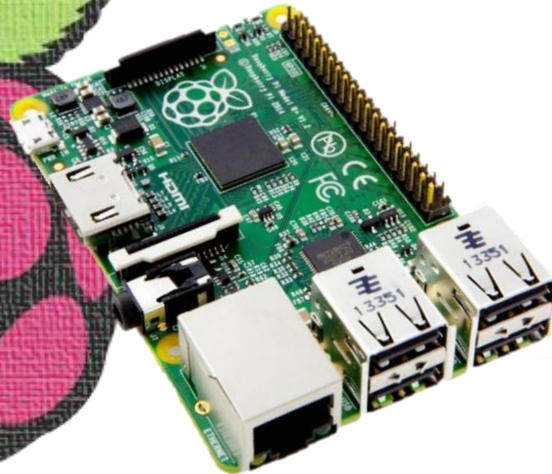
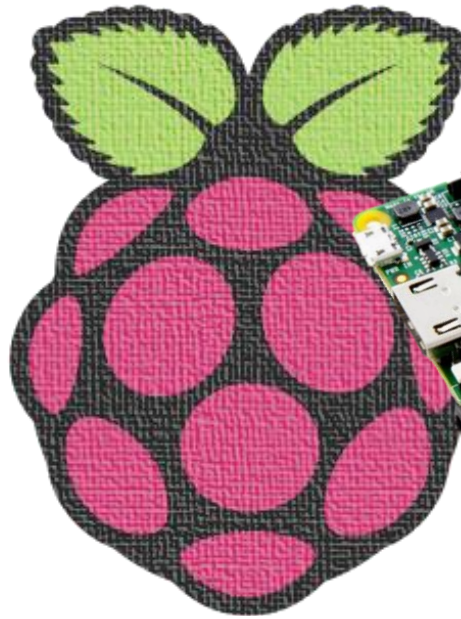


# DÉCOUVERTE DE LA RASPBERRY PI



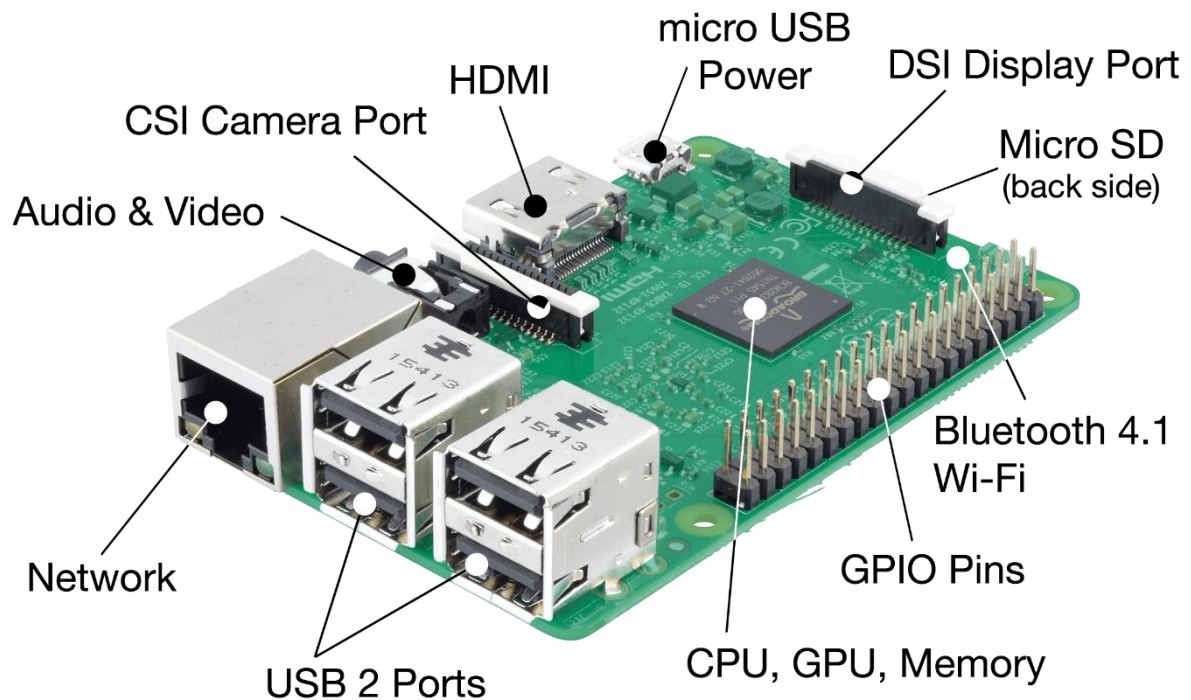
## TRAVAUX PRATIQUES SÉANCE N°1

## Préambule :

Depuis quelques années maintenant, la carte Raspberry Pi fait beaucoup parler d'elle. C'était en effet la première carte à embarquer un SoC ('System On a Chip') soit un mini-ordinateur tournant sous Linux, vendue aussi bon marché.

Nous allons utiliser au cours de ces séances la version 3 en modèle B+ sortie en février 2016.

Même si cela vous a déjà été présenté, revenons rapidement sur ses principales caractéristiques :



(source : [www.cours-ordinateur.fr](http://www.cours-ordinateur.fr))

### Principales caractéristiques :

- SoC BCM2837 contenant un ARM Cortex-A53 64bits 4 cœurs – 1,2GHz
- 1Go de RAM
- 85.60mm x 56mm x 21mm
- Carte micro-SD (OS + données) 64 Go max
- 1 port ETHERNET
- 4 ports USB
- 1 port HDMI
- 1 sortie audio-vidéo Jack 3,5mm
- Alimentation par micro-USB (5V-310mA min / 750mA le plus souvent)
- 1 connecteur mâle GPIO 40 broches (3,3V) : I/O, PWM, UART, I2C, SPI
- Deux ports dédiés : DSI (pour un écran LCD) et CSI (pour une camera)
- Modules BlueTooth 4.1 & Wi-Fi

## Objectifs de la séance :

- Découvrir la carte et ses périphériques
- Configurer une distribution Linux pour la Raspberry Pi
- Se familiariser avec l'environnement Linux
- Configurer la Raspberry Pi pour une connexion internet
- Installer et gérer de nouveaux paquets
- Découvrir le langage Python

Cette première séance va plutôt se dérouler en mode pas à pas alors que les suivantes vous laisseront davantage d'autonomie.

## Mise en place de la carte et de ses périphériques :

### Etape 1 :

A chaque début de séance, voici à quoi doit ressembler votre poste de travail :



Un PC est déjà installé mais nous n'utiliserons que son écran et le switch Ethernet disposé juste à côté de l'unité centrale. Vous n'aurez aucune manipulation de connectique à effectuer au niveau de ce PC, de l'écran ou du switch.

Seuls les deux câbles (HDMI et Ethernet) accrochés seront à câbler au moment où on vous l'indiquera.

**Veillez à remettre ce poste de travail dans le même état lorsque vous quitterez la salle à chaque fin de TP, notamment à bien ranger ces deux câbles afin qu'ils ne traînent pas sur la table !!**

Vous devez également récupérer :

- Une carte Raspberry Pi protégée par un boîtier
- Une alimentation 5V DC pour la Raspberry
- Un clavier USB
- Une souris USB
- Une carte SD (avec votre numéro de poste)

**Ce matériel sera également à ranger en fin de séance !!**





### Etape 2 :

Installer la carte RPI à plat sur votre table et libérer de l'espace autour d'elle.

*Si vous utilisez un jour une Raspberry sans boîtier, veillez à ne laisser aucun objet métallique à proximité (notamment dessous !!) au risque de provoquer sans le vouloir des courts-circuits irrémediables.*

### Etape 3 :

Repérer la carte SD qui vous a été fournie (elle n'est pas nécessairement identique à celle-ci-contre).

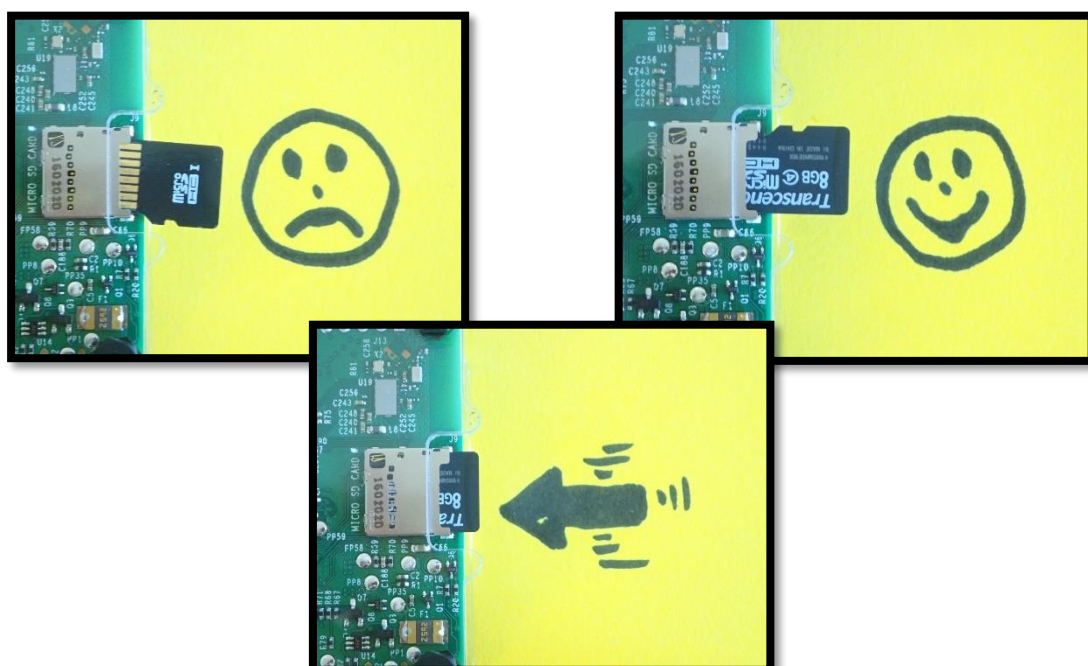
**Cette carte vous est attribuée pour l'ensemble des séances de TP. Veuillez ne pas repartir avec en fin de séance et la redonner aux enseignants qui vous encadrent.**

Retirer la carte micro-SD de son adaptateur.



### Etape 4 :

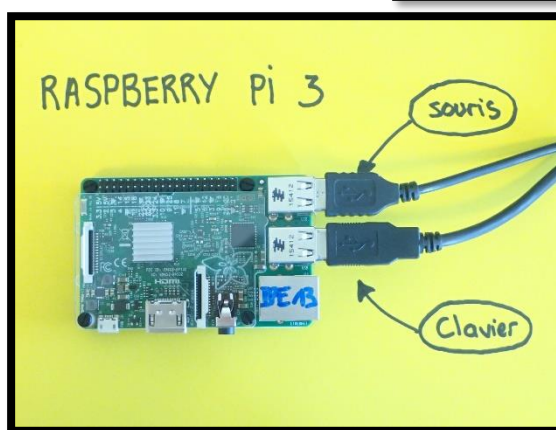
Retourner la carte RPI et repérer le connecteur micro-SD. Insérer alors au maximum la carte **dans le bon sens** (elle doit légèrement dépasser) et retourner à nouveau la carte Raspberry Pi.



### Etape 5 :

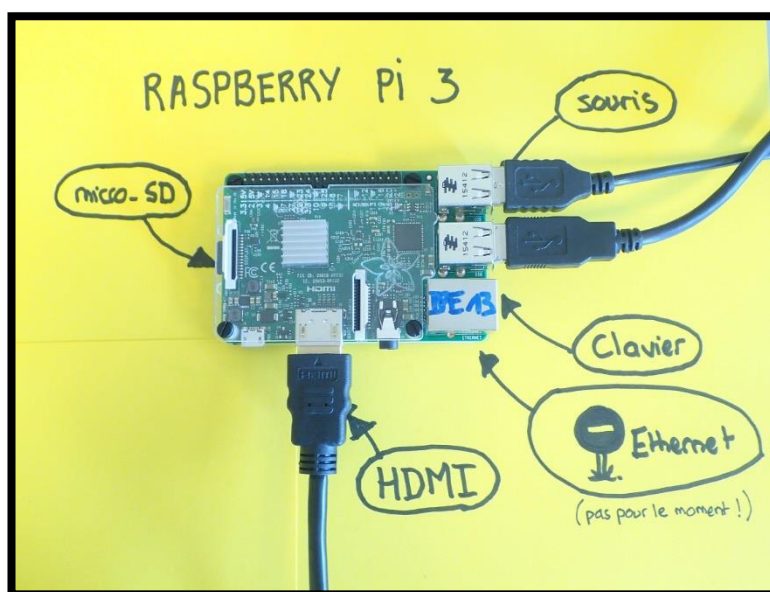
Brancher sur les ports USB disponibles de la carte RPI le clavier et la souris disponibles sur votre table.

*Remarque :* On pourrait aussi brancher un dongle RF en USB ou utiliser le module Bluetooth intégré pour disposer ainsi d'un clavier et d'une souris sans fil. Vérifier toutefois la compatibilité du modèle...



### Etape 6 :

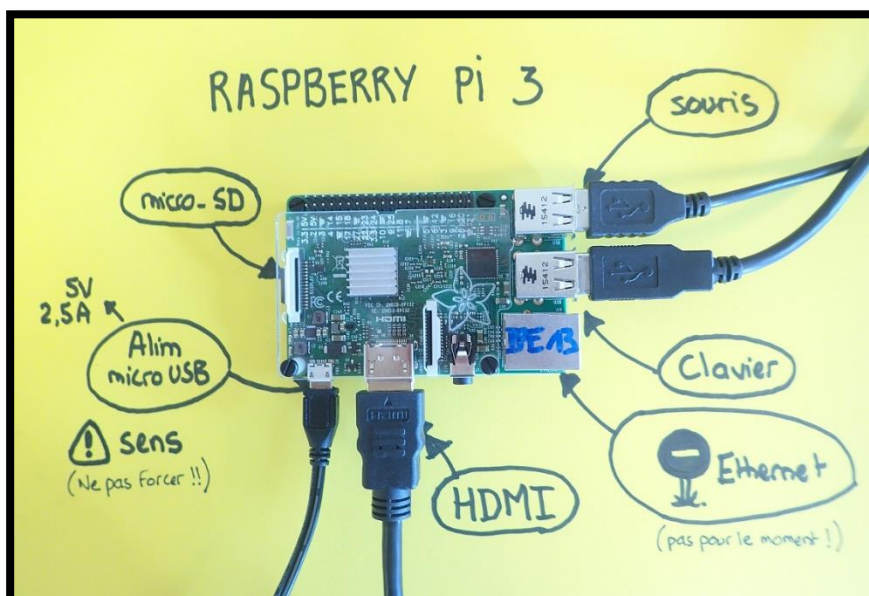
Détacher le câble HDMI de son support et brancher ensuite le connecteur HDMI du câble vidéo disponible sur la Raspberry.



ATTENTION !! Pour le moment, aucun câble réseau ne doit être branché !!

## Etape 7 :

Repérer ensuite le bloc secteur qui nous servira d'alimentation. Il dispose d'un connecteur micro-USB que vous connecterez à la Raspberry à côté du port HDMI.

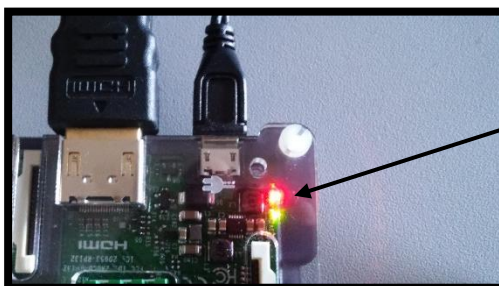


*Remarque :* La carte Raspberry seule consomme facilement environ 700mA au meilleur de sa forme !! Mais si l'on ajoute des périphériques (clavier, souris, webcam, clef USB, ...), il faut prévoir de la marge. Une alimentation de plus de 2000mA est vivement conseillée. La tension régulée doit être de 5V.

Brancher maintenant cette alimentation au secteur :



Si tout se déroule bien, la LED 'power' rouge doit être allumée et la LED 'activity' verte doit clignoter à la manière d'un disque dur. Et vous devriez voir la phase de 'boot' se dérouler à l'écran...



**REMARQUE :** Prenez bien note que pour le moment, aucun câble réseau n'a été branché sur la Raspberry !!

## Configuration du système d'exploitation à la mise en route :

Pour des raisons d'organisation et d'horaires, la carte SD qui vous a été fournie est déjà préinstallée avec un système d'exploitation.

Cet OS est la dernière version de **RASPBIAN STRETCH**, une version modifiée pour la RPI de la distribution Linux DEBIAN.

Toutefois, si vous êtes peut-être familier avec la démarche d'installation d'un OS sur le disque dur de votre ordinateur, c'est un peu différent ici puisqu'il s'agit ici de l'installer sur une carte SD bootable. Vous trouverez très facilement sur internet des tutoriels sur la méthode à suivre.

Nous allons maintenant procéder à la configuration pas-à-pas du système d'exploitation **RASPBIAN STRETCH** lors de la mise en route de notre Raspberry Pi.

**A la fin de la phase de boot, vous arrivez directement sur le bureau et un outil d'aide à la configuration apparaît au centre de l'écran :**



Remarque : Dans le cas où vous auriez fermé sans le vouloir cette fenêtre, demandez de l'aide d'un enseignant pour la faire réapparaître ou si vous êtes à l'aise avec Linux, lancez vous-même un terminal pour y entrer la commande '**sudo piwiz**'.

L'étape suivante est la configuration de la localisation, c'est-à-dire le pays, la langue et le fuseau horaire :





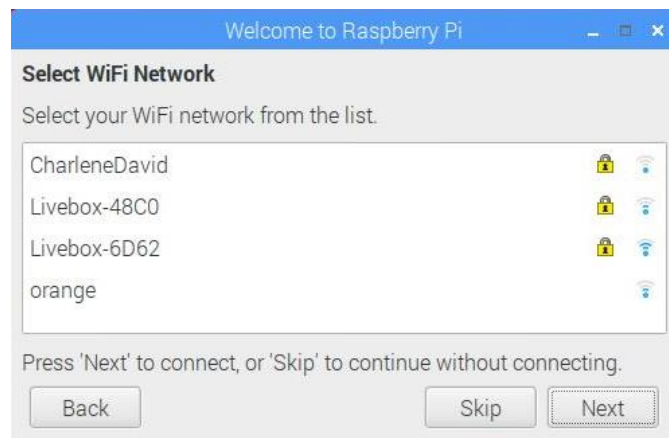
Vient ensuite l'étape où l'on fixe le mot de passe d'utilisateur. Respectez bien les consignes sur l'écriture du mot de passe, car nous en aurons besoin à plusieurs moments au cours des TP :



Le mot de passe doit être '**RPI\_xx**' où '**xx**' représente votre numéro de poste de '**01**' à '**14**'.

**ATTENTION** encore une fois, si vous mettez quelque chose de différent et que vous ne vous en souvenez plus au cours de ce module, nous n'aurons pas d'autre choix que de formater votre carte SD et vous perdrez donc tous vos fichiers !!

L'étape suivante consiste à configurer le réseau WiFi. Comme nous ne l'utiliserons pas durant ces séances (connexion internet filaire via un câble Ethernet), cliquez sur '**Skip**' :



L'étape suivante (Cf. capture d'écran page suivante) est très importante car il s'agit de la mise à jour du système d'exploitation **Raspbian Stretch**. En effet, l'OS est mis à jour en ligne tous les 2 à 3 mois environ. Entre deux, des modifications mineures et parfois majeures sont toujours disponibles en ligne et permettent parfois la correction de bugs gênants.

Toutefois, comme nous n'avons pas configuré de connexion internet pour le moment, nous ne pouvons pas effectuer de mise à jour. Nous l'effectuerons tout à l'heure en mode manuel. Vous cliquerez donc sur '**Skip**' également.





La dernière fenêtre de l'outil de configuration est simplement celle vous indiquant que le système d'exploitation est maintenant configuré. Vous pouvez donc cliquer sur '**Done**' pour terminer cet assistant.

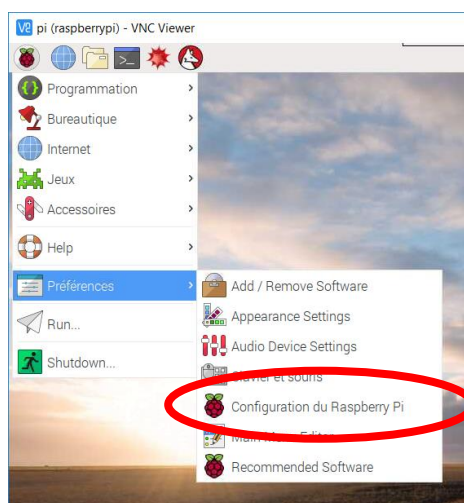


Il vous demande alors de rebooter la machine pour prendre en compte les modifications. Faites-le.

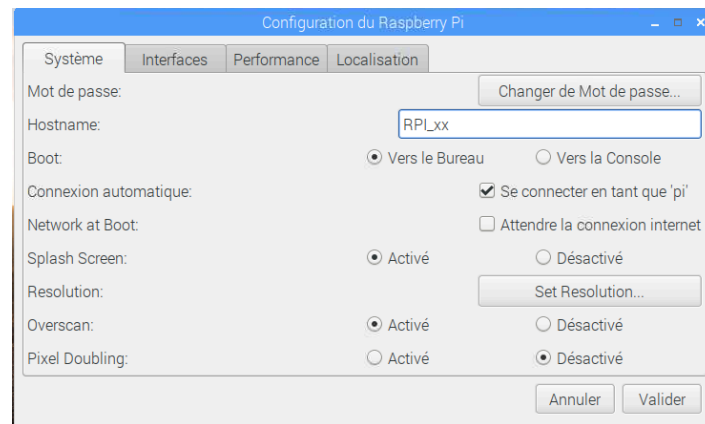
Une fois que le système a redémarré et que vous êtes sur le bureau, les modifications ont été prises en compte.

Toutefois, cette configuration ne suffit que dans le cas d'une utilisation assez basique de la carte Raspberry. Dans notre cas, nous allons utiliser ses broches d'entrée-sortie, la piloter depuis le réseau, ... Toutes ces actions nécessitent une configuration plus poussée. Nous allons donc la compléter.

Pour cela, allez dans le menu 'Démarrer' et choisissez 'Préférences > Configuration du Raspberry Pi' :



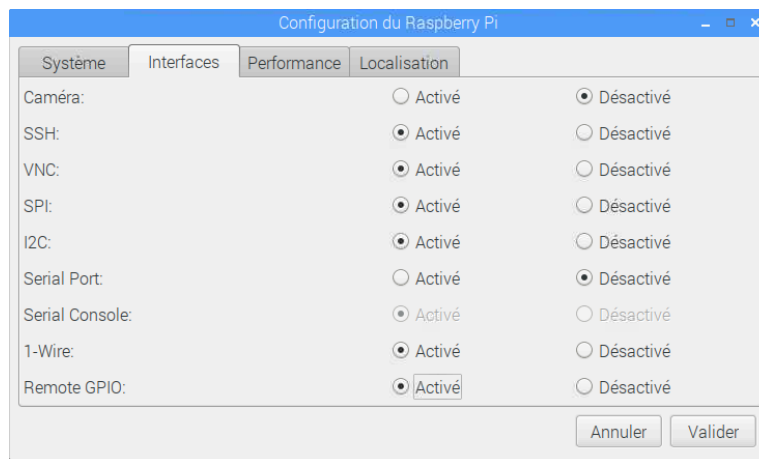
Dans l'onglet '**Système**' de la fenêtre qui vient de s'ouvrir, vous allez modifier le nom de votre Raspberry. C'est le nom qu'elle aura sur le réseau. Il est important de pouvoir l'identifier étant donné que l'on va avoir sur notre réseau plus d'une dizaine de cartes qui par défaut, porteraient toutes le nom 'Raspberrypi' !!



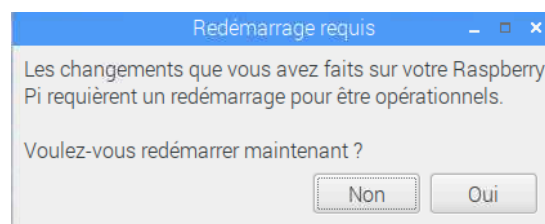
Vous allez mettre dans le champ '**Hostname**' le même nom que pour votre mot de passe, c'est-à-dire '**RPI\_xx**' avec '**xx**' représentant votre numéro de poste. Ce n'est pas conseillé en termes de sécurité, mais dans notre cas, cela va nous simplifier la tâche.

Vous laisserez toutes les autres options de configuration sur leur valeur par défaut.

Dans l'onglet '**Interfaces**', vous allez activer les différentes interfaces selon la copie d'écran ci-dessous. Cela nous permettra de ne pas devoir y revenir au cours de ce module.



Vous pouvez alors cliquer sur le bouton '**Valider**'. L'OS va vous proposer de rebooter une nouvelle fois :



Faites-le et attendez le retour à l'écran du bureau...

## Installation de la connexion à internet :

**NE BRANCHEZ ENCORE AUCUN CABLE RESEAU, ATTENDEZ QUE LE TEXTE VOUS LE DEMANDE !!**

Par défaut, la Raspberry est configurée pour une connexion par câble Ethernet ou par Wi-Fi à un routeur de type Freebox ou Livebox par exemple (DHCP – IP dynamique).

Ainsi, sur votre réseau personnel chez vous, dès que vous brancherez un câble réseau, votre Raspberry sera automatiquement connecté au réseau avec une adresse IP fournie par votre routeur (votre box). Pour le Wi-Fi, c'est comme sous un autre OS, les réseaux disponibles s'affichent et l'on se connecte avec la clef WEP.

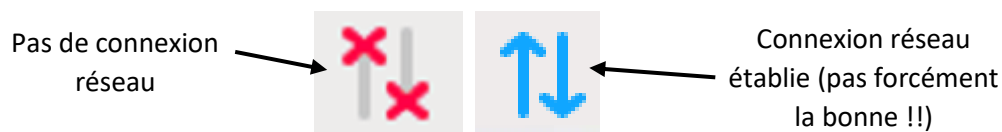
Mais l'utilisation à l'IUT se fait sur le réseau privé de l'université (qui n'est d'ailleurs, pour information, pas accessible de l'extérieur). Sur ce réseau, toutes les machines sont déclarées et connues. Dans le cadre de ces TP, on a donc une plage d'adresses attribuée pour les Raspberry de cette salle.

Il va donc falloir travailler non pas en adresse IP dynamique (configuration par défaut de la Raspberry au démarrage, attribuée par le routeur), mais en IP fixe ou statique.

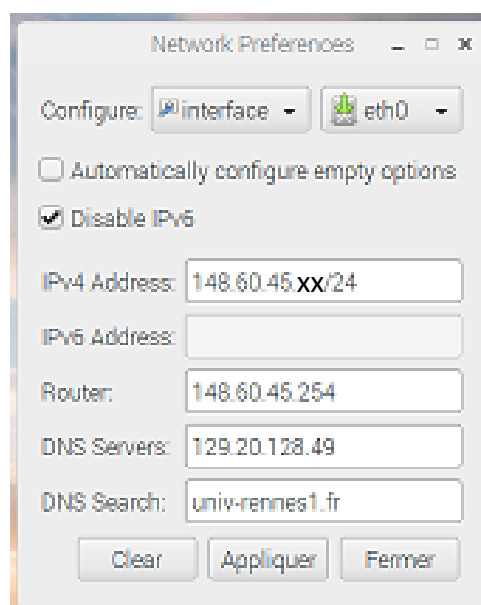
Pour ceux et celles qui utiliseraient une ancienne version de Raspbian, cette opération était assez fastidieuse car elle nécessitait de modifier des fichiers système comme le fichier 'interfaces'.

Depuis la version de septembre 2016, Raspbian Jessie intègre un outil de configuration du réseau beaucoup plus convivial que l'on va utiliser maintenant.

1. Faire un clic droit sur l'icône symbolisant le réseau (dans la barre de tâches, vers la droite) et choisissez '**Wireless & Wired Network Settings**' (Tout n'est pas encore traduit !!) :



2. Dans la fenêtre qui s'ouvre, sélectionnez l'interface '**eth0**', et remplissez les paramètres comme ci-dessous :



**ATTENTION !!** Pour l'adresse IP, chaque Raspberry doit avoir  
Une adresse différente.

**148.60.45.XX** : XX représentant votre poste  
(81 = poste 1, 82 = poste 2, ..., 93 = poste 13)

3. Cliquez sur 'Appliquer' et fermez la fenêtre.
4. Branchez alors le câble réseau Ethernet disponible. Vous devriez voir les LEDs de la partie inférieure du connecteur s'éclairer et clignoter. L'icône symbolisant le réseau doit également changer d'aspect.



5. Taper ensuite dans un terminal '**\$ ifconfig eth0**' pour vérifier que le contrôleur réseau a bien démarré. Vous devriez obtenir les informations suivantes (avec bien sûr l'adresse IP qui correspond à la vôtre).

```
pi@RPI_xx: ~  
Fichier  Édition  Onglets  Aide  
pi@RPI_xx:~ $ ifconfig eth0  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500  
    inet 148.60.45.64  netmask 255.255.255.0  broadcast 148.60.45.255  
    inet6 fe80::ba27:ebff:fe72:7afb  prefixlen 64  scopeid 0x20<link>  
    ether b8:27:eb:72:7a:fb  txqueuelen 1000  (Ethernet)  
    RX packets 20292  bytes 1694676 (1.6 MiB)  
    RX errors 0  dropped 453  overruns 0  frame 0  
    TX packets 68  bytes 6758 (6.5 KiB)  
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0  
pi@RPI_xx:~ $
```

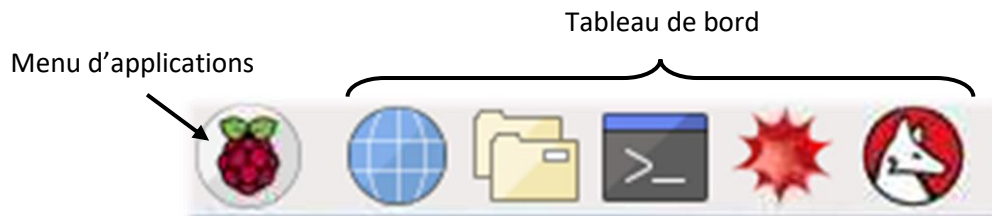
6. Pour terminer, lancez le navigateur internet préinstallé ('**Chromium**') pour vérifier que vous êtes bien connecté au reste du monde !!



## Découverte rapide du bureau :

L'interface graphique de cette distribution Linux rappelle celle de Windows et vous permettra de vous familiariser plus rapidement avec Linux pour ceux et celles qui n'en ont pas l'habitude.

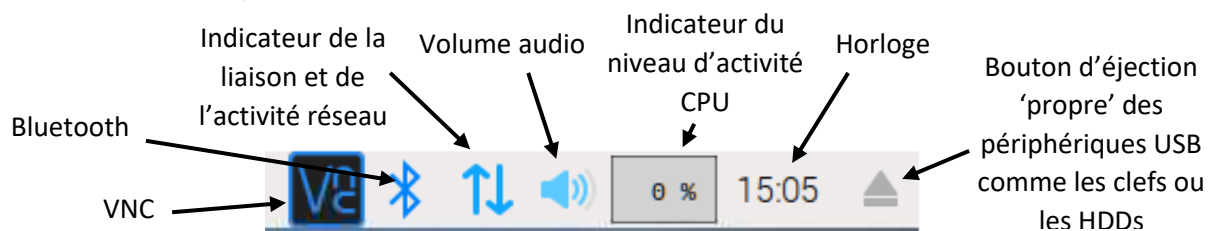
En haut de l'écran, vous retrouvez l'équivalent de la barre des tâches de Windows avec sur la gauche :



Le tableau de bord offre des raccourcis rapides vers des applications.  
Par défaut, on a dans l'ordre : navigateur web (Chromium), explorateur de fichiers, terminal de commande, Mathematica, Wolfram

Vous pouvez bien évidemment personnaliser les applications du tableau de bord pour adapter cela à votre convenance.

Et sur la droite de l'écran, on a :



On ne va pas balayer nécessairement tout le menu d'application, mais sachez donc que vous disposez déjà d'applications préinstallées et que celles-ci sont classées par thème dans l'arborescence du menu.

Cet environnement graphique familier est très pratique pour l'utilisation de la Raspberry en tant que mini-ordinateur ou pour développer du code de manière conviviale.

Toutefois, l'intérêt principal de la Raspberry ou des autres cartes similaires est d'offrir une grande puissance de traitement des informations et une connectivité au réseau dans un volume très réduit, ce qui en fait un outil très adapté à l'électronique embarquée, à l'internet des objets.

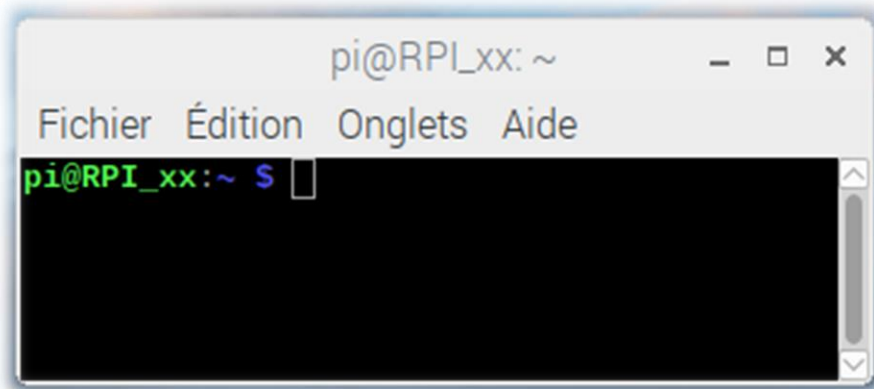
Or, bon nombre de systèmes embarqués ne disposent pas d'écran ou d'IHM (Interface Homme-Machine). On va donc s'habituer un peu à utiliser la Raspberry en ligne de commande.

Pour cela, ouvrez un terminal X en cliquant sur l'icône suivant du tableau de bord :

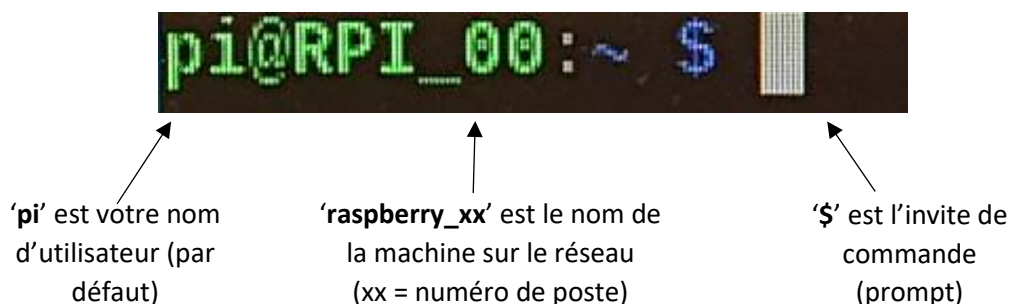


(ou menu > accessoires > LXTerminal)

Vous devriez voir s'ouvrir la fenêtre suivante :



Que signifie cette première ligne qui est affichée dans la fenêtre ?



Le '~' a aussi une signification. Dans l'arborescence de Linux, le répertoire 'utilisateurs' est le répertoire '/home'. Etant par défaut l'utilisateur 'pi', votre répertoire d'utilisateur est '/home/pi'. Par défaut, au démarrage du système, vous vous trouvez dans votre répertoire personnel. Ce dernier est également matérialisé par '~' (raccourci).

On aura fréquemment besoin au cours de ces séances de taper des commandes dans ce terminal, même si beaucoup de choses peuvent se faire via les interfaces graphiques du système d'exploitation ou des applications qui ont été installées.

Mais encore une fois, l'utilisation du terminal est indispensable lorsque l'on veut créer un système 'Headless' (entendre par ce terme un système sans dispositif d'affichage de type écran et sans périphérique utilisateur de type clavier ou souris).

On peut quasiment tout faire en ligne de commande, mais c'est bien sûr plus laborieux et moins convivial. Pour le moment, étant maintenant connecté au web, on va se contenter d'installer une application bien utile qui nous permettra d'avoir le clavier numérique verrouillé au démarrage de session.

Pour cela, tapez l'instruction suivant :

**\$ sudo apt-get install numlockx**

Décrivons rapidement le contenu de cette ligne de commande :

- '**sudo**' : spécifie que cette commande doit se faire en mode 'super utilisateur' ou 'administrateur'
- '**apt-get**' : spécifie que l'on va aller chercher via le réseau un 'paquet' (entendre par là un ensemble de fichiers) sur les serveurs dédiés au système d'exploitation Raspbian.
- '**install**' : spécifie que l'on va lancer l'installation de ce 'paquet'.
- '**numlockx**' : spécifie le nom du paquet à rechercher et à télécharger depuis le serveur.

Ce type de commande est très utile sur Raspberry puisqu'elle permet d'installer des applications, de nouveaux langages de programmation, des bibliothèques pour ces langages de programmation, ...

La seule contrainte est qu'il faut connaître la syntaxe exacte de paquet à installer et savoir si son installation ne dépend d'autres paquets qui doivent être installés au préalable.

Des informations complémentaires très utiles concernant cette commande sont disponibles à l'adresse suivante : <https://manpages.debian.org/jessie/apt/apt-get.8.fr.html>

## Mettre à jour le système :

On va également profiter de ce premier TP et de notre connexion au web pour maintenir à jour notre système d'exploitation et toutes les applications et bibliothèques déjà installées.

1. Dans le terminal de commande, taper la commande '**\$ sudo apt-get -y update**'. Cette commande se charge de mettre à jour la liste des paquets (entendre par là indirectement les logiciels, les bibliothèques, les drivers ...). Cette opération est assez rapide (environ 2') et il convient de la faire régulièrement.
2. Taper ensuite la commande '**\$ sudo apt-get -y dist-upgrade**'. Cette commande se charge quant à elle de mettre à jour tous les paquets déjà installés. Cette opération est plus ou moins longue selon l'ancienneté de la dernière mise à jour (ou de la dernière installation).

On se contentera d'effectuer ces actions une seule fois pour l'ensemble de nos TPs, car il ne devrait pas y avoir de mise à jour majeure sur la courte durée de ce module.

## Récupération des fichiers utiles pour cette séance :

Au début de chaque séance de TP, vous aurez à récupérer des fichiers utiles pour l'écriture de vos programmes. Ces fichiers sont stockés en ligne via un outil de gestion de code très utilisé dans l'industrie : **Github**. Il permet notamment une gestion toujours à jour des versions de code et un travail collaboratif.

L'utilisation de Github sera transparente pour vous, vous aurez juste à cloner le répertoire en ligne vers votre Raspberry pour une utilisation en local.

Pour cela, tapez la commande suivante dans le terminal :

```
$ sudo git clone https://github.com/MbottinIUT/IUT\_IPE\_TP1.git
```

[rem : il n'y a pas d'espace dans ce lien, juste des '\_']

Une fois la tâche effectuée, vous devriez retrouver, en ouvrant un explorateur de fichiers, un dossier '**IUT\_IPE\_TP1**' sous '**/home/pi**'.

## Découvrir l'IDE Python :

Dans le cadre de ces TP, vous allez découvrir un nouveau langage de programmation, le **PYTHON**.

La raspberry étant un ordinateur, en installant un compilateur ou une machine virtuelle, vous pourriez quasiment programmer dans n'importe quel langage connu. On aurait donc pu programmer en C puisque c'est un langage que vous utilisez déjà, puis dériver vers le C++ pour l'approche objet.

Mais le **PYTHON** reste le langage le plus utilisé sur la Raspberry, la communauté est très importante et c'est un langage facile tout en restant très puissant.

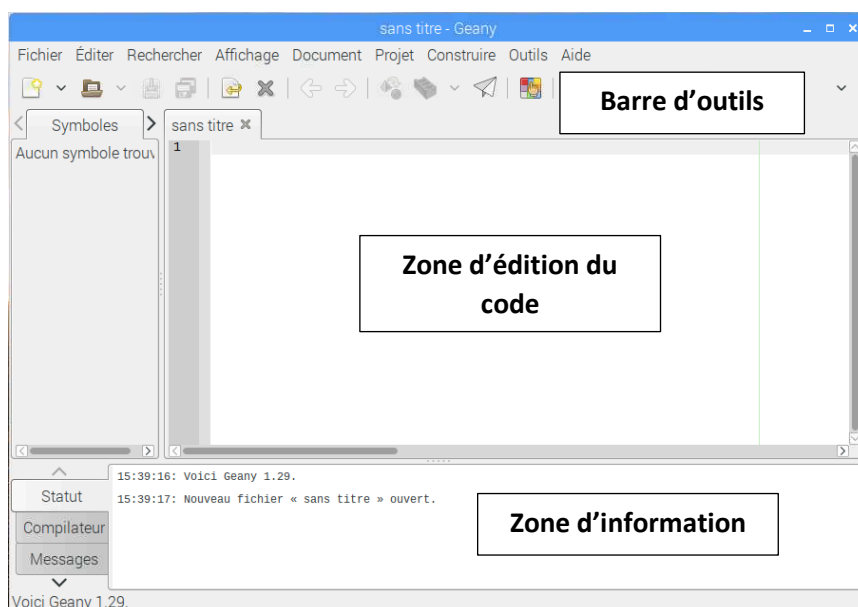
A l'heure actuelle, Python coexiste en deux versions : la **2.x** et la **3.x**. L'industrie passe progressivement en 3.x, mais beaucoup de bibliothèques disponibles sont encore en 2.x. Donc pour nous faciliter la tâche, nous resterons pour le moment sur une version 2.x.

Pour développer en Python, il existe des éditeurs graphiques qui traduisent le code graphique en code Python : Scratch (avec Pyblock <http://mathematiques.discipline.ac-lille.fr/accueil/ressources/pyblock> ), Wylidrin, Snap (<https://snap.berkeley.edu/snapsource/snap.html> ), Blockly de Google (avec Sofuspy), Trinket, ... Ces éditeurs permettent de développer d'une manière simple et rapide des programmes sur Raspberry.

Toutefois, dans le cadre de votre formation DUT, il est important de savoir programmer directement dans un langage sans passer par des outils de simplification qui sont très peu utilisés dans l'industrie.

Vous allez donc écrire ici vos premiers vrais programmes en Python sous un éditeur (IDE) populaire sur Raspberry appelé '**Geany**' et préinstallé dans le système d'exploitation.

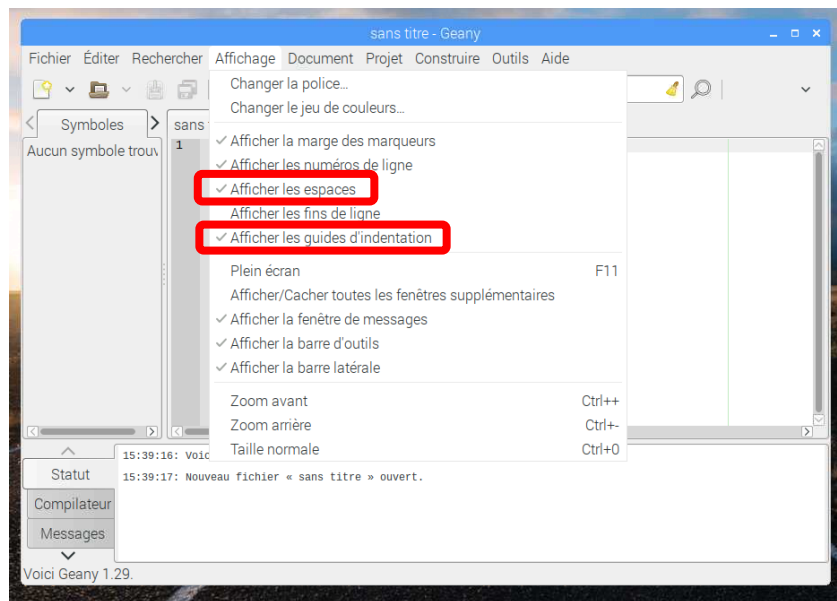
Vous lancez cet éditeur via le menu '**Programmation**'. Comme nous allons en avoir besoin à chaque séance de TP, je vous conseille de faire un clic droit sur l'entrée du menu et de choisir '**Add to Desktop**' afin d'obtenir un raccourci sur votre bureau :



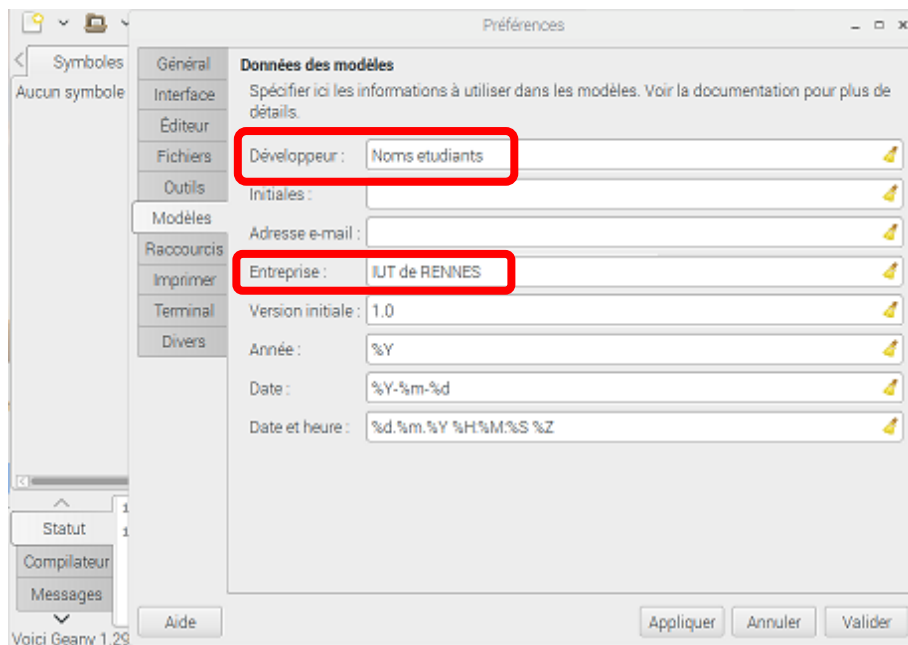


Encore une fois, nous allons faire un peu de configuration afin de nous faciliter l'écriture de programme par la suite.

Pour commencer, ouvrez le menu '**Affichage**' et sélectionnez les deux options suivantes :



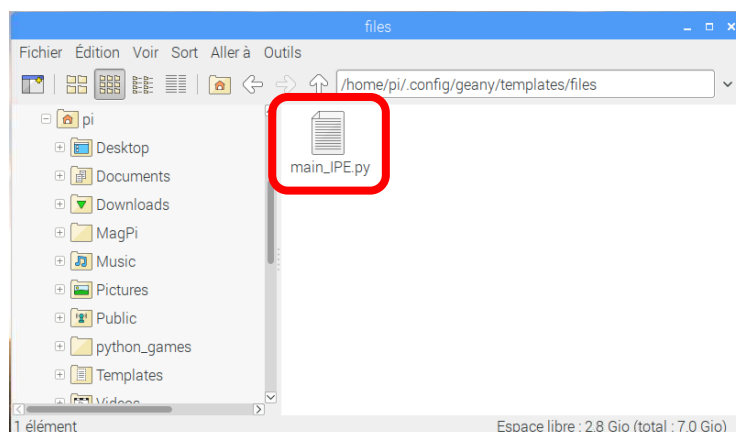
Allez ensuite dans les préférences (menu '**Editer > Préférences**') et dans l'onglet '**Modèles**', rentrez vos informations :



Vous pouvez également dans l'onglet '**Editeur**' cocher le renvoi à la ligne. Cela évitera d'avoir des lignes de plus de 72 caractères.

Enfin, ouvrez un explorateur de fichiers. Allez dans le répertoire '**IUT\_IPE\_TP1/Templates\_Python**' et copiez les fichiers '**main\_TP1.py**' et '**main\_IPE.py**'.

Collez ensuite ces fichiers dans le répertoire '**/home/pi/.config/geany/templates/files**' (vous aurez à taper le chemin dans la barre d'adresse car il s'agit d'un répertoire caché) :



Voilà l'éditeur est maintenant configuré pour notre usage. Pensez à le fermer et le redémarrer pour valider ces changements.

**Remarque :** 'Geany' est un éditeur multi-langages. Il n'est donc pas spécialisé pour Python. En conséquence, il n'y a pas de débogueur intégré pour faciliter le dépannage de code, mais nous verrons comment se débrouiller malgré tout. Un autre éditeur 'Thonny' est également disponible via le menu 'programmation'. Celui-ci est exclusivement dédié à Python et dispose d'un débogueur intégré. Toutefois, il ne travaille qu'avec la version 3.x de Python, et donc pour les raisons évoquées un peu plus haut, nous ne l'utiliserons pas dans le cadre de ce module.

## Ecriture des premiers programmes en Python :

Vous n'avez pas eu de cours sur le langage de programmation Python, car le but ici est de simplement vous initier à ce langage dont la popularité ne cesse d'augmenter...

Du coup, il va falloir se jeter dans le vide, plonger dans l'inconnu, mais avec les bases de programmation que vous avez déjà acquises (en C) et la facilité d'utilisation du Python, cela devrait bien se passer !!

Aussi, pour écrire les programmes demandés, vous vous aiderez du glossaire des principales instructions Python dans le document '**Mémento Raspberry - Linux-Python**' disponible en salle de TP ainsi qu'en version numérique dans le dossier '**IUT\_IPE\_TP1/Documents**'. Mais vous aurez également intérêt à utiliser les pages web de référence suivantes :

- Syntaxe générale du langage Python 2.x : <https://docs.python.org/fr/2/index.html>  
(Le lien '**Référence du langage**' de la page vous donnera la syntaxe des instructions de base Python)
- Syntaxe des modules et fonctions natives : <https://docs.python.org/fr/2/library/index.html>
- Condensé du langage Python en français : <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python>

**ATTENTION**, ceci est valable pour l'ensemble des TP et intervient fortement dans votre évaluation :

- Vos programmes doivent être clairs et aérés.
- Utilisez des noms de variables explicites (pas de toto ou de temp1, temp2, ...) lorsque celles-ci ne vous sont pas imposées
- Mettez un maximum de commentaires. Un commentaire en Python est précédé du caractère '#'. Il peut être en début de ligne ou à la suite d'une instruction. Commentez tout un bloc en le commençant par " " " et en le terminant par " " "
- Si votre ligne de code est trop longue, vous pouvez l'interrompre par '\ ' et la finir sur la ligne suivante.
- Ne mettez qu'une seule instruction par ligne.
- L'indentation est primordiale en Python, utilisez toujours pour cela la touche '**TAB**' et surtout pas la touche '**espace**' à la place.

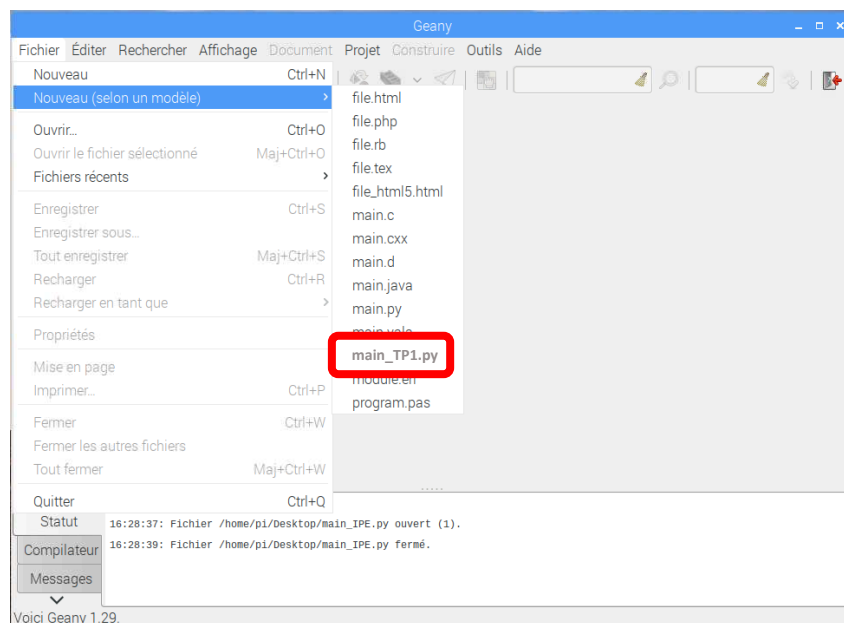
- La sauvegarde de chacun de vos programmes se fera dans le dossier correspondant (ex : 'IUT\_IPE\_TP1' pour le TP n°1) et le nom des programmes sera celui de l'exercice (ex : 'Ex01.py').

Forts de toutes ces recommandations, on peut commencer nos premiers programmes.

## EXERCICE 1 : 'Ex01.py'

L'énoncé de ce premier exercice va être un peu long, mais il servira de base à tous les suivants puisque l'on va y décrire la démarche d'écriture d'un programme en Python et son exécution.

- Dans 'Geany', faire 'Fichier > Nouveau (selon un modèle) > main\_TP1.py' :



- Dans l'onglet qui vient de s'ouvrir dans la zone d'édition du code, le squelette de votre premier programme est déjà en place :

```
sans titre.py ✕
1  #!/usr/bin/python
2  # -*- coding : utf-8 -*-
3
4  """
5  Module IPE - TPs Raspberry
6  auteurs : Noms des étudiants
7  Etablissement : IUT de Rennes - Dpt GEII
8  Date : 20.07.2018 15:52:52 Paris, Madrid (heure d'été)
9
10 Commentaires :
11 """
12
13 # Programme principal
```

Description des lignes de ce modèle de programme :

- **Ligne 1** : indique à 'Geany' d'utiliser l'interpréteur 'Python 2.x' lors de l'exécution du programme.
- **Ligne 2** : précise le codage utilisé (**UTF8**) qui va nous permettre d'utiliser des caractères spécifiques à notre alphabet (accents, cédille, ...)

- **Lignes 4 à 11 : ‘docstring’**, ce sont des lignes de commentaires en début de programme qui fourniront des informations importantes via la fonction ‘**help()**’ dans un shell (terminal) Python.
- **Ligne 13** : commentaire indiquant le début du programme
- A partir de la ligne 14, créez une variable ‘**table**’ qui est initialisée pour le moment à 7.  
Remarque : En Python, le type de la variable n’est jamais précisé, c’est l’interpréteur qui se débrouille. Par conséquent, *une variable n’a pas besoin d’être déclarée avant de lui affecter une valeur !!*
- Affichez le texte ‘**Voici la table de multiplication de X**’ En utilisant pour ‘**X**’ la variable ‘**table**’ que vous avez précédemment déclarée. Pour cela, vous utiliserez l’instruction ‘**print**’.  
Remarque : Le langage python est un langage très évolué et particulièrement souple. Par exemple, pour concaténer une chaîne de caractères avec une variable, plein de solutions s’offrent à vous :
  - print ‘Bonjour’, nom
  - print ‘Bonjour ’, nom
  - print ‘Bonjour %s’ % nom
  - print (‘Bonjour %s’ % nom)
  - print ‘Bonjour ’ + str(nom)
  - (...)
- Vous pouvez alors enregistrer votre programme (‘**Fichier > Enregistrer sous**’) avec le nom ‘**Ex01.py**’ dans le dossier ‘**IUT\_IPE\_TP1**’, puis le lancez en cliquant sur l’icône :



## EXERCICE 2 : ‘Ex02.py’

- Enregistrez le programme précédent sous le nom ‘**Ex02.py**’ avant de le modifier. Vous ferez de même à chaque nouveau programme afin de conserver un fichier par exercice.
- On va compléter le programme précédent en affichant maintenant la table de multiplication associée à la variable ‘**table**’. Il faut donc créer une boucle qui balaye de 0 à 9. Comme en C, il existe les instructions ‘**for**’ ou ‘**while**’ (pas de ‘do...while’ ici). A l’aide du memento ou du site web de référence, créez une boucle de 0 à 9 à la suite du code précédent.
- A l’intérieur de cette boucle (attention à l’indentation !!), affichez le message suivant ‘**A x B = C**’ où A est notre variable ‘**table**’, B notre index de boucle et C le résultat de la multiplication.  
Remarque : Pour l’affichage, on va toujours faire appel à la fonction ‘**print**’, mais elle doit intégrer plusieurs variables dynamiques. Là aussi Python est très souple et plusieurs solutions sont possibles :
  - print A, ‘x’, B, ‘=’, C
  - print str(A) + ‘ x ’ + str(B) + ‘ = ’ + str(C)
  - print (‘%s x %s = %s’ %(A,B,C))
  - print("{} x {} = {}".format(A, B, C))
  - (...)

## EXERCICE 3 : ‘Ex03.py’

- Au lieu de fixer la valeur de ‘**table**’, on va maintenant demander à l’utilisateur la valeur qu’il souhaite. Pour cela, on va utiliser la fonction ‘**input()**’ (information sur le deuxième lien de la page 18 et sur la page allez au chapitre 2). Affichez la question ‘**Quelle table de multiplication souhaitez-vous ?**’ et stockez le résultat dans ‘**table**’. N’oubliez pas de retirer l’initialisation de la variable ‘**table**’ que l’on utilisait dans les deux précédents exercices.



## EXERCICE 4 : 'Ex04.py'

- Lorsque le programme vous demande un nombre, essayez donc de rentrer une ou plusieurs lettres. Que se passe-t-il ? Identifiez le type d'erreur qui s'est produite.
- Au lieu de planter avec un message d'erreur pas vraiment compréhensible pour les néophytes, il serait intéressant de pouvoir gérer ces erreurs dans notre programme et informer l'utilisateur. Python a un bloc d'instructions dédié pour cela. Il s'agit du **'try : .... Except Error : ....'**. Inspirez-vous de ce qui est décrit dans la page web suivante pour modifier le programme afin de gérer cette erreur : <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/231688-les-exceptions>

Faites contrôler votre programme par un enseignant.

## EXERCICE 5 : 'Ex05.py'

Après avoir révisé nos tables de multiplication, on va continuer de travailler autour des opérations mathématiques.

Vous allez écrire un programme dont le cahier des charges est le suivant :

- Demander à l'utilisateur de rentrer deux nombres en utilisant l'exercice 4 pour la gestion des erreurs.
- Demander à l'utilisateur de rentrer une opération : '+', '-', '\*' ou '/'
- Afficher alors l'opération des deux nombres et le résultat

Dans ce programme, vous n'utilisez que des instructions vues dans les programmes précédents.

*Remarque :* pour la division, si vous avez entré deux nombres entiers, le résultat de la division en Python sera entier même s'il ne l'est pas dans la réalité. Pour forcer la division à avoir un résultat décimal, il faut forcer un des deux nombres à être considéré comme décimal. Cela s'appelle le forçage de type (en anglais 'cast')

nombre = 5  
float(nombre) donne 5.0

## EXERCICE 6 : 'Ex06.py'

On va modifier le programme précédent pour le rendre plus lisible. Pour cela, on va utiliser des fonctions comme on peut le faire en C.

La création de fonctions en Python est relativement simple et ne nécessite pas de prototypage. Elle doit simplement avoir été créée avant d'être utilisée. Voici l'exemple de la création d'une fonction de multiplication :

```
def multiplication (nb1=0, nb2=0) :  
    resultat = nb1 * nb2  
    return resultat
```

- Le mot **'def'** est le mot clef dédié pour la création d'une fonction
  - Vient ensuite le nom de la fonction avec des parenthèses ( )
- S'il y a passage de paramètres, les paramètres sont spécifiés à l'intérieur des parenthèses sans préciser leurs types. Vous pouvez, en faisant suivre un paramètre du signe '=' lui attribuer une valeur par défaut.

- Si la fonction renvoie un paramètre, cela se fait par le mot clef **'return'**.
- Comme en C, une variable utilisée pour la première fois dans la fonction, sera locale à cette fonction. On peut la faire précéder du mot clef **'global'** pour indiquer qu'elle est déjà définie ailleurs dans le code.

Modifiez donc le programme précédent en gérant le calcul via une fonction **'traitement\_operation'** qui aura trois paramètres : les deux nombres choisis par l'utilisateur et l'opération choisie. Cette fonction renverra le résultat de l'opération sans gérer l'affichage.

**Faites contrôler votre programme par un enseignant.**

## EXERCICE 7 : 'Ex07.py'

On va aborder une fonctionnalité avancée offerte par la Raspberry et très en vogue en ce moment avec les objets connectés type 'virtual assistant' ou 'machine-learning' : la synthèse vocale  
On va donc donner de la voix à notre précédent programme.

Pour offrir une lecture sonore, deux sorties audios sont disponibles sur la Raspberry :

- Une sortie numérique HDMI (activée par défaut) utile si vous disposez d'un moniteur avec haut-parleur ou sur un téléviseur HDMI.
- Une sortie analogique via un connecteur Jack 3.5mm, de moins bonne qualité mais utilisable dans tous les autres cas notamment en embarqué.

Dans notre cas, on va utiliser la sortie analogique. Vous devez commencer par l'activer en faisant un clic droit sur l'icône de volume et en choisissant **'analog'** :



Il va donc falloir installer un haut-parleur sur cette sortie audio. Vous allez y brancher un haut-parleur de ce type :



Pour éviter qu'il ne se décharge, vous brancherez le câble USB sur un port USB libre de la Raspberry en plus du câble jack audio.

Le langage Python ne dispose pas en natif (i.e. dans la version standard de base du langage) d'instructions de synthèse vocale. On va donc faire appel à une 'bibliothèque' (classe d'objets) externe.

Il existe un nombre important de solutions pour générer une voix numérique.

Nous allons commencer par une solution hors-ligne, c'est-à-dire une solution qui n'a pas besoin d'internet pour générer le son. En contrepartie, on ne peut pas s'attendre à une synthèse de qualité étant donné que les 'voix' sont stockés sur la carte SD et qu'elles doivent rester peu volumineuses en termes d'espace disque.

Ouvrez donc un terminal et tapez la commande suivante :

```
$ sudo pip install pyttsx
```

[rem : on a utilisé en page 14 la commande '**apt-get**' pour récupérer un paquet. C'est la commande la plus souvent utilisée, mais tous les paquets existants ne sont pas toujours disponibles au même endroit. D'autres commandes existent donc comme ici '**pip**']

Vous trouverez des exemples d'utilisation de cette classe à la page suivante :

<https://pyttsx.readthedocs.io/en/latest/engine.html>

Vous pouvez commencer par faire un test en vous appuyant sur le premier exemple fourni ('**Speaking text**')

Modifier alors votre précédent programme (n'oubliez pas de le renommer en '**Ex07.py**' en l'enregistrant) pour y intégrer ces nouvelles fonctionnalités :

- En plus d'afficher le résultat dans la console, écrire une fonction qui annoncera l'opération et son résultat dans le haut-parleur. Exemple : '5+6=11' sera énoncé 'cinq plus six égal onze'
- Modifier les propriétés de l'objet 'engine' (en reprenant le nom donné dans l'exemple en ligne) de façon à ralentir le rythme (paramètre '**rate**') et à changer la voix pour qu'elle soit réglée sur le français (paramètre '**voice**' → '**french**')

**Faites contrôler votre programme par un enseignant.**

## EXERCICE 8 : 'Ex08.py'

Comme vous avez pu le constater, la qualité n'est pas forcément au rendez-vous. Cette synthèse est donc plutôt adapté à un robot non connecté ou à tout autre système embarqué non connecté.

Voyons maintenant comment utiliser une solution en ligne pour améliorer la qualité.

Là aussi, il existe un certain nombre de solutions :

- Google TTS API
- IBM TTS Watson API
- Amazon TTS Polly
- Android TTS Pico
- (...)

Elles se différencient surtout par le nombre de langues 'parlées', la gratuité ou le coût du service, le nombre de requêtes autorisées, ...

On va ici utiliser le module de Google. Le nom du package Python est '**gTTS**'.

Vous pouvez l'installer via un terminal par la commande :

```
$ sudo pip install gTTS
```

Toutefois, cette installation ne va pas suffire. En effet, cet outil de synthèse vocale passe par la génération d'un fichier mp3 qui est ensuite téléchargé et sauvegardé en local sur votre disque (ici dans le dossier de votre code Python).

Il nous faut donc savoir comment lire un tel fichier. Là aussi, Python n'est pas équipé en natif. Il vous faut donc installer le module suivant via le terminal :

```
$ sudo apt-get install mpg321
```

Un exemple d'utilisation de ces deux librairies vous est proposé ci-dessous :

```
from gtts import gTTS
import os

tts = gTTS(text=u'Bonjour, je déteste les tomates', lang='fr')

tts.save("bonjour.mp3")

os.system("mpg321 bonjour.mp3")
```

Modifier votre programme précédent pour utiliser maintenant le moteur de synthèse vocale '**gTTS**'. Comme vous pouvez le constater, la qualité est bien meilleure.

Gardez ces méthodes dans un coin de votre tête. Leur simplicité d'utilisation (quelques lignes de code suffisent) pourront vous permettre d'agrémenter vos codes au gré des prochains TPs !!

**Faites contrôler votre programme par un enseignant.**