**Основы python**

**Интерпретатор и интерпретируемость**

Интерпретатор python – это система, состоящая из компилятора и виртуальной машины. Компилятор переводит программный код в байт-код, проверяя синтаксические ошибки, в свою очередь виртуальная машина построчно выполняет байт-код.  
Байт-код кешируется (файл \_\_pycache\_\_)

**Изменяемые и неизменяемые типы данных**

**Изменяемые (mutable)** — объекты, содержимое которых можно менять после создания.  
**Неизменяемые (immutable)** — объекты, содержимое которых нельзя менять; любое изменение создаёт новый объект.  
Неизменяемые:

* int, float, complex
* bool
* str
* tuple
* frozenset
* bytes

Изменяемые:

* list
* dict
* set
* bytearray

**Основные структуры данных в python**

**1. list (список) = [1, 2, 3]**

* Упорядоченный контейнер, допускает дубликаты.
* Изменяемый (можно добавлять, удалять, изменять элементы).
* Индексация и срезы поддерживаются.

**2. tuple (кортеж) = (1, 2, 3)**

* Упорядоченный контейнер, допускает дубликаты.
* Неизменяемый — после создания элементы нельзя менять.
* Может содержать изменяемые объекты.

**3. set (множество) = {1, 2, 3}**

* Неупорядоченный контейнер.
* Изменяемый, элементы уникальные (дубликаты автоматически удаляются).
* Быстрые операции: объединение, пересечение, разность.

**4. frozenset = frozenset([1, 2, 3])**

* То же, что set, но неизменяемый.
* Можно использовать как ключ словаря или элемент множества.

**5. dict (словарь) = {“first\_name”: “Nikita”, “second\_name”: “Zhidkov”}**

* Ассоциативный массив: ключ → значение.
* Ключи должны быть неизменяемыми (int, str, tuple, frozenset).
* Значения могут быть любыми.

**Основные встроенные функции в python**

**1. map() = map(lambda x: x\*\*2, nums)**

* Применяет функцию ко всем элементам итерируемого объекта.
* Возвращает итератор.

**2. lambda (лямбда-функции) = lambda x: x\*\*2**

* Анонимная функция, создаётся в одну строку.
* Синтаксис: lambda аргументы: выражение.

**3. copy и deepcopy**

* copy.copy(obj) — поверхностная копия, вложенные объекты остаются ссылками.
* copy.deepcopy(obj) — глубокая копия, рекурсивно копирует все вложенные объекты.

**4. super()**

* Позволяет вызвать метод родительского класса без явного имени родителя.
* Важен при множественном наследовании.

**Классы и ООП в python**

**Основные ключевые слова**

1. \_\_new\_\_ и \_\_init\_\_ - система, переопределяющая конструктор класса. New инициализирует атрибуты, init их задает.