

Programmation en Python de la factorielle à l'aide de la récursivité

```
In [1]: def fact(n):  
        if n==0:  
            return 1  
        else:  
            return n*fact(n-1)
```

La fonction `fact()` d'argument `n` s'auto-appelle afin de générer le résultat. On parle de fonction récursive lorsqu'une fonction définie s'utilise elle-même dans son corps de programme.

```
In [2]: for number in range(6):  
        print(f"{number}! = {fact(number)}")
```

```
0! = 1  
1! = 1  
2! = 2  
3! = 6  
4! = 24  
5! = 120
```

Construisons une fonction `coeff_bin(k, n)` qui renverra le coefficient binomial associé à `k` succès parmi `n` répétitions d'une épreuve de Bernoulli.

```
In [3]: def coeff_bin(k, n):  
        return int(fact(n)/(fact(k)*fact(n-k)))
```

Déterminons les cinq coefficients binomiaux associés à une répétition de quatre épreuves de Bernoulli identiques, successives et indépendantes.

```
In [4]: L = []  
        for i in range(5):  
            L.append(coeff_bin(i,4))
```

```
In [5]: L
```

```
Out[5]: [1, 4, 6, 4, 1]
```

```
In [6]: print(f"La liste des coefficients binomiaux pour n = 4 est : {L}.")
```

La liste des coefficients binomiaux pour n = 4 est : [1, 4, 6, 4, 1].

Un programme plus élaboré

```
In [7]: def fact(n):
        if n==0:
            return 1
        else:
            return n*fact(n-1)

        def coeff_bin(k, n):
            return int(fact(n)/(fact(k)*fact(n-k)))

        n = int(input("Entrer le paramètre n de l'expérience de Bernoulli : "))
        L = []
        for i in range(n+1):
            L.append(coeff_bin(i,n))
        print(f"La liste des coefficients binomiaux pour n = {n} est : {L}.")
```

Entrer le paramètre n de l'expérience de Bernoulli : 8

La liste des coefficients binomiaux pour n = 8 est : [1, 8, 28, 56, 70, 56, 28, 8,1].

Elle est pas belle la vie?

```
In [ ]:
```