

Découverte et prise en main



Caractérisation du PC utilisé

Principaux composants d'une machine numérique

Un PC ouvert est mis à votre disposition. L'ordinateur est un matériel communicant.

☞ Observer les éléments du panneau arrière.

Désignation		Fonction	Entrée/ sortie	Vitesse	
Port PS/2		Communication clavier souris	S		Supplanté par USB ou bluetooth
Port Série COM1		Communication série	E/S	19200 bps	Supplanté par USB
Port Parallèle		Communication parallèle (imprimante)	S	16 Mbps	Supplanté par USB
Port USB		Communication série	E/S	USB2 60 Mo/s	
Port VGA de la carte vidéo intégré		Communication vidéo analogique	S		
Port DVI		Communication vidéo numérique non HD	S		
Port HDMI		Communication vidéo numérique HD	S		
Port Réseau RJ45		Connexion filaire au réseau ethernet	E/S	De 128 Mo/s à 12800 Mo/s	
Connecteurs sons		Connexion aux hauts parleurs externes et micro	E/S		

Q1 : Identifier les ports de communication présents et absents.

Caractéristiques principales de la machine numérique utilisée

☞ Allumer l'ordinateur à votre disposition.

☞ Utiliser vos identifiants pour pouvoir ouvrir une session et utiliser le PC.

Q2 : Rechercher le nom et la version du système d'exploitation de la machine.

Q3 : Chercher si le processeur de cette machine est un processeur 32 bits ou 64 bits.

Q4 : L'ordinateur dispose-t-il de mémoire(s) de masse ? Quelle est la taille de celle(s)-ci ?

Q5 : Quelle est la taille de la mémoire vive ?

Organisation des fichiers

☞ Cliquer sur Démarrer, puis Mes Documents.

☞ Ce répertoire est votre espace de stockage personnel accessible depuis chaque poste.

Q6 : Ce répertoire est-il physiquement situé sur le poste de travail ou ailleurs ?

☞ Créer un répertoire Mes Fichiers Python . (clic droit, puis nouveau / dossier)

☞ Créer un fichier texte bidon dans ce répertoire. (clic droit, puis nouveau, document texte)

☞ Aller sur la fenêtre Ordinateur.

☞ Consulter les dossiers du disque dur C : et ceux du disque dur D :

- ☞ Aller dans le répertoire Utilisateurs
- ☞ Essayer d'accéder à un répertoire. Normalement vous n'avez les droits d'accès que pour votre compte.
- ☞ Retrouver votre fichier bidon dans l'arborescence de C :

Présentation de WinPython dans l'informatique pour tous

Ouvrir le disque D : ; Dans l'arborescence de celui-ci trouver le dossier WinPython. Lancer le programme **Ipython QT Console**.

Interpreteur interactif

Q1 : Relever dans la Console si la version de Python utilisée est une version 32 bits ou 64 bits. Relever également la version de Python utilisée.

L'interpréteur s'utilise comme une calculatrice : Il suffit de taper une **instruction**, pour qu'elle s'exécute. S'il y a une réponse, elle sera affichée.

L'interpréteur interactif présente l'intérêt d'afficher l'aide de la fonction en cours d'utilisation. N'hésitez pas à lire ce qui est indiqué.

Il est aussi possible d'invoquer l'aide d'une fonction en tapant :
 Help(*nom de la fonction*)

Découverte et distinction des types

Les valeurs en Python sont *typées*, autrement dit, elles sont classées selon l'objet qu'elles représentent. Une valeur peut ainsi être de type entier, flottant, booléen, chaîne de caractères, etc.

Pour connaître le type d'une valeur, il est possible d'utiliser la commande (ou fonction type)

type(object) -> string	Renvoie le type de l'objet
------------------------	----------------------------

Exemple : taper :

```
| >>> type(2)
```

Où l'on constate que le type de la valeur de a est le type entier (*integer*, en anglais).

Q2 : En utilisant la même procédure, déterminer le type des valeurs suivantes:

valeurs	type
False	
486	
True	
-32	
-32.	
3.14	
"PCSI rules"	

Pour modifier le type de variable, il est possible d'utiliser, lorsque cela est possible, les instructions :

<code>int(x)</code>	conversion en entier
<code>float(x)</code>	Conversion en nombre à virgule flottante

Manipulation d'entiers relatifs

Nous allons utiliser la console interactive de Python où chaque ligne tapée est immédiatement exécutée. Exécuter les commandes suivantes :

```
>>> 3 + 5
>>> 6 - 8
>>> 2 * 3
>>> 2 ** 3
```

Listes des opérations

+	addition
-	soustraction
*	produit
/	quotient flottant
<code>x // y</code>	Quotient entier de x par y
<code>x % y</code>	reste de la division euclidienne
-x	opposé
<code>abs(x)</code>	valeur absolue
<code>x ** y</code>	puissance

Exemple

```
>>> 125 * 2 + 5 % 3
>>> 5**2
```

Application

Un angle est mesuré avec un rapporteur avec une précision de 1° . Cette mesure est alors codée par un entier.

Lors d'un premier déplacement, la mesure indique 5282° . Puis un second déplacement est mesuré à -1287° .

Q3 : Déterminer le nombre de tours parcourus à la fin des déplacements.

Q4 : Déterminer l'angle final modulo 360°

Q5 : Justifier le résultat de l'instruction : `2**(1//2)`

Manipulation des nombres à virgules flottantes

Opérations

On retrouvera les mêmes opérations qu'avec les entiers.

Taper :

```
>>> type(42 + 3.1415)
```

Q6 : Justifier l'appellation « typage automatique »

Nombre de chiffres affichés

L'instruction `format` permet de préciser le nombre de chiffres affichés après la virgule. Ce nombre apparaît entre deux apostrophes, après un point.

Exemple : taper :

```
>>> format(1.2569, '.1f')
```

Q7 : Comparer les résultats des opérations :

```
>>>format(455**45, '.100f')
>>>format(455**45, '.100e')
```

Q8 : Afficher $2,451.10^6$ avec trois chiffres significatifs.

Quelques surprises à justifier

Q9 : Comparer et justifier les résultats des opérations :

```
>>> 7 / 3
>>> 7. / 3
>>> 7 // 3
>>> 7. // 3
```

Q10 : Comparer les résultats des opérations :

```
>>> 2 ** 100
>>> 2.0 ** 100
```

Commenter

Quelques surprises à justifier ultérieurement

Q11 : Effectuer instructions suivantes et commenter les résultats sans les justifier

```
>>> 1. / 3**6
>>> (1. / 3**6) * 3**6
>>> 1+2**-52-1
>>> 1+2**-54-1
>>> 1-1+2**-54
>>> 0.1+0.1-0.2
>>> 0.1+0.1+0.1-0.3
```

Déclaration de variables

Une variables est :

- un nom, qui permet de l'identifier (commence par une lettre)
- un type (typage automatique de Python)
- une valeur

Déclaration

Pour de nombreux langages de programmation, il est nécessaire d'annoncer dès le début les variables qui seront utilisées par la suite. Cette étape s'appelle la déclaration. Il s'agit alors de définir le nom et le type de variable qui sera utilisé par la suite.

Python permettant un typage automatique, cette opération n'est pas nécessaire.

Affectations

On peut d'ores et déjà utiliser des variables pour stocker des valeurs. L'affectation s'écrit avec le symbole = et n'affiche aucune valeur ; mais la variable mémorise la valeur qu'on lui a donnée et peut être utilisée dans la suite de la session.

Exécuter les commandes suivantes :

```
>>> a = 2
>>> print(a)
>>> a
>>> b = 6
>>> b = b + 2
>>> b = b + 2
>>> b
```

Application

x=1

y=4

Q12 : Proposer quelques instructions pour échanger les valeurs de x et y.

Manipulations des booléens

Un booléen est le codage d'une information sur un bit : vrai (True) ou faux (False).

Il peut être obtenu par des opérations de comparaisons entre deux valeurs.

Exemple : taper :

```
>>> 2>3
```

Opération	Signification
<	Strictement inférieur
<=	Inférieur ou égal
>	Strictement supérieur
>=	Supérieur ou égal
==	Egal
!= (ou <>)	Différent
is	Objet égal

Q13 : Comparer et justifier le résultat des comparaisons :

```
1 == 1.0
```

```
1 is 1.0
```

Q14 : Commenter le résultat de l'instruction :

```
>>> 0.1+0.1+0.1 == 0.3
```

Les opérations de bases sur les booléens sont :

Opération	Résultat
x or y	est vrai si x ou y est vrai
x and y	est vrai si x et y sont vrais

Opération	Résultat
<code>not x</code>	est vrai si x est faux, et inversement

Q15 : En utilisant une commande sous Python, déterminer si 678/123 est plus petit que 6789/1234.

Codage des caractères

Le codage des caractères est défini par le tableau ASCII. Chaque caractère est codé par un entier sur 7 bits.

Il est possible de retrouver le caractère associé avec l'entier en utilisant l'instruction `chr`, et inversement avec l'instruction `ord`.

```
>>> chr(97)
>>> chr(33)
>>> chr(128)
>>> ord(a)
>>> ord('a')
```

☞ Maintenir enfoncer Alt, taper 97 (avec le pavé numérique), relâcher Alt
Il est donc possible d'utiliser le code ASCII directement à partir d'un clavier.

Un mot, une phrase sont définis comme une chaîne de caractère, qui est un tableau d'entier.
Taper :

```
>>> a = "yop"
>>> len(a)
>>> a.upper()
```

Concaténation de chaînes

En Python, il est possible de concaténer plusieurs chaînes de caractères afin d'en créer une nouvelle. Pour cela, on utilise l'opérateur `+`. Essayer les instructions suivantes.

```
>>> "Concaténer " + "deux chaînes"
>>> 10+2
>>> '10'+2'
>>> "10"+"2"
```

Q16 : Quelle est la chaîne de caractère associée à la liste d'entier : 80 67 83 73 33.

Utilisation d'un environnement de développement interactif (IDE).

Dès qu'on veut écrire une suite d'instructions plus évoluées, en garder la trace voire la modifier, on utilise un éditeur.

Quitter la console Python

Lancer **Spyder** depuis le répertoire WinPython du disque D :

Trois fenêtres apparaissent.

- en bas à droite, la console ;
- en haut à droite, la fenêtre interactive composée de quatre onglets :

- l'inspecteur d'objet = l'aide interactive
 - l'explorateur de variable = liste de variable et de leur valeur
 - l'explorateur de fichier
 - les points d'arrêt (utilisé en mode débogage)
- à gauche : votre programme (ou script).

Prise en main : entrées & sorties standards

Pour l'instant on va se limiter à l'interface homme machine à notre disposition. L'entrée sera le clavier et la sortie l'écran.

Écrire le script suivant :

```
7 print("Hello word")
```

Enregistrer votre script dans votre répertoire de travail Python (sur votre disque personnel)
Taper et exécuter ce script. Pour cela, cliquer Exécution puis Exécution.

Écrire le script suivant :

```
7 print("Donner la valeur de a")
8 a=input()
9 print("a =")
10 print(a)
```

Exécuter

Écrire le script suivant :

```
7 a=input("Donner la valeur de a")
8 print("a =", a)
```

Exécuter ce script.

Instruction conditionnelle

Écrire le script suivant :

```
7 x=int(input("Donner un entier x = "))
8 y=int(input("Donner un entier y = "))
9 if x==y :
10     print("x = y")
11 else :
12     if x>y :
13         print("x > y")
14     else :
15         print("x < y")
```

Exécuter ce script.

Écrire le script suivant :

```
7 x=int(input("Donner un entier x = "))
8 y=int(input("Donner un entier y = "))
9 if y!=0 :
10     print("x/y = ",x/y)
11 else :
12     print("erreur")
```

Exécuter ce script.