

Mémorisation – 1ère – Spécialité NSI

Chapitre 1 – 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	Quel est l'opérateur d'affectation ? Y a-t-il un ordre ?	opérateur <code>=</code> , variable = valeur ou expression (jamais l'inverse)
6	Incrémenter / décrémenter une variable	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 – 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 ?	relier les 2 chaînes opérateur <code>+</code>

Chapitre 3 – 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 <code>cdt1 and cdt2</code> ?	Elle est vraie si les 2 conditions le sont
20	Que vaut l'expression <code>cdt1 or cdt2</code> ?	Elle est vraie si l'une des 2 conditions l'est
21	Que vaut l'expression <code>not cdt</code> ?	Elle est vraie si la condition <code>cdt</code> est fausse

Chapitre 4 – Boucles

22	Quels sont les 2 types de boucle vus en cours ?	Les boucles <code>for</code> et <code>while</code>
23	Structure d'une boucle <code>for</code>	<pre> if condition1 : bloc1 elif condition2 : bloc2 else : bloc3 </pre>
24	Structure d'une boucle <code>while</code>	<pre> while condition : bloc </pre>
25	Que faut-il vérifier dans la boucle <code>while</code> ?	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 <code>for i in range(a,b,c)</code>	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 formattage
29	Qu'est-ce que les post-conditions ?	les données en sortie du programme et leur formattage
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 ?	LSB : bit de poids le plus faible MSB : bit de poids le plus fort
36	Comment savoir si un nb binaire est pair ?	LSB = 0
37	Dans toute base, rajouter des <input type="text"/> devant un nombre ne <input type="text"/> pas ce nombre.	<input type="text"/> zéros / <input type="text"/> change
38	Quelle est la plus grande valeur que l'on peut coder sur N bits ?	$2^N - 1$
39	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
40	Complément à 2 d'un nb binaire ?	inverser tous les bits : $0 \leftrightarrow 1$
41	Un entier <input type="text"/> utilise le MSB comme bit de <input type="text"/>	<input type="text"/> relatif / <input type="text"/> signe
42	Comment trouver le code binaire d'un entier négatif ?	complément à 2 puis ajouter 1 oubli éventuelle retenue

Chapitre 7 – 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 → base 2 et 16	fonctions bin() et hex()
46	(py) conversion base 2 et 16 → base 10	fonctions int('0b...', 2) et int('0x...', 16)

Chapitre 8 – 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 <input type="text"/> ou <input type="text"/> , ce qui signifie qu'on peut l'utiliser, respectivement, dans une portion précise du programme ou dans le programme <input type="text"/> .	<input type="text"/> locale / <input type="text"/> globale / <input type="text"/> entier
51	Bonne habitude : on minimise au maximum les variables <input type="text"/> .	<input type="text"/> globales

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 <input type="text"/> : la solution doit être <input type="text"/> et obtenue de manière <input type="text"/> .	<input type="text"/> un problème / <input type="text"/> exacte / <input type="text"/> efficace
54	Une première mesure de l'efficacité de l'algorithme est <input type="text"/> .	<input type="text"/> son temps d'exécution
55	Comme le temps d'exécution de l'algorithme dépend de paramètres <input type="text"/> , on ne regarde que le <input type="text"/> du temps d'exécution en fonction de <input type="text"/> .	<input type="text"/> externes <input type="text"/> taux de croissance <input type="text"/> la taille n des entrées
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 <input type="text"/> qu'une constante multipliée par la fonction $f(n)$. Elle donne le comportement <input type="text"/> du taux de croissance du temps d'exécution de <input type="text"/> .	<input type="text"/> n'est jamais pire <input type="text"/> asymptotique <input type="text"/> l'algorithme