

Погружение в Python

Урок 2

Простые типы данных





Содержание урока





План курса





Что будет на уроке сегодня

- 📌 Простые типы данных и коллекции
- 📌 Аннотация типов
- 📌 Объект, его атрибуты и методы
- 📌 Простые объекты
- 📌 Математика в Python





Простые типы данных и коллекции





Строгая динамическая типизация Python



Функция `type(object)`

Возвращает класс объекта, его тип



Функция `id(object)`

Возвращает адрес объекта в оперативной памяти



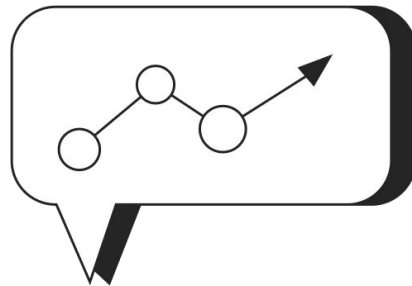
Функция `isinstance(object, classinfo)`

Принимает на вход объект и класс и возвращает истину, если объект является экземпляром прямого или косвенного подкласса



Оператор `is`

Сравнивает пару объектов на идентичность





Изменяемые и неизменяемые типы

Основные математические операторы представлены в таблице.

Неизменяемые	Изменяемые
None	
Числа: int, bool, float, complex	
Последовательности: str, tuple, bytes	Последовательности: list, bytearray
Множества: set	Множества: frozenset
	Отображения: dict



Хэш как проверка на неизменяемость

Хеш — это криптографическая функция хеширования, которую обычно называют просто хэшем.

Хеш — функция представляет собой алгоритм, который может преобразовать произвольный массив данных в набор бит фиксированной длины.



Функция `hash(object)`

Возвращает hash объекта в виде целого числа





Напишите небольшую программу.
Результат работы пришлите в чат. У вас 5 минут.

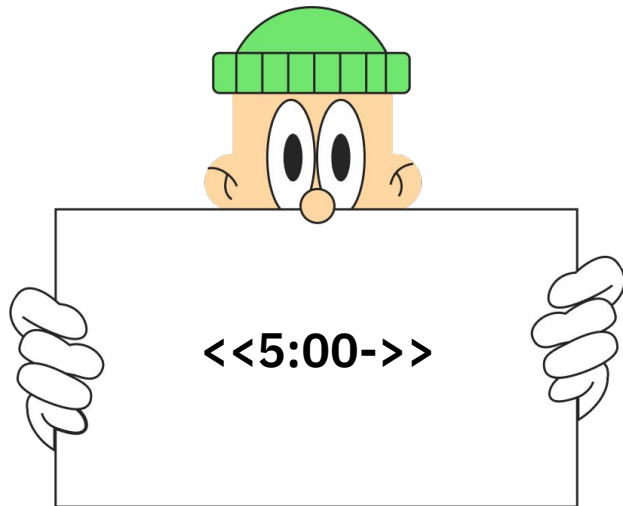


Типы данных

Напишите небольшую программу, которая запрашивает у пользователя любой текст и выводит о нём следующую информацию:

- ✓ тип объекта,
- ✓ адрес объекта в оперативной памяти,
- ✓ хеш объекта.

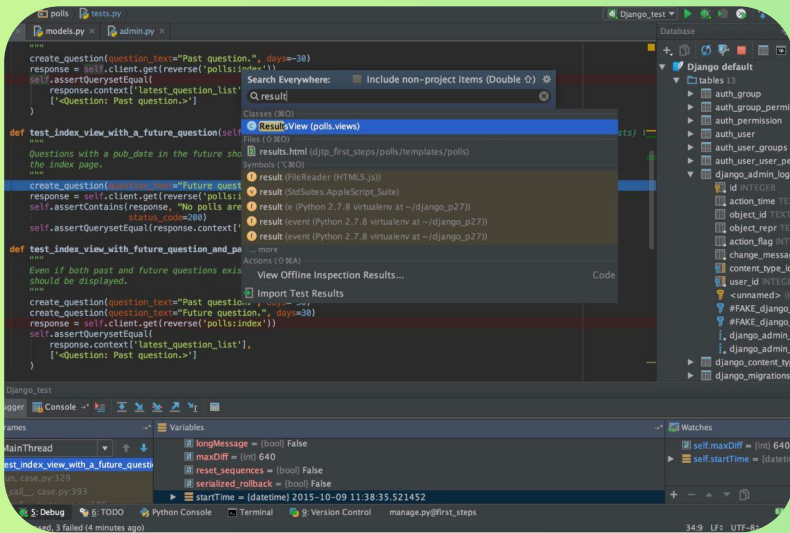
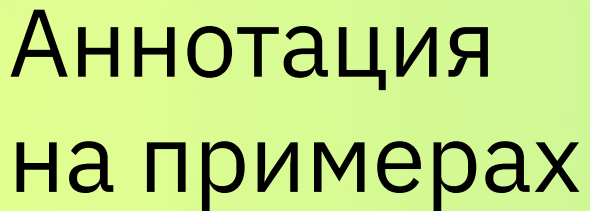
Результат работы пришлите в чат.





Аннотация типов







Модуль typing

- ✓ **Примитивы супер специального типа:** Annotated, Any, Callable, ClassVar, Final, ForwardRef, Generic, Literal, Optional, Protocol, Tuple, Type, TypeVar, Union
- ✓ **Абсолютные типы из collections.abc:** AbstractSet, ByteString, Container, ContextManager, Hashable, ItemsView, Iterable, Iterator, KeysView, Mapping, MappingView, MutableMapping, MutableSequence, MutableSet, Sequence, Sized, ValuesView, Awaitable, AsyncIterator, AsyncIterable, Coroutine, Collection, AsyncGenerator, AsyncContextManager
- ✓ **Структурные проверки, протоколы:** Reversible, SupportsAbs, SupportsBytes, SupportsComplex, SupportsFloat, SupportsIndex, SupportsInt, SupportsRound
- ✓ **Коллекция конкретных типов:** ChainMap, Counter, Deque, Dict, DefaultDict, List, OrderedDict, Set, FrozenSet, NamedTuple, TypedDict, Generator
- ✓ **Другие конкретные типы:** BinaryIO, IO, Match, Pattern, TextIO
- ✓ **Одноразовые вещи:** AnyStr, cast, final, get_args, get_origin, get_type_hints, NewType, no_type_check, no_type_check_decorator, NoReturn, overload, runtime_checkable, Text, TYPE_CHECKING



Объект, его атрибуты
и методы





Атрибуты и методы

Атрибуты и методы есть практически у всех объектов в Python.

1

Атрибуты — это переменные, конкретные характеристики объекта, такие как цвет поля или имя пользователя.

2

Методы — это функции, которые описаны внутри объекта или класса. Они относятся к определенному объекту и позволяют взаимодействовать с ними или другими частями кода.



Функции для получения информации об атрибутах и методах



Функция `dir(object)`

Попытается вернуть список допустимых атрибутов для объекта.

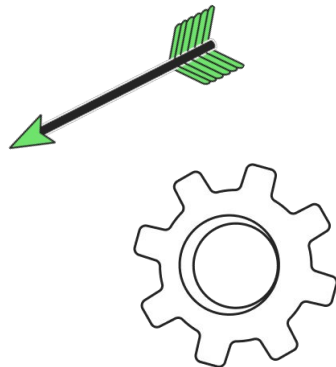
Если объект не передавать — список имен в текущей локальной области



Функция `help(object)`

Если объект не указан, запускается интерактивная справочная система.

Если аргумент является строкой, то эта строка ищется как имя модуля, функции, класса, метода, ключевого слова или раздела документации и далее выводится страница справки. Если аргументом является объект любого другого типа, создается страница справки по этому объекту.



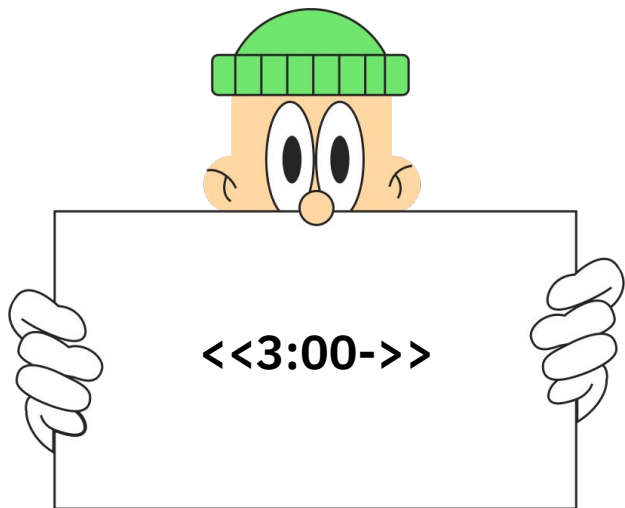


Запустите интерактивный режим справки
и проведите два небольших исследования.
У вас по три минуты на каждое исследование.



Интерактивная работа со справкой, `help()`

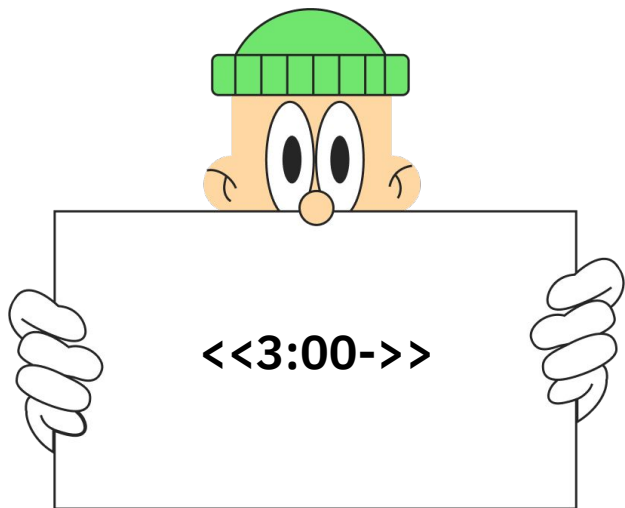
Введите команду `keywords`, далее любое интересное вам ключевое слово из списка. Прочитайте описание и напишите в чат пару слов о том, что узнали.





Интерактивная работа со справкой, `help()`

Введите команду `symbols`, далее любой заинтересовавший вас символ из списка. Прочитайте описание и напишите в чат пару слов о том, что узнали.





«Простые» объекты





Целые числа

Разберём на примерах.

- ✓ **int(x, base=10)** — возвращает целочисленный объект, созданный из числа или строки `x`, или возвращает значение 0, если аргументы не заданы. **base** — основание системы счисления, от 2 до 36.
- ✓ **bin(x)** — преобразует целое число в двоичную строку с префиксом «0b».
- ✓ **oct(x)** — преобразует целое число в восьмеричную строку с префиксом «0o».
- ✓ **hex(x)** — преобразует целое число в строчную шестнадцатеричную строку с префиксом «0x».





Вещественные числа

Разберём на примерах.

- ✓ **float(x)** — возвращает число с плавающей запятой, составленное из числа или строки `x`.

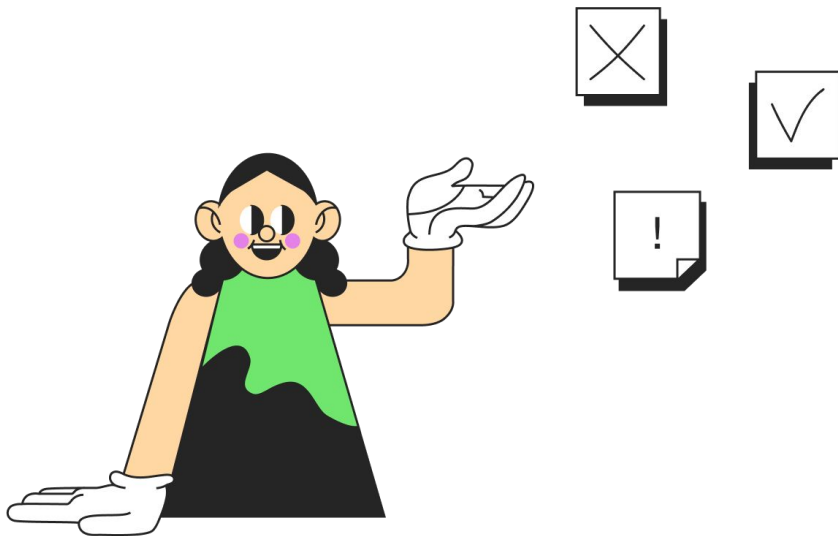




Логические типы

Разберём на примерах.

- ✓ **bool(x)** — возвращает логическое значение, т.е. одно из двух: True или False.





Строки и способы их записи

`str(object='')` — возвращает строковую версию объекта.

Три вида кавычек

```
txt = 'Книга называется "Война и мир".'
```

```
class str(object):
```

```
    """
```

```
    str(object='') -> str
```

```
    str(bytes_or_buffer[, encoding[,  
errors]]) -> str
```

```
    ...
```

```
    """
```

Склеивание и обратный слеш

```
text = 'Привет.' 'Как ты, друг?' 'Рад тебя  
видеть.'
```

```
very_long_text = 'Lorem ipsum dolor sit amet,  
consectetur adipisicing elit. A ab alias animi  
assumenda at aut ' \
```

```
                'commodi, consequatur cumque  
ea harum, hic id illum ipsam itaque laboriosam  
magnam minus nam nulla ' \
```

```
                'numquam obcaecati officia  
officiis porro possimus praesentium quaerat  
temporibus ullam veniam? '
```




Конкатенация строк

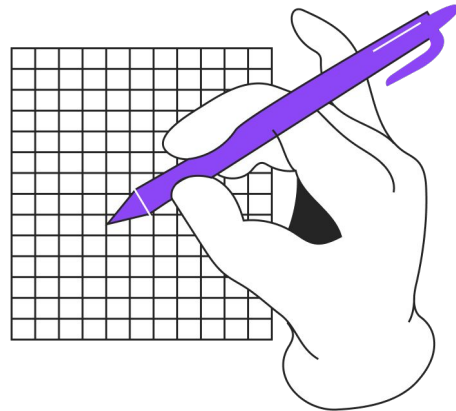


Сложение строк через плюс

```
LIMIT = 120
ATTENTION = 'Внимание!'
name = input('Твоё имя? ')
age = int(input('Твой возраст? '))

text = ATTENTION + ' Хотя тебе и осталось ' + \
       str(100 - age) + " до ста лет, но длинна" \
       " строки не должна превышать " + \
       str(LIMIT) + ' символов.'

print(text)
```





Размер строки в памяти

`object.__sizeof__()` — метод возвращает размер объекта в байтах

```
empty_str = ''  
en_str = 'Text'  
ru_str = 'Текст'  
unicode_str = '😄😍😞😏'
```

```
print(empty_str.__sizeof__())  
print(en_str.__sizeof__())  
print(ru_str.__sizeof__())  
print(unicode_str.__sizeof__())
```



Размер строки в памяти. 48 байт служебной информации для x64 Python. Далее 1, 2 или 4 байта на каждый символ в зависимости от кодировки.



Модуль typing

- ✓ **str.isalnum()** — возвращает True, если все символы в строке буквенно-цифровые. Символ является буквенно-цифровым, если одно из следующих значений возвращает True: `c.isalpha()`, `c.isdecimal()`, `c.isdigit()` или `c.isnumeric()`.
- ✓ **str.isalpha()** — возвращает True, если все символы в строке являются буквенными. Алфавитные символы — это символы, определенные в базе данных символов Юникода как «буква».
- ✓ **str.isdecimal()** — возвращает True, если все символы в строке являются десятичными символами
- ✓ **str.isdigit()** — возвращает True, если все символы в строке являются цифрами. Цифры включают десятичные символы и цифры, требующие специальной обработки, например цифры надстрочного индекса совместимости.
- ✓ **str.isnumeric()** — возвращает True, если все символы в строке являются числовыми символами. Числовые символы включают цифровые символы и все символы, которые имеют свойство числового значения Unicode.



Модуль typing

- ✓ **str.isascii()** — возвращает True, если строка пуста или все символы в строке ASCII.
- ✓ **str.islower()** — возвращает True, если все символы в строке в нижнем регистре.
- ✓ **str.istitle()** — возвращает True, если строка является строкой с заглавным регистром и содержит хотя бы один символ.
- ✓ **str.isupper()** — возвращает True, если все символы в строке в верхнем регистре.
- ✓ **str.isprintable()** — возвращает True, если все символы в строке доступны для печати или строка пуста. Непечатаемые символы — это символы, определенные в базе данных символов Unicode как «Другие» или «Разделители», за исключением пробела ASCII (0x20), который считается печатаемым.
- ✓ **str.isspace()** — возвращает True, если в строке есть только пробельные символы.



Напишите небольшую программу.
Результат работы отправьте в чат. У вас 7 минут.

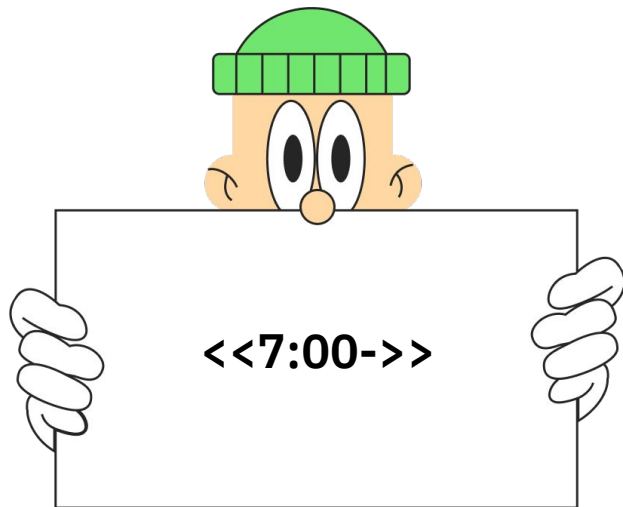


«Простые» объекты

Напишите небольшую программу, которая запрашивает у пользователя текст.

Если текст можно привести к целому числу, выведите его двоичное, восьмиричное и шестнадцатеричное представление.

А если преобразование к целому невозможно, сообщите написан ли текст в ASCII или нет.





Математика в Python





Математические модули

В Python есть несколько модулей в стандартной библиотеке, которые облегчают математические расчёты. Для доступа к ним необходимо выполнить импорт в начале файла.

```
import math  
import decimal  
import fractions
```

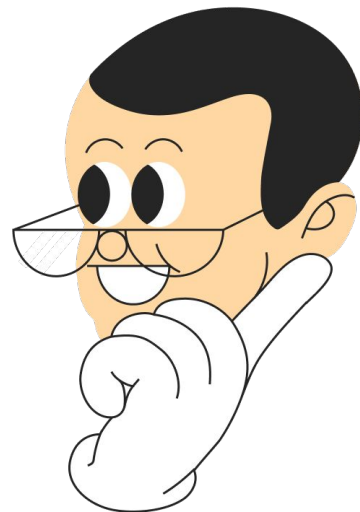




Модуль math

Константы: e, inf, nan, pi, tau

Математические функции: acos, acosh, asin, asinh, atan, atan2, atanh, ceil, comb, copysign, cos, cosh, degrees, dist, erf, erfc, exp, expm1, fabs, factorial, floor, fmod, frexp, fsum, gamma, gcd, hypot, isclose, isfinite, isinf, isnan, isqrt, lcm, ldexp, lgamma, log, log10, log1p, log2, modf, nextafter, perm, pow, prod, radians, remainder, sin, sinh, sqrt, tan, tanh, trunc, ulp





Модуль decimal



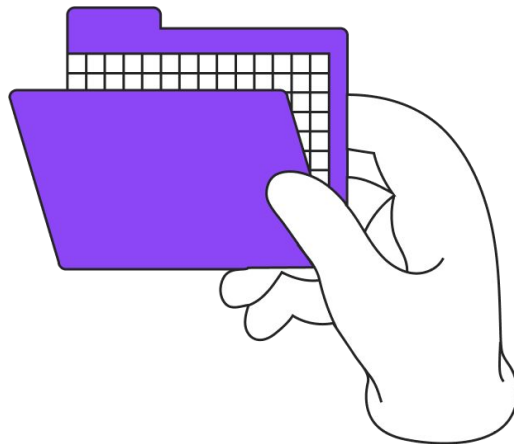
`num = decimal.Decimal(object)`

Получаем вещественное число
с точностью 28 знаков (до и после запятой).



`decimal.getcontext().prec = dec`

Задаём точность в dec знаков
для будущих операций.

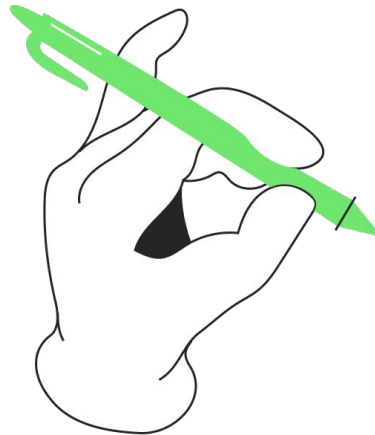




Модуль fraction

? **Запись дробей вида $\frac{1}{3}$, $\frac{3}{5}$ и т.п.**

```
f1 = fractions.Fraction(1, 3)
print(f1)
f2 = fractions.Fraction(3, 5)
print(f2)
print(f1 * f2)
```





Комплексные числа

`complex([real[, imag]])` — комплексное число из действительной `real` и мнимой `imag` частей.

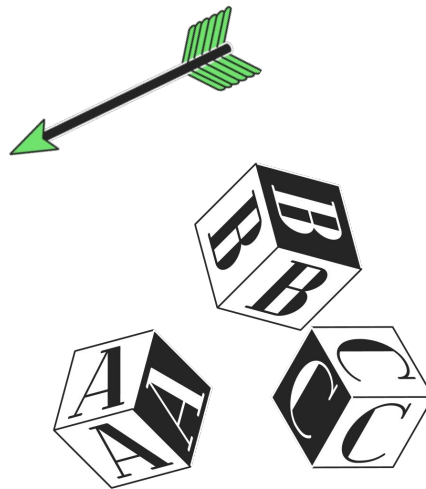
```
a = complex(2, 3)
b = complex('2+3j')
print(a, b, a == b, sep='\n')
```





Математические функции «из коробки»

- ✓ **abs(x)** — возвращает абсолютное значение числа x , число по модулю.
- ✓ **divmod(a, b)** — функция принимает два числа в качестве аргументов и возвращает пару чисел — частное и остаток от целочисленного деления. Аналогично вычислению $a // b$ и $a \% b$.
- ✓ **pow(base, exp[, mod])** — при передаче 2-х аргументов возводит $base$ в степень exp . При передаче 3-х аргументов, результат возведения в степень делится по модулю на значение mod .
- ✓ **round(number[, ndigits])** — округляет число $number$ до $ndigits$ цифр после запятой. Если второй аргумент не передать, округляет до ближайшего целого.





Итоги занятия





На этой лекции мы

- 📌 Познакомились со строгой динамической типизацией языка Python.
- 📌 Изучили понятие объекта в Python. Разобрались с атрибутами и методами объектов.
- 📌 Рассмотрели способы аннотации типов.
- 📌 Изучили «простые» типы данных, такие как числа и строки.
- 📌 Узнали про математические возможности Python.

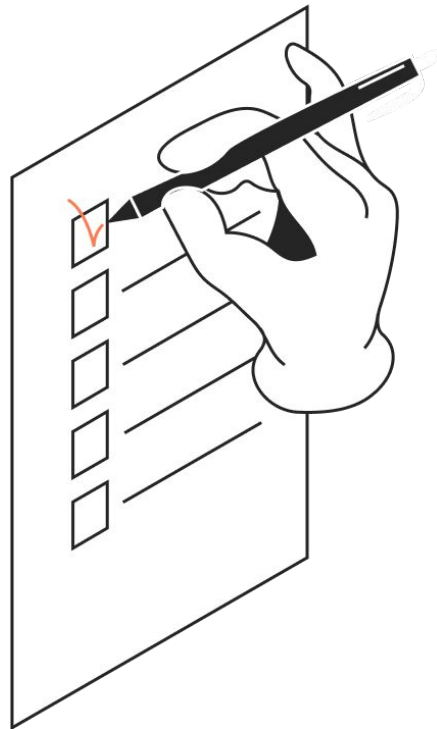




Задание

Поработайте со справочной информацией в Python, функцией `help()`. Попробуйте найти зарезервированные слова, функции и модули, которые прошли за две лекции и почитать про них.

Если вы плохо читаете на английском, воспользуйтесь любым онлайн переводчиком.





Спасибо за внимание