In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import warnings
from scipy import special
import math
%matplotlib inline
warnings.simplefilter('ignore')
```

Тема 1 “Введение в математических анализ”

1. Как относятся друг к другу множество и последовательность? (в ответе использовать слова типа: часть, целое, общее, частное, родитель, дочерний субъект и т.д.)

Множество - это набор неких объектов обладающих общим свойством. Например, множество комплексных чисел, множество жителей страны и т.п.

Последовательность - это пронумерованный набор объектов, причем порядок имеет значение. Фактически, это функция одной переменной.

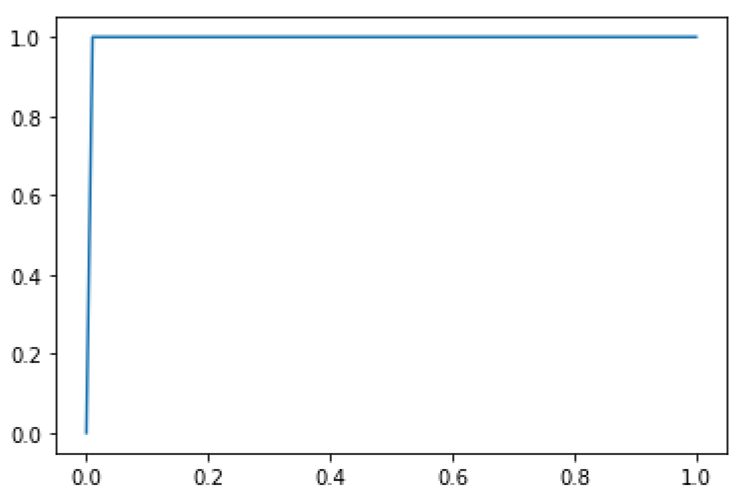
Можно считать множество абстракцией более высокого уровня (родителем), потому как любую последовательность можно считать множеством, но не любое множество можно назвать последовательностью

1. Прочитать высказывания математической логики, построить их отрицания и установить истинность.

- $\forall y \in [0;1] : sgn(y) = 1$

Для каждого y принадлежащего отрезку [0; 1] значение функции sgn(y) равно 1

In [33]:
```
x = np.linspace(0, 1, 100)
y = np.sign(x)
plt.plot(x, y);
```



Высказывание ложно, так как

$$sgn(y) = \begin{cases} 1, & \text{x} > 0 \\ 0, & \text{x} = 0 \\ -1, & \text{x} < 0 \end{cases}$$

а следовательно при $y = 0 : sgn(y) \neq 1$

- $\forall n \in \mathbb{N} > 2 : \exists x, y, z \in \mathbb{N} : x^n = y^n + z^n$

Для каждого n принадлежащего множеству натуральных чисел больше 2 существуют такие x, y, z принадлежащие множеству натуральных чисел, при которых $x^n = y^n + z^n$

Выражение истинно

- $\forall x \in \mathbb{R} \exists X \in \mathbb{R} : X > x$

Для каждого x принадлежащего множеству рациональных чисел, существует X, также принадлежащий множеству рациональных чисел, который больше x

Высказывание истинно, так как для любого числа из множества рациональных чисел можно найти большее число

- $\forall x \in \mathbb{C} \nexists y \in \mathbb{C} : x > y || x < y$

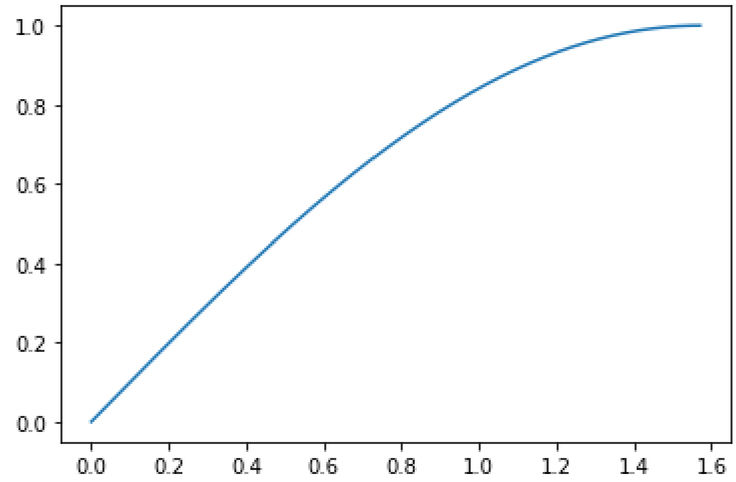
Для каждого x принадлежащего множеству комплексных чисел не существует y, принадлежащий этому же множеству, которое или больше x или меньше

Выражение ложно, так как для любого числа в этом множестве можно найти число большее, меньшее или равное другому

- $\forall y \in [0; \frac{\pi}{2}] \exists \epsilon > 0 : \sin y < \sin(y + \epsilon)$

Для каждого y принадлежащего отрезку $[0; \frac{\pi}{2}]$ существует ϵ больше нуля, при котором $\sin(y)$ будет меньше $\sin(y + \epsilon)$

In [28]:
```
x = np.linspace(0, np.pi / 2, 1000)
y = np.sin(x)
plt.plot(x, y);
```

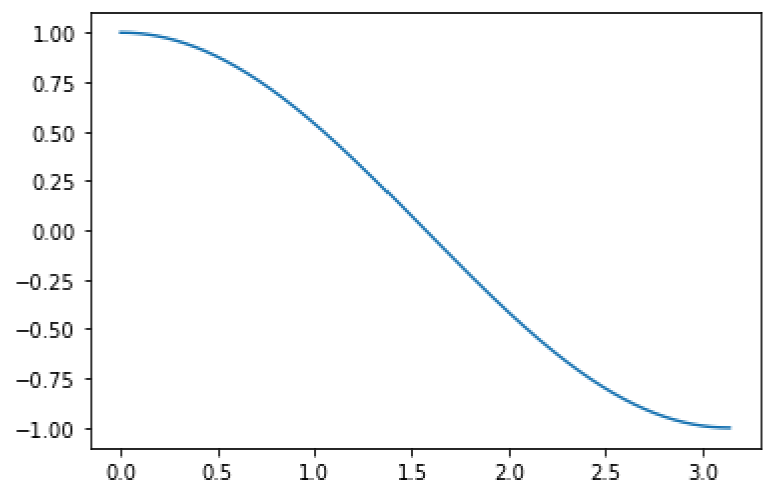


Выражение ложно, так как при значении $y = \frac{\pi}{2}, \sin y = 1$, а это максимальное значение функции

- $\forall y \in [0; \pi] \exists \epsilon > 0 : \cos y > \cos(y + \epsilon)$

Для каждого y принадлежащего отрезку $[0; \pi]$ существует ϵ больше нуля, при котором $\cos(y)$ будет больше $\cos(y + \epsilon)$

In [27]:
```
x = np.linspace(0, np.pi, 1000)
y = np.cos(x)
plt.plot(x, y);
```



Выражение ложно, так как при $y = \pi, \cos y = -1$, а это минимальное значение функции

- $\exists x : x \notin \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$

Существует x не входящий ни в одно числовое множество

Выражение ложно, так как не существует число (считаем x числом, иначе выражение истинно) не входящее ни в одно из перечисленных множеств

Тема 2 “Множество”

1. Даны три множества A, B и C. Необходимо выполнить все изученные виды бинарных операций над всеми комбинациями множеств.

$$\{A, B, C\}$$

- конъюнкция
  $A \wedge B$
  $A \wedge C$
  $B \wedge C$
  $A \wedge B \wedge C$

- дизъюнкция
  $A \vee B$
  $A \vee C$
  $B \vee C$
  $A \vee B \vee C$

- инверсия
  $\neg A, \neg B, \neg C$

- импликация
  $A \Rightarrow B$
  $A \Rightarrow C$
  $B \Rightarrow A$
  $B \Rightarrow C$
  $C \Rightarrow A$
  $C \Rightarrow B$

- эквивалентность
  $A \iff B$
  $A \iff C$
  $B \iff C$

1. Выполнить задание 1 на языке Python

In [57]:
```
A = set([1, 2, 3])
B = set([1, 2, 3])
C = set([6, 7, 8])
a = True
b = False
c = False
```

In [48]:
```
# Конъюнкция
A & B, A & C, B & C, A & B & C
```

Out[48]:
```
({1, 2, 3}, set(), set(), set())
```

In [49]:
```
# Дизъюнкция
A | B, A | C, B | C, A | B | C
```

Out[49]: ({1, 2, 3}, {1, 2, 3, 6, 7, 8}, {1, 2, 3, 6, 7, 8}, {1, 2, 3, 6, 7, 8})

In [54]:
```
# Эквивалентность
A == B, A == C, B == C
```

Out[54]: (True, False, False)

In [61]:
```
# отрицание
not(a), not(b), not(c)
```

Out[61]: (False, True, True)

In [66]:
```
# Импликация
not(a) or b, not(a) or c, not(b) or a, not(b) or c, not (c) or a, not(c) or b
```

Out[66]: (False, False, True, True, True, True)

Тема 3 “Последовательность”

1. Даны 4 последовательности. Необходимо:

- a. исследовать их на монотонность;
- b. исследовать на ограниченность;
- c. найти пятый по счету член.

Сначала определим, какая последовательность является монотонной.

Монотонной считается последовательность, элементы которой с увеличением номера не убывают, или, наоборот, не возрастают. То есть, последовательность $\{x_n\} \in \mathbb{N}$ является монотонной, если удовлетворяет одному из следующих условий:

- $\forall n \in \mathbb{N} : x_n \leq x_{n+1}$ - неубывающая последовательность
- $\forall n \in \mathbb{N} : x_n \geq x_{n+1}$ - невозрастающая последовательность

Очевидно, что возрастающая или убывающая последовательности, для которых

- $\forall n \in \mathbb{N} : x_n < x_{n+1}$ - возрастающая последовательность
- $\forall n \in \mathbb{N} : x_n > x_{n+1}$ - убывающая последовательность

являются монотонными

$$\{a_n\}_{n=1}^{\infty} = 2^n - n \tag{1}$$

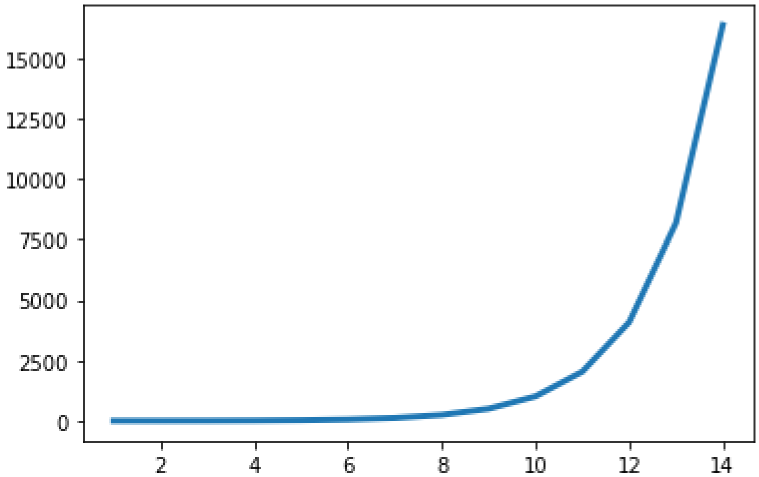
Определим характер последовательности, для чего вычтем из n+1 члена последовательности n-й

$a_{n+1} - a_n = (2^{n+1} - (n+1)) - (2^n - n) = 2^{n+1} - n - 1 - 2^n + n = 2^{n+1} - 2^n - 1 = 2^n -$

Очевидно, что $\forall n \in [1; \infty) : 2^n - 1 > 0 \Rightarrow a_n < a_{n+1}$, а значит последовательность (1) является возрастающей, следовательно строго монотонной

Представим это графически

In [4]:
```python
n = np.arange(1, 15, 1)
a = np.power(2, n) - n
plt.plot(n, a, lw=3)
plt.show()
print(f'пятый элемент последовательности равен {a[4]}')
```




```
пятый элемент последовательности равен 27
```

Последовательность (1) является ограниченой снизу, поскольку $2^n - n \geq 1 : \forall n \in [1; \infty)$

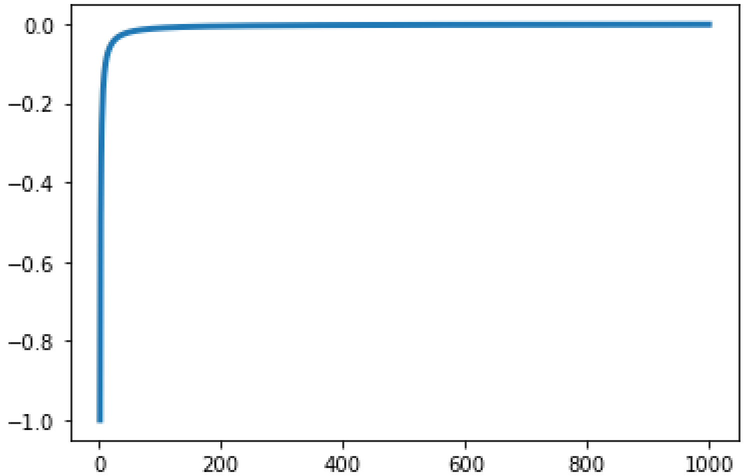
$$\{b_n\}_{n=2}^{\infty} = \frac{1}{1-n} \tag{2}$$

Аналогично,

$$b_{n+1} - b_n = \frac{1}{1-(n+1)} - \frac{1}{1-n} = \frac{1}{-n} - \frac{1}{1-n} = \frac{1-n}{-n(1-n)} - \frac{-n}{-n(1-n)} = \frac{1}{-n(1-n)} = \frac{1}{n^2-n} > 0 : \forall n \in$$

,
а значит последовательность (2) также является возрастающей

In [68]:
```python
n = np.arange(2, 1000, 1)
b = 1 / (1 - n)
plt.plot(n, b, lw=3)
plt.show()
print(f'пятый элемент последовательности равен {b[4]}')
```




```
пятый элемент последовательности равен -0.2
```

Последовательность (2) является ограниченой сверху, поскольку $\frac{1}{1-n} < 0 : \forall n \in [2;\infty)$

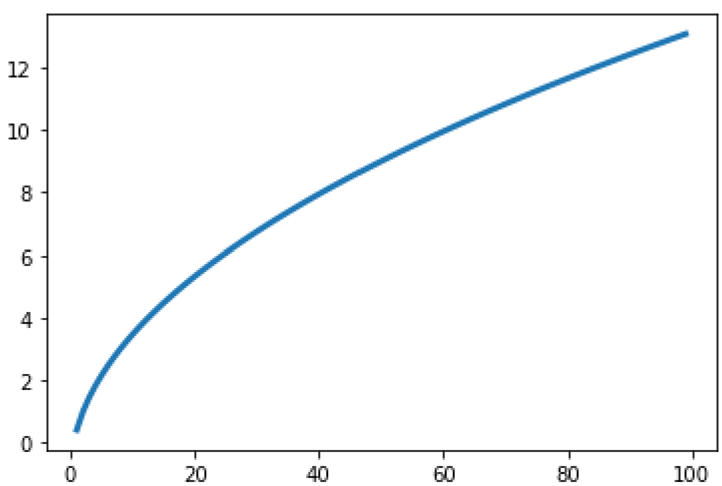
А также ограниченой снизу, так как при первом значении n = 2 значение функции равно -1

$$\{c_n\}_{n=1}^{\infty} = -1^n + \sqrt{2n} \tag{3}$$

Последовательность (3) можно представить как:
$\{c_n\}_{n=1}^{\infty} = -1^n + \sqrt{2n} = \sqrt{2n} - 1$

Следовательно,
$c_{n+1} - c_n = (\sqrt{2(n+1)} - 1) - (\sqrt{2n} - 1) = \sqrt{2(n+1)} - \sqrt{2n} > 0$, так как $\sqrt{2(n+1)} > \sqrt{2n} : \forall n \in [1;\infty)$,
а значит последовательнось является возрастающей

In [6]:
```
n = np.arange(1, 100 ,1)
c = np.sqrt(2 * n) - 1
plt.plot(n, c, lw=3)
plt.show()
print(f'пятый элемент последовательности равен {c[4]}')
```




```
пятый элемент последовательности равен 2.1622776601683795
```

Последовательность (3) является ограниченой снизу, поскольку
$\sqrt{2n} - 1 > 0.4 : \forall n \in [1; \infty)$

$$\{d_n\}_{n=1}^{\infty} = (-1)^{2n} + \frac{1}{n^2} \tag{4}$$

Последовательность (4) можно представить как:

$\{d_n\}_{n=1}^{\infty} = (-1)^{2n} + \frac{1}{n^2} = 1 + \frac{1}{n^2}$

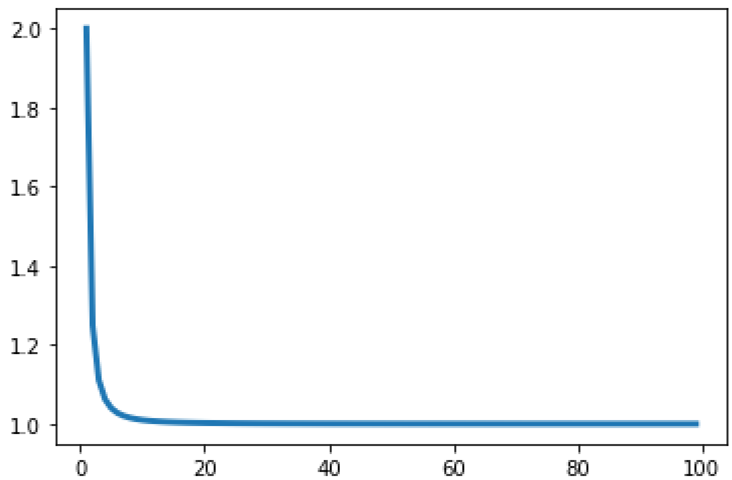
Следовательно,

$d_{n+1} - d_n = (1 + \frac{1}{(n+1)^2}) - (1 + \frac{1}{n^2}) = \frac{1}{(n+1)^2} - \frac{1}{n^2} = \frac{n^2}{n^2(n+1)^2} - \frac{(n+1)^2}{n^2(n+1)^2} = \frac{n^2-(n^2+2n+1)}{n^2(n+1)^2} =$

,

а значит последовательность (4) является убывающей

In [70]:
```
n = np.arange(1, 100, 1)
d = 1 + 1 /np.power(n, 2)
plt.plot(n, d, lw=3)
plt.show()
print(f'пятый элемент последовательности равен {d[4]}')
```




```
пятый элемент последовательности равен 1.04
```

Последовательность (1) является ограниченой снизу, поскольку
$1 + \frac{1}{n^2} > 1 : \forall n \in [1; \infty)$

А также ограниченой сверху, так как при первом значении n = 1 значение функции равно 2

---

2. Найти 12-й член заданной неявно последовательности

$$a_1 = 128,$$

$$a_{n+1} - a_n = 6$$

Последовательность можно представить в виде:

$$\{a_n\}_{n=1}^{\infty} = 128 + 6(n-1)$$

Следовательно, при

$$n = 12$$

$$a_{12} = 128 + 6(12-1) = 128 + 66 = 194$$

In [8]:
```
# То же самое в виде функции на python
def seq(n:int)->int:
    return 128 + 6 * (int(n) - 1)

seq(12)
```

Out[8]: 194

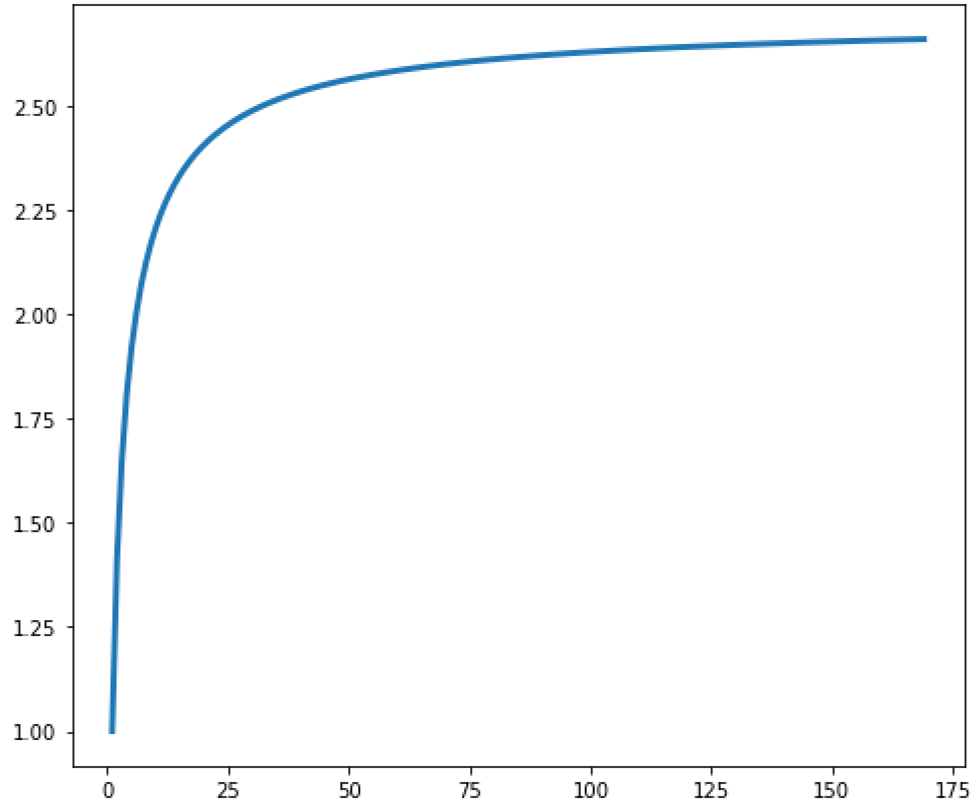
---

3. На языке Python предложить алгоритм вычисляющий численно предел с точностью $\epsilon = 10^{-7}$

$$\lim_{n \to +\infty} \frac{n}{\sqrt[n]{n!}} \tag{5}$$

Попробуем нарисовать функцию (5)

In [9]:
```
plt.figure(figsize=(8, 7))
n = np.arange(1, 170, 1)
y = n / np.power(special.factorial(n), (1 / n))
plt.plot(n, y, lw=3)
plt.show()
print(f'Последний элемент {y[-1]}')
```




```
Последний элемент 2.6628110780187835
```

При значении n > 170 упираемся в ограничение на размер типа данных. Попробуем упростить выражение

Обозначим значение, которое нам нужно найти за L. Тогда,

$$L = \lim_{n\to+\infty} \frac{n}{\sqrt[n]{n!}}$$

Прологарифмируем обе части равенства:

$$\ln L = \ln \lim_{n\to+\infty} \frac{n}{\sqrt[n]{n!}}$$

Пользуясь тем, что логарифм является непрерывной функцией, вынесем предел за него

$$\ln L = \lim_{n\to+\infty} \ln \frac{n}{\sqrt[n]{n!}}$$

Далее, пользуясь свойствами логарифмов, получим

$$\ln L = \lim_{n\to+\infty} \ln \frac{n}{\sqrt[n]{n!}} = \lim_{n\to+\infty} (\ln n - \ln \sqrt[n]{n!}) = \lim_{n\to+\infty} (\ln n - \ln(n!)^{\frac{1}{n}}) = \lim_{n\to+\infty} (\ln n - \frac{1}{n} \ln n!$$

Воспользуемся теоремой Штольца, которая гласит:

Для последовательностей x_n, y_n, причем $y_n > 0, y_n \to +\infty, y_{n+1} > y_n$

$$\lim_{n\to+\infty} \frac{x_n}{y_n} = \lim_{n\to+\infty} \frac{x_{n+1} - x_n}{y_{n+1} - y_n}$$

У нас получается, что

$$x_n = n \ln n - \ln n!$$

$$y_n = n$$

$$\lim_{n\to+\infty} (\frac{n \ln n - \ln n!}{n}) = \lim_{n\to+\infty} (\frac{(n+1)\ln(n+1) - \ln(n+1)! - (n \ln n - \ln n!)}{(n+1) - n}) = \lim_{n\to+\infty} ((n+1)\ln(n+1) - \ln($$

$$\lim_{n\to+\infty} [n \ln(\frac{n+1}{n}) + \ln(\frac{(n+1)n!}{(n+1)!})]$$

$\frac{(n+1)n!}{(n+1)!} = 1$, так как мы умножаем n! на следующее за n число (n+1), а значит это (n+1)!, то же самое, что имеем в знаменателе. Логарифм от 1 равен 0, а значит

$\ln L = \lim\limits_{n \to +\infty} [n \ln(\frac{n+1}{n})] = \lim\limits_{n \to +\infty} [\ln(1 + \frac{1}{n})^n] = \ln \lim\limits_{n \to +\infty} [(1 + \frac{1}{n})^n] \Rightarrow$

$L = \lim\limits_{n \to +\infty} [(1 + \frac{1}{n})^n]$

In [10]:
```
e = 1e-7
def lim_1(precision: float) -> tuple:
    n = 1
    dif = 10 # любое число большее чем точность
    prev = 0
    while dif > precision:
        current = np.power((1 + (1 / n)), n)
        dif = abs(current - prev)
        prev = current
        n += 1
    return current, n
```

In [11]:
```
%%timeit
lim_1(e)
```

```
7.76 ms ± 75.4 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

In [12]:
```
lim_result, iter_number = lim_1(e)

print(f'За {iter_number} итераций значение предела равно {lim_result} при точнос
```

```
За 3688 итераций значение предела равно 2.7179132895130134 при точности 1e-07
```

4. Предложить оптимизацию алгоритма, полученного в задании 3, ускоряющую его сходимость.

$L = \lim\limits_{n \to +\infty} [(1 + \frac{1}{n})^n]$, воспользуемся тем, что это второй замечательный предел, а значит

$L = e$

In [13]:
```
%%timeit
result = np.e
```

```
41.8 ns ± 0.473 ns per loop (mean ± std. dev. of 7 runs, 10000000 loops each)
```

Первый вариант вычислений выполнялся за 8 мс, текущий - 42 нс. Хотя, назвать это вычислением сложно :)

In [14]:
```
print(f'Итоговый результат: {np.e}')
```

```
Итоговый результат: 2.718281828459045
```