



Знакомство с Python

Курс “Python для начинающих”

О преподавателе



Серов Николай Евгеньевич

О себе

Backend-разработчик и аналитик данных (Python) | Преподаватель высшей школы | Кандидат технических наук

Специализация:

Разработка серверных решений, миграция и оптимизация БД, обучение IT-дисциплинам.

🔧 Технологический стек:

Backend:

- Python: **Django**, **FastAPI**, Flask, Litestar (REST API, микросервисы, коробочные решения)
- Брокеры сообщений: **RabbitMQ**, Kafka
- Серверная разработка: **PL/pgSQL**, асинхронные решения

Базы данных:

- Реляционные: **PostgreSQL**, MySQL, Oracle, MariaDB (миграции, оптимизация запросов)
- NoSQL: ****MongoDB**, Redis *******
- Аналитика: **OLAP** (ClickHouse) и **OLTP**-системы

🎓 Преподавательская экспертиза: Разработка курсов по:

- Backend-разработке (Python, API, микросервисы)
- Базам данных (SQL/NoSQL, проектирование, оптимизация)
- Аналитике данных (Pandas, визуализация)
- Подготовка учебных программ для ВУЗов и корпоративного обучения.

💡 Ключевые проекты:

- ✓ Миграция Oracle → PostgreSQL для госучреждения (снижение затрат на 40%)
- ✓ Разработка API для интернет-магазинов (FastAPI + RabbitMQ, 5K RPS)
- ✓ Оптимизация OLAP-системы для аналитики Big Data (ускорение отчётов)

Почему стоит сотрудничать?

- **Гибкость:** Готов работать с legacy-кодом и modern-стеком.
- **Опыт преподавания:** Умею объяснять сложное просто (доказано студентами!).
- **Фокус на результат:** Автоматизирую рутину, выбираю технологии под бизнес-задачи.

Опыт

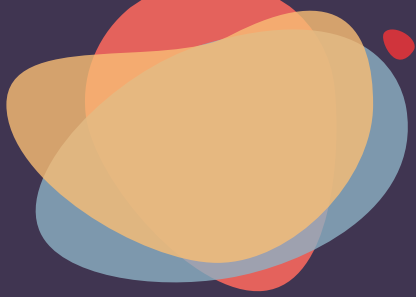
- ✓ МИРЭА — Российский технологический университет, доцент • с 2023 г. (2 года)
- ✓ На сервисе с июня 2024 г. (1 год)

Достижения

- ✓ Войсковая часть 44026, учёная степень кандидата технических наук • 2009 г.
- ✓ Python Institute, PCEP™ — Certified Entry-Level Python programmer • 2024 г.

Дополнительная информация

- Международный сертификат Python Institute PCEP • 2024 г.
- Сертификация "Разработка серверной части приложений PostgreSQL 16" (DEV-1) • 2025 г.
- Сертификация "Разработка серверной части приложений PostgreSQL 16" (DEV-2) • 2025 г.

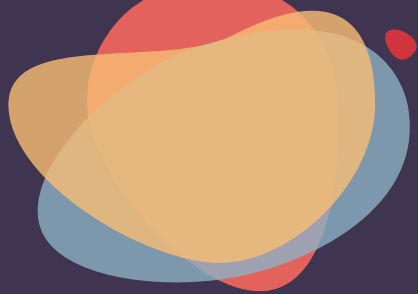


Программирование

Программирование — это процесс создания набора инструкций, которые компьютер может понять и выполнить.

Оно включает в себя написание кода с использованием языка программирования для выполнения компьютером определенных действий.

Программирование позволяет создавать программы, приложения и системы, автоматизировать процессы и разрабатывать новые технологии.

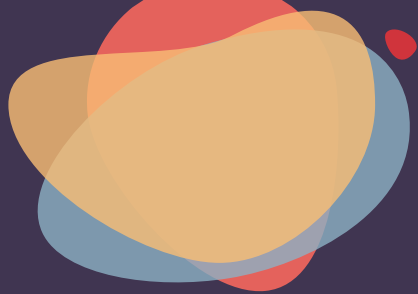


Python

Python читается как ПАЙТОН

Простой в изучении, имеет чистый и понятный синтаксис.

Использование: Широко применяется в веб-разработке, научных вычислениях, анализе данных и автоматизации.



Програма на Python

```
print("Hello, world!")
```

Програма на C++

```
#include <iostream>
```

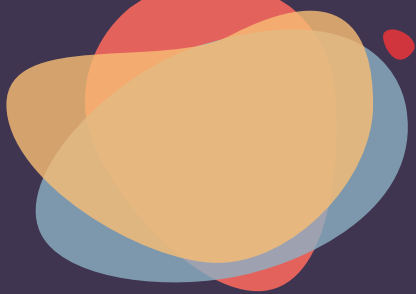
```
int main() {
```

```
    std::cout << "Hello, world!" <<
```

```
std::endl;
```

```
    return 0;
```

```
}
```



Сравнение языков

```
console.log("Hello, world!");
```

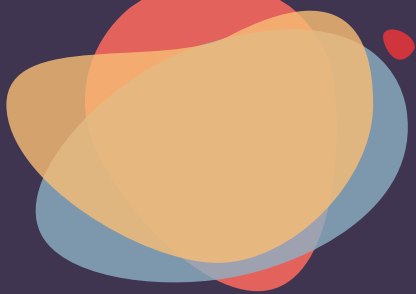
JavaScript

```
<?php  
echo "Hello, world!";  
?>
```

PHP

```
print("Hello, world!");
```

Swift



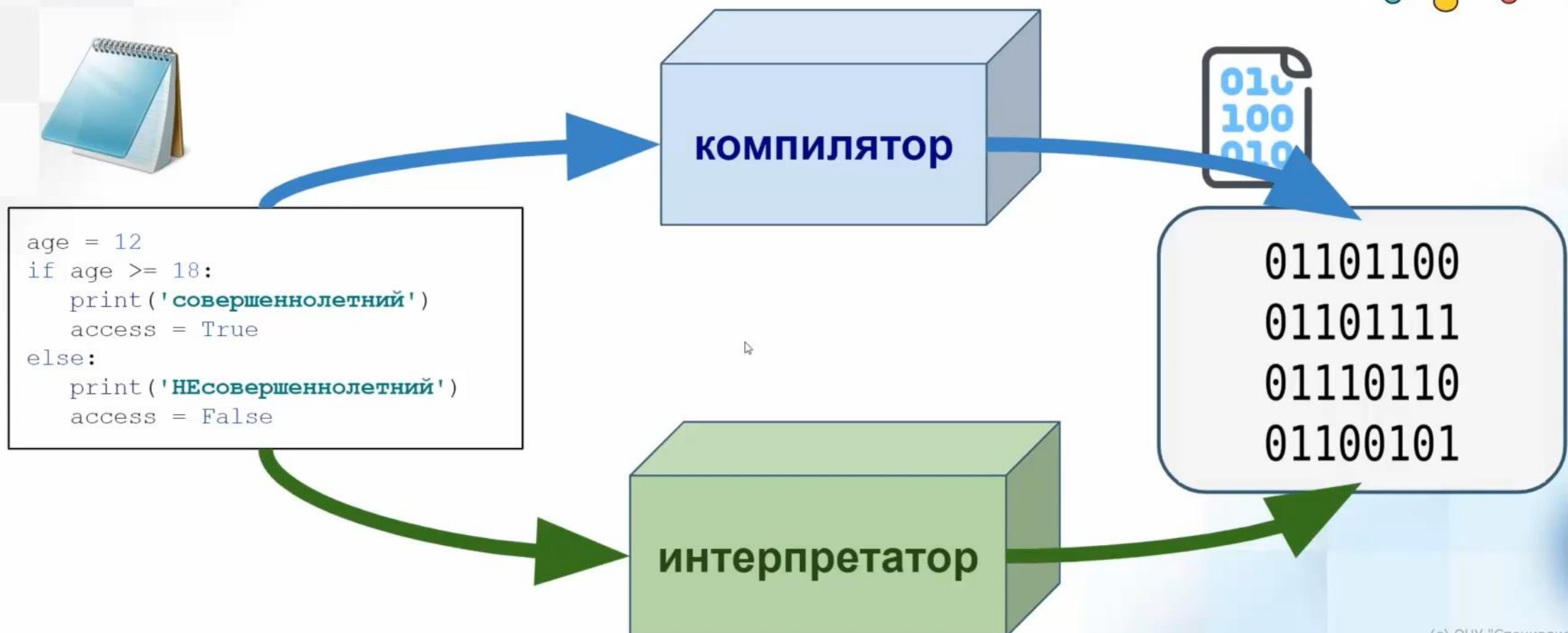
Интерпретатор

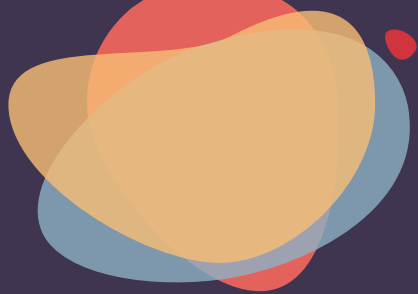
Интерпретатор превращает в машинный код не всю программу сразу, а построчно.

Программы на Python часто запускают с помощью интерпретатора.

Интерпретатор

Как выполняется программа?





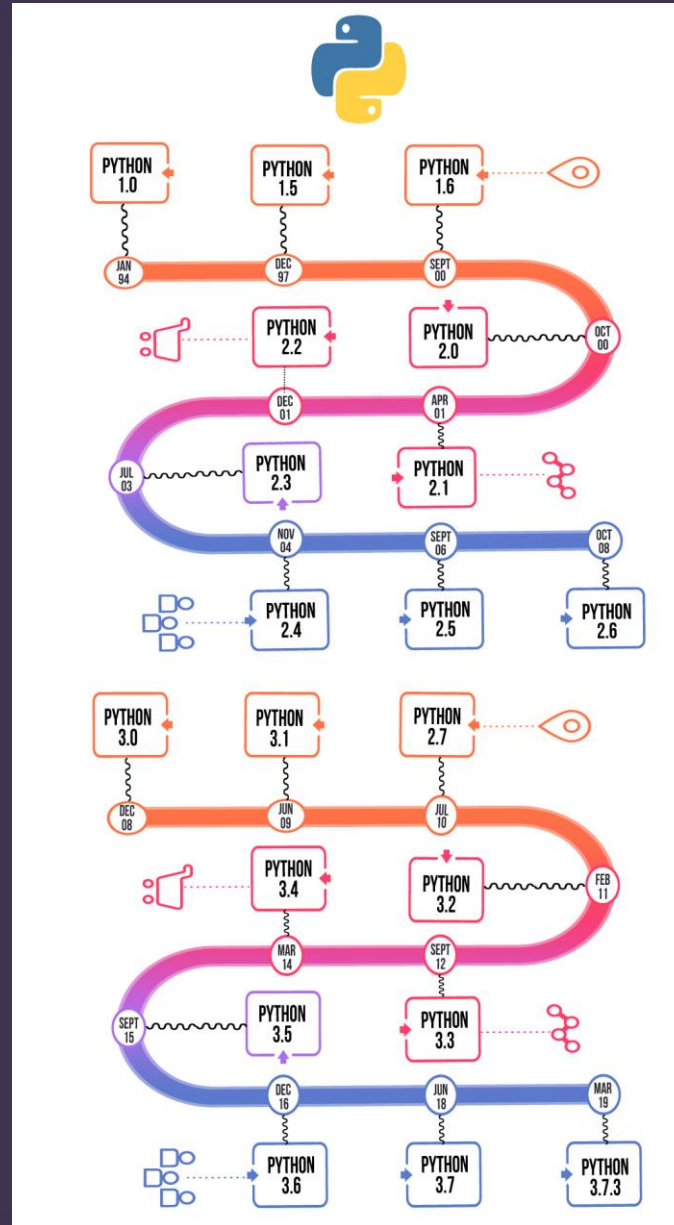
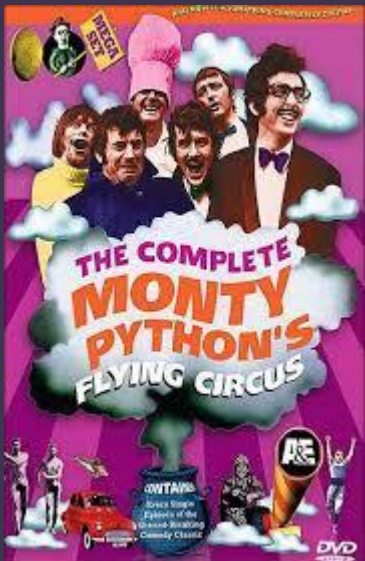
История Python

Язык программирования Python был создан в конце 1980-х и начале 1990-х годов Гвидо ван Россумом, нидерландским программистом. История его создания началась в декабре 1989 года, когда Гвидо ван Россум начал работать над новым языком программирования. Название "Python" происходит от уважения к комедийному шоу BBC под названием "Monty Python's Flying Circus".

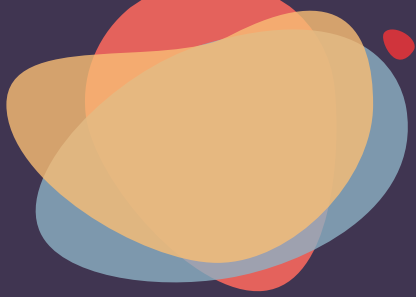
Ван Россум стремился создать язык, который был бы простым в изучении и чтении кода, одновременно обладал мощными возможностями для решения задач. Он хотел создать язык, который был бы эффективным как для профессиональных разработчиков, так и для новичков в программировании. Python вышел из первоначальной версии в феврале 1991 года. С тех пор он претерпел множество изменений и улучшений, став одним из самых популярных и востребованных языков программирования благодаря своей простоте, читаемости кода и разнообразным возможностям применения.



Создатель языка – Guido van Rossum



Гвидо ван Россум приступил к работе над Python в 80-х годах XX века, задумав его как потомка языка [ABC](#). Язык программирования был впервые опубликован в версии 0.9 в феврале 1991 г., после чего в мире зародилось сообщество его разработчиков. Версия 1.0 появилась в январе 1994 г, до 2.0 язык обновился в октябре 2000 г., а до 3.0 – в декабре 2008 г.



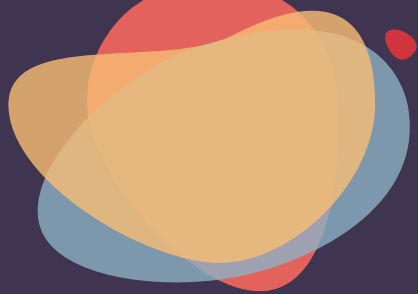
Применение Python

Анализ данных: Python используется для анализа больших объемов данных, создания статистических моделей, визуализации данных и машинного обучения.

Разработка веб-приложений: С помощью фреймворков, таких как Django и Fastapi, Python используется для создания веб-приложений, включая сайты, веб-сервисы и RESTful API.

Автоматизация задач: Python позволяет создавать скрипты для автоматизации рутинных задач, обработки файлов, управления данными и процессами.

Искусственный интеллект и машинное обучение: Python широко применяется в области искусственного интеллекта и машинного обучения для создания моделей, обучения и анализа данных.



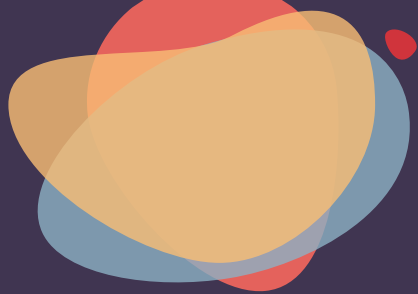
Применение Python

Научные и инженерные расчеты: Python используется для численных вычислений, моделирования, решения уравнений и создания графиков.

Разработка игр: С помощью библиотек, таких как Pygame, Python используется для создания игр и развлекательного программного обеспечения.

Обработка текста и регулярные выражения: Python может помочь в обработке и анализе текстовых данных с использованием мощных инструментов для работы с регулярными выражениями.

Скриптинг и администрирование: Python может быть использован для написания скриптов для автоматизации системного администрирования, управления серверами и сетевыми устройствами.



Кто использует Python

- Data Scientist / Аналитик данных
- Разработчик веб-приложений
- Инженер машинного обучения
- Инженер по автоматизации
- Системный администратор
- Разработчик игр
- Инженер по тестированию программного обеспечения
- Исследователь / Ученый
- Разработчик искусственного интеллекта
- Финансовый аналитик

Переменные

Способ получить доступ к информации



программа:

```
name = "Василий"
```

```
age = 32
```

```
rate = 12.5
```



память:

"Василий"

32

12.5

Оператор присваивания. Типизация данных

Ссылка на объект
(переменная)



Объект с числом



Объект со строкой

Оператор присваивания. Типизация данных

Ссылка на объект
(переменная)



var_a



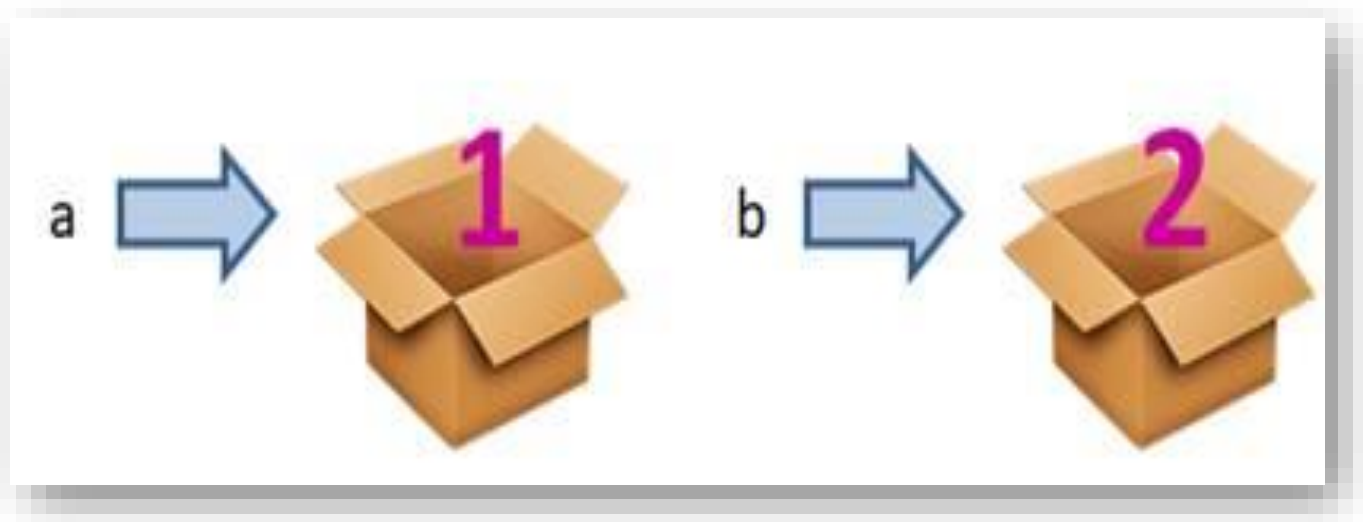
Оператор присваивания. Типизация данных



Каскадное присваивание



Оператор присваивания. Типизация данных



Множественное присваивание

Оператор присваивания. Типизация данных

Переменные

Динамическая типизация

```
a = 5
type(a)  # int
a = -3.5
type(a)  # float
a = "I'm string"
type(a)  # str
a = 10
type(a)  # снова int
```



Функция **type()** – определяет тип переменной

– комментарий, игнорируется интерпретатором

Оператор присваивания. Типизация данных

Правильные имена переменных

1. Имена следует брать существительными (отвечают на вопросы: кто, что).
2. Имена должны быть осмысленными и отражать суть данных.
3. Допустимые символы в именах: первый символ – любая буква латинского алфавита a-z, A-Z и символ подчеркивания `_`. В качестве второго и последующих символов еще цифры 0-9.

Оператор присваивания. Типизация данных

Переменные

Именованное

Python **регистрозависимый**: Name и name - разные переменные

bbb = "Иванов" - плохо

surname = "Иванов" - хорошо

В хорошей программе имя переменной говорит, что за данные хранятся в ней.

Оператор присваивания. Типизация данных

Переменные

Примеры именования

name

user_age

age

2to3

my name

my-name

my_name

myName

voзраст

print

ИМЯ

a~b

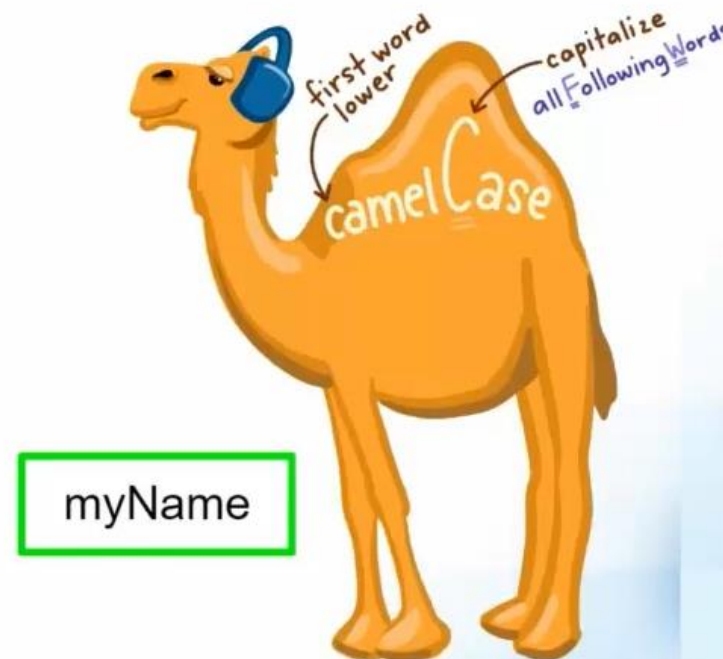
Оператор присваивания. Типизация данных

Примеры именования

name	age	2to3
my name	my-name	
vozrast	print	имя
		a~b



my_name	user_age
---------	----------



myName

Неизменяемые типы данных: int, float, str, bool

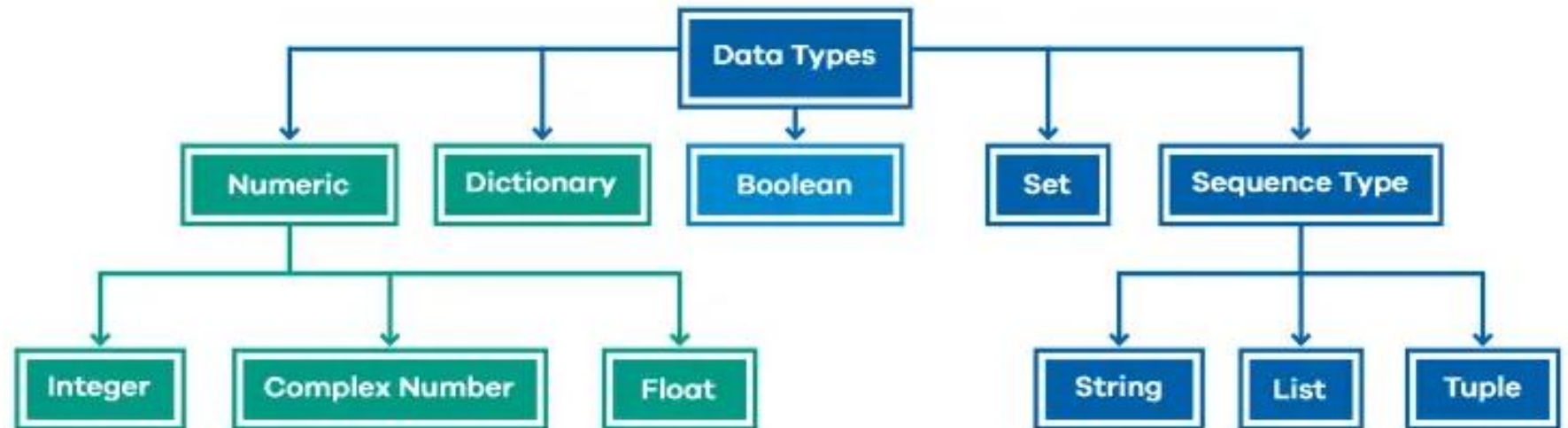
Правильные имена переменных

msg = "Сообщение"

count = 0

arg = 0

Arg = 0



1.2 Оператор присваивания. Типизация данных

Три базовых типа для представления чисел

- `int` – для целочисленных значений;
- `float` – для вещественных;
- `complex` – для комплексных

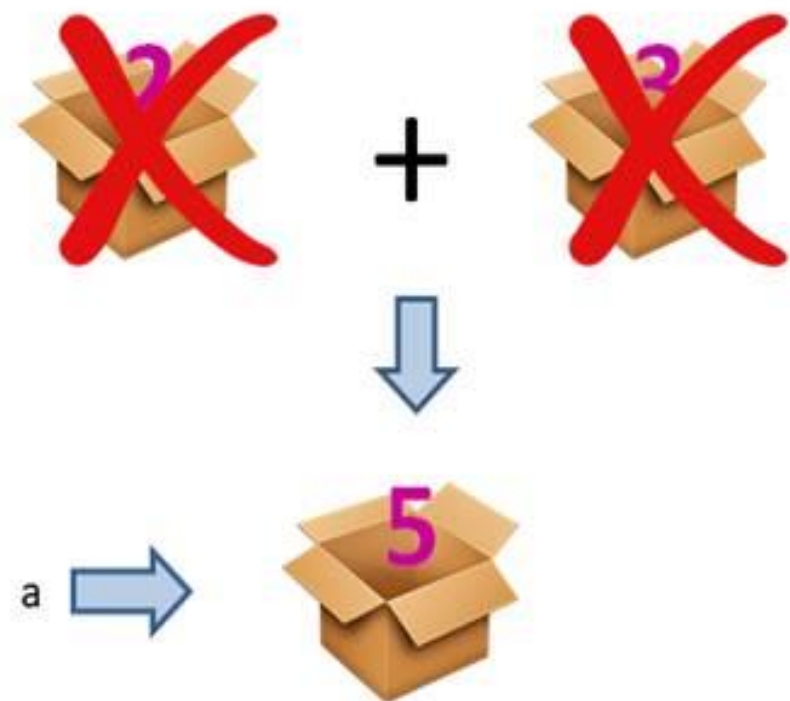
Тип	Длина на 32 bit	Длина на 64 bit
<code>int</code>	12	24
<code>float</code>	16	24
<code>str</code>	21+длина	37+длина (49*)
<code>tuple</code>	12+4*длина	24+8*длина
<code>list</code>	20+4*длина	40+8*длина
<code>Set/ frozenset</code>	(<=5 элементов) 100 (>5 элементов) 100+8*длина	(<=5 элементов) 200 (>5 элементов) 200+16*длина
<code>dict</code>	(<=5 элементов) 124 (>5 элементов) 124+12*длина	(<=5 элементов) 248 (>5 элементов) 248+24*длина

Математические операции

Оператор	Описание	Приоритет
+	сложение	2
-	вычитание	2
*	умножение	3
/, //	деление	3
%	остаток деления	3
**	возведение в степень	4

1.3 Математические операции

$$a = 2 + 3$$



Введение в строки. Ввод-вывод данных.
Преобразование строк в числа

```
text = "Я Python бы выучил только за то,  
что есть популярные курсы.  
Много хороших курсов!"
```



Спасибо за внимание!