

# Recommandations générales pour les séances de TD-TP

## Soyez honnêtes et comprenez ce que vous faites.

- Les TP sont posés de manière à ce que plusieurs méthodes pour aborder un même problème soient parfaitement valides et toutes légitimes. Il serait dommage que tout le monde réponde de la même manière.
- Il est bien plus facile de recommencer quelque chose depuis le départ, même si vous travaillez à plusieurs ou si vous vous aidez de bouts de code faits par d'autre, plutôt que de partir d'un énorme bout de code plagié (qui ne correspondrait pas à la manière dont vous auriez abordé le problème) en essayant de le déboguer.
- Masquer un plagiat informatique demande énormément plus de compétences informatiques que ce dont vous avez besoin pour avoir une excellente note à vos évaluations de TP.
- Même si vous n'arrivez pas à avancer aussi vite que vous le souhaitez, vous aurez une note très correcte si vous venez en séance et que vous êtes sérieux.
- Nous n'hésiterons par contre pas à vous sanctionner si vous arrivez en début de séance avec un code (qui marche ou pas) mais que vous êtes incapables de comprendre. On préfère largement vous aider à avancer à partir de zéro que de vous aider à comprendre un code qui n'est pas de vous.

## Structurez le document.

- Validez souvent les morceaux de code que vous écrivez, même si toutes les instructions ne sont pas présentes, cela permet au moins de s'affranchir régulièrement des problèmes de syntaxe et d'indentation.
- Dès qu'un morceau de code fonctionne, passez à une autre cellule (n'hésitez pas à faire de très petites cellules). Si vous avez défini une fonction par exemple et qu'elle fonctionne, vous n'avez pas besoin de la valider de nouveau pour la tester, testez-la dans d'autres cellules.
- Ayez toujours une cellule vide sous la main, pour pouvoir faire des tests, comme afficher la valeur d'une variable, etc.
- De temps en temps, et au moins une fois 10 minutes avant de rendre votre document, redémarrez le kernel, et validez les cellules dans l'ordre, pour vérifier que les choses fonctionnent et qu'elles se déroulent dans un sens qui soit cohérent.
- N'hésitez pas à faire des cellules de texte pour expliquer votre démarche, signaler vos soucis par rapport à des résultats qui vous paraîtraient incohérents.
- Nous ne validerons pas les cellules à votre place, l'évaluation consistera à lire votre document dans l'ordre, et à vérifier la cohérence de ce que vous obtenez.

## Utilisez de l'aide interne avant l'aide externe.

- Commencez par voir si le simple fait de faire Maj.+Tab. après une commande ne donne pas des exemples suffisants pour la syntaxe de la dite commande.
- Vérifiez dans vos TP précédents si vous ne l'avez pas déjà utilisée, allez voir dans le TP d'introduction s'il n'y a pas un exemple.
- Demandez à vos voisins ou votre chargé de TP/TD.
- Enfin si aucune des pistes ci-dessus ne fonctionnent, vous pouvez vous aider d'Internet, mais cela doit être un dernier recours : toutes les réponses peuvent se faire uniquement à l'aide des bibliothèques `numpy` et `matplotlib` qui sont déjà chargées par la commande `%pylab inline`,

et on vous demande de n'utiliser que ça, pour que ce soit lisible. Il existe des tonnes de solutions utilisant des myriades de bibliothèques diverses et variées, et chercher sur Internet vous amènera au hasard sur une de ces possibilités, mais ne vous aidera pas à progresser dans le cadre qu'on s'est fixés. Si vous cherchez sur Internet, rajoutez les mots-clés `numpy` ou `matplotlib` à vos recherches, cela vous restreindra au moins un peu les possibilités. Par contre les exemples sont souvent donnés en abrégant `numpy` par `np` et `matplotlib` par `plt`. Souvenez-vous que nous n'en avons pas besoin, tout a été chargé directement (c'est la signification de la phrase « *Populating the interactive namespace from numpy and matplotlib* » qui s'affiche lors de la validation de la commande magique `%pylab inline`). Vous écrirez donc par exemple `array` et non pas `np.array`.