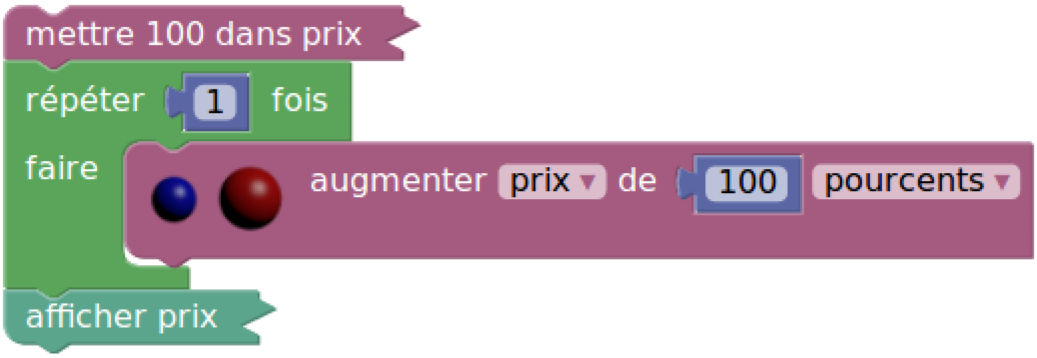
# Le nombre e

## I/ Intérêts composés

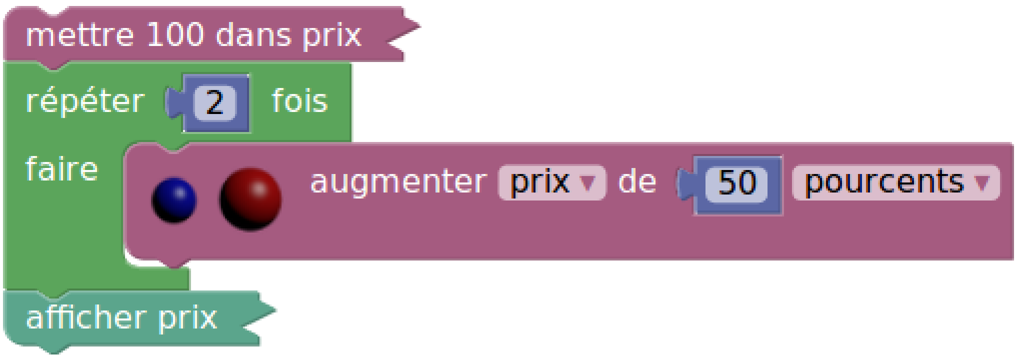
En 1685, dans la revue scientifique *Ephemeris Eruditorum Gallinacœ*, Jakob Bernoulli publiait un texte titré *Quæstiones nonnullæ de usuris, cum solutione problematis de sorte alearum, propositi* où il se posait la question du devenir d'un placement à intérêts composés si, au lieu de verser les intérêts en une seule fois à la fin de l'année, on les versait en continu. Pour illustrer le problème, on suppose qu'on place 100 € avec pour but de doubler le placement à la fin de la période. Pour faire court, la variable donnant le placement a été nommée `prix`.

### 1) En une seule fois




Doubler une variable, c'est l'augmenter de 100 %.

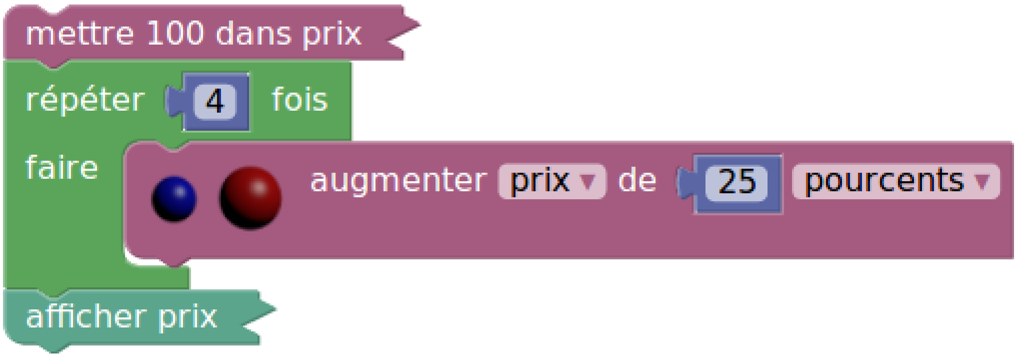
### 2) En deux fois




Le coefficient multiplicateur est 1,5 donc le `prix` a été multiplié par 1,5×1,5=2,25 : il a plus que doublé puisque l'augmentation est de 125 % (plus que 100 %).

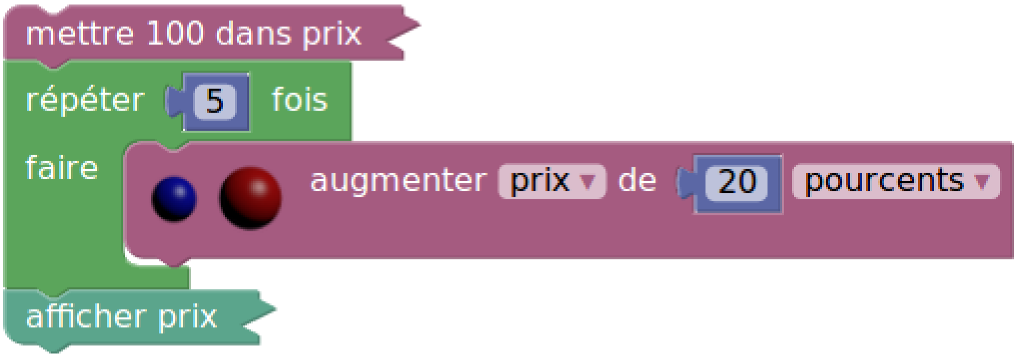
### 3) En quatre fois

On a une suite géométrique de raison 1,25 (le quart de 100 c'est 25) :



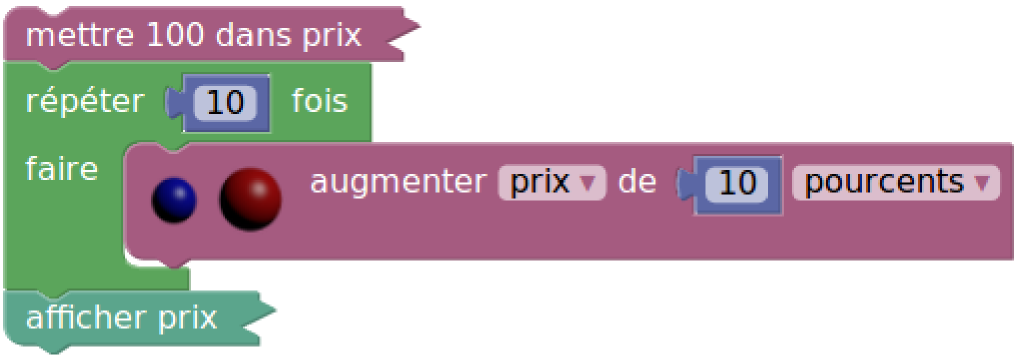

## 4) En cinq fois

La raison de la suite géométrique est 1,2 :




## 5) En dix fois

La raison de la suite géométrique est 1,1 :

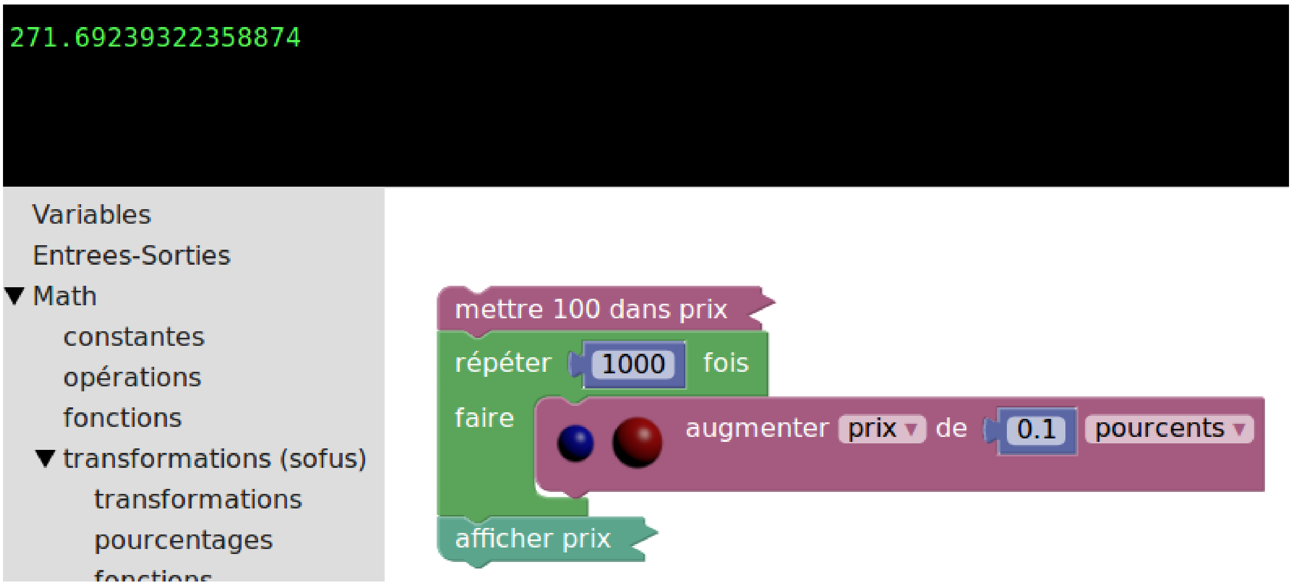



On a plus que 200 € à la fin, et même plus que 250 €. Est-il possible en divisant en plus que 10 l'augmentation de 100 €, de dépasser 300 € ? Bernoulli dit que non :

si a = b , debebitur plusquam 2 1/2 a , et minus quam 3a .

# 6) En mille fois

La raison de la suite géométrique est 1,001 mais on calcule son 1001$^{e}$ terme :




On a bien plus que 200 € à la fin, on a même dépassé les 271 €. Est-il possible en divisant encore plus les 100 % d'augmentation, de dépasser 272 € ? Pour le savoir, on va faire appel à Python. La traduction du script précédent en Python donne

```
prix = 100
for count in range(1000):
    prix = prix + prix * 0.1 / float(100 )
print(prix)
```

Mais en divisant 0,1 par 100 on a un millième :

```
prix = 100
for count in range(1000):
    prix = prix + prix * 1./1000
print(prix)
```

Le pseudo-code s'écrit

```
prix ← 100
pour n allant de 1 à 1000
    prix ← prix + prix × 1/1000
fin pour
```

## 7) En dix mille fois

```
prix = 100
for count in range(10000):
    prix = prix + prix * 1./10000
```

Mais on peut factoriser la dernière ligne :

```
prix = 100
for count in range(10000):
    prix = prix * (1 + 1./10000)
```

Après il suffit d'ajouter un zéro aux lignes 2 et 4 du script pour passer à 100000 itérations :

```
prix = 100
for count in range(100000):
    prix = prix * (1 + 1./100000)
```

La raison de la suite géométrique est alors $1+1/100000=1+10^{-5}=1{,}00001$.

## 8) En des millions de fois

Un million, c'est $10^6$ (en Python, `10**6`). On trouve le résultat pour k=6 dans

```
for k in range(1,10):
     n = 10**k
     x = (1+1./n)**n
     print(k,x)
```

qui va jusqu'à $10^9$ et même là, on n'a pas dépassé 272 €.

En fait on a une suite géométrique de raison $1+1/n$ et on s'intéresse au $n^e$ terme qui est $100\times(1+1/n)^n$. Il s'agit là d'une suite et cette suite semble converger vers environ 2,718. Bernoulli a découvert un nouveau nombre auquel il s'agit de donner un nom.

## II/ Le nom du nombre

En 1713, Rémont de Montmort effectue une analyse mathématique de jeux de hasard dans son livre *Essay d'analyse sur les jeux de hazard*. Parmi ces jeux il y en a un appelé *jeu de treize* où une probabilité fait intervenir le nombre découvert par Bernoulli. L'analyse de ce jeu est reprise en 1741 par Leonhard Euler dans cet article :

CALCUL DE LA PROBABILITÉ

DANS LE JEU DE RENCONTRE,

PAR M. EULER.

(on remarque au passage que le jeu de treize est devenu jeu de rencontre). Euler démontre que la probabilité de gain d'un des joueurs au jeu de rencontre est proche de l'inverse du nombre découvert par Bernoulli :

posant $e$ pour marquer le nombre, dont le logarithme est $= 1$, on fait que $\frac{1}{e}$ exprime cette dernière serie.

Euler a donné un nom au nombre découvert par Bernoulli, il l'a appelé e.

### 1) Le logarithme d'Euler

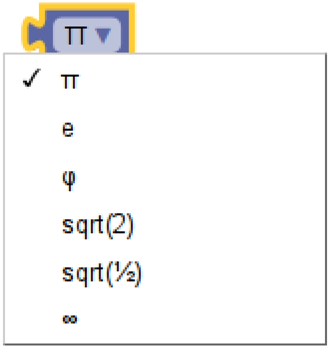
Le logarithme auquel Euler fait allusion est celui découvert par Grégoire de Saint-Vincent[1]. On l'obtient avec la touche `ln` de la calculatrice. Comme c'est un logarithme, on a

$\ln(1+1/n)^n = n \times \ln(1+1/n)$. Mais une propriété de ce logarithme est que $\ln(1+dx)=dx$ (avec la notation de Leibniz). Alors $\ln(1+1/n)$ est proche de $1/n$ et $\ln(1+1/n)^n$ a la même limite que $n\times 1/n=1$. Euler parle donc bien du même nombre que Bernoulli puisque le logarithme de ce nombre est 1.

### 2) En Python

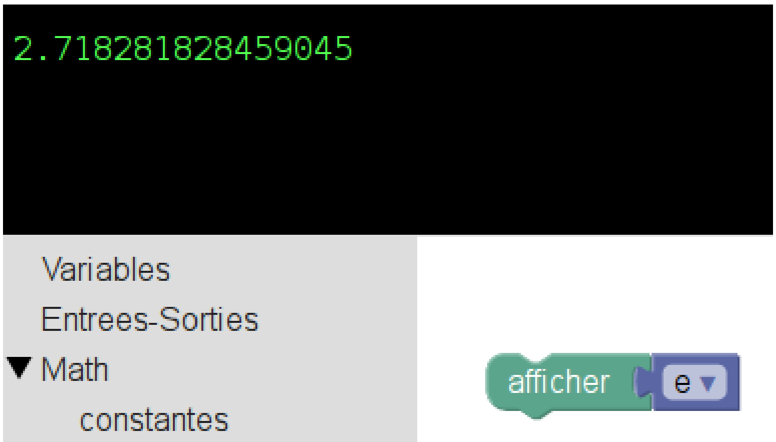
Le nombre e est si important qu'il existe dans SofusPy974, comme constante mathématique, juste en-dessous de π :

1 Et appelé aujourd'hui ***logarithme népérien*** bien que ce ne soit pas Neper qui l'ait inventé.



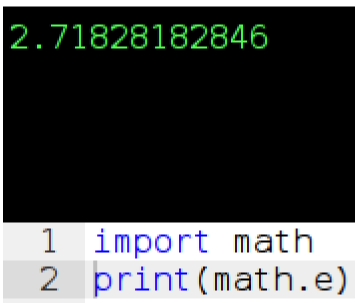

(on remarque que l'infini est également une constante qui peut intervenir dans des calculs).

Et on peut afficher cette constante :




En cliquant sur


on peut voir la version Python du nombre e :




En fait e est une constante (nommée `e` en Python) qui se trouve dans le module `math`. On peut donc aussi l'avoir par import direct depuis le module `math` :

```
from math import e

print(e)
```

## 3) Sur calculatrices Ti

La séquence de touches 2nde ÷ donne aussi le nombre e (écrit en petit au-dessus de la touche ÷).
Pour vérifier le définition d'Euler on peut donc faire ln 2nde ÷ entrer qui donne bien 1 à l'affichage.

## 4) Calcul par Euler

Dans son article de 1741, Euler écrit

$$e = 2,718281828459045235360.$$