

In [1]:

```
from IPython.core.display import HTML
css_file = './custom.css'
HTML(open(css_file, "r").read())
```

Out[1]:

M62_CM1 Introduction aux Notebooks IPython.

Table of Contents

- ▼ [1 Notebook Jupyter](#)
 - [1.1 Installation simple](#)
 - [1.2 Exporter un notebook](#)
- ▼ [2 Manipulation du notebook](#)
 - [2.1 Mode commande \(presser la touche \[Esc\].\)](#)
 - [2.2 Mode édition \(presser la touche \[Enter\].\)](#)
- ▼ [3 Mode édition : les cellules *Markdown* et sa syntaxe](#)
 - [3.1 La syntaxe](#)
 - [3.2 Les titres](#)
 - [3.3 Les paragraphes](#)
 - [3.4 L'emphase](#)
 - [3.5 Les citations](#)
 - [3.6 Les listes non ordonnée](#)
 - [3.7 Les listes ordonnées](#)
 - [3.8 Le bloc de code](#)
 - [3.9 Les filets ou barres de séparation](#)
 - [3.10 Les liens](#)
 - [3.11 Les images](#)
 - [3.12 Les tableaux](#)
 - [3.13 Échappement des caractères](#)
- ▼ [4 Mode édition: les équations](#)
 - [4.1 Les modes mathématiques](#)
 - [4.2 Indices et exposantes](#)
 - [4.3 Symboles d'usage courant](#)
 - [4.4 Racine carrée, racine n-ième](#)
 - [4.5 Fractions et coefficients binomiaux](#)
 - [4.6 Lettres grecques](#)
 - [4.7 Texte dans une formule *displaystyle*](#)
 - [4.8 Fonctions mathématiques](#)

- [4.9 Grands opérateurs: intégrales, sommes, produits, etc.](#)
- [4.10 Accents mathématiques](#)
- [4.11 Délimiteurs](#)
- [4.12 Alphabets mathématiques](#)
- [4.13 Tableaux mathématiques](#)
- [4.14 Distinction de cas](#)
- [4.15 Matrices](#)
- [4.16 Alignement de formules](#)
- [4.17 Flèches extensibles](#)
- [4.18 Modules de congruences](#)
- [4.19 Placer au-dessus ou en-dessous](#)
- ▼ [5 Mode édition : les cellules Code](#)
 - [5.1 Les exceptions](#)
 - [5.2 Entrée et sortie standard](#)
 - [5.3 L'aide en ligne IPython](#)
 - [5.4 Les commandes systèmes](#)
 - ▼ [5.5 Les commandes magiques de IPython](#)
 - [5.5.1 %matplotlib inline & %matplotlib notebook](#)
 - [5.5.2 %autosave 300](#)
 - [5.5.3 %who , %whos et %reset](#)
- [6 Références](#)

1 Notebook Jupyter

Jupyter est une application Web qui regroupe intimement deux fonctionnalités très différentes :

- Un outil qui permet de créer des documents multimédia intégrant du texte, des formules mathématiques, des graphiques, des images, voire des animations et des vidéos.
- Une interface qui permet d'exécuter du code informatique. Pour cela *Jupyter* s'appuie sur des programmes indépendants capables d'interpréter le langage dans lequel est écrit ce code. Dans la terminologie de *Jupyter* ces interpréteurs sont appelés des noyaux (kernel en anglais). Nous utiliserons le noyau pour le langage *Python 3*.

Les documents *Jupyter* sont appelés des **notebooks**. Un fichier notebook est reconnaissable par son extension `.ipynb`. Le document que vous lisez actuellement est un notebook.

Pour commencer, dans un terminal, lancer la commande `jupyter notebook &` (ou `anaconda-navigator &`). Cela ouvrira automatiquement un navigateur dans lequel on peut alors commencer à travailler. L'onglet principal représente l'arborescence des fichiers.

1.1 Installation simple

La façon la plus simple d'installer *Python* et *Jupyter* est d'installer *Anaconda*. Les procédures d'installations détaillées selon chaque système d'exploitation sont décrites à l'adresse :

<https://docs.anaconda.com/anaconda/install/>

(<https://docs.anaconda.com/anaconda/install/>). Les procédures suivantes sont un résumé rapide de la procédure d'installation.

- Installation sous Windows.
 1. Télécharger Anaconda 5.2 (ou plus récent) pour Python 3.6 (ou plus récent) à l'adresse :
<https://www.anaconda.com/download/#windows>
(<https://www.anaconda.com/download/#windows>) ;
 2. Double cliquer sur le fichier téléchargé pour lancer l'installation d'Anaconda, puis suivre la procédure d'installation (il n'est pas nécessaire d'installer VS Code).
 3. Une fois l'installation terminée, lancer Anaconda Navigator à partir du menu démarrer.
- Installation sous macOS.
 1. Télécharger Anaconda 5.2 (ou plus récent) pour Python 3.6 (ou plus récent) à l'adresse :
<https://www.anaconda.com/download/#macos>
(<https://www.anaconda.com/download/#macos>) ;
 2. Double cliquer sur le fichier téléchargé pour lancer l'installation d'Anaconda, puis suivre la procédure d'installation (il n'est pas nécessaire d'installer VS Code).
 3. Une fois l'installation terminée, lancer Anaconda Navigator à partir de la liste des applications.
- Installation sous Linux.
 1. Télécharger Anaconda 5.2 (ou plus récent) pour Python 3.6 (ou plus récent) à l'adresse :
<https://www.anaconda.com/download/#linux>
(<https://www.anaconda.com/download/#linux>) ;
 2. Exécuter le fichier téléchargé avec bash puis suivre la procédure d'installation (il n'est pas nécessaire d'installer VS Code).
 3. Une fois l'installation terminée, taper anaconda-navigator dans un nouveau terminal pour lancer Anaconda Navigator.

1.2 Exporter un notebook

Lorsque vous avez terminé la rédaction de votre document, vous pouvez le télécharger sur l'ordinateur local dans le format qui vous convient.

Choisir pour cela `File/Download as...`

Pour mettre vos notebooks en ligne : [nbviewer](http://nbviewer.ipython.org/)
(<http://nbviewer.ipython.org/>).

2 Manipulation du notebook

Le notebook est constitué d'une succession de cellules comportant

- soit du texte en *Markdown* comme ici,
- soit du code comme dans la cellule suivante (Python pour nous):

In [2]:

```
x=range(3)
```

Un liseré repère la cellule actuellement sélectionnée.

Les cellules peuvent être dans le *mode commande* ou le *mode édition*:

- *mode commande* : permet de se déplacer d'une cellule à l'autre et d'exécuter les cellules
- *mode édition* : permet de modifier le contenu d'une cellule.

2.1 Mode commande (presser la touche **[Esc]**)

Pour entrer dans le mode commande de la cellule sélectionnée, il suffit de presser la touche **[Esc]** . Quand vous êtes dans le mode commande, vous pouvez ajouter ou supprimer des cellules mais vous ne pouvez pas saisir de texte dans une cellule.

Voici les raccourcis principaux disponibles en mode commande :

- **[ESC]** : Passe dans le mode commande
- **M** : Transforme la cellule en une cellule de type Markdown.
- **Y** : Transforme la cellule en une cellule de type Code.
- **[Ctrl-Enter]** : Exécute la cellule.
- **[Shift-Enter]** : Exécute la cellule et sélectionne la cellule suivante. L'appui répété de cette touche permet ainsi d'exécuter pas à pas toutes les cellules du notebook.
- **[Alt-Enter]** : Exécute la cellule et insère une nouvelle cellule juste en dessous.
- **[Ctrl-S]** ou **S** : Save and Checkpoint
- **I, I** : Interrompt l'exécution du code.
- **0, 0** : Redémarre l'interpréteur. Il se retrouve alors dans son état initial.
- **A** et **B** : Insèrent une nouvelle cellule, respectivement au-dessus ou

au-dessous de la cellule sélectionnée.

- X et C : Respectivement coupe ou copie la cellule sélectionnée.
 - V et [Shift-V] : Colle la cellule copiée/coupée respectivement au-dessus ou au-dessous.
 - D, D : Supprime la cellule sélectionnée.
 - Z : Annule la dernière suppression de cellule.
-
- [Alt-Down] ou [Alt-Up] : Déplace la cellule sélectionnée respectivement au-dessus ou en-dessous
 - [Shift-Down] ou [Shift-Up] : sélectionne la cellule respectivement au-dessus ou en-dessous
 - [Shift-J] ou [Shift-K] : étende la sélection aux cellules respectivement au-dessus ou en-dessous
 - [Shift-M] : merge selected cells, or current cell with cell below if only one cell is selected
 - [Shift-space] ou [space] : scroll notebook up or down
-
- F : find and replace
 - L : toggle line numbers
 - [Shift-L] : toggles line numbers in all cells, and persist the setting
 - O : toggle output of selected cells
 - [Shift-O] : toggle output scrolling of selected cells
-
- H : Affiche la liste de tous les raccourcis clavier.
 - [Shift-A] : Insert a heading cell above the selected cell
 - [Shift-B] : Insert a heading cell below the selected cell's section

2.2 Mode édition (presser la touche **[Enter]**)

Pour entrer dans le mode édition de la cellule sélectionnée, il suffit de presser la touche [Enter] ou de double-cliquer à l'intérieur de la cellule. Quand une cellule est en édition vous pouvez saisir du texte comme dans un éditeur classique.

Lorsque le curseur est en début de ligne ou lorsque vous avez sélectionné du texte, l'appui sur la touche [Tab] (respectivement [Shift-TAB]) indente (respectivement désindente) les lignes correspondantes.

Voici d'autres raccourcis clavier :

- [Ctrl-A] : Sélectionne tout le texte de la cellule.
- [Ctrl-Z] : Annule les dernières saisies de texte.
- [Ctrl-Enter] : Exécute la cellule.
- [Shift-Enter] : Exécute la cellule et sélectionne la cellule suivante. L'appui répété de cette touche permet ainsi d'exécuter pas à pas toutes les cellules du notebook.
- [Alt-Enter] : Exécute la cellule et insère une nouvelle cellule juste en dessous.
- [ESC] : Passe dans le mode commande

3 Mode édition : les cellules *Markdown* et sa syntaxe

La touche **M** transforme la cellule sélectionnée en type *Markdown*. Vous pouvez alors rédiger du texte enrichi (titres, sous-titres, **gras**, *italique*, alinéas, tableaux, liens hypertexte, etc).

Le texte de la cellule doit être rédigé en langage *Markdown* qui est un langage de balisage léger. La syntaxe markdown est facile à apprendre (le plus simple est d'ailleurs de regarder des exemples de documents). Voici quelques liens :

- [Markdown Cheatsheet \(https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet\)](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
- [Markdown for Jupyter notebooks cheatsheet \(https://medium.com/ibm-data-science-experience/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed\)](https://medium.com/ibm-data-science-experience/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed)

Il est également possible d'insérer des morceaux de texte en langage *TEX* pour composer des expressions mathématiques.

3.1 La syntaxe

Markdown est un système d'édition et de formatage de texte. La philosophie du système veut que le texte écrit soit lisible sans interpréteur particulier en mode texte. Il est léger et épuré de l'essentiel de la verbosité d'un langage balisé.

Les éléments de syntaxe sont des caractères de ponctuation qui font sens visuellement même non convertis. Une fois converti, le navigateur web (qui joue alors le rôle d'interpréteur) en rendra la lecture plus claire.

Vous pouvez tout à fait introduire directement des balises HTML dans votre texte. Attention, le formatage markdown ne sera pas appliqué à l'intérieur de ces balises.

3.2 Les titres

Il est possible de titrer en ajoutant des dièses (#) au début de la ligne. Il n'y a pas de limite au nombre de niveaux de titres qu'il est possible d'utiliser (mais n'allez pas au delà de 4).

```
# Title 1
## Title 2
### Title 3
#### Title 4
##### Title 5
```

3.3 Les paragraphes

Pour afficher un paragraphe, laissez une ligne vide.

Effectuer un saut de ligne simple dans votre texte markdown n'aura aucun effet. Sauf si vous terminez votre ligne par un double espace (ou plus que ça)...
comme ici.

3.4 L'emphase

Pour formater une partie de votre texte comme emphase, entourez le par des astérisques `*` ou des underscores `_`. Entourer par un signe unique passe en *italique* et par un double signe en **gras**. Il est possible de **combiner** les deux.

Un double tildes `~~` vous permettent de barrer le texte.

3.5 Les citations

Pour afficher un bloc de citation, commencez le paragraphe par un chevron fermant `>`. Si votre bloc contient plusieurs lignes, vous pouvez faire des sauts de lignes à la main et toutes les ouvrir par un chevron fermant, mais ce n'est pas nécessaire. Ces bloc peuvent contenir d'autres éléments markdown comme des titres ou des listes.

3.6 Les listes non ordonnée

Pour afficher une liste, commencez la ligne par une astérisque `*`, un moins `-` ou un plus `+`. Là encore, le choix n'a pas d'importance, mais il faut rester cohérent dans votre document.

```
+ item  
+ item  
+ item
```

donne

- item
- item
- item

3.7 Les listes ordonnées

Pour afficher une liste ordonnée, commencez la ligne par un nombre suivit d'un point.

1. item
123. item
17. item

donne

1. item
2. item
3. item

3.8 *Le bloc de code*

Pour afficher un bloc de code, sautez deux lignes comme pour un paragraphe, puis indentez avec 4 espaces ou une tabulation. Pour afficher du code dans une ligne, il faut l'entourer par des guillemets simples (```).

3.9 *Les filets ou barres de séparation*

Pour afficher un filet de séparation, entrez dans votre texte au moins 3 astérisques `*` ou 3 moins `-` sur une ligne entourée de sauts de lignes. Il est possible de les séparer par des espaces.

3.10 *Les liens*

Il y a deux façons d'afficher un lien.

1. De manière automatique en encadrant un lien par des chevrons, par exemple `<http://www.google.com>` donne <http://www.google.com> (<http://www.google.com>). Il est alors cliquable et affiche l'url indiquée entre chevrons.
2. Ou en ajoutant des paramètres. Le texte à afficher est alors indiqué entre crochets suivi de l'adresse du lien entre parenthèses, par exemple `[google](http://www.google.com "link to google")` donne [google](http://www.google.com) (<http://www.google.com>). Dans les parenthèses, à la suite du lien, on peut indiquer un titre entre guillemets. Ce titre sera affiché lors du survol du lien dans le navigateur. Il sera également lu par les navigateurs textuels pour les déficients visuels.

3.11 *Les images*

Pour afficher une image en markdown,

- commencez par un point d'exclamation,
- puis indiquez le texte alternatif entre crochets. Ce dernier sera affiché si l'image n'est pas chargée et lue par les moteurs de recherche.
- Terminez par l'URL de l'image entre parenthèses:

- Cette URL peut être un lien vers le web
<https://upload.wikimedia.org/wikipedia/commons/3/>
- ou un chemin local de ce type :
 /dossier_images/nom_de_mon_image.jpg .
- Après le lien vers l'image, il est possible d'ajouter un titre lu par les navigateurs textuels et affiché au survol de l'image par les autres.
 Exemple:
 ![Jupyter logo]
 (<https://upload.wikimedia.org/wikipedia/commons/3/38/>
 donne



Si l'image doit être redimensionnée, utiliser directement du code html.
 Comparer les codes suivants

- `![Google logo]`
 (<https://www.google.fr/images/srpr/logo11w.png>
 "google logo") donne



- `<img src =`
`"https://www.google.fr/images/srpr/logo11w.png"`
`title = "google logo" alt = "Google logo"`
`width="200">` donne

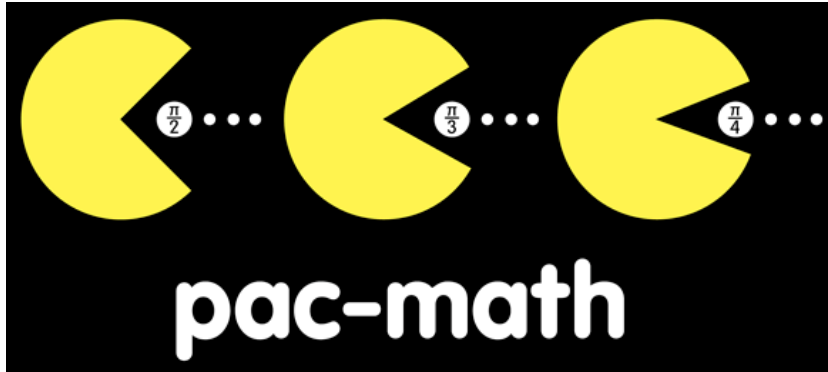


Si on veut qu'une image locale soit directement sauvegardée dans le notebook (par exemple pour partager le notebook sur le web), on peut utiliser une cellule Python et plusieurs méthodes sont possibles:

In [3]:

```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "http://www.apprendre-en-ligne.net/blog/images/pacm
```

Out[3]:



In [4]:

```
# You retain the ability to use HTML tags to resize, etc...
Image(url= "http://www.apprendre-en-ligne.net/blog/images/pacm
```

Out[4]:



In [5]:

```
# You can also display images stored locally, either via relat
PATH = "./Images/"
Image(filename = PATH + "pacmath.png", width=100, height=100)
```

Out[5]:



3.12 Les tableaux

L'idée globale est de "dessiner" des colonnes en les entourant avec des pipes | . Le nombre de colonnes est défini dans la première ligne du tableau et vous devez pour chaque ligne avoir le même nombre de colonnes, même si certaines sont vides.

La première ligne sera votre en-tête. La seconde ligne sépare cet en-tête du corps du tableau, elle ne contient que des tirets - Il n'est plus possible de définir l'alignement des colonnes dans markdown.

Header 1	header 2	header 3
1 Online	1	value
Line 2	2	value
3 Online	3	value

Header 1	header 2	header 3
1 Online	1	value
Line 2	2	value
3 Online	3	value

Assistant en ligne : [Markdown Tables Generator](https://www.tablesgenerator.com/markdown_tables)
(https://www.tablesgenerator.com/markdown_tables)

3.13 Échappement des caractères

Les caractères spéciaux ayant un sens en markdown doivent être échappés. Si vous souhaitez utiliser dans votre texte des astérisques, accolades, dièses... à une position indiquant à markdown que vous désirez un formatage particulier, vous devez les échapper en les faisant précéder d'un antislash `\`. Sinon markdown les masquera et appliquera le formatage correspondant. Les caractères suivants sont à échapper :

`\ * ` - _ [] () {} # + . !`

4 Mode édition: les équations

Les équations peuvent être directement données au format TeX , on peut donc par exemple écrire x_i au milieu d'une ligne ou bien écrire des équations dans une nouvelle ligne et centrées:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

Quelques commandes TeX de base sont expliquées sur [cette page](http://www.tuteurs.ens.fr/logiciels/latex/maths.html#s2) (<http://www.tuteurs.ens.fr/logiciels/latex/maths.html#s2>), et pour plus de détails, vous pourrez consulter mon polycopié d'Initiation à $TeX2e$, disponible [ici](http://faccanoni.univ-tln.fr/user/enseignements/20142015/InitiationLaTeX.pdf) (<http://faccanoni.univ-tln.fr/user/enseignements/20142015/InitiationLaTeX.pdf>), en particulier le chapitre 3.

4.1 Les modes mathématiques

Pour taper des formules mathématiques avec \LaTeX , il y a besoin d'entrer en mode mathématiques.

Il existe deux modes mathématiques : les formules dans le texte (appelée *inline*) et les formules en évidence (dites en *displaystyle*).

- Pour mettre une formule dans le texte, comme $y = f(x)$, il suffit de mettre la formule entre deux dollars: $\$y=f(x)\$$.
Remarque la différence entre le mode texte et le mode mathématique : avec des dollars on a $y = f(x)$ tandis que sans dollars on a $y=f(x)$
- Pour mettre une formule en évidence, comme

$$y = f(x)$$
il faut mettre la formule entre deux doubles dollars $\$y=f(x)\$$

4.2 Indices et exposantes

Deux opérations fondamentales en mode mathématique sont la mise en exposant et la mise en indice.

- Pour obtenir un **indice**, il faut utiliser la touche `_` qui s'obtient avec la touche «8». Par exemple, taper $\$x_1\$$ donnera x_1 . Attention, taper $\$x_{12}\$$ ne donne pas x_{12} mais x_12 : seul le premier caractère tapé après `_` est mis en indice; pour obtenir x_{12} , il faut taper $\$x_{\{12\}}\$$.
- Pour obtenir un **exposant**, il faut utiliser `^` (accent circonflexe qui s'obtient en tapant deux fois la touche «^»). Par exemple, $\$x^2\$$ donne x^2 . De même que pour les indices, $\$x^{23}\$$ donne x^{23} tandis que $\$\{x^{23}\}\$$ donne x^{23} .
- On peut bien sûr **combinaison des deux**, dans l'ordre que l'on veut : $\$x_1^2\$$ ou $\$x^2_1\$$ donnent x_1^2

4.3 Symboles d'usage courant

Résultat	Commande	Résultat	Commande	Résultat	Commande
∞	<code>\infty</code>	\leq	<code>\le</code>	\Leftrightarrow	<code>\iff</code>
\forall	<code>\forall</code>	\leq	<code>\leq</code>	\nearrow	<code>\nearrow</code>
\exists	<code>\exists</code>	\geq	<code>\ge</code>	\searrow	<code>\searrow</code>
\nexists	<code>\nexists</code>	\geq	<code>\geq</code>	\in	<code>\in</code>
∂	<code>\partial</code>	\approx	<code>\approx</code>	\subset	<code>\subset</code>
ℓ	<code>\ell</code>	\simeq	<code>\simeq</code>	\cup	<code>\cup</code>
\pm	<code>\pm</code>	\equiv	<code>\equiv</code>	\cap	<code>\cap</code>
\mp	<code>\mp</code>	\rightarrow	<code>\to</code>	\setminus	<code>\setminus</code>
\times	<code>\times</code>	\mapsto	<code>\mapsto</code>	\emptyset	<code>\emptyset</code>
\neq	<code>\neq</code>	\Rightarrow	<code>\implies</code>		

Pour la négation d'un symbole, on peut utiliser `\not`. Par exemple, $\$F \not\subset E\$$ fournit $F \not\subset E$.

4.4 Racine carrée, racine n-ième

La racine carrée s'obtient par `\sqrt{...}` et la racine n-ième par `\sqrt[n]{...}` :

```

$$
\sqrt{1+x} + \sqrt[3]{1+x}
$$

```

$$\sqrt{1+x} + \sqrt[3]{1+x}$$

4.5 Fractions et coefficients binomiaux

Pour les fractions, on utilise la commande `\frac{}{}` qui prend deux arguments, dans l'ordre le numérateur et le dénominateur.

Pour les coefficients binomiaux on utilise la commande `\binom{}{}`.

Le rendu change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*:

`\frac{1}{2}+1` donne $\frac{1}{2} + 1$

`$$\frac{1}{2}+1$$` donne

$$\frac{1}{2} + 1$$

4.6 Lettres grecques

Pour taper les lettres grecques, il suffit de précéder le nom de la lettre par un *backslash*; par exemple `\alpha` donne α . Voici une liste complète des lettres grecques disponibles sous *TEX*:

Résultat	Commande	Résultat	Commande	Résultat	Commande
α	<code>\alph</code>	ν	<code>\nu</code>	Γ	<code>\Gamma</code>
β	<code>\beta</code>	ξ	<code>\xi</code>	Δ	<code>\Delta</code>
γ	<code>\gamma</code>	π	<code>\pi</code>	Θ	<code>\Theta</code>
δ	<code>\delta</code>	ϖ	<code>\varpi</code>	Λ	<code>\Lambda</code>
ϵ	<code>\epsilon</code>	ρ	<code>\rho</code>	Ξ	<code>\Xi</code>
ε	<code>\varepsilon</code>	ϱ	<code>\varrho</code>	Π	<code>\Pi</code>
ζ	<code>\zeta</code>	σ	<code>\sigma</code>	Σ	<code>\Sigma</code>
η	<code>\eta</code>	τ	<code>\tau</code>	Υ	<code>\Upsilon</code>
θ	<code>\theta</code>	υ	<code>\upsilon</code>	Φ	<code>\Phi</code>
ϑ	<code>\vartheta</code>	χ	<code>\chi</code>	Ψ	<code>\Psi</code>
ι	<code>\iota</code>	ϕ	<code>\phi</code>	Ω	<code>\Omega</code>
κ	<code>\kappa</code>	φ	<code>\varphi</code>		

Résultat	Commande	Résultat	Commande	Résultat	Commande
λ	<code>\lambda</code>	ψ	<code>\psi</code>		
μ	<code>\mu</code>	ω	<code>\omega</code>		

4.7 Texte dans une formule *displaystyle*

La commande `\text{}` permet d'insérer du texte dans une formule *displaystyle* (les commandes `\quad` et `\qquad` insèrent de l'espace autour du texte):

```

$$
y = x^2 \quad \text{et donc} \quad x = \pm\sqrt{y}
$$

```

donne

$$y = x^2 \quad \text{et donc} \quad x = \pm\sqrt{y}$$

4.8 Fonctions mathématiques

Résultat	Commande	Résultat	Commande	Résultat	Commande	Résultat	Commande
exp	<code>\exp</code>	cos	<code>\cos</code>	cosh	<code>\cosh</code>	lim	<code>\lim</code>
ln	<code>\ln</code>	sin	<code>\sin</code>	sinh	<code>\sinh</code>	lim sup	<code>\limsup</code>
lg	<code>\lg</code>	tan	<code>\tan</code>	tanh	<code>\tanh</code>	lim inf	<code>\liminf</code>
log	<code>\log</code>	cot	<code>\cot</code>	coth	<code>\coth</code>	max	<code>\max</code>
inf	<code>\inf</code>	arccos	<code>\arccos</code>	gcd	<code>\gcd</code>	sup	<code>\sup</code>
det	<code>\det</code>	arcsin	<code>\arcsin</code>	hom	<code>\hom</code>	min	<code>\min</code>
deg	<code>\deg</code>	arctan	<code>\arctan</code>	ker	<code>\ker</code>	arg	<code>\arg</code>
dim	<code>\dim</code>						

4.9 Grands opérateurs: intégrales, sommes, produits, etc.

Le rendu des grands opérateurs change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*. Certaines de ces commandes prennent, tout comme `\max` ou `\lim`, des bornes. Le principe est le même, on utilise des indices ou des exposants pour les taper. Le placement des indices et exposants dépend à nouveau de si la formule est mise en évidence ou pas.

`\int_a^b f(x) dx` et `\sum_{i=0}^n u_n` donnent
 $\int_a^b f(x)dx$ et $\sum_{i=0}^n u_n$
`\int_a^b f(x) dx` et `\sum_{i=0}^n u_n` donnent

$$\int_a^b f(x)dx$$

et

$$\sum_{i=0}^n u_n$$

Ponctuellement et seulement si absolument nécessaire, dans une formule *inline*

- on peut forcer le positionnement des indices avec `\limits` :
 $\int_a^b f(x) dx$ et $\sum_{i=0}^n u_n$ donnent $\int_a^b f(x)dx$ et $\sum_{i=0}^n u_n$
- on peut forcer la dimension et le positionnement des indices avec `\displaystyle`
 $\int_a^b f(x) dx$ et $\sum_{i=0}^n u_n$ donnent $\int_a^b f(x)dx$ et $\sum_{i=0}^n u_n$

Résultat <i>inline</i>	Commande	Résultat <i>displaystyle</i>	Commande
\int	<code>\int</code>	\int	<code>\$\$\int\$\$</code>
\iint	<code>\iint</code>	\iint	<code>\$\$\iint\$\$</code>
\iiint	<code>\iiint</code>	\iiint	<code>\$\$\iiint\$\$</code>
Σ	<code>\sum</code>	Σ	<code>\$\$\sum\$\$</code>
Π	<code>\prod</code>	Π	<code>\$\$\prod\$\$</code>
\cup	<code>\bigcup</code>	\cup	<code>\$\$\bigcup\$\$</code>
\cap	<code>\bigcap</code>	\cap	<code>\$\$\bigcap\$\$</code>
\sqcup	<code>\bigsqcup</code>	\sqcup	<code>\$\$\bigsqcup\$\$</code>
\oplus	<code>\bigoplus</code>	\oplus	<code>\$\$\bigoplus\$\$</code>
\otimes	<code>\bigotimes</code>	\otimes	<code>\$\$\bigotimes\$\$</code>
\amalg	<code>\coprod</code>	\amalg	<code>\$\$\coprod\$\$</code>

Pour mettre plusieurs lignes dans les indices, il faut utiliser `\substack{}` ; à l'intérieur de l'argument de `\substack`, on passe à la ligne avec `\\`.

```

$$
L_i(x)=\prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}
$$

```

donne

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

4.10 Accents mathématiques

Voici les accents pouvant se mettre sur une lettre seule:

Résultat	Commande	Résultat	Commande	Résultat	Commande
\tilde{a}	<code>\tilde{a}</code>	\mathring{a}	<code>\mathring{a}</code>	\dot{a}	<code>\dot{a}</code>
\vec{a}	<code>\vec{a}</code>	\grave{a}	<code>\grave{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\hat{a}	<code>\hat{a}</code>	\acute{a}	<code>\acute{a}</code>	\dddot{a}	<code>\dddot{a}</code>
\bar{a}	<code>\bar{a}</code>	\breve{a}	<code>\breve{a}</code>	\ddddot{a}	<code>\ddddot{a}</code>
\check{a}	<code>\check{a}</code>				

Il existe aussi des accents extensibles:

Résultat	Commande	Résultat	Commande	Résultat	Commande
\widehat{abc}	<code>\widehat{abc}</code>	\widetilde{abc}	<code>\widetilde{abc}</code>	\overleftarrow{abc}	<code>\overleftarrow{abc}</code>
\overline{abc}	<code>\overline{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>	\overleftarrow{abc}	<code>\overleftarrow{abc}</code>
\underline{abc}	<code>\underline{abc}</code>	\overrightarrow{abc}	<code>\overrightarrow{abc}</code>	\overleftrightarrow{abc}	<code>\overleftrightarrow{abc}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	\underrightarrow{abc}	<code>\underrightarrow{abc}</code>	$\underleftrightarrow{abc}$	<code>\underleftrightarrow{abc}</code>

Pour `\underbrace` il est possible de placer du matériel en-dessous en utilisant `_` et pour `\overbrace` du matériel au-dessus en utilisant `^` :

```

$$
\underbrace{x^3 + x^2 + x + 1}_ {\to 0}
\quad
\overbrace{x^3 + x^2 + x + 1}^{=0}
$$

```

donne

$$\underbrace{x^3 + x^2 + x + 1}_{\rightarrow 0} \quad \overbrace{x^3 + x^2 + x + 1}^{=0}$$

4.11 Délimiteurs

Résultat	Commande	Résultat	Commande	Résultat	Commande
((/	/	⌊	<code>\lfloor</code>
))	\	<code>\backslash</code>	⌋	<code>\rfloor</code>
[[<code>\lvert</code>	⌈	<code>\lceil</code>
]]		<code>\rvert</code>	⌉	<code>\rceil</code>
{	<code>\{</code>	∥	<code>\lVert</code>	⟨	<code>\langle</code>
}	<code>\}</code>	∥	<code>\rVert</code>	⟩	<code>\rangle</code>
rien	.				

Il est important de comprendre que, même si `\lvert` et `\rvert` se ressemblent, ils ne peuvent pas être interchangés: `\lvert` doit toujours être utilisé pour ouvrir et `\rvert` pour refermer. Par exemple, $|x|$ se tape `\lvert x \rvert`. Les seuls délimiteurs à n'être ni ouvrant ni fermant mais médian sont le slash `/`, l'anti-slash `\` et le `\middle`.

Pour avoir des délimiteurs qui sont de la même taille que ce qu'ils entourent, il faut précéder le délimiteur ouvrant par `\left` et le délimiteur fermant par `\right`:

```
$$
(1+\frac{1}{n})^n
$$
```

donne

$$\left(1 + \frac{1}{n}\right)^n$$

```
$$
\left(1+\frac{1}{n}\right)^n
$$
```

donne

$$\left(1 + \frac{1}{n}\right)^n$$

Voici un exemple d'utilisation d'un délimiteur vide:

```
$$
\left.\frac{\partial P}{\partial T}\right|_V=
\frac{nR}{V}
$$
```

donne

$$\left.\frac{\partial P}{\partial T}\right|_V = \frac{nR}{V}$$

On peut aussi utiliser `\middle` pour mettre un délimiteur médian au milieu d'un couple `\left` et `\right`, comme dans la définition d'un ensemble:

```

$$
\left\{ x \middle| x^2 < \frac{1}{2} \right\}
$$

```

donne

$$\left\{ x \middle| x^2 < \frac{1}{2} \right\}$$

On note qu'ici la barre verticale est trop proche de ce qu'il y a autour d'elle et il faut rajouter à la main deux petits espaces avec la commande `\,` :

```

$$
\left\{ x \,, \middle| \,, x^2 < \frac{1}{2} \right\}
$$

```

donne

$$\left\{ x \middle| x^2 < \frac{1}{2} \right\}$$

4.12 Alphabets mathématiques

Voici un résumé des alphabets mathématiques disponibles:

Résultat	Commande	Résultat	Commande
\mathbb{C}	<code>\mathbb{C}</code>	x	<code>x</code>
\mathbb{C}	<code>\mathrm{C}</code>	x	<code>\mathrm{x}</code>
\mathcal{C}	<code>\mathcal{C}</code>	\mathbf{x}	<code>\mathbf{x}</code>
\mathscr{C}	<code>\mathscr{C}</code>	$\boldsymbol{\sigma}$	<code>\boldsymbol{\sigma}</code>
\mathfrak{C}	<code>\mathfrak{C}</code>	PGL	<code>\mathit{PGL}</code>

Ne pas confondre `\mathrm` et `\text` : le premier est pour mettre des maths en romain tandis que le second est pour insérer du texte dans une formule.

4.13 Tableaux mathématiques

Pour composer des tableaux mathématiques on utilise l'environnement `{array}`.

Cet environnement prend un argument obligatoire qui est la spécification du type des colonnes.

Les types de colonnes de base sont `c` (centré), `l` (aligné à gauche) et `r` (aligné à droite).

Le texte des différentes colonnes est séparé par un `&` et on passe à la ligne suivante en utilisant `\\`.

Pour rajouter un trait vertical entre deux colonnes, il suffit de mettre un `|` (qu'on obtient avec la combinaison de touches "AltGr+6") dans la spécification des colonnes.

Pour tracer un trait horizontal, il y a la commande `\hline`. Cette commande doit être soit au tout début du tableau soit après un `\\`.

```

$$
\begin{array}{|c|c|}
\hline
f(x) & f'(x) \\
\hline
x^n & nx^{n-1} \\
\hline
\ln(x) & \frac{1}{x} \\
\hline
\end{array}
$$

```

donne

$f(x)$	$f'(x)$
x^n	nx^{n-1}
$\ln(x)$	$\frac{1}{x}$

4.14 Distinction de cas

Pour définir une fonction par morceaux on peut utiliser l'environnement `{cases}`, qui s'utilise de la façon suivante:

```

$$
H(x) =
\begin{cases}
0 & \text{si } x < 0, \\
1 & \text{si } x \geq 0.
\end{cases}
$$

```

donne

$$H(x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 & \text{si } x \geq 0. \end{cases}$$

4.15 Matrices

On tape les matrices comme des tableaux, sauf qu'on utilise les environnements `{matrix}`, `{pmatrix}`, `{bmatrix}`, `{Bmatrix}`, `{vmatrix}`, `{Vmatrix}` ou `{smallmatrix}`:

```

$$
\begin{matrix} a & b \\ c & d \end{matrix}
\quad
\begin{pmatrix} a & b \\ c & d \end{pmatrix}
\quad
\begin{bmatrix} a & b \\ c & d \end{bmatrix}
\quad
\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}
\quad
\begin{vmatrix} a & b \\ c & d \end{vmatrix}
\quad
\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}
\quad
\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}
\quad
\end{pre}

```

donne

$$\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

4.16 Alignement de formules

Pour aligner plusieurs formules, on dispose des environnements `{align}` et `{multline}` (il n'est pas nécessaire d'utiliser les `$$...$$`). Ne jamais mettre de `\\` sur la dernière ligne de ces environnements.

- L'environnement `{align}` permet d'aligner plusieurs signes d'égalité à l'intérieur d'une même formule:

```

\begin{align}
A &= B \\
&= C+D \\
&= E
\end{align}

```

donne

$$\begin{aligned}
 A &= B \\
 &= C + D \\
 &= E
 \end{aligned}$$

On peut au besoin mettre plusieurs équations en colonne:

```

\begin{align}
A &= B & E &= F & I &= J \\
&= C+C & &= G & &= K \\
&= D & & & &
\end{align}

```

donne

$$\begin{array}{lll}
 A = B & E = F & I = J \\
 = C + C & = G & = K \\
 = D & &
 \end{array}$$

- L'environnement `{multline}` permet de découper une formule trop grande en plusieurs morceaux sans contrôle précis de l'alignement:

```

\begin{multline}
A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J
\end{multline}

```

donne

$$\begin{array}{c}
 A + B + C + D + E + F + G + H + I + J \\
 + A + B + C + D + E + F + G + H + I + J \\
 + A + B + C + D + E + F + G + H + I + J \\
 + A + B + C + D + E + F + G + H + I + J
 \end{array}$$

Le package `amsmath` de \LaTeX définit d'autres environnements, vous pouvez les utiliser (après avoir lu la documentation du package et compris dans quels cas il faut utiliser un environnement plutôt qu'un autre). En revanche, l'utilisation de l'environnement `{eqnarray}` pour mettre en forme une formule sur plusieurs lignes, qu'on peut trouver dans certains vieux manuels, sera considéré comme une faute grave. Cet environnement a plusieurs défauts bien illustrés dans ce document: [Avoid eqnarray!](http://tug.org/pracjourn/2006-4/madsen/madsen.pdf) (<http://tug.org/pracjourn/2006-4/madsen/madsen.pdf>).

4.17 Flèches extensibles

On dispose de deux flèches extensibles, `\xrightarrow[]{}{}` et `\xrightarrow[]{}{}`. Pour mettre une formule dessus, on utilise l'argument obligatoire entre accolades tandis que pour mettre une formule dessous, on utilise l'argument optionnel entre crochets:

```

$$
f(x) \xrightarrow{\text{d'après } (H)} a
\quad
f(x) \xrightarrow[x \rightarrow 0]{} a
\quad
f(x) \xrightarrow[x \rightarrow 0]{\text{d'après } (H)} a
$$

```

donne

$$f(x) \xrightarrow{\text{d'après } (H)} a \quad f(x) \xrightarrow[x \rightarrow 0]{} a \quad f(x) \xrightarrow[x \rightarrow 0]{\text{d'après } (H)} a$$

4.18 Modules de congruences

Selon l'apparence voulue, il y a trois façon d'écrire les modules de congruence:

```
$a \equiv b \pmod m$
$a \equiv b \pmod m$
$a \equiv b \pmod m$
```

donnent

```
 $a \equiv b \pmod m$ 
 $a \equiv b \pmod m$ 
 $a \equiv b \pmod m$ 
```

Il y a aussi la commande `\bmod` qu'on peut utiliser dans le contexte suivant

```
 $\gcd(n, m \bmod n)$  donne  $\gcd(n, m \bmod n)$ 
```

4.19 Placer au-dessus ou en-dessous

Les commandes `\underset{en-dessous}{symbole}` et `\overset{au-dessus}{symbole}` permettent de placer du matériel arbitraire en-dessous ou au-dessus de n'importe quel symbole.

```
$$
A
\overset{\text{déf}}{=}
\frac{1}{n} \underset{n \to +\infty}{\in} o\left(\frac{1}{n^2}\right)
$$
```

donne

$$A \stackrel{\text{déf}}{=} \frac{1}{n} \underset{n \rightarrow +\infty}{\in} o\left(\frac{1}{n^2}\right)$$

5 Mode édition : les cellules

Code

Lorsqu'on ouvre un terminal, un programme interpréteur de commandes système est lancé et attend les commandes pour les exécuter. De la même façon, en arrière-plan d'un notebook, un interpréteur de code a été lancé et attend lui aussi qu'on lui donne du code. Dans le coin supérieur droit vous pouvez voir qu'il s'agit ici de l'interpréteur Python 3.

Un notebook est constitué de cellules successives. Les cellules pour écrire du code sont repérables par le prompt `In[]`. Essayons :

In [6]:

```
2 / 3
```

Out[6]:

```
0.6666666666666666
```

Le texte `2 / 3` a été transmis à l'interpréteur Python 3. Il s'agit d'une expression Python 3 valide. L'interpréteur a donc pu l'évaluer. La valeur de l'expression (dont le type est ici un nombre flottant) est alors récupérée et représentée sous la forme du texte affiché à droite du prompt de sortie

N'oubliez pas que vous êtes dans un notebook. Vous pouvez donc modifier l'expression ci-dessus et la ré-exécuter en appuyant sur `[Shift-ENTER]`. Le numéro entre crochet est un compteur. Il est incrémenté à chaque sollicitation de l'interpréteur.

In [7]:

```
h = 2+2
```

L'exécution de l'instruction ci-dessus n'a produit aucun résultat en sortie. Cette instruction a cependant eu un effet. Elle a modifié l'état de l'interpréteur. En interne, l'interpréteur a associé la valeur de l'expression `2+2` (c'est-à-dire le type entier `4`) au nom `hauteur`.

On peut alors exécuter :

In [8]:

```
h=2*h  
h
```

Out[8]:

```
8
```

Lorsque vous ouvrez un notebook vous le visualisez tel qu'il a été sauvegardé. Vous voyez en particulier les résultats des séquences de code qui ont été exécutées. Cependant, à l'ouverture du notebook, un nouvel interpréteur tout neuf est lancé. Vous devez donc exécuter à nouveau les cellules. Mais dans quel ordre? La réponse naturelle est "dans l'ordre où apparaissent les cellules"... mais cela suppose que la personne qui a réalisé le notebook a fait correctement les choses.

Pour commenter une ou plusieurs lignes, les sélectionner et appuyer sur `[Ctrl-/]` :

5.1 Les exceptions

Dans certaines situations, l'interpréteur peut s'interrompre en levant une exception. Les exceptions n'ont rien d'exceptionnelles. On en rencontre souvent, en particulier lorsque l'on commet des erreurs de programmation.

In [9]:

```
T = [18, 5, 3]
# T[3] # Décommenter la ligne pour voir l'exception
```

Pour comprendre ce qui a produit l'exception il faut :

- identifier le nom de l'exception, ici `IndexError`,
- lire le message associé, ici `'list index of range'`,
- identifier l'instruction qui l'a provoqué, ici l'évaluation de `T[3]` à la ligne 2.

Il se peut aussi que le code que l'on exécute ne se termine pas :

In [10]:

```
k = 1
# while k > 0: # Décommenter pour voir l'exception
#     k = k + 1
```

Lorsqu'une cellule de code s'exécute le prompt comporte une étoile `In[*]`. Pour interrompre l'interpréteur, il suffit d'appuyer deux fois sur la touche `I`. Si cela s'avère nécessaire, il est également possible de redémarrer l'interpréteur. Pour cela il faut appuyer deux fois sur la touche `0`. L'interpréteur se retrouve alors dans son état initial.

5.2 Entrée et sortie standard

Les programmes, quels qu'ils soient, lisent des données en provenance de sources variées (fichiers, réseaux, ports usb, etc). Ils envoient eux-mêmes des données vers diverses destinations. Dans tous les cas, cela se traduit par la lecture ou l'écriture de flux d'octets.

Dans un programme Python la fonction `print` permet d'écrire sur la sortie standard.

In [11]:

```
print("Accueil s'écrit :")
for x in "accueil":
    print(x)
```

Accueil s'écrit :

a
c
c
u
e
i
l

Remarquez que l'exécution du code ci-dessus n'a retourné aucune valeur. **Il n'y a pas de prompt Out []** . Remarquez aussi que chaque fonction print a également envoyé un caractère saut de ligne sur la sortie standard.

5.3 L'aide en ligne IPython

- *L'auto-complétion* Lorsque vous commencez à saisir un nom connu de l'interpréteur, l'appui sur la touche [TAB] complète le nom automatiquement. Si plusieurs noms sont possibles un menu contextuel vous propose de choisir. Ceci économise de la frappe tout en évitant les erreurs d'orthographe dans les noms des variables.
- *Les infobulles* Lorsque le curseur de saisie est sur un nom connu de l'interpréteur (ou immédiatement à droite), l'appui sur [Shift-TAB] affiche une infobulle qui donne un descriptif succinct de l'objet désigné par ce nom. C'est un moyen rapide de vérifier par exemple quels sont les arguments qui sont attendus par une fonction.
- *Documentation en ligne* Pour lire la documentation en ligne concernant un objet python (module, fonction, classe, etc), il suffit d'ajouter un ? juste après le nom et d'appuyer sur la touche [Enter] . Un pager s'ouvre alors avec la dite documentation.

In [12]:

```
?abs
```

5.4 Les commandes systèmes

Lorsque le texte de la cellule débute par un ! alors IPython en déduit que le reste du texte n'est pas du code python mais une commande système qui doit être exécutée par le Shell. Autrement dit IPython peut se substituer au terminal.

In [13]:

```
# Liste le contenu du dossier courant
!ls -l
```

```
total 10316
-rw-rw-r-- 1 minnolina minnolina 36617 févr. 18
10:39 chevre1.png
-rw-rw-r-- 1 minnolina minnolina 29100 févr. 18
10:38 chevre2.png
-rw-rw-r-- 1 minnolina minnolina 32188 févr. 18
10:38 chevre3.png
-rw-rw-r-- 1 minnolina minnolina 3034 mars 20
17:20 custom.css
-rw-rw-r-- 1 minnolina minnolina 32205 juil. 28
2015 erreurEuler.png
-rw-rw-r-- 1 minnolina minnolina 36739 févr. 18
11:14 fraise.png
drwxrwxr-x 2 minnolina minnolina 4096 mars 21
10:22 Images
-rw-rw-r-- 1 minnolina minnolina 118694 mars 22
10:59 M62-CM1-M.ipynb
-rw----- 1 minnolina minnolina 1122526 mars 13
10:14 M62-CM1.pdf
-rw-rw-r-- 1 minnolina minnolina 1164211 mars 22
11:02 M62-CM1-P.ipynb
-rw-rw-r-- 1 minnolina minnolina 647241 mars 22
10:25 M62-CM2.ipynb
-rw----- 1 minnolina minnolina 459926 mars 13
10:16 M62-CM2.pdf
-rw-rw-r-- 1 minnolina minnolina 83597 mars 25
09:57 M62-CM3.ipynb
-rw----- 1 minnolina minnolina 256441 mars 13
10:32 M62-CM3.pdf
-rw-rw-r-- 1 minnolina minnolina 35195 mars 25
09:57 M62-CM4.ipynb
-rw----- 1 minnolina minnolina 196378 mars 13
11:36 M62-CM4.pdf
-rw-rw-r-- 1 minnolina minnolina 642639 mars 25
09:37 M62-CM5.ipynb
-rw----- 1 minnolina minnolina 721739 mars 17
17:02 M62-CM5.pdf
-rw-rw-r-- 1 minnolina minnolina 209083 mars 25
09:54 M62-CM6-7.ipynb
-rw----- 1 minnolina minnolina 369125 mars 19
17:50 M62-CM6-7.pdf
-rw-rw-r-- 1 minnolina minnolina 961199 mars 25
21:19 M62-CM8.ipynb
-rw-rw-r-- 1 minnolina minnolina 1860 mars 25
20:56 M62-CM9.ipynb
-rw-rw-r-- 1 minnolina minnolina 73541 mars 25
21:06 M62-T0-D0.ipynb
-rw-rw-r-- 1 minnolina minnolina 238677 mars 25
21:04 M62-TP1.ipynb
-rw----- 1 minnolina minnolina 381758 mars 11
13:28 M62-TP1.pdf
-rw-rw-r-- 1 minnolina minnolina 82103 mars 21
10:30 M62-TP2.ipynb
-rw----- 1 minnolina minnolina 305266 mars 11
13:33 M62-TP2.pdf
```

```

-rw-rw-r-- 1 minnolina minnolina 271168 mars 25
09:58 M62-TP3.ipynb
-rw----- 1 minnolina minnolina 378306 mars 13
17:34 M62-TP3.pdf
-rw-rw-r-- 1 minnolina minnolina 220779 mars 21
10:30 M62-TP4.ipynb
-rw----- 1 minnolina minnolina 360561 mars 19
16:38 M62-TP4.pdf
-rw-rw-r-- 1 minnolina minnolina 185728 mars 25
21:13 M62-TP5.ipynb
-rw----- 1 minnolina minnolina 347221 mars 19
16:45 M62-TP5.pdf
-rw-rw-r-- 1 minnolina minnolina 30543 mars 25
21:07 M62-TP-Examen.ipynb
-rw-rw-r-- 1 minnolina minnolina 829 mars 20
11:42 pendule.pgf
-rw-rw-r-- 1 minnolina minnolina 34371 mars 20
11:42 pendule.png
drwxrwxr-x 2 minnolina minnolina 4096 févr. 18
14:49 __pycache__
-rw-rw-r-- 1 minnolina minnolina 12954 avril 23
2018 tikzmagic.py
-rw-rw-r-- 1 minnolina minnolina 1509 mars 4
13:31 Untitled1.ipynb
-rw-rw-r-- 1 minnolina minnolina 12332 mars 7
11:44 Untitled2.ipynb
-rw-rw-r-- 1 minnolina minnolina 1922 mars 13
16:13 Untitled3.ipynb
-rw-rw-r-- 1 minnolina minnolina 192583 févr. 28
13:10 Untitled.ipynb

```

5.5 Les commandes magiques de IPython

5.5.1 `%matplotlib inline` & `%matplotlib notebook`

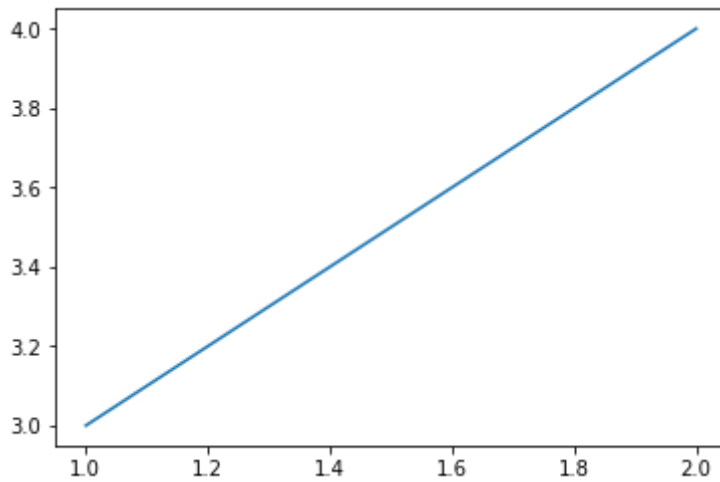
Si la commande magique `%matplotlib` est appelée sans argument, la sortie d'une commande de traçage est affichée à l'aide du backend `matplotlib` par défaut dans une fenêtre séparée.

Alternativement, le backend peut être explicitement demandé en utilisant, par exemple:

- `%matplotlib inline` : la sortie est affichée directement sous la cellule de code qui l'a produite,
- `%matplotlib notebook` : la sortie est affichée directement sous la cellule de code qui l'a produite et, de plus, permet des figures interactives.

In [14]:

```
%matplotlib inline
from matplotlib.pyplot import *
plot([1,2],[3,4]);
```



5.5.2 %autosave 300

Sauvegarde automatiquement le notebook tous les 300 secondes (=5 minutes)

5.5.3 %who , %whos et %reset

%who , %whos affichent les objets qui sont actuellement définis dans l'interpréteur Python.

%reset efface les variables définies; %reset -f efface les variables définies sans demander confirmation.

In [15]:

```
%who
```

```
ALLOW_THREADS      Annotation      Arrow      Artis
t      AutoLocator      Axes      AxisError
      BUFSIZE      Button
CLIP      Circle      ComplexWarning      DAILY      DataS
ource      DateFormatter      DateLocator      DayLo
cator      ERR_CALL
ERR_DEFAULT      ERR_IGNORE      ERR_LOG
      ERR_PRINT      ERR_RAISE      ERR_W
ARN      FLOATING_POINT_SUPPORT      FPE_DIVIDEBYZ
ERO      FPE_INVALID
FPE_OVERFLOW      FPE_UNDERFLOW      FR      False
      Figure      FigureCanvasBase      Fixed
Formatter      FixedLocator      FormatStrForm
atter
Formatter      FuncFormatter      GridSpec
      HOURLY      HTML      HourLocator      Image
      IndexDateFormatter      IndexLocator

Inf      Infinity      LinAlgError      Line2
n      LinearLocator      Locator      Locator
```

In [16]:

```
%reset -f
```

In [17]:

```
%who
```

Interactive namespace is empty.

6 Références

- https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html (https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html)
- Ce qu'on peut écrire en *Markdown* et en *LaTeX* dans les notebooks [Jupyter \(jupyter.org\)](http://jupyter.org) et ce qu'on peut faire dans les cellules de code dans cette [série de tutoriels \(https://www.youtube.com/playlist?list=PLRJx8WOUx5XcDMOxSQegCJUjTJePTIF9Z\)](https://www.youtube.com/playlist?list=PLRJx8WOUx5XcDMOxSQegCJUjTJePTIF9Z)

