

Вбудовані функції Python

```
print(dir(__builtins__))
```

```
import builtins  
print(dir(builtins))
```

Математичні функції

abs(int/float/complex) — абсолютне значення числа, повертає модуль (додатнє значення)

```
print(abs(-5)) #Поверне 5
```

divmod(a, b) — ціла частина і залишок від ділення, повертає кортеж із двох значень: цілої частини та залишку від ділення a на b.

```
print(divmod(10, 3)) # Результат: (3, 1) (3 — ціла частина, 1 — залишок)
```

max(iterable, *args) / min(iterable, *args) — максимальне/мінімальне значення, повертає найбільше/найменше значення з переданих аргументів (числа) або об'єкта ітерації (список/кортеж, що складається з чисел).

```
numbers = [1, 5, 3, 9, 2]
```

```
print(max(numbers)) # Результат: 9
```

```
print(max(3, 7, 2)) # Результат: 7
```

```
print(min(numbers)) # Результат: 1
```

```
print(min(3, 7, 2)) # Результат: 2
```

Математичні функції (продовження)

pow(x, y, mod) — піднесення до степеня. Функція обчислює $x**y$. Якщо вказано третій аргумент **mod**, повертається $(x**y)\%mod$.

```
print(pow(2, 3)) # Результат: 8 (2**3)
```

```
print(pow(2, 3, 5)) # Результат: 3 ((2**3) % 5)
```

round(x, ndigits) — округлення числа, округлює число **x** до найближчого цілого або до **ndigits** знаків після коми, якщо вказано.

```
print(round(3.14159)) # Результат: 3
```

```
print(round(3.14159, 2)) # Результат: 3.14
```

sum(iterable, start=0) - обчислює суму елементів ітерації (кортеж чи список), додаючи їх до початкового значення start (за замовчуванням 0)

```
numbers = [1, 2, 3, 4, 5]
```

```
print(sum(numbers)) # Результат: 15
```

```
print(sum(numbers, 10)) # Результат: 25 (15 + 10)
```

hash(immutable) - отримання геш-значення незмінного об'єкта.

Типи даних та їх перетворення

bool() — перетворює значення у булеве, повертає True чи False.

bytearray() - перетворює значення у змінний масив байтів.

bytes() - перетворює значення у незмінний об'єкт типу байтів.

complex(a, b) - створює комплексне число $a+bj$, де j - уявна одиниця.

dict() — створення словника.

float() — перетворення в число з плаваючою точкою.

frozenset() — незмінна множина.

int() — перетворення в ціле число.

list() — створення списку.

memoryview() — створює об'єкт, який дозволяє отримувати доступ до байтових даних без їх копіювання. **memoryview()** працює лише з об'єктами, які підтримують інтерфейс буферу пам'яті, наприклад: **bytes**, **bytearray**, **array.array**.

object() - базовий клас для всіх типів, усі класи та типи в Python наслідуються від object, навіть якщо це явно не вказано.

range(start=0, stop не включно, step=1) - створює послідовність цілих чисел.

set() — створення множини.

Типи даних та їх перетворення (продовження)

slice() — Об'єкт для нарізки послідовностей.

```
data = [0, 1, 2, 3, 4, 5, 6]
```

```
s = slice(1, 5, 2)
```

```
# Вибір елементів з 1 до 5 з кроком 2
```

```
print(data[s]) # Результат: [1, 3]
```

str() — Перетворення в рядок.

tuple() — Створення кортежу.

type() — Отримання або створення типу.

zip() — об'єднання ітераційних об'єктів у кортежі, довжина визначається за найкоротшим, створює zip-об'єкт, який потрібно перетворити на список, словник, кортеж

Робота з об'єктами (екземплярами класу) та їх атрибутами

callable() — перевірка, чи об'єкт можна викликати.

classmethod() — створення методу класу.

getattr() — отримання атрибута об'єкта.

hasattr() — перевірка наявності атрибута у об'єкта.

property() — створення властивості (дозволяє створювати властивості (properties), які виглядають як звичайні атрибути, але їх значення обчислюється або задається через методи (getter, setter, deleter)).

setattr() — встановлення атрибута об'єкта.

staticmethod() — створення статичного методу.

super() — доступ до батьківського класу.

vars() — повертає атрибути об'єкта у вигляді словника.

Робота з кодом

compile() — використовується для компіляції рядка з кодом Python, блоку байткоду або AST (Abstract Syntax Tree) у code object. Цей об'єкт може бути виконаний за допомогою функцій `exec()` або `eval()`.

eval() — використовується для обчислення виразів, записаних у вигляді рядка, в поточному контексті Python.

exec() — використовується для виконання Python-коду, представленого у вигляді рядка або об'єкта коду.

filter(function, iterable) — фільтрація ітерації за функцією.

```
def is_even(number):
```

```
    return number % 2 == 0
```

```
numbers = [1, 2, 3, 4, 5, 6]
```

```
result = filter(is_even, numbers)
```

```
print(list(result))
```

map() — застосування функції до ітерації.

Робота з функціями

(приймають функції й ітеративні об'єкти у якості аргумента, повертають об'єкт-ітератор, щоб побачити результати, потрібно перетворити його на список, кортеж або інший тип ітерації)

filter(function, iterable) — фільтрація ітеративного об'єкта за функцією,
numbers = [1, 2, 3, 4, 5, 6]

```
def is_even(number):  
    return number % 2 == 0  
result = filter(is_even, numbers)
```

map(function, iterable) — застосування функції до ітеративного об'єкта.

```
def square(x):  
    return x**2  
result = map(square, numbers)
```


Управління ввід/вивід

chr() — приймає ціле число, яке є кодом символу в таблиці Unicode (діапазон від 0 до 1114111) - повертає символ.

input(prompt) — отримання вводу від користувача, prompt - рядок тексту, який буде виведено перед запитом на введення..

open(file, mode='r', buffering=-1, encoding=None, errors=None) — відкриття файлу.

- file: Ім'я файлу або шлях до нього.
- mode: Режим відкриття файлу:
 1. 'r' — читання (за замовчуванням),
 2. 'w' — запис (створення нового файлу або очищення існуючого),
 3. 'a' — додавання до кінця файлу,
 4. 'b' — двійковий режим (додається до інших режимів, наприклад, 'rb'),
 5. 'x' — створення нового файлу (помилка, якщо файл існує).
- encoding (опціонально): Кодування тексту (наприклад, 'utf-8')
- errors - інструкція з обробки помилок

print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False) виведення на екран

- *objects: Дані для виведення (розділяються комами).
- sep: Рядок, який розділяє об'єкти (за замовчуванням пробіл ' ').
- end: Рядок, який додається в кінці (за замовчуванням новий рядок '\n').
- file: Потік, у який буде здійснюватись виведення (за замовчуванням sys.stdout).
- flush: Якщо True, потік буде очищено після виведення.

Робота з рядками

ascii() — представлення об'єкта у вигляді ASCII.

bin() — бінарне представлення числа.

format() — форматування рядків.

hex() — шістнадцяткове представлення числа.

oct() — восьмеричне представлення числа.

ord() — код символу Unicode.

repr() — офіційне строкове представлення об'єкта.

Дебаг/перевірка

breakpoint() - виконання програми зупиняється в цьому місці, можна взаємодіяти з кодом через інтерактивну консоль дебагера

isinstance(object, class_or_tuple) - використовується для перевірки, чи є об'єкт екземпляром певного класу (або будь-якого класу / типу даних з кортежу класів)

issubclass(class, class_or_tuple) перевіряє, чи є один клас підкласом іншого.

id(object) - повертає унікальний ідентифікатор (адресу в пам'яті) переданого об'єкта

Робота з ітераційними об'єктами

all(iterable) — перевіряє, чи всі елементи є істинними (True). Повертає True/False.

any(iterable) — перевіряє, чи хоча б один елемент є істинним (True). Повертає True/False.

enumerate(iterable, start=0) — додає індекси до елементів ітерації, **start** — число, з якого починається індексація.

iter(object, sentinel) - створює ітератор з об'єкта, **sentinel** — значення, при досягненні якого ітерація зупиняється

next(iterator, default) - повертає наступний елемент з ітератора, **default** - значення, яке повертається, якщо більше немає елементів.

len(s) - повертає кількість елементів у об'єкті.

reversed(seq) - повертає ітератор, який перебирає елементи у зворотному порядку.

sorted(iterable, key=None, reverse=False) - повертає новий список із відсортованими елементами, **key** — функція для визначення критерію сортування, **reverse** — **True** для сортування у зворотному порядку (за замовчуванням **False**).

Системні функції

globals() - повертає словник усіх глобальних змінних у поточній області видимості.

locals() - повертає словник усіх локальних змінних у поточній області видимості.

help(object) - викликає вбудовану систему довідки Python для отримання інформації про об'єкт (модуль, функцію, клас, метод тощо).

dir() - повертає список атрибутів і методів для переданого об'єкта. Якщо об'єкт не передано, вона повертає список імен змінних, доступних у поточній області видимості.

exit() та **quit()** - завершують виконання програми.

copyright, credits, license - виводить інформацію про авторські права Python, подяки розробникам Python, текст ліцензії Python