuelle à l'enseignant du TP fini au format .ipynb, .html ou .pdf.

En voici deux exemples :

- 1. TP tracer une courbe avec Matplotlib : <u>https://mybinder.org/v2/gh/gitbenrida/Tracer Courbe de Fonction avec Python.git/HEAD</u>
- 2. TP méthode de Monte-Carlo : <u>https://mybinder.org/v2/gh/lucponsonnet/TP1_Monte_Carlo.git/HEAD</u>

Quelques remarques :

- L'utilisation du TP Jupyter Notebook est très efficace auprès des élèves et d'une grande facilité d'utilisation.
- Les élèves n'ont pas à retaper le script. Cela permet de gagner du temps et d'éviter les trop nombreuses erreurs de syntaxes lors de l'écriture des lignes du programme. Mais il est parfois utile que les élèves réécrivent le code, cela permet de faciliter la mémorisation. C'est pourquoi, l'énoncé devra contenir des scripts sous forme d'images pour éviter les copier-coller.
- Les explications sont « dynamiques » et les pages du TP interactives. On peut illustrer très facilement un exemple ou nos propos à l'aide d'une *cellule Code* qu'ils auront juste à exécuter. Les images et les liens hypertextes sont facilement intégrables dans le TP.
- Tout est présent sur un même support, les énoncés (*cellule Markdown*), les zones de programmation ou de Shell (*cellule Code*), les zones de réponse (*Cellule Texte brut*).
- Les exploitations pédagogiques d'une *cellule Code* sont nombreuses (aide à faire découvrir une instruction, l'utiliser sous forme de Shell ou de programmation, permet de demander de corriger un script faux ou de compléter un script incomplet...).

2) Installer Anaconda ou Miniconda. Comment créer un Jupyter Notebook ?

Etape 1 : installer Jupyter Notebook sur votre ordinateur personnel

Le plus facile est d'installer Anaconda qui contient Jupyter Notebook. Choisir la bonne version (sous Windows, généralement la version 64 bits mais cela dépend de l'architecture de l'ordinateur) de l'exécutable au lien suivant : <u>https://www.anaconda.com/products/individual</u>

Home	Applications on base (root)	✓ Channels				Refre
Environments	*	\$	\$	\$	\$	
Learning	\bigcirc	E	Ŭ.	lab	Jupyter	
Community	CMD.exe Prompt	Datalore	IBM Watson Studio Cloud	JupyterLab	Notebook	
	0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated	Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.	IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and	2.2.6 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	6.1.4 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	
	Launch	Launch	science tools or visual modeling.	Launch	Launch	
	° °	¢ IP[y]:	*	ı Î	*	
	Powershell Prompt	Qt Console	Spyder	Glueviz	Orange 3	
	Run a Powershell terminal with your current environment from Navigator activated	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Scientific PYthon Development EnviRonment: Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	
Join Now	Launch	Launch	Launch	Install	Install	
cover premium data science content	*	•				
Documentation		R				
Anaconda Blog	PyCharm Professional	RStudio				
- V	A full-fledged IDE by JetBrains for both	1.1.456 A set of integrated tools designed to help you be more productive with P. Includes P.				

Il est possible d'installer Jupyter Notebook par l'intermédiaire de Miniconda, mais c'est beaucoup plus compliqué.

Voir annexe 2 : installation de Jupyter Notebook avec les distributions Anaconda ou Miniconda sous Windows.

Etape 2 : Créer un Jupyter Notebook

Vous pourrez alors lancer Jupyter Notebook en cliquant sur Launch de l'icône

Jupyter Notebook s'ouvre dans votre navigateur par défaut à l'adresse <u>http://localhost:8888/tree#notebooks</u> :

💭 jupyter	Quit Logout
Files Running Clusters	
Select items to perform actions on them.	Upload New - 3
0 -	Name Last Modified File size
Ch 3D Objects	il y a 4 mois
anaconda3	il y a 13 jours
Desktop	il y a 2 mois
Coursents	il y a 2 mois
Downloads	il y a 2 jours
Dropbox	il y a 2 mois
E Favorites	il y a 4 mois
C Links	il y a 4 mois
mes_notebooks	il y a 6 jours

Nous vous conseillons de créer un dossier *mes_notebooks* dans « *C* : *Utilisateurs**votre_nom* » pour stocker tous vos notebooks. Pour créer ce dossier il faudra aller dans le répertoire qui porte votre nom *votre_nom* car il ne sera pas possible de le faire avec l'interface de Jupyter. Il est aussi possible de le créer à l'aide de Jupyter Notebook par *New/Folder*.

Cliquer alors sur New/Python 3 pour créer un nouveau Jupyter Notebook :

Uploa	New -
Notebook:	
Python 3	e
	Créer un nouveau notebook avec Python 3
Other:	
Text File	
Folder	
Terminal	



La première cellule qui apparaît est une cellule Code.

Vous pouvez ajouter + ou supprimer autant de cellules que vous souhaitez aux trois « formats » suivants :

- cellule Code : Shell ou zone de programmation
- cellule Texte brut : zone où les élèves pourront taper leurs réponses sans format de texte (ni éditeur d'équations).
- cellule Markdown : zone où vous allez écrire votre texte en langage Markdown, et les formules mathématiques en Latex. Le langage html est aussi supporté mais ne sera pas conservé si vous souhaitez convertir votre Jupyter Notebook au format pdf (via un fichier Latex ou directement). Vous pouvez aussi insérer dans ce type de cellule des liens hypertextes et des images.

Voici quelques liens pour apprendre à compléter votre cellule Markdown à l'aide :

- du langage de balisage Markdown :
 - <u>https://github.com/capytale/capytale/blob/master/PRES%20-</u> %20Bases%20pour%20rediger%20un%20notebook.ipynb
 - <u>https://seps.flibuste.net/markdown_help</u>
 - <u>https://wprock.fr/blog/markdown-syntaxe/</u>
- de formules mathématiques écrites en Latex :
 - <u>https://fr.wikipedia.org/wiki/Aide:Formules_TeX</u>
 - o <u>Online Latex Equation Editor Sciweavers</u>

Quelques remarques utiles :

• Il est possible de numéroter les lignes de code des cellules Code par View/Toggle Line Number



Avant de déposer votre notebook sur GitHub, penser à bien effacer les *Output* de vos scripts par *Kernel/Restart* & *Clear Output* (ou *Cell/All Output/Clear*).

Kernel	Widgets	Help
Interrupt Restart		'n
Restart	& Clear Outp	ut
Restart Reconne	& Run _{Restart} act	the Kernel and clear all output
Shutdow	/n	ateurs de c
Change	kernel	nit True soit Fale

3) Créer un compte GitHub et un environnement en ligne avec Binder

Etape 1 : ouvrir un compte GitHub

Créer un compte gratuit sur la plateforme GitHub où nous allons stocker nos Jupyter Notebooks.



Au besoin, vous pouvez lire la page suivante : <u>https://openclassrooms.com/fr/courses/5641721-utilisez-git-et-github-pour-vos-projets-de-developpement/6113011-demarrez-votre-projet-avec-github</u>

Etape 2 : créer un New repository et y déposer votre TP Jupyter Notebook

Une fois connecté à votre compte Git	Hub par Sign in , cliquer sur 🖳 New ou sur +, e	et sélectionner New repository :
	Q + ▼ ● New repository Import repository New gist New organization New project New project	
	Create a new repository	
	Owner * Repository name *	1- Ecrire le nom de votre TP
	Great repository names are short and memorable. Need inspiration? How about psychic-eure Description (optional)	ka?
	Public Anyone on the internet can see this repository. You choose who can commit. Private You choose who can see and commit to this repository.	2- Laisser par défaut : <i>Public</i>
	Initialize this repository with: Skip this step if you're importing an existing repository.	
	Add a README file This is where you can write a long description for your project. Learn more.	
	Add .gitignore Choose which files not to track from a list of templates. Learn more.	3 – Sélectionner Add a README file
	Choose a license A license tells others what they can and can't do with your code. Learn more.	
	This will set 🐉 main as the default branch. Change the default name in your settings.	
	Create repository	

Vous pouvez alors glisser-déposer tous vos éléments (généralement votre fichier au format ipynb, des images et éventuellement un fichier *requirements.txt* (voir remarque ci-après) qui composent votre TP ou bien sélectionner *Add file/Upload files* par :



Une remarque très importante :

Il sera parfois nécessaire d'ajouter dans le dépôt GitHub un fichier texte qui sera nommé obligatoirement *requirements.txt* (que l'on pourra créer par exemple avec *Bloc-note* ou à l'aide GitHub même par *Add file/Create new file*) pour que l'environnement généré par Binder (voir étape suivante) puisse accéder aux librairies *Matplotlib* ou *Numpy*. Il n'est pas nécessaire de le faire pour les modules *Math* et *Random*.

🖵 lucpons	sonnet / TP	1_Monte_Carlo		
<> Code	() Issues	🖏 Pull requests 🕑 Actions 🛄 Pr	rojects 🖽 Wiki 🕕 Security	🗠 Insights 🖇 Settings
		🐉 main 👻 🐉 1 branch 💿 0 tags	5	Go to file Add file - 💆 Code -
		🐣 lucponsonnet Update requirement	s.txt	bf92211 11 days ago 🛛 36 commits
		🗅 image1.png	Add files via upload	16 days ago
		🗅 image2.png	Add files via upload	16 days ago
		🗅 image3.png	Add files via upload	16 days ago
		🗅 image4.png	Add files via upload	12 days ago
		🗅 image5.png	Add files via upload	16 days ago
		🗅 image6.png	Add files via upload	16 days ago
		🗅 requirements.txt 🗡	Update requirements.txt	11 days ago
		tp_monte_carlo.ipynb	Add files via upload	14 days ago
		Help people interested in this repository	r understand your project by adding a R	EADME. Add a README

Voici par exemple le contenu du fichier requirements.txt du TP1_Monte_Carlo :

requirements - Bloc-notes	_	0 ×
Fichier Edition Format Affichage Aide		
matplotlib		^
<		>
Ln 2, Col 1 100% Wind	lows (CRLF) UTF-8	

Remarque :

Avec la commande *help('modules')*, il est possible d'obtenir les modules de Python « natifs » dans Jupyter Notebook.

💭 jupyte	er Untitled Dernie	ère Sauvegarde : il y a une n	ninute (modifié)				~	Visit repo	Copy Binder
File Edit	View Insert	Cell Kernel Widg	ets Help					Fiable	Python
8 + %	∞ 16 ↑ ↓	H Exécuter	Code 🗸	🖾 🛓 Download	0 0	O GitHub	% Binder	Memo	ry: 235 / 2048
Entrée [1]: help('modules')							
	Please wait a	moment while I gather	a list of all available	modules					
	/srv/conda/env has been depre "You should /srv/conda/env fore Setuptool rectly or at 1 "Distutils w	s/notebook/lib/python cated since IPython 4 import from ipykernel s/notebook/lib/python s. This usage is disc east import Setuptool as imported before Se	3.7/site-packages/IPythe .0.You should import fro or jupyter_client inste 3.7/site-packages/setupt ouraged and may exhibit s first. tuptools. This usage is	<pre>in/kernel/init m ipykernel or ju ad.", ShimWarning cools/distutils_pa undesirable behav discouraged "</pre>	.py:13: S pyter_cli :) tch.py:26 iors or e	ShimWarning lent instea 5: UserWarn errors. Ple	: The `IPyt d. ing: Distut: ase use Set	hon.kernel` ils was imp uptools' ob	package orted be jects di
	IPython OpenSSL future _abc	asyncio asyncore atexit attr	<pre>ipykernel_launcher ipython_genutils ipywidgets itertools</pre>	requests resource rlcompleter rmagic					
	_ast _asyncio	audioop autoreload	jedi jinja2	runpy sched					

Etape 3 : créer l'environnement en ligne grâce à Binder

Binder va déployer un environnement Jupyter en ligne à partir du TP notebook qui a été déposé sur GitHub. Il pourra être utilisé par les élèves sans aucune installation préalable par un simple clic sur un lien Binder.

Lorsque vous êtes dans le répertoire de votre TP sur la plateforme GitHub. Copier l'adresse GitHub de ce TP.

Ici : <u>https://github.com/lucponsonnet/TP1 Monte Carlo</u>

Nouvel onglet	X C Home Page - Select or create a n X	tp_bool - Jupyter Notebook ×	Iucponsonnet/TP1_Monte_Carlo × +		- 0 ×
$\leftarrow \rightarrow C \cap (= github.c$	com/lucponsonnet/TP1_Monte_Carlo				🔤 🖈 🖻 i
Applications					
Search or jump to	Pull requests Issues	Marketplace Explore			Q +• 😔•
🖟 lucponsonnet / TP1	1_Monte_Carlo			⊙ Unwatch → 1	A Star 0 양 Fork 0
<> Code () Issues	👫 Pull requests 🕞 Actions 🗐 Proj	ects 🖽 Wiki 🕕 Security	∠ Insights 爺 Settings		
	양 main 👻 양 1 branch 💿 0 tags		Go to file Add file - Code -	About	\$
	B lucponsonnet Update requirements.	xt	6492211 11 days ago 🛛 36 commits	No description, website, or topics provided.	
	🗅 image1.png	Add files via upload	16 days ago		
	🗅 image2.png	Add files via upload	16 days ago	Releases	
	🗅 image3.png	Add files via upload	16 days ago	No releases published	
	🗅 image4.png	Add files via upload	12 days ago	Create a new release	
	🗅 image5.png	Add files via upload	16 days ago		
	🗅 image6.png	Add files via upload	16 days ago	Packages	
	requirements.txt	Update requirements.txt	11 days ago	No packages published Publish your first package	
	tp_monte_carlo.ipynb	Add files via upload	14 days ago		

Puis aller sur Binder en suivant ce lien : <u>https://mybinder.org/</u> et copier l'adresse GitHub de votre TP dans la barre intitulée *Github repository name ou URL*.

	C	8 binder	
	Turn a Git rep	o into a collection of	f interactive
		notebooks	
	Have a repository full of Jupyt environment, making	er notebooks? With Binder, open those no your code immediately reproduciple by an	tebooks in an executable yone, anywhere.
	New to Binder? Get s	arted with a Zero-to-Binder tutorial in Ju	lia, Python or R.
	Build and launch a repository GitHub repository name or URL GitHub • https://github.com/lucponso	nnet/TP1_Monte_Carlo	
	Git ref (branch, tag, or commit)	Path to a notebook file (optional)	
	HEAD	Path to a notebook file (optional)	File - launch
	Copy the URL below and share your Binder	r with others:	
	https://mybinder.org/v2/gh/lucponso	nnet/TP1_Monte_Carlo/HEAD	Ê
	Copy the text below, then paste into your 1	README to show a binder badge; 🔗 launch binder	•
Cliquer sur <i>launch</i> .			

Le lien suivant pourra être donné à vos élèves !

- 4) Comment créer un pdf à partir d'un Jupyter Notebook ?
- a) Sur votre ordinateur personnel
- Vous devez avoir préalablement installé MikTex : <u>https://miktex.org/download</u> . Mettre à jour MiKTek : par *MiKTek Console /Updates / Check for Updates*
- On peut obtenir un pdf grâce à la commande File/Download as/PDF via LaTex (.pdf) :



Il sera parfois nécessaire d'installer la dernière version de Jupyter en suivant la démarche suivante :

- Dans Qt Console d'Anaconda, taper : conda update --all.



- Valider en appuyant sur la touche Entrée.



- Attendre un moment, il apparaît alors la fenêtre suivante :

```
Jupyter QtConsole
                                                                               \times
File Edit View Kernel Window Help
Jupyter OtConsole 4.7.7
                                                                                         ~
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.
In [1]: conda update --all
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done
## Package Plan ##
  environment location: C:\Users\USER\anaconda3
The following packages will be downloaded:
    package
                                             build
    backports.functools_lru_cache-1.6.1 pyhd3eb1b0_0
                                                                     12 KB
    conda-build-3.21.4 | py38haa95532_0 552 KB
                                            Total:
                                                           564 KB
The following packages will be UPDATED:
  conda-build
                                               3.20.5-py38_1 --> 3.21.4-py38haa95532_0
The following packages will be DOWNGRADED:
  backports.functoo~
                                                  1.6.1-py_0 --> 1.6.1-pyhd3eb1b0_0
Downloading and Extracting Packages
backports.functools_ | 12 KB
                                                   9%
backports.functools_ | 12 KB
backports.functools_ | 12 KB
                                  ######### | 100%
                                  ######### | 100%
conda-build-3.21.4 | 552 KB
                                                  0%
conda-build-3.21.4
                     552 KB
                                  Т
                                   #1
                                                  12%
conda-build-3.21.4
                    552 KB
                                                 20%
                                  ##
conda-build-3.21.4 | 552 KB
                                  ####3
                                               43%
                    552 KB
                                  ######3
                                                64%
conda-build-3.21.4
conda-build-3.21.4
                     552 KB
                                   ########1
                                  Т
                                                 81%
conda-build-3.21.4 | 552 KB
                                  ########## | 100%
                                  | ######### | 100%
conda-build-3.21.4 552 KB
Note: you may need to restart the kernel to use updated packages.
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done
In [2]:
```

- Redémarrer Anaconda par Ctrl + R.

Remarques importantes :

- Pour ceux qui s'y connaissent un peu en Latex, il sera parfois préférable d'obtenir à partir du Notebook de Jupyter le fichier .tex intermédiaire (*File/Download as/Latex (.tex*)) avant de générer le pdf. Ce fichier .tex nous permettra de modifier :
 - la taille des images en remplaçant l'instruction \includegraphics {image1.png} par \includegraphics[width=5cm]{ image1.png}. Cela qui permettra de redimensionner la taille de l'image comme on le souhaite.
 - Concernant les titres #, ##, ##..., si vous avez ajouté des numérotations (exemple : ## 1-Introduction), il faudra les enlever car lors de la compilation en pdf, une numérotation propre au fichier Latex sera mise en lien avec l'écriture Markdowon des titres.
 - Seuls le langage Markdown et les formules Latex seront correctement convertis en pdf donc pas le langage en html.

b) Sur l'ordinateur d'un élève

- Créer un fichier au format html par File/Download as/HTML (.html).
- Déposer ce fichier .html dans un dossier qui contient toutes les images du TP (ils pourront les télécharger de l'environnement Binder).
- Puis imprimer au format pdf avec le navigateur cette page web. Par exemple avec Chrome :

	0	-		5	×	
	ଷ୍	¥		e		
Nouvel onglet				Ct	rl+T	
Nouvelle fenêtre de	navigation	privée	Ct	rl+Ma	n+N j+N	
Historique						Þ
Téléchargements				C	irl+J	
Favoris						۲
Zoom	-	150 %	+		53	
Imprimer				Ct	rl+P	
Caster						
Rechercher				Ct	rl+F	
Plus d'outils						۲
Modifier	Couper	Cop	oier	0	Oller	
Paramètres						
Aide						۲

S'ils n'ont pas de créateur de pdf, ils pourront installer pdf creator : https://pdfcreator.fr/

c) Remarque

Il est possible d'obtenir un pdf à partir d'un notebook via un fichier html. Mais l'enseignant aura intérêt à installer MikTex car ainsi il pourra faire des modifications via ce fichier Tex avant la compilation en un pdf de qualité (voir un exemple, **annexe 3** : **exemple de pdf créé en passant par un fichier .tex**).

5) Annexes

a) Annexe 1 : consignes élèves concernant les TP Jupyter Notebooks

Ouvrir le TP

- Cliquer sur le lien suivant et attendre que l'environnement de travail soit créé par Binder : <u>https://mybinder.org/v2/gh/lucponsonnet/TP1_Monte_Carlo.git/HEAD</u>
- Puis cliquer sur le fichier d'extension .*ipynb* : *tp_monte_carlo.ipynb*.

1) Pour lancer une *cellule code* et compléter une *cellule texte*

• Cellule Code (où il est marqué Entrée []) :

Cliquer dans la *cellule Code*, puis cliquer sur le bouton *Exécuter* (vous pouvez aussi utiliser le raccourci clavier CTRL + Entrée) :

Cjupyter tp_monte_carlo (modifié)	Visit repo Copy Binder link
File Edit View Insert Cell Kernel Widgets Help	Non fiable 🖋 Python 3 O
🖺 🕂 🎉 🛍 🖍 🔸 🕅 Exécuter 🔳 C 🅨 Code	V Download 🛆 🗠 🕥 GitHub % Binder Memory: 115 / 2048 MB

• Cellule Texte brut :

Vous devrez saisir dans la *cellule Texte* la réponse à la question posée. Parfois, il ne sera pas possible d'écrire un exposant. A vous d'être astucieux et d'écrire par exemple OA^2 au lieu de OA^2 qui n'est pas possible d'écrire dans le texte brut.

De temps en temps :

- A l'aide de l'icône disquette, il sera nécessaire d'enregistrer votre travail.
- D'actualiser la page si la connexion avec le serveur s'est interrompue (touche F5 ou l'icône ^C du navigateur).

2) Pour enregistrer le TP une fois terminé au format html

Pour enregistrer votre TP au format html

• Cliquer sur le titre du TP, et renommer le TP en y ajoutant votre NOM et prénom :



Renommer le Notebook	3
Saisir le nouveau nom du notebook	
tp_monte_carlo_PONSONNET_Luc	
	Annuler Renommer

• Enregistrer votre TP par *File/Download as/HTML (.html)*. Il sera enregistré par défaut dans le dossier *Téléchargements*. Si on le souhaite, il est possible de faire choisir le lieu de téléchargement.

Pour déposer le fichier sur Pronote ou Moodle •

Récupérer votre fichier TP en cliquant sur la flèche puis sur le dossier.	

\leftrightarrow \rightarrow C $rac{1}{2}$	Q about:blank	× III\ □ © ≡
6 Débuter avec Firefox		tp_monte_carlo.html Ouvrir le dossier contenant le fichier
		Afficher tous les téléchargements

Vous pouvez déposer votre fichier tp monte carlo.html dans un dépôt Moodle ou Pronote selon les consignes de votre enseignant.

Remarque : il se peut que les images n'apparaissent plus dans le fichier .html car le fichier html n'est plus dans le répertoire qui contient les images du TP.

3) Pour enregistrer le TP au format pdf

- Créer un fichier au format html par File/Download as/HTML (.html). _
- Déposer ce fichier .html dans un dossier qui contient toutes les images du TP (vous pourrez les télécharger de l'environnement Binder).
- Puis imprimer au format pdf avec le navigateur cette page web. Par exemple avec Chrome :

		-	1	5	×
	O	$\dot{\nabla}$		P	:
	~	м	~		•
Nouvel onglet				Ctrl+	T
Nouvelle fenêtre				Ctrl+	N
Nouvelle fenêtre d	e navigation	privée	Ctr	l+Maj+	N
Historique					•
Téléchargements				Ctrl	+J
Favoris					•
Zoom	-	150 %	+	1	3
Imprimer				Ctrl+	P
Caster					
Rechercher				Ctrl	• F
Plus d'outils					•
Modifier	Couper	Сор	ier	Co	ller
Paramètres					
Aide					•
Quitter					

S'ils n'ont pas de créateur de pdf, ils pourront installer pdf creator : https://pdfcreator.fr/

Pour enregistrer le TP au format ipynb 4)

- Enregistrez votre travail sous la forme TP_NOM_Prenom, -
- Vous pouvez aussi remettre votre TP à votre professeur au format .ipynb par :



b) Annexe 2 : installation de Jupyter Notebook avec les distributions Anaconda ou Miniconda sous Windows.

A) Anonconda VS Miniconda ?

Le plus simple pour installer *Jupyterlab* et *Jupyter Notebook* est d'installer la distribution *Anaconda* (qui contient le gestionnaire de paquets *Conda*, plus les bibliothèques scientifiques, plus un environnement de développement...), seulement *Anaconda* nécessite un espace disque de plusieurs giga-octets.

La distribution *Miniconda* est une version allégée d'*Anaconda* qui contient aussi le gestionnaire de paquets *Conda*. Elle occupe moins d'espace disque mais comme vous le verrez, l'installation de *Jupyterlab* et *Jupyter Notebook* est beaucoup plus difficile par Miniconda.

B) Avec Anaconda

- 1) <u>Télécharger le fichier qui va vous permettre d'installer Anaconda</u>
 - Télécharger le fichier *Anaconda3-2020.11-Windows-x86_64.exe* (ou 32 bits selon votre ordinateur) au lien suivant : <u>https://www.anaconda.com/products/individual#windows</u>
 - Lancer l'installation de Anaconda en double cliquant sur ce fichier, et suivre les indications de chacune des fenêtres d'installation.
 - Vous constaterez dans le menu Démarrer la nouvelle rubrique suivante :

📙 Anaconda3 (64-bit)	🕨 🌔 Anaconda Navigator (anaconda3) 🛛 🙀
	📟 Anaconda Powershell Prompt (anaconda3)
	🕋 Anaconda Prompt (anaconda3)
	💭 Jupyter Notebook (anaconda3)
	📄 Reset Spyder Settings (anaconda3)
	🛞 Spyder (anaconda3)

2) Lancer JupyterLab et Jupyter Notebook dans le navigateur web par défaut

Le plus facile est de démarrer le navigateur d'Anaconda en cliquant sur *Anaconda Navigator (anaconda3)*. On obtient alors :



C) Avec Miniconda

- 1) <u>Télécharger le fichier qui va vous permettre d'installer Miniconda</u>
 - Télécharger le fichier *Miniconda3-latest-Windows-x86_64.exe* (ou la version 32 bits selon votre ordinateur) au lien suivant : <u>https://docs.conda.io/en/latest/miniconda.html</u>
 - Lancer l'installation de Miniconda en double cliquant sur ce fichier, et suivre les indications de chacune des fenêtres d'installation.
 - Vous constaterez dans le menu Démarrer la nouvelle rubrique suivante :

Anaconda3 (64-bit)	►	📟 Anaconda Powershell Prompt (miniconda3)
		🔤 Anaconda Prompt (miniconda3)

- 2) <u>Utilisation de Conda pour installer des modules complémentaires dont Jupyterlab</u>
 - Lancer le Shell ci-dessus en cliquant sur Anaconda Powershell prompt (miniconda3), écrire dans le Shell la ligne de commande suivante, et appuyer sur la touche Entrée :



- A la question **Proceed ([y]/n)** ?, répondre y.
- 3) <u>Vérifier la bonne installation des modules complémentaires</u>
 - Dans le Shell, lancer l'interpréteur de Python en entrant python, on obtient alors :

Anaconda Powershell Prompt (miniconda3)	-	×
(base) PS C:\Users\USER> python		^
Python 3.8.3 (default, May 19 2020, 06:50:17) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on w	in32	
Type "help", "copyright", "credits" or "license" for more information.		
>>>		

• Puis dans l'interpréteur Python, vérifier la bonne importation des modules par :



S'il n'apparaît aucune erreur comme ici, cela signifie que les modules ont bien été installés. Pour quitter l'interpréteur de Python et revenir au Shell, taper *exit()*.

- 4) Lancer JupyterLab dans le navigateur web par défaut
 - Une fois dans le Shell entrer (attention à bien respecter l'espace) jupyter lab comme ci-dessous :



• Cette commande ouvre votre navigateur web par défaut et démarre JupyterLab.

0	JupyterLab	x +
	ightarrow O G	D localhost:8888/lab/workspaces/auto-e
С	File Edit View Run	Kernel Tabs Settings Help
	+ 🗈 🛨 C	☑ Launcher ×
_	I	
0	Name 🔶	
	3D Objects	Notebook
耹	Contacts	
_	Desktop	
2	Documents	
~	Downloads	Python 3
	Dropbox	- Junio
	Favorites	
	Links	>_ Console
*	miniconda3	
	Saved Games	
	Searches	
	Videos	Python 3
		the Other
		Terminal Text File Markdown File Show Contextual

- On peut obtenir un *Notebook* de *Jupyter* en cliquant sur l'icône *Python 3*.
- Attention de ne pas fermer la fenêtre suivante du Shell :



- 5) Lancer Jupyter Notebook dans le navigateur web par défaut
 - Taper et valider la commande ci-dessous dans le Shell :

Anaconda Powershell Prompt (miniconda3)	_	×
(base) PS C:\Users\USER> jupyter-notebook_		^

• On obtient l'interface de Jupyter Notebook dans le navigateur web par défaut :

C Home Page - Select or create a r x +	- 🗆 X
$\leftrightarrow \rightarrow$ O $\widehat{\omega}$ O localhost:8888/tree	🗴 🏚 🚇 …
💭 jupyter	Quit Logout
Files Running Clusters	
Select items to perform actions on them.	Upload New - 2
	Name 4 Python 3
🗇 🗅 3D Objects	Other:
Contacts	Text File
Desktop	Folder
Documents	Terminat
🗇 🗁 Downloads	il y 4 heures
🗋 🗀 Dropbox	y a un mois
Constant Favorites	il y a 4 mois
Ci Links	il y a 4 mois
D miniconda3	il y a 2 heures
🗋 🗅 Music	il y a 4 mois
CneDrive	il y a 6 heures
Saved Games	il y a 4 mois
C Searches	il y a 4 mois
C Videos	il y a 3 mois

• On obtient par la suite un lien direct vers Jupyter Notebook dans le menu Démarrer :

📟 Anaconda Powershel	l Promp	t (miniconda3)
📟 Anaconda Prompt (m	ninicond	a3)
💭 Jupyter Notebook		

• Pour créer un *Notebook de Jupyter*, cliquer sur *New*, puis sélectionner *Python 3*. On obtient alors le *Notebook* vierge avec une seule cellule et dans un autre onglet du navigateur :

🔿 Home Page - Select or create a 🕫 🗴 🧧 Untitled 1 - Jupyter Notebook 🛛 🗙 🕂			-	×
$\leftrightarrow \rightarrow$ O for localhost:8888/notebooks/Untitled1.ipynb?kernel_name=python3	Å	s∕≡	匝	
C Jupyter Untitled1 Dernière Sauvegarde : il y a quelques secondes (modifié)	e	Logo	ut	
File Edit View Insert Cell Kernel Help	Fiable	Python 3	0	
B + S< 2 K ↑ ↓ ► Exécuter ■ C → Code				
Entrée []:]	*

• Donner un nom à votre *Notebook* en cliquant sur *Untitled*. Si le nom est *mon_notebook*, le fichier aura pour extension *.ipynb*, et il sera enregistré dans le répertoire qui contient le dossier *miniconda3*.

💢 jupyter	Quit	Logout
Files Running Clusters		
Select items to perform actions on them.	Upload	New 🗸 🎗
	Name 🕹 Last Modified	File size
C 3D Objects	il y a 4 mois	
Contacts	il y a 4 mois	
Ca Desktop	il y a 25 jours	
Documents	il y a 25 jours	
Downloads	il y a 5 heures	
🗋 🗅 Dropbox	il y a un mois	
Favorites	il y a 4 mois	
C Links	il y a 4 mois	
🗋 🗅 miniconda3	il y a 4 heures	
C Music	il y a 4 mois	
ConeDrive	il y a 7 heures	
Saved Games	il y a 4 mois	
Carches	il y a 4 mois	
C Videos	il y a 3 mois	
🗋 🧧 mon_notebook.ipynb	Actif il y a une minute	555 B

c) Annexe 3 : exemple de pdf créé en passant par un fichier .tex

Le logiciel MikTek et Anaconda ont été installés et mis à jour sur l'ordinateur personnel (voir pages 11 et 12).

tp_monte_carlo_correction

January 23, 2021

1 Correction du TP informatique : la méthode de Monte-Carlo

1.1 Algorithmes exigibles d'après le programme scolaire

D'après le programme de première spé maths, nous avons à traiter les deux algorithmes suivants :

Méthode de Monte-Carlo : estimation de l'aire sous la parabole, estimation du nombre π .

1.2 Estimation du nombre π par la méthode de Monte-Carlo

1.2.1 Déterminer un critère pour savoir si un point M appartient au quart de disque (D)

Rappel 1 : On considère les points $A(x_A; y_A)$ et $B(x_B; y_B)$ dans un repère (O; I, J) alors $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$ et donc en élevant le tout au carré : $AB^2 = (x_B - x_A)^2 + (y_B - y_A)^2$ (1). En particularisant (1) aux points M(x; y) et O(0, 0), on obtient $OM^2 = x^2 + y^2$ (2). On considère un quart de disque (**D**) de rayon 1 dont le centre O est l'origine d'un repère orthonormé (O; I, J).



Question 1 : On considère les points A(0, 1; 0, 8) et B(0, 7; 0, 8) du repère (O; I, J). A l'aide de la formule (2) et de votre calculatrice, taper dans la *cellule texte* ci-dessous : * les valeurs de OA^2 et OB^2 , * et dire si les points A et B appartiennent au quart de disque (**D**) de centre O et de rayon 1.

 $OA^2 = 0, 1^2 + 0, 8^2 = 0, 65$ donc $OA^2 \le 1$, et on déduit que $A \in (\mathbf{D})$.

 $OB^2 = 0,7^2 + 0,8^2 = 1,13$ donc $OB^2 > 1$, et on déduit que $B \notin (\mathbf{D})$.

En résumé :

Soit M(x;y) un point du plan doté d'un repère orthonormé (O; I, J).

 $M \in (\mathbf{D}) \Leftrightarrow OM \le 1 \Leftrightarrow OM^2 \le 1 \Leftrightarrow x^2 + y^2 \le 1.$

Autrement dit : si $x^2 + y^2 \le 1$ alors le point M(x; y) appartient au quart de disque (D), frontière comprise.

1.2.2 La méthode de Monte-Carlo

Le principe de la méthode de Monte-Carlo consiste à tirer au hasard les coordonnées (x, y) d'un point M avec $x \in [0;1]$ et $y \in [0;1]$.

Le point M appartient soit au quart de disque (D), soit au carré OIKJ.



 $M(x;y) \in (\mathbf{D}) \Leftrightarrow x^2 + y^2 \le 1$ d'après le résumé précédent.

Soit *N* est le nombre de points M(x, y) tirés au hasard donc avec *x* et *y* des réels aléatoires de [0;1]. Soit *n* le nombre de points M(x, y) qui appartiennent au quart de disque **(D)**.

On admettra l'affirmation suivante :

Le rapport $\frac{n}{N}$ donne une approximation du quotient $\frac{\text{Aire de (D)}}{\text{Aire du carré OIKJ}}$.

Or
$$\frac{\text{Aire de (D)}}{\text{Aire du carré }OIKJ} = \frac{\frac{\pi \times 1^2}{4}}{1^2}$$
 car Aire de (D) $= \frac{\pi \times 1^2}{4}$ et Aire du carré $= 1^2$.

En simplifiant, on obtient que $\frac{\text{Aire de (D)}}{\text{Aire du carré }OIKJ} = \frac{\pi}{4}$.

En invoquant l'affirmation (3), on déduit que $\frac{n}{N} \approx \frac{\pi}{4}$. Ainsi $\pi \approx \frac{4n}{N}$.

Question 2 : On suppose que sur 1000 tirages de points M(x, y) de manière aléatoire avec $x \in [0;1]$ et $y \in [0;1]$, 800 d'entre eux appartiennent au quart de disque **(D)**. En déduire une valeur approchée de π que vous saisirez dans la *cellule texte* ci-après :

Nous avons N = 1000 et n = 800 donc $\frac{4 \times 800}{1000} = 3, 2$. Une approximation de π par la méthode de Monte-Carlo est 3, 2.

1.2.3 Implémentation en langage Python de la méthode de Monte-Carlo pour donner une approximation du réel π

Exécuter plusieurs fois la cellule code suivante :

2

```
[1]: from random import *
   random()
```

```
[1]: 0.23980463858600443
```

Question 3 : A l'aide du mémento Python (cliquer sur le lien) dire ce que renvoie précisément la fonction *random()* ? Taper votre réponse dans la *cellule texte* ci-après :

Lorsque nous exécutons plusieurs fois la *cellule code*, nous constatons que la fonction *random()* renvoie une valeur aléatoire comprise entre 0 et 1. Le *Mémento Python* précise que la fonction *random()* renvoie une valeur pseudo-aléatoire qui appartient à [0;1[. On peut remarquer que la borne 1 n'est pas prise.

```
[2]: from random import *
```

```
Monte_Carlo(1000000)
```

[2]: 3.13782

Donner une valeur approchée de π à l'aide de l'appel de fonction *Monte_Carlo(1000000)*. Saisir votre résultat dans la *cellule texte* ci-après :

Le script précédent donne par exemple $\pi \approx 3,13782$.

Exécuter le programme suivant qui : - affiche 1000 points aléatoires. - donne une approximation de π par la méthode de Monte-Carlo.

```
[3]: from random import *
    import matplotlib.pyplot as plt

def Monte_Carlo(N):
    plt.axis('equal')
    plt.xlabel("x")
    plt.ylabel("y")
    n=0
    for i in range(N):
        x=random()
        y=random()
        if x**2+y**2<=1:
            plt.plot(x,y,"or")
            n=n+1</pre>
```

3

```
else :
    plt.plot(x,y,"ob")
plt.grid()
plt.show()
return round((4*n)/N,5)
```

Monte_Carlo(1000)



[3]: 3.108

1.3 Estimation de l'aire de la surface sous la courbe d'une parabole par la méthode de Monte-Carlo

1.3.1 La problématique

Soit C_f la parabole représentative de la fonction f définie sur \mathbb{R} par $f(x) = x^2$.

On considère la surface bleutée (S) de la figure ci-dessous, définie par $\{M(x,y) \ / \ 0 \le x \le 1$ et $0 \le x \le x^2\}$.



On souhaite obtenir une approximation par la méthode de Monte-Carlo, de l'aire de la surface bleutée (S) délimitée par la parabole C_f , l'axe des abcisses et la droite d'équation x = 1. ### L'implémentation en langage Python d'une fonction aire_parabole(N) Saisir le script ci-contre from random import *

```
2
                          3
                             def aire parabole(N):
                          4
                                  n=0
                          5
                                  for i in range(N):
                          6
                                       x=random()
                          7
8
                                       y=random()
                                       . . .
                                       - - -
dans la cellule code ci-dessous : 11 aire_parabole (1000000)
```

Compléter les lignes 8), 9) et 10) pour que le script affiche une approximation de l'aire de la surface (S) avec deux chiffres après la virgule :

```
[5]: from random import *
```

```
def aire_parabole(N):
   n=0
   for i in range(N):
        x=random()
        y=random()
        if y<=x**2:
            n=n+1
   return round(n/N,2)
aire_parabole(1000000)
```

[5]: 0.33

Tapper dans la cellule texte suivante la valeur que vous obtenez :

Une approximation de l'aire de la surface (S) avec deux chiffres après la virgule est par exemple de : 0,33.

Quelques remarques en lien avec ce TP informatique 2

2.0.1 Qui a inventé la méthode de Monte-Carlo ?

"Le terme méthode de Monte-Carlo, ou méthode Monte-Carlo, désigne une famille de méthodes algorithmiques visant à calculer une valeur numérique approchée en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. Le nom de ces méthodes, qui fait allusion aux jeux de hasard pratiqués au casino de Monte-Carlo, a été inventé en 1947 par Nicholas Metropolis, et publié pour la première fois en 1949 dans un article coécrit avec Stanislaw Ulam". D'après Wikipédia

Nicholas Metropolis (1915-1999) : physicien gréco-américain

5



Stanislaw Ulam (1909-1984) : mathéméticien polonais



2.0.2 Une correction vidéo du 3. b)

Voici une correction vidéo de la question 3. b) pour récompenser ceux qui sont allés jusqu'au bout de ce TP informatique !

Lien vers la vidéo

6