

Mémorisation – 1ère – Spécialité NSI

Chapitre 1 – Python – Variables, affectation, entrées / sorties

1	Une variable python commence par	une lettre ou <code>_</code> , mais pas par un chiffre
2	Qu'est-ce que le type d'une variable ?	La nature de l'information stockée dans la variable
3	Quels sont les 4 types de base en python ?	<code>str</code> = chaîne de caractère <code>int</code> = entier (integer) <code>float</code> = nombre réel <code>boolean</code> = booléen (True / False)
4	Quelle est la fonction python qui renvoie le type d'un objet ?	La fonction <code>type()</code>
5	Opérateur d'affectation ? Affectation ? ordre important ?	opérateur = variable = valeur ou expression (ordre) nom de variable toujours à gauche
6	Incrémenter / décrémenter une variable (+ opérateurs)	ajouter ou soustraire 1 au contenu de la variable (opérateurs <code>+=</code> ou <code>-=</code>)
7	Déf : transtypage d'une variable	changer le type d'une variable
8	Qu'est-ce qu'une expression ?	Ensemble de variables combinées à l'aide d'opérateurs
9	Quelle fonction python permet de récupérer ce qui est entré par l'utilisateur au clavier ?	La fonction <code>input()</code>
10	Quelle fonction python permet d'afficher le contenu d'une variable ?	La fonction <code>print()</code>

Chapitre 2 – Python – Chaîne de caractère

11	Pourquoi dit-on qu'une chaîne est itérable ?	On peut la parcourir
12	L'index d'un caractère dans une chaîne <code>s</code> varie entre <code>0</code> et <code>len(s)-1</code> .	<code>0</code> / <code>len(s)-1</code>
13	<code>s='azertyuiop'</code> donner <code>s[2]</code> et <code>s[-2]</code>	<code>s[2]='e'</code> (3ème élément) <code>s[-2]='o'</code> (en partant de la fin)
14	<code>s='azertyuiop'</code> donner <code>s[:2]</code> et <code>s[:-2]</code>	<code>s[:2]='az'</code> <code>s[:-2]='azertyui'</code>
15	<code>s='azertyuiop'</code> → donner <code>s[2:4]</code>	<code>s[2:4]='er'</code> (indice de fin exclu)
16	concaténer 2 chaînes de caractères ? opérateur de concaténation ?	ajouter la 2ème chaîne après la première opérateur <code>+</code>

Chapitre 3 – Python – Conditions

17	Quelle est la structure complète d'un bloc test en python ?	<pre> if condition1 : bloc1 elif condition2 : bloc2 else : bloc3 </pre>
18	Quels sont les opérateurs de comparaison égal, différent et appartient ?	<p>égal : ==, différent : != appartient : in</p>
19	Que vaut l'expression cdt1 and cdt2 ?	Elle est vraie si les 2 conditions le sont
20	Que vaut l'expression cdt1 or cdt2 ?	Elle est vraie si l'une des 2 conditions l'est
21	Que vaut l'expression not cdt ?	Elle est vraie si la condition cdt est fausse

Chapitre 4 – Python – Boucles

22	Quels sont les 2 types de boucle vus en cours ?	Les boucles for et while
23	Structure d'une boucle for	<pre> for variable in liste : bloc </pre>
24	Structure d'une boucle while	<pre> while condition : bloc </pre>
25	Que faut-il vérifier dans la boucle while ?	Que la condition devienne <i>fausse</i> à un moment pour sortir de la boucle
26	Que représentent les valeurs a,b,c dans l'instruction for i in range(a,b,c)	i parcourt toutes les valeurs entières depuis a inclu jusqu'à b exclu par pas de c

Chapitre 5 – Spécifications d'un programme

27	Quelles sont les spécifications d'un programme ?	Les pré-conditions et les post-conditions
28	Qu'est-ce que les pré-conditions ?	les données en entrée du programme et leur formatage
29	Qu'est-ce que les post-conditions ?	les données en sortie du programme et leur formatage
30	Avant de coder un programme sur ordinateur, ...	il faut le découper en bloc élémentaire ; et pour chaque bloc, préciser les spécifications

Chapitre 6 – Représentation des entiers en informatique

31	Quels sont les chiffres de base en base 2, en base 10 et en base 16 ?	base 2 : 0 et 1 base 10 : 0 1 2 3 4 5 6 7 8 9 base 16 : 0 1 2 3 4 5 6 7 8 9 A B C D E F
32	Qu'est-ce qu'un bit (binary digit) ?	Un chiffre valant 0 ou 1
33	Convertir 1101_2 en base 10	$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
34	Convertir $A3F_{16}$ en base 10	$10 \times 16^2 + 3 \times 16^1 + 15 \times 16^0$
35	Signification LSB ? MSB ? Exemple sur 110110	LSB : bit de poids le plus faible MSB : bit de poids le plus fort (non nul) MSB \rightarrow 110110 \leftarrow LSB
36	Comment savoir si un nb binaire est pair ?	LSB = 0
37	Quelle est la plus grande valeur que l'on peut coder sur N bits ?	$2^N - 1$
38	Comment convertir un nb binaire en hexadécimal ?	On groupe les bits par 4 en commençant par le LSB, à chaque groupe son code hexadécimal correspondant
39	Complément à 2 d'un nb binaire ?	inverser tous les bits : $0 \leftrightarrow 1$
40	Un entier <input type="text"/> utilise le MSB comme bit de <input type="text"/>	<input type="text"/> relatif / <input type="text"/> signe
41	Comment trouver le code binaire d'un entier négatif ?	complément à 2 entier positif puis ajouter 1 oubli éventuelle retenue
42	Schéma circulaire des entiers relatifs sur 4 bits	

Chapitre 7 – Python – Représentation des entiers en python

43	(py) Les entiers en base 10 sont de type <input type="text"/>	<input type="text"/> int
44	(py) Les entiers en base 2 et 16 sont de type <input type="text"/> et commencent par <input type="text"/> ou <input type="text"/> .	<input type="text"/> str / <input type="text"/> '0b...' ou <input type="text"/> '0x...'
45	(py) conversion base 10 \rightarrow base 2 et 16	fonctions bin() et hex()
46	(py) conversion base 2 et 16 \rightarrow base 10	fonctions int('0b...', 2) et int('0x...', 16)

Chapitre 8 – Python – Les fonctions

47	Quelle est la structure complète d'une fonction en python ?	<pre>def nom_fonction (paramètres) : """ docstring """ ... return résultat</pre>
48	Que renvoie <code>help(nom_fonction)</code> en python ?	La « <i>docstring</i> »
49	Qu'est-ce que la portée d'une variable ?	C'est la « zone » du programme où on peut l'utiliser
50	Une variable peut être <code>locale</code> ou <code>globale</code> , ce qui signifie qu'on peut l'utiliser, respectivement , dans une portion précise du programme ou dans le programme <code>entier</code>	<code>locale</code> / <code>globale</code> / <code>entier</code>
51	Bonne habitude : on minimise au maximum les variables <code>globales</code>	<code>globales</code>

Chapitre 9 – Algorithmique – Partie 1

52	Qu'est-ce qu'un algorithme ?	Ensemble d'étapes qui permettent d'accomplir une tâche
53	Un algorithme résoud <code>un problème</code> : la solution doit être <code>exacte</code> et obtenue de manière <code>efficace</code>	<code>un problème</code> / <code>exacte</code> / <code>efficace</code>
54	Une première mesure de l'efficacité de l'algorithme est <code>son temps d'exécution</code>	<code>son temps d'exécution</code>
55	Comme le temps d'exécution de l'algorithme dépend de paramètres <code>externes</code> , on ne regarde que le <code>taux de croissance</code> du temps d'exécution en fonction de <code>la taille n des entrées</code>	<code>externes</code> <code>taux de croissance</code> <code>la taille n des entrées</code>
56	Définir le logarithme en base 2 Formule	Fonction inverse de la fonction exponentielle $n = 2^x \Leftrightarrow x = \lg n$
57	Classer par ordre croissant le comportement asymptotique des fonctions constante / linéaire / carrée / exponentielle / logarithmique	$1 \leq \lg n \leq n \leq n^2 \leq 2^n$
58	La notation $\mathcal{O}(f(n))$ est utilisée pour indiquer qu'un temps d'exécution <code>n'est jamais pire</code> qu'une constante multipliée par la fonction <code>asymptotique</code> $f(n)$. Elle donne le comportement <code>l'algorithme</code> du taux de croissance du temps d'exécution de <code>l'algorithme</code> .	<code>n'est jamais pire</code> <code>asymptotique</code> <code>l'algorithme</code>

Chapitre 10 – Algorithmique – Partie 2

59	Expliquer le tri par sélection sur le tableau $A[1..n]$	min =indice valeur min sous-tableau $A[i..n]$ échanger $A[i]$ et $A[\text{min}]$
60	Expliquer le tri par insertion sur le tableau $A[1..n]$	insertion de $A[i+1]$ à la bonne place dans le sous-tableau $A[1..i]$
61	Tri par sélection : comportement asymptotique	$\mathcal{O}(n^2)$
62	Tri par insertion : comportement asymptotique	$\mathcal{O}(n^2)$
63	La recherche dichotomique est efficace car elle travaille sur un tableau de n valeurs en \dots .	trié / $\mathcal{O}(\lg n)$
64	Un algorithme récursif comprend nécessairement 2 points. Lesquels ?	1) un cas de base 2) chaque appel récursif doit se faire sur une instance plus petite du même problème .
65	Algorithme des k plus proches voisins Connaissant une certaine \dots des données, cet algorithme prédit la propriété d'une donnée inconnue par \dots des propriétés des k plus proches voisins. Il est donc nécessaire de calculer une certaine \dots .	propriété / moyenne / distance

Chapitre 11 – Portes Logiques

66	Quelle est la table de vérité de la porte NOT ?	<table border="1"> <thead> <tr> <th>A</th> <th>NOT A</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	NOT A	0	1	1	0									
A	NOT A																
0	1																
1	0																
67	Quelle est la table de vérité de la porte AND ?	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A AND B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A AND B	0	0	0	1	0	0	0	1	0	1	1	1
A	B	A AND B															
0	0	0															
1	0	0															
0	1	0															
1	1	1															
68	Quelle est la table de vérité de la porte OR ?	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A OR B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A OR B	0	0	0	1	0	1	0	1	1	1	1	1
A	B	A OR B															
0	0	0															
1	0	1															
0	1	1															
1	1	1															
69	Quelle est la table de vérité de la porte XOR ?	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A XOR B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	A XOR B	0	0	0	1	0	1	0	1	1	1	1	0
A	B	A XOR B															
0	0	0															
1	0	1															
0	1	1															
1	1	0															
70	Les \dots permettent de faire des portes logiques qui, à leur tour, permettent de faire des \dots logiques qui, à leur tour, permettent de faire des \dots électroniques.	transistors / circuits / composants															

Chapitre 12 – Histoire des ordinateurs – Ruptures technologiques

71	Date du premier ordinateur (ENIAC)	1945
72	Date de découverte du transistor	1947
73	Date du premier circuit intégré	1958
74	Date Microprocesseur 4004	1971
75	Date du langage C / de python	1970 / 1991

Chapitre 13 – Architecture matérielle – Système d'exploitation

76	Quels sont les 3 points du modèle de von Neumann ?	1) séparation mémoire / unité centrale 2) séparation calculs / séquençage 3) la mémoire contient données et instructions
77	Schéma de von Neumann d'un ordinateur	
78	Quelles sont les 3 fonctions d'un système d'exploitation ?	Abstraction / Optimisation / Protection
79	Qu'est-ce qu'un processus ? Caractérisé par ?	Un processus représente l'exécution d'un programme. Il est caractérisé par son PID.
80	Qui gère les processus ?	Le système d'exploitation
81	Quels sont les 2 types de processus ?	processus système / processus utilisateur
82	Le processeur exécute <input type="checkbox"/> processus à la fois, <input type="checkbox"/> attendent.	<input type="checkbox"/> 1 seul / <input checked="" type="checkbox"/> tous les autres

Chapitre 14 – Commandes de base du shell Bash

83	[Bash] Commande pour avoir des infos sur la commande cmd ?	man cmd
84	[Bash] lister le contenu d'un répertoire	ls (list)
85	[Bash] changer de répertoire	cd (change directory)
86	[Bash] copier (fichier/répertoire)	cp (copy)
87	[Bash] afficher le répertoire de travail	pwd (print working directory)
88	[Bash] déplacer/renommer un fichier/répertoire	mv (move)
89	[Bash] créer/supprimer un répertoire vide	mkdir / rmdir

Chapitre 15 – Représentation d'un nombre réel (float) en informatique

90	Sous quelle forme écrit-on un nombre réel en base 2 ?	$(-1)^s \times m \times 2^n$
91	Sous quelle forme est écrit un nombre réel en mémoire (64 bits) ?	signe (1 bit) / exposant (11 bits) / mantisse (52 bits)
92	Quels liens entre la représentation mathématique $(-1)^s \times m \times 2^n$ d'un nombre réel en base 2 et son écriture en mémoire ?	bit de signe = 0 si nb positif $m = 1 + \text{mantisse}$ $n + 1023 = \text{exposant}$

Chapitre 16 – Mathématiques

93	Division euclidienne de a par b Définition + noms Comment faire en python ?	$a = b \times q + r$ (quotient q et reste r) (py) $q = a//b$ (py) $r = a\%b$
94	Définition de factorielle(n) + notation Exemple sur factorielle(5)	$n! = n \times (n - 1) \times \dots \times 3 \times 2 \times 1$ $5! = 5 \times 4 \times 3 \times 2 \times 1$
95	$1 + 2 + \dots + n = \sum_{i=1}^n i = ?$	$\frac{n(n+1)}{2}$

NOM :
Prénom :

Classe :

Pour préparer un devoir surveillé (fiche DS SPC)

Cette fiche poursuit 2 objectifs à propos des **erreurs stupides**.

Cette page doit être conservée avec les fiches de mémorisation.

Objectif 1 : on oublie parfois les consignes habituelles en devoir. Cette fiche est là pour se les remettre en tête.

- Utiliser un brouillon pour réfléchir (la copie doit être propre)
- Répondre avec les données de l'énoncé (au brouillon, j'écris comme je veux)
- Utiliser les notations de l'énoncé
- Ne pas utiliser les nombres dans les formules : on écrit la formule littérale + unités de chaque grandeur (\rightarrow conversions si nécessaire) + application numérique
- On justifie tout ce qu'on peut justifier (une réponse non justifiée n'a aucune valeur, une réponse juste obtenue par un raisonnement faux ne sera pas comptée.)
- Rien au crayon à papier !

Objectif 2 : notre cerveau a des **automatismes** qui nous jouent parfois des tours. Il faut apprendre à les repérer et en construire d'autres ! Le reste de cette page est donc à vous. Après chaque ds, relire son ds et noter ici toutes les erreurs stupides que vous avez faites. Avant chaque ds, on relit cette page et on se rappelle où on fait des erreurs stupides.