

COURS:

Attention sur une ligne de programme on ne peut pas commencer où l'on veut.

(ce qui est sur une ligne n'est pas flottant)

Si la ligne précédente se termine par : le retour ne se fait pas au même niveau.

En cas de doute mettez le curseur à droite de : puis appuyer sur la touche Entrée.

Le curseur se mettra là où il faut écrire.

Thème :

- `for i in range (1,n+1):` veut dire << pour tout entier i allant de 1 à n >>

(Voir feuille précédente)

Attention: i s'arrête à (n+1)- 1

- `b=int(input("Entrez un entier : "))` Cela affiche Entrez un entier
puis attend la saisie de cet entier
si le nombre saisi n'est pas un entier
b sera sa partie entière en raison de `int`

- *if* Condition:

.....

Si :

sous entendu alors

(le *alors* c-à-d *then* ne s'écrit pas en Python 2.7,

le : le remplace)

(Dans le cas de plusieurs *si*, on met *elif* pour chaque
si supplémentaire)

Une Condition comportant une égalité se met avec ==

Le symbole = est donc doublé

•

•

L'instruction qui suit est décalée (indentation)

le double astérix signifie ^

- * un seul astéris mis entre deux réels est la croix \times de la multiplication

$$2^{**}3 = 2 \times 2 \times 2 = 8$$

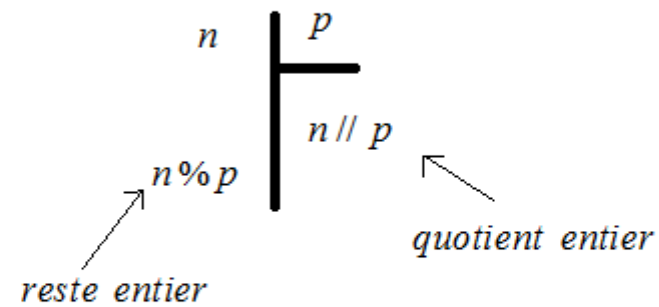
$$5^{**}(2^{**}3)=5^{2^3}$$

$$5^{2^3} = 5^{2 \times 2 \times 2} = 5^8 = 390625$$

- $n\%p$ *donne le reste (entier) de la division de n par p*

Notations en Python2.7: Soit p est un entier naturel non nul .

Soit n un entier naturel.



$n \% p = 0$ signifie que p est un diviseur de n

Par exemple : $7 // 3$ donne 2

$7 \% 3$ donne 1

$$7 = 3 \times 7 // 3 + 7 \% 3$$

c-à-d

$$7 = 3 \times 2 + 1$$

COURS:

Thème:

- **from random import *** *se met au dessus de* **def nomprogramme():**
pour faire appel au module random
s'il est installé
- **while Condition:** **Tant que Condition :**
.....
- **if Condition:** **Si Condition:**
.....
- elif Condition:** **Si Condition:**

- else:** **Sinon:**

- `floor` (*valeur*)

Donne la partie entière de la valeur

c-à-d "le plus grand entier relatif inférieur ou égal à la valeur"

En maths c'est $E(\text{valeur})$

Cela nécessite de faire appel au module `math`

(Donc avant il faut mettre `from math import*`)

Par exemple : `from math import*`

`floor(5.4) = 5`

`floor(- 3.4) = - 4`

Remarque: On peut aussi utiliser plus facilement

pour les réels positifs

`int(valeur positive)`

Pour les nombres réels positifs cela donne leur partie entière.

Mais attention cela *ne marche pas* pour les nombres réels négatifs.

(En fait cela donne la partie non décimale.)

- `randint(0,100)` Cela donne un entier entre 0 et 100 au hasard
décidé par l'ordinateur
(Cela nécessite avant le menu `from random import*`)
- `!=` Pour dire \neq
Par exemple: `6!=5`
veut dire $6 \neq 5$
- `float(a)` transforme *a* en un décimal
Par exemple: `float(5)` donne 5.0
Si *a* est déjà un nombre décimal cela ne fait rien
Par exemple: `float(5.)` donne 5.0
`float(a)/b` permet d'avoir *a* / *b* comme
nombre décimal .
Par exemple: `5 / 2` donne 2
`5.0 / 2` donne 2.5
`5 / 2.` donne 2.5
`float(5)/2` donne 2.5

`float(5/2) = float(2) = 2.0`

Si a et b ne sont pas déjà des nombres décimaux

*a. / b comme a /b. donne aussi le résultat de la division
de a par b sous forme décimale*

Parfois il faut avant faire appel au module decimal

Par exemple

`float(11)/2` donne 5.5

Cela revient à 11.0 /2

L'ordinateur donne un décimal car 11.0 l'est

Cela peut parfois avant nécessiter d'écrire:

`from decimal import*`

- `random()` donne un réel au hasard de l'intervalle [0;1[

Cela necessite avant `from decimal import`*

COURS:

Thème :

- | | |
|--------------------------|---------------------------------------|
| if Condition1: | <i>veut dire</i> Si Condition1 |
| <i>..Conséquence1...</i> | <i>alors</i> Conséquence1 |
| elif Condition2: | <i>veut dire</i> Si Condition2 |
| <i>..Conséquence2</i> | <i>alors</i> Conséquence2 |
| else: | <i>veut dire</i> Sinon |
| <i>..Conséquence3</i> | <i>alors</i> Conséquence3 |

(On peut utiliser plusieurs fois **elif** :)
- *float* (x) *Pour une écriture décimale de x*
Cela nécessite un module complémentaire
- *raw_input*(" texte ") *Affiche le texte et attend la saisie d'une chaîne*

