

Informatique Tronc Commun

PTSI-PT*

Séquence

Cahier de vacances

Thème : Préparer sa rentrée en informatique

Objectifs

- Avoir une connaissance, même vague, des outils essentiels à la programmation informatique sous Python ;
- Préparer en amont les premiers chapitres de la rentrée ;
- Appréhender le fonctionnement de cet enseignement en classe CPGE.

Table des matières

1 Introduction

1.1 Message d'accueil

Félicitations pour votre admission en classe préparatoire aux grandes écoles. Vous vous apprêtez à débiter une formation exigeante mais particulièrement enrichissante, au cours de laquelle vous développerez de nombreuses compétences scientifiques, méthodologiques et intellectuelles.

L'informatique occupe aujourd'hui une place importante dans la formation des ingénieurs et des scientifiques. En première année de CPGE, il n'est pas attendu que vous soyez déjà un programmeur expérimenté. En revanche, il est essentiel d'aborder les premières semaines de CPGE avec des bases solides en algorithmique et en programmation afin de pouvoir se concentrer rapidement sur les raisonnements et les méthodes propres à l'enseignement supérieur.

Ce recueil a pour objectif de vous accompagner dans cette transition entre le lycée et la classe préparatoire. Les exercices proposés vous permettront de réviser/travailler les notions fondamentales de programmation en Python.

Travaillez régulièrement, prenez le temps de comprendre les mécanismes étudiés et n'hésitez pas à expérimenter par vous-même. En programmation, la maîtrise s'acquiert avant tout par la pratique.

1.2 Objectif de ce recueil

L'objectif principal de ce recueil de recommandations et d'exercices n'est pas de vous faire devenir un programmeur durant les vacances d'été. Il vous est donné afin de vous aider à appréhender au mieux cette matière, et notamment le côté écriture de code informatique/programmation, pour ceux ou celles d'entre vous qui ne seraient pas complètement à l'aise avec cela. De plus, si certains des exercices, vous paraissent difficiles, ne vous frustrez pas et ne vous inquiétez pas. Le parcours d'enseignement en CPGE est repris de zéro.

2 Enjeux du début d'année en Informatique en CPGE-PTSI au lycée Chaptal

Au début de l'année, les enseignements d'informatique s'appuient principalement sur les bases de la programmation en Python. L'objectif n'est pas seulement d'apprendre à écrire du code, mais surtout d'acquérir une méthode rigoureuse permettant de concevoir, analyser et corriger un algorithme.

2.1 Les premières notions abordées

2.1.1 Les séquences d'instructions

Une séquence d'instructions correspond à une suite d'actions exécutées les unes après les autres dans l'ordre où elles sont écrites. Les objectifs attendus sont les suivants :

- comprendre le déroulement séquentiel d'un programme ;
- savoir utiliser les fonctions d'entrée et de sortie usuelles ;
- manipuler des variables et leur affecter des valeurs ;
- réaliser des calculs simples à l'aide d'expressions arithmétiques ;
- prévoir le résultat produit par un programme donné ;
- traduire un algorithme simple en langage Python.

2.1.2 Les instructions conditionnelles

Les instructions conditionnelles permettent à un programme de prendre des décisions en fonction d'une condition logique. Les objectifs attendus sont les suivants :

- comprendre le rôle des expressions booléennes ;
- savoir utiliser les opérateurs de comparaison ;

- maîtriser les structures `if`, `if...else` et `if... elif...else` ;
- déterminer les différentes issues possibles d'un programme ;
- écrire un traitement adapté à plusieurs situations distinctes ;
- construire des algorithmes faisant intervenir des choix logiques.

2.1.3 Les boucles conditionnelles et inconditionnelles

Les boucles permettent de répéter automatiquement certaines instructions. Pour les boucles inconditionnelles (`for`), les objectifs sont :

- parcourir une suite de valeurs ;
- répéter un traitement un nombre connu de fois ;
- réaliser des calculs itératifs ;
- utiliser correctement la fonction `range()`.

Pour les boucles conditionnelles (`while`), les objectifs sont :

- répéter un traitement tant qu'une condition est vérifiée ;
- comprendre la notion de condition d'arrêt ;
- éviter les boucles infinies ;
- construire des algorithmes de recherche ou de simulation simples.

Dans les deux cas, il sera important de savoir suivre l'évolution des variables au cours des différentes itérations.

2.1.4 Les variables et les structures de données élémentaires

Une attention particulière sera portée à la gestion des variables, qui constituent la mémoire de travail d'un programme. Les objectifs attendus sont les suivants :

- comprendre le principe de l'affectation ;
- distinguer les différents types de données usuels (entiers, flottants, booléens, chaînes de caractères) ;
- suivre l'évolution de la valeur d'une variable au cours de l'exécution ;
- utiliser les variables pour stocker et exploiter des résultats intermédiaires.

2.1.5 Les listes (et variables itérables en générale)

Les listes constituent l'une des structures de données les plus utilisées en Python. Elles permettent de regrouper plusieurs valeurs au sein d'un même objet. Les objectifs attendus sont les suivants :

- créer et manipuler une liste ;
- accéder aux éléments d'une liste grâce à leur indice ;
- modifier les valeurs contenues dans une liste ;
- parcourir une liste à l'aide d'une boucle ;
- calculer des grandeurs simples à partir des éléments d'une liste (somme, moyenne, maximum, minimum, etc.) ;
- rechercher une information dans une liste ;
- comprendre la différence entre une variable simple et une structure contenant plusieurs données.

La maîtrise de ces notions constitue une base indispensable pour aborder sereinement les chapitres plus avancés qui seront étudiés au cours de l'année de PTSI.

2.2 Et si je n'ai pas tout compris à ce qui vient d'être écrit ?

Ce n'est pas bien grave. (Mais, ça ne vous empêche pas de vous renseigner sur Internet pour comprendre ce qui est écrit précédemment ;-)). Ce qui vous est présenté ici sont les premières notions vues en CPGE-PTSI à Chaptal. La compréhension et la maîtrise de ces notions vont s'étaler sur plusieurs semaines.

3 Intérêt de ce recueil et début d'année

3.1 Début d'année en CPGE pour l'enseignement de l'informatique

En CPGE, au 1^{er} semestre, il n'y a pas de cours d'informatique, mais uniquement une heure, par semaine, de travaux dirigés (TD) sur ordinateur. Les TDs sont prévus afin de s'entraîner à la résolution d'exercices. Afin que ces TDs se passent de manière efficace, il est important de connaître les notions traitées. C'est pour cela qu'en classe de PTSI, l'organisation sera la suivante au début d'année :

- Vous recevrez au moment voulu par votre professeur, un polycopié de cours et de TD.
- Le cours sera à lire et à travailler de façon autonome, seul(e) ou en groupe. Le cours devra être travaillé avant la future séance de TD.
- Afin de valider vos connaissances et votre compréhension du cours, un ou des QCM d'auto-évaluations vous seront proposés. Vous aurez l'obligation de répondre à ces auto-évaluations, afin que votre professeur puisse vérifier que le travail de lecture/compréhension du cours est bien fait.
- Lors de la séance de TD, les exercices sont travaillés sur plusieurs semaines (spécifiquement au début d'année).
- D'autres QCM d'auto-évaluations et exercices auto-évalués seront alors également à votre disposition pour vous permettre de valider la compréhension des notions et des mécanismes de programmation traités en TD.

3.2 Quel est l'intérêt de ce recueil proposé ?

Le niveau des élèves arrivant en CPGE-PTSI est très hétérogène sur les connaissances et les compétences en programmation informatique. C'est pourquoi l'ensemble des notions élémentaires (séquences d'instructions, instructions conditionnelles, boucles, variables, listes) sont revues en début d'année. Mais, comme dit précédemment, il faudra travailler en partie celles-ci, de façon autonome. Ce recueil a donc pour vocation (comme dit précédemment également) de vous aider à appréhender au mieux celles-ci afin que certain(e)s d'entre vous ne sentent pas mal à l'aise avec cet enseignement. Ne vous inquiétez pas quant à votre niveau actuel, il n'est pas représentatif de celui que vous pourrez avoir après quelques semaines en CPGE.

4 Quoi travailler ?

Vous pouvez tout d'abord vous documenter sur Internet sur la programmation en Python pour appréhender les notions décrites dans le paragraphe **2.1 Les premières notions abordées**. Comme, il a déjà été écrit dans ce document, le niveau des élèves est très différent lors du début d'année, donc il est tout à fait possible, qu'il faille un peu vous documenter pour appréhender le code informatique.

Pour les exercices, je vous propose de travailler sur le site <https://codex.forge.apps.education.fr/> ou de taper dans un moteur de recherche : « Codex : le code par l'exercice ».

Ce site agréé par l'éducation nationale va vous permettre de réviser/travailler les notions présentées précédemment à votre rythme et selon votre niveau et ne nécessite aucune inscription.

Les exercices que je vous recommande sont du niveau « débutant » ou « facile » mais libre à vous de faire des exercices plus "durs" si vous êtes à l'aise avec ce qui est demandé. **Attention**, le niveau des exercices « débutant » ou « facile » ne l'est que par la longueur de code à écrire et par le niveau de programmation proposée par rapport à des compétences de fin de cycle de Terminale avec spécialité NSI... Donc, si même des exercices « débutant » ne vous paraissent pas évidents du point de vue de la programmation, cela peut être normal. (Certains élèves arrivant en CPGE ne maîtrisent pas la création d'une simple fonction Python, par exemple). Le but de ces exercices est de vous familiariser avec ces notions élémentaires si vous ne les connaissez pas bien ou de les réviser pour ceux se sentant le moins à l'aise avec. *A priori*, le contexte mathématique proposé, est lui par contre simple, et ne doit pas vous poser de problèmes. On se concentre donc ici, uniquement sur l'aspect programmation informatique. Certains de ces exercices ressemblent à ceux que vous ferez en tout début d'année. S'il

vous semble que des notions vous sont manquantes pour répondre à certaines questions (par exemple, comment faire une boucle `for` en utilisant `range`), n'hésitez pas à vous documenter sur Internet.

Mais pas de panique si vous n'arrivez pas à tout faire, ou même si vous ne faites que peu de choses. Comme, il a été écrit précédemment, à la rentrée, des cours vous seront donnés pour travailler pendant plusieurs semaines, les différentes notions abordées ici.

4.1 Exercices proposés

- Séquences d'instructions et fonctions
 - Somme de deux entiers (débutant)
 - Consommation d'énergie (débutant)
 - Fonctions simples (débutant)
 - Différentes distances (débutant)
 - Fonctions affines (débutant)
- Instructions conditionnelles
 - Autour des booléens (débutant)
 - Opérateurs booléens (facile)
 - Années bissextiles (débutant)
 - Multiple (facile)
- Boucles
 - Autour du range (débutant)
 - Détermination d'un seuil (1) (facile)
 - Détermination d'un seuil (2) (facile)
- Les listes/tableaux
 - Longueur d'un objet (débutant)
 - Somme d'une liste d'entiers (débutant) (pour celui-ci il faut connaître les boucles `for` et savoir comment accéder aux éléments de la liste);
 - Autour du second degré (1) (débutant)
 - Autour du second degré (2) (débutant)

4.2 Comment accéder aux exercices proposés ? et comment les traiter ?

Sur le site <https://codex.forge.apps.education.fr/>, cliquer sur l'onglet « Recherche », puis taper le nom de l'exercice directement (en choisissant éventuellement le niveau de difficulté correspondant).









Pour répondre à l'exercice, vous devez taper le programme demandé et le valider en cliquant sur le bouton entouré, ci-dessous. Vous disposez de 10 essais avant que la correction n'apparaisse.

```


1- def somme(a, b):
2-     return ...| # ← compléter ici
3-
4- # Tests
5-
6- assert somme(10, 32) == 42
7- assert somme(100, 7) == 107
8-
>>>

```

Evaluations restantes : 10/10


Certains exercices proposent de l'aide, ou alors différentes versions de code pour vous aider à répondre à la question.

Version vide **Version à trous** 

```

1- def est_circulaire(distribution):
2-     ...
3-
4- # Tests
5-
6- distribution_a = {
7-     "Anne": "Elodie",
8-     "Bruno": "Fidel",

```

Version vide **Version à trous** 

```

1- def est_circulaire(distribution):
2-     personnes = [prenom for prenom in distribution]
3-     premiere = personnes[0]
4-     destinataire = distribution[...]
5-     nb_destinataires = 1
6-
7-     while destinataire != ...:
8-         destinataire = ...
9-         nb_destinataires = ...
10-
11-     return nb_destinataires == ...
12-
13- # Tests

```