МФТИ

ПРОГРАММИРОВАНИЕ НА PYTHON

МФТИ

Модуль 1. Базовые конструкции языка Python

Переменные и основные операторы

Содержание

- Принцип работы переменных в Python
- Базовые типы данных
- Основные арифметические операции
- Логические операторы и выражения

Актуализация материала. Работа в группах.
Проблемный вопрос.

У вас есть коробка, в которую вы складываете разные вещи.

Вопрос:
«Если вы положите в коробку книгу, а потом закроете её, как вы найдёте эту книгу через неделю, не открывая все коробки подряд?»

Часть 1

МФТИ

Принцип работы переменных в Python

Определение

Переменные – области в памяти компьютера, предназначенные для хранения данных и проведения с этими данными различных операций.

Имена переменных

Можно:

- Английские буквы a-z, A-Z
- Цифры
- Нижнее подчёркивание (underscore)

Имена переменных

Можно:

- Английские буквы a-z, A-Z
- Цифры
- Нижнее подчёркивание (underscore)

Нельзя:

- Неанглийские буквы
- Начинать с цифры
- Зарезервированные слова синтаксиса Python

Имена переменных

Зарезервированные слова:

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Имена переменных

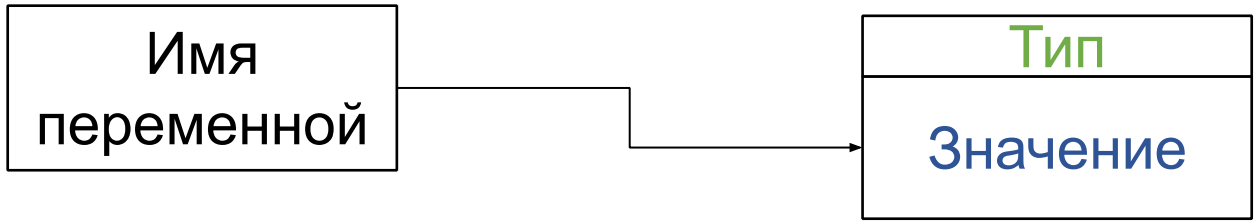
Следует:

- Писать разумные названия
- Использовать snake case
  Пример: number_of_cats

Принцип работы

Пространство имён

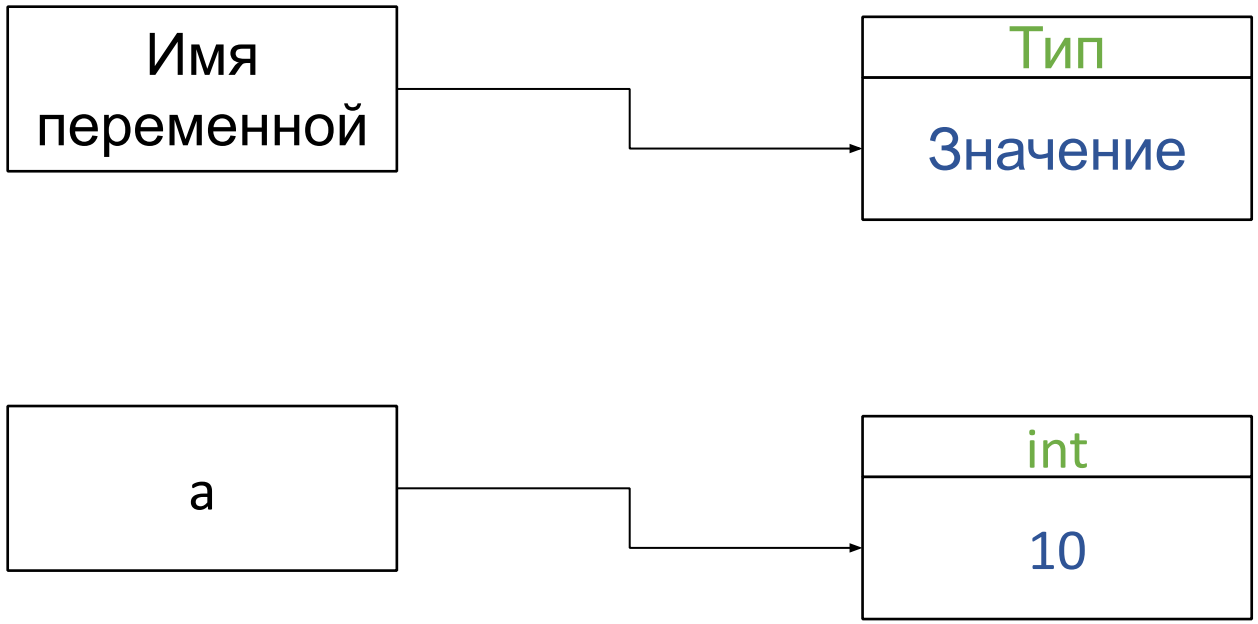
Пространство объектов



Принцип работы

Пространство имён

Пространство объектов



Структура переменной

- Имя (идентификатор) — придумывается программистом и подчиняется стилю именования lower_case_with_underscores
- Значение — информация, которая хранится в памяти компьютера и с которой работают программы
- Адрес — это номер ячейки памяти, в которой хранится значение переменной

Структура переменной

МФТИ

Пример

```
name = 'Анна'
print(type(name))   # Вывод: <class 'str'>
print(id(name))
# Вывод: 138945339796144 (адрес в памяти)
a = 123
print(type(a))      # Вывод: <class 'int'>
print(id(a))        # Вывод: 11645992
```

Множественное присваивание и обмен переменных

Python позволяет присваивать одно значение нескольким переменным одновременно. Таким образом, вы можете инициализировать несколько переменных, а позже переназначить их:

```
x = y = z = 0
```

Множественное присваивание и обмен переменных

МФТИ

Пример

```
a = 100
b = 200
a, b = b, a  # Магия Python!
print(a, b)  # Вывод: 200 100
```

Часть 2

МФТИ

Базовые типы данных

Типы данных

На курсе нам предстоит работать с:

- Numbers (числа)
- Strings (строки)
- Lists (списки)
- Dictionaries (словари)
- Tuples (кортежи)
- Sets (множества)
- Boolean (логический тип данных)

Базовые типы данных

МФТИ

Сейчас нам понадобятся:

- str
  - string - хранит строку текста
- int
  - integer - хранит целое число
- float
  - floating-point number - хранит вещественное число
- bool
  - boolean data type - хранит True или False

Базовые типы данных

МФТИ

Пример

```
num_int = 10          # int
num_float = 3.14      # float
text = "Привет"       # str
flag = True           # bool
print(type(num_int), type(num_float),
type(text), type(flag))
```

Конвертация типов

Чтобы преобразовать строку в число, воспользуемся функцией int().

Преобразование в вещественное число организуется с помощью функции float().

Конвертация типов

МФТИ

Пример

```
num = 210.3
print(type(num))      # <class 'float'>
num = int(num)        # Превращаем в целое
print(num)
# Вывод: 210 (дробная часть потеряна!)
```

Часть 3

МФТИ

Основные арифметические операции

Арифметические операции

- сложение «+»
- вычитание «-»
- умножение «*»
- деление «/»
- возведение в степень «**»
- целочисленное деление «//»
- остаток от деления «%»

Арифметические операции

Как и в математике, скобки влияют на приоритет операций:

```
print(10 * 2 + 2)
print(10 * (2 + 2))

# Вывод:
# 22
# 40
```

Задание

Напишите программу, которая запрашивает у пользователя два числа (через input()) и выводит на экран их целочисленное деление. Не забудьте преобразовать строки в числа!

Бинарное представление числа

Двоичное число — это число, состоящее из двоичных цифр.

Python не считается подходящим языком для низкоуровневого программирования, инструменты для работы с двоичным представлением данных у него всё же есть. Например, чтобы вывести число в бинарном виде, воспользуемся функцией bin():

```
bin(5)

# Вывод: 0b101
```

Часть 4

МФТИ

Логические операторы и выражения

Операторы сравнения

Используются для сравнения значений.
Результатом всегда является True (истина) или False (ложь).

Оператор	Значение	Пример (True)
==	равно	10 == 10
!=	не равно	2 != 3
>	больше	2 > 5 → False
<	меньше	8 < 3 → False
>=	больше или равно	7 >= 7
<=	меньше или равно	3 <= 3

Логические операторы

Служат для создания составных условий

Оператор	Логика	Пример
and (И)	True только если **оба** выражения истинны.	(5 > 2) and (3 < 4) → True
or (ИЛИ)	True, если **хотя бы одно** выражение истинно.	(5 > 2) or (3 > 4) → True
not (НЕ)	Инвертирует значение.	not (5 > 2) → False

«Ленивая» обработка логических выражений

Python не вычисляет всё выражение целиком, если результат уже ясен.

Для and: если первое значение False, второе не проверяется.

Для or: если первое значение True, второе не проверяется.

Пример

```
x = 13
y = False
print(x or y)  # Вывод: 13 (число, а не True!)
```

Битовые операторы

Побитовые операторы предназначены для изменения строк с двоичным кодом, что может понадобиться для работы с криптографическими алгоритмами, драйверами различных устройств или, например, с сетевой инфраструктурой.

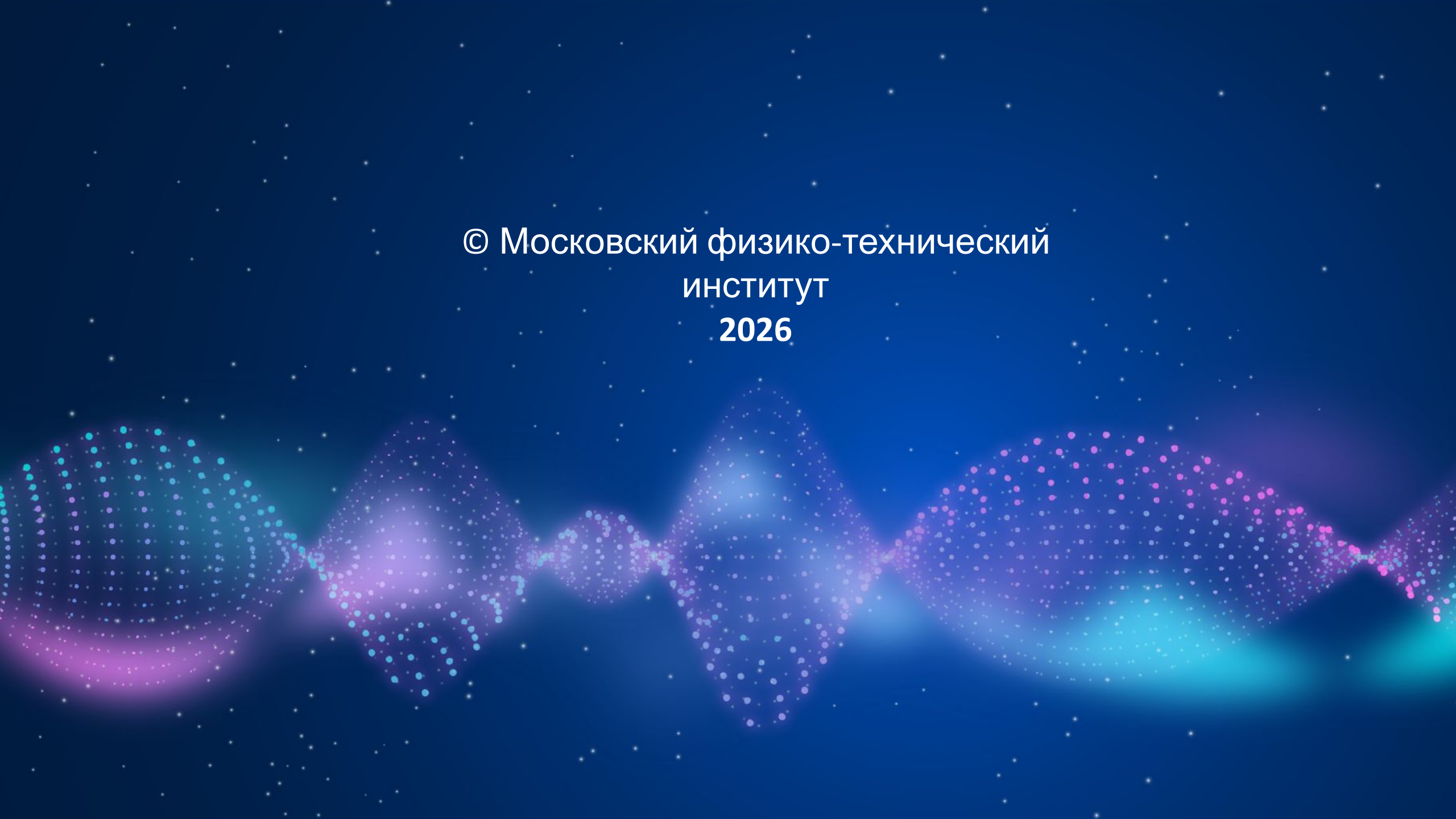
Пример

Возьмем числа 4 = 100 и 3 = 011. Побитовое «ИЛИ» — это операция по применению логического «ИЛИ» к каждой паре битов, которые стоят на одинаковых позициях. Если записать наши числа друг над другом и применить логическую операцию «ИЛИ», мы получим

```
100
или
011
---
```

Итоги

- Переменная — ссылка на объект в памяти, имеющий адрес, тип и значение
- Базовые типы данных для Python — это строки, целые числа, вещественные числа и логический тип данных bool
- Python поддерживает все основные арифметические операции, включая вычисление остатка от деления
- Для сравнения чисел и составления логических выражений используются базовые математические и логические операторы типа


© Московский физико-технический
институт
2026