

www.freemaths.fr

TLE

# Technologique Mathématiques

(STI2D & STL)

Algorithmes  $\exp(x)$



**CORRIGÉ** DE L'EXERCICE

# L'étrange nombre $e$

## Correction

1. Recopions la fonction *factorielle* et utilisons la pour calculer le nombre  $6!$ :

On recopie la fonction dans un éditeur et on écrit dans la console l'instruction :

```
>>> factorielle(6)
720
```

On obtient :  $6! = 720$

2. Complétons le tableau d'état des variables Python :

La fonction contient une variable *for*. La variable  $i$  varie de 1 à  $n$  par définition du range.

Donc pour  $n = 7$ , la variable  $i$  varie de 1 à 7.

La variable  $S$  est initialisée à 1.

Pour  $i = 1$ , la fonction calcule le terme  $\frac{1}{i!}$  et le rajoute à la variable  $S$  précédente.

La variable  $S$  est donc affectée de la valeur  $S = 1 + \frac{1}{1!} = 2$ .

Puis  $i$  passe à la valeur suivante, soit la valeur 2.

Pour  $i = 2$ , la fonction calcule le terme  $\frac{1}{i!}$  et le rajoute à la variable  $S$  précédente.

La variable  $S$  est donc affectée de la valeur  $S = 2 + \frac{1}{2!} = 2,5$ .

La boucle continue ainsi jusqu'à la valeur  $i = 7$ .

On peut donc compléter le tableau :

$i$	1	2	3	4	5	6	7
$S$	2.0	2.5	2.66666	2.70833	2.71666	2.71805	2.71825

3. Expliquons ce que renvoie la fonction `nbre_e` :

La fonction renvoie le terme de rang  $n$  de la suite  $(u_n)$ .

4. Utilisons cette fonction avec  $n = 1000$  et comparons le résultat avec la valeur de  $e$  donnée dans l'énoncé :

On écrit dans la console l'instruction suivante :

```
>>> nbre_e(1000)
2.7182818284590455
```

Le terme de rang 1000 de la suite est :

$$u_{1000} \approx 2,718\ 281\ 828\ 459\ 045\ 5$$

Nous constatons que la valeur approchée donnée dans l'énoncé est la même jusqu'à la 13<sup>ième</sup> décimale.