

Performance Insights

[Get Started →](#)

Python для маркетинга: старт карьеры в дата- ориентированном мире

Вводный курс для будущих студентов программы профессиональной переподготовки "Основы программирования в медиакоммуникациях"

Цель этого вводного курса



Познакомить с основами

Разобраться в ключевых понятиях программирования и анализа данных, которые необходимы современному маркетологу



Python в маркетинге

Объяснить, как Python применяется в современном маркетинге и какие проблемы помогает решать



Базовые навыки

Дать необходимый минимум знаний для уверенного старта в основной программе переподготовки



Профориентация

Помочь понять, интересен ли вам путь дата-ориентированного маркетолога и аналитика

Структура курса

Курс рассчитан на 8 часов самостоятельного изучения и разделен на 4 логических модуля:

Модуль 1

Мир данных и Python

Знакомство с основными концепциями и принципами работы языка программирования

2 часа

Модуль 2

Основы программирования

Переменные, функции и простые программы для решения маркетинговых задач

2 часа

Модуль 3

Структуры данных

Как хранить и обрабатывать большие объемы маркетинговой информации

2 часа

Модуль 4

Библиотеки и итоги

Обзор инструментов для анализа данных и подготовка к основной программе

2 часа

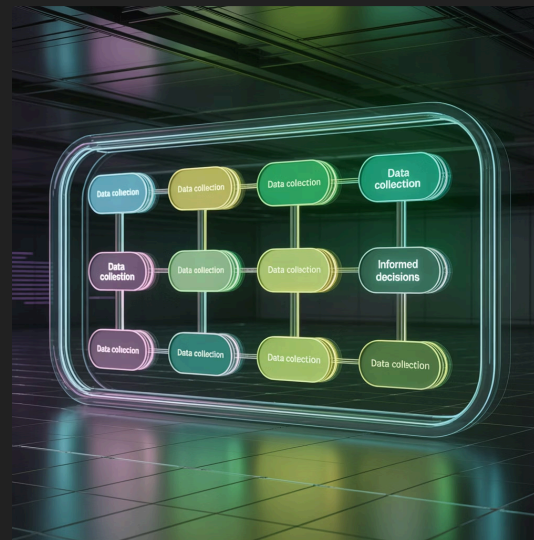


Data-Driven Маркетинг: почему данные — это новая нефть?

Data-driven маркетинг — это подход, при котором решения принимаются на основе **анализа данных**, а не интуиции или предположений.

Например:

- Распределение рекламного бюджета происходит не «на глазок», а по точным данным о конверсии каждого канала
- A/B-тестирование заголовков email-рассылки позволяет выбрать вариант с лучшим открытием
- Сегментация аудитории основана на реальном поведении пользователей, а не на догадках



Сбор данных

Получение информации из разных источников



Анализ

Выявление закономерностей и инсайтов



Решение

Обоснованные выводы на основе данных



Результат

Измеримое улучшение показателей

Что входит в основную программу?

После завершения вводного курса вас ждет погружение в более сложные темы:

1

Модуль 1: Основы кибербезопасности

Работа с персональными данными в маркетинге требует соблюдения строгих правил. Вы научитесь:

- Безопасно хранить данные клиентов
- Соблюдать требования законодательства (152-ФЗ, GDPR)
- Защищать данные от несанкционированного доступа

2

Модуль 2: Углубленное программирование на Python

Расширение базовых знаний и изучение продвинутых техник:

- Функции и модули для организации кода
- Работа с файлами различных форматов (CSV, Excel, JSON)
- Обработка ошибок и отладка программ

3

Модуль 3: Парсинг данных

Автоматизация сбора информации из различных источников:

- Извлечение данных с веб-сайтов
- Работа с API социальных сетей
- Этичный скрейпинг и соблюдение правил сайтов





Модуль 4: Анализ и визуализация данных

Превращение сырых данных в полезные инсайты с помощью мощных библиотек:

- Pandas для манипуляции данными
- Matplotlib и Seaborn для создания наглядных графиков
- Построение интерактивных дашбордов



Модуль 5: Автоматизация маркетинговых процессов

Создание инструментов для экономии времени и ресурсов:

- Разработка чат-ботов для коммуникации с клиентами
- Автоматическая генерация отчетов и рассылок
- Интеграция с CRM-системами и рекламными платформами



По окончании программы вы сможете автоматизировать рутинные задачи, анализировать большие объемы данных и создавать собственные инструменты для повышения эффективности маркетинга.

Почему именно Python?

Простой и читаемый синтаксис

Код Python напоминает английский язык и легко воспринимается даже новичками без технического бэкграунда

Огромное сообщество и библиотеки

Тысячи готовых библиотек для решения любых задач и активное сообщество, готовое помочь

Идеален для анализа данных

Стандарт де-факто в области обработки больших данных, машинного обучения и аналитики

Универсальность применения

От веб-разработки до искусственного интеллекта — один язык для множества задач



«Python стал незаменимым инструментом для современного маркетолога, который хочет принимать решения на основе данных, а не интуиции»

Ключевые понятия: Программа и Код

Что такое программа?

Программа — это последовательность инструкций для компьютера, которая решает определенную задачу.

Например, программа может:

- Анализировать эффективность рекламных кампаний
- Создавать персонализированные предложения для клиентов
- Автоматически публиковать посты в социальных сетях

Что такое код?

Код — это текст программы, написанный на языке программирования (в нашем случае Python).

```
# Пример кода на Python
campaign_name = "Летняя
распродажа"
budget = 50000
ctr = 2.5
```

```
print("Кампания:", campaign_name)
print("Бюджет:", budget, "руб.")
print("CTR:", ctr, "%")
```

Аналогия: Если представить программирование как приготовление блюда, то **код** — это рецепт, а **программа** — процесс готовки по этому рецепту.

Как начать? Установка Python

Шаг 1: Скачайте установщик

Посетите официальный сайт python.org и скачайте последнюю версию Python для вашей операционной системы.

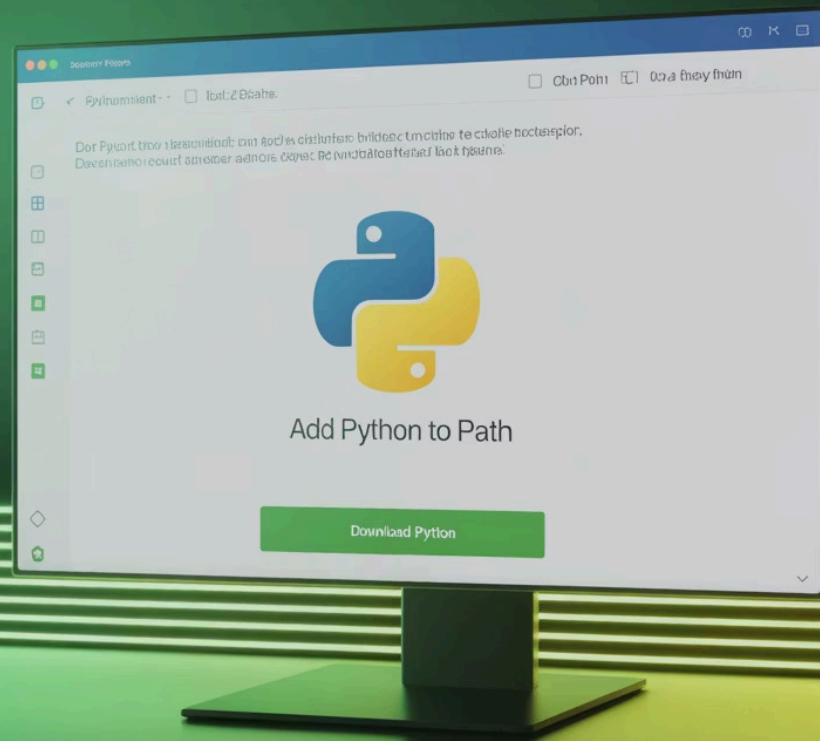
Шаг 2: Запустите установщик

Откройте загруженный файл и обязательно отметьте галочку «Add Python to PATH» — это важно для корректной работы.

Шаг 3: Завершите установку

Нажмите «Install Now» и дождитесь завершения процесса установки.
После этого Python готов к использованию!

После установки вы можете открыть программу IDLE (она устанавливается вместе с Python) для написания и запуска своего первого кода. Это простой редактор, который отлично подходит для начала работы.



Переменные: контейнеры для данных

Переменная — это именованный контейнер, в котором хранится значение. Представьте ее как коробку с ярлыком, где содержится какая-то информация.

```
# Создание переменных
brand_name = "Nike"
followers = 150000
engagement_rate = 3.5

# Использование переменных
print("Бренд:", brand_name)
print("Подписчиков:", followers)
print("Вовлеченность:", engagement_rate, "%")
```

Переменные позволяют хранить и обрабатывать данные, а также делают код более понятным и поддерживаемым.



Правила именования переменных:

- Могут содержать буквы, цифры и подчеркивания
- Начинаются с буквы или подчеркивания
- Чувствительны к регистру (name и Name — разные переменные)
- Лучше использовать осмысленные имена

Типы данных: текст и числа

Строки (str) Текстовые данные в кавычках <pre>name = "Анна" message = 'Добро пожаловать!' slogan = "Just Do It"</pre> Используются для: - Имен и названий - Сообщений и текстов - URL-адресов	Целые числа (int) Числа без дробной части <pre>followers = 5000 clicks = 327 year = 2023</pre> Используются для: - Подсчета количества - Индексов и ID - Целых значений	Дробные числа (float) Числа с десятичной точкой <pre>ctr = 2.7 conversion = 0.035 price = 999.99</pre> Используются для: - Процентов и долей - Денежных сумм - Точных измерений
--	--	--

Важно: с числами (int, float) можно производить математические операции, а со строками (str) — нельзя!

Тип данных	Пример	Можно делать	Нельзя делать
Строка (str)	"100"	Объединять с другими строками	Использовать в расчетах как число
Число (int, float)	100	Складывать, вычитать, умножать	Использовать напрямую как текст

Функция `print()`: наш первый помощник

`print()` — это встроенная функция Python, которая выводит информацию на экран. Это один из самых простых способов "общения" с компьютером.

```
# Простой вывод текста
print("Привет, мир!")

# Вывод переменных
clicks = 150
print("Количество кликов:", clicks)

# Вывод нескольких значений
campaign = "Черная пятница"
budget = 10000
print("Кампания:", campaign, "Бюджет:", budget)

# Вывод с форматированием
ctr = 3.7
print(f"CTR кампании: {ctr}%")
```

Функция `print()` незаменима для отладки программ и проверки промежуточных результатов.



Результат выполнения кода слева:

```
Привет, мир!
Количество кликов: 150
Кампания: Черная пятница Бюджет: 10000
CTR кампании: 3.7%
```


Функция `input()`: диалог с пользователем

`input()` — функция, которая позволяет получить данные от пользователя во время выполнения программы.

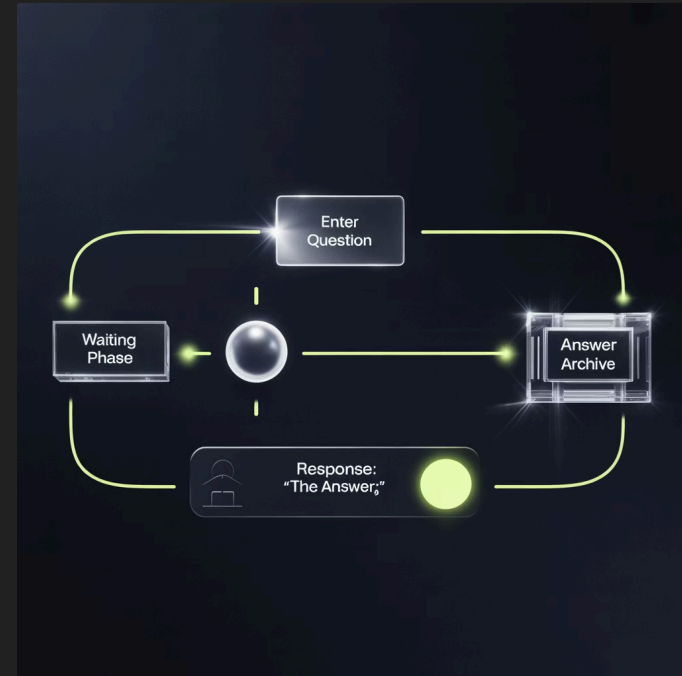
Принцип работы:

1. Программа выводит сообщение-приглашение
2. Останавливается и ждет ввода от пользователя
3. После нажатия Enter введенное значение сохраняется в переменную

```
# Запрос имени пользователя
user_name = input("Как вас зовут? ")

# Запрос числового значения
budget_str = input("Укажите бюджет: ")
budget = int(budget_str) # Преобразование строки в число

# Приветствие с использованием полученных данных
print(f"Здравствуйте, {user_name}!")
print(f"Ваш бюджет: {budget} рублей")
```



Важно помнить: функция `input()` всегда возвращает строку (`str`), даже если пользователь ввел число. Если вам нужно работать с числовым значением, необходимо преобразовать строку в число с помощью функций `int()` или `float()`.

```
# Неправильно (ошибка типов)
budget = input("Бюджет: ")
roi = budget * 0.2 # Ошибка!

# Правильно (с преобразованием)
budget = input("Бюджет: ")
roi = float(budget) * 0.2 # Работает!
```

Пример: простая программа-анкета

Объединим изученные функции в простую программу для сбора маркетинговой информации:

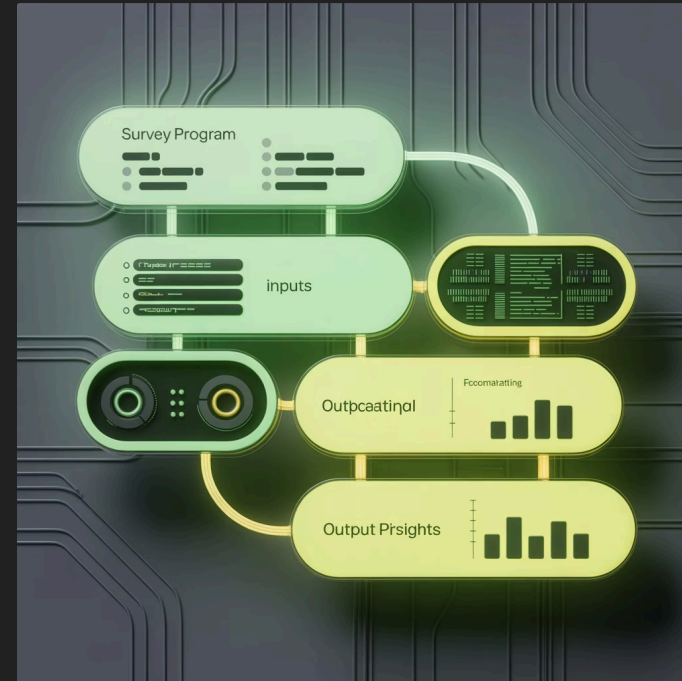
```
# Маркетинговая мини-анкета

# Сбор данных о пользователе
user_name = input("Как вас зовут? ")
age_str = input("Сколько вам лет? ")
favorite_brand = input("Ваш любимый бренд? ")

# Преобразование возраста в число
age = int(age_str)

# Персонализированное сообщение
print("\nСпасибо за участие в опросе!")
print(f"Привет, {user_name}!")
print(f"Вам {age} лет.")
print(f"Ваш любимый бренд — {favorite_brand}.")

# Дополнительная логика
if age < 25:
    print("Вы относитесь к молодой аудитории.")
else:
    print("Вы относитесь к основной аудитории.")
```



Результат выполнения программы:

```
Как вас зовут? Мария
Сколько вам лет? 28
Ваш любимый бренд? Zara

Спасибо за участие в опросе!
Привет, Мария!
Вам 28 лет.
Ваш любимый бренд — Zara.
Вы относитесь к основной аудитории.
```

Это простой пример того, как можно использовать Python для сбора и обработки данных в маркетинговых целях.

Структуры данных: как хранить много данных?

В маркетинге приходится работать с большими объемами информации. Для их эффективного хранения и обработки используются специальные структуры данных.

Список (List)

Упорядоченная коллекция элементов, доступ к которым осуществляется по индексу (номеру позиции).

Аналогия: список покупок, где важен порядок, но нет явных названий для каждого пункта.

```
# Список брендов
brands = ["Nike", "Adidas", "Puma", "Reebok"]

# Список цен
prices = [1990, 2490, 1790, 2990]
```

Словарь (Dictionary)

Коллекция пар "ключ-значение", где каждое значение доступно по уникальному ключу.

Аналогия: анкета, где у каждого поля есть название и значение.

```
# Словарь с данными о клиенте
customer = {
    "name": "Мария",
    "age": 28,
    "city": "Москва",
    "purchases": 5
}
```

Эти структуры данных позволяют эффективно организовать информацию и упрощают ее обработку в маркетинговых задачах.

Списки (Lists)

Список — это упорядоченная коллекция элементов, которые могут быть разных типов. Элементы списка нумеруются начиная с 0.

```
# Создание списка
channels = ["Instagram", "Facebook", "TikTok", "YouTube"]

# Доступ к элементам по индексу
print("Первый канал:", channels[0]) # Instagram
print("Последний канал:", channels[3]) # YouTube

# Изменение элемента
channels[1] = "Meta"
print(channels) # ['Instagram', 'Meta', 'TikTok', 'YouTube']

# Добавление нового элемента
channels.append("Telegram")
print(channels) # ['Instagram', 'Meta', 'TikTok', 'YouTube', 'Telegram']

# Удаление элемента
channels.remove("TikTok")
print(channels) # ['Instagram', 'Meta', 'YouTube', 'Telegram']

# Длина списка
print("Количество каналов:", len(channels)) # 4
```



Полезные операции со списками:

- **append()** — добавить элемент в конец
- **insert()** — вставить элемент по индексу
- **remove()** — удалить элемент по значению
- **pop()** — удалить элемент по индексу
- **sort()** — отсортировать список
- **len()** — получить длину списка

Словари (Dictionaries)

Словарь — это коллекция пар "ключ-значение", где каждый ключ уникален. Словари удобны для хранения данных, которые логически связаны между собой.

```
# Создание словаря
campaign = {
    "name": "Летняя распродажа",
    "budget": 150000,
    "channels": ["Instagram", "Facebook"],
    "conversion_rate": 2.7,
    "active": True
}

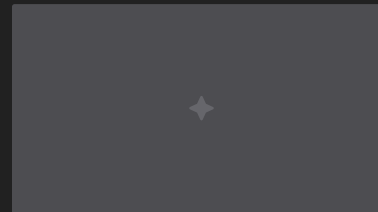
# Доступ к значениям по ключу
print("Название:", campaign["name"])
print("Бюджет:", campaign["budget"], "руб.")

# Изменение значения
campaign["budget"] = 200000

# Добавление новой пары ключ-значение
campaign["start_date"] = "2023-06-01"

# Удаление пары
del campaign["active"]

# Проверка наличия ключа
if "conversion_rate" in campaign:
    print("Конверсия:", campaign["conversion_rate"], "%")
```



Преимущества словарей:

- Быстрый доступ к данным по ключу
- Хранение разнородной информации
- Понятная структура данных
- Возможность вложенности (словари внутри словарей)

Словари особенно полезны при работе с API социальных сетей и аналитических систем, которые часто возвращают данные именно в таком формате.

Библиотеки: суперспособности Python

Библиотека — это набор готового кода, который можно использовать в своих программах для решения типичных задач.

Использование библиотек позволяет:

- Не "изобретать велосипед" и экономить время
- Использовать проверенные решения от экспертов
- Решать сложные задачи с минимальным кодом

```
# Пример использования библиотеки для работы с данными
import pandas as pd
```

```
# Загрузка данных из CSV-файла
data = pd.read_csv("marketing_data.csv")
```

```
# Быстрый анализ данных
print(data.head()) # первые 5 строк
print(data.describe()) # статистика по числовым колонкам
```



Ключевые библиотеки для маркетинга и аналитики:

Pandas

Анализ и обработка табличных данных

Matplotlib

Создание графиков и визуализаций

NumPy

Математические вычисления

Requests

Работа с API и веб-запросами

Pandas: мощь для работы с данными

Pandas — главная библиотека для работы с табличными данными в Python. Ее название происходит от "Panel Data" — панельные данные.

Основной объект Pandas — **DataFrame** (таблица с строками и столбцами), аналог Excel-таблицы, но с более мощными возможностями обработки.

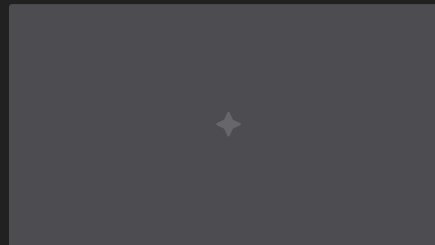
```
import pandas as pd

# Создание DataFrame из словаря
data = {
    'campaign': ['FB_summer', 'IG_promo', 'YT_brand'],
    'budget': [50000, 75000, 100000],
    'clicks': [1200, 3400, 890],
    'conversions': [45, 82, 12]
}

df = pd.DataFrame(data)

# Расчет CTR для каждой кампании
df['ctr'] = df['clicks'] / df['impressions'] * 100

# Фильтрация данных
successful = df[df['conversions'] > 50]
print(successful)
```



Что можно делать с Pandas:

- Загружать данные из CSV, Excel, JSON
- Фильтровать и сортировать данные
- Объединять таблицы (как в SQL)
- Обрабатывать пропущенные значения
- Группировать и агрегировать данные
- Проводить статистический анализ
- Визуализировать результаты

Pandas особенно полезен для анализа результатов рекламных кампаний, поведения пользователей и других маркетинговых метрик.

Matplotlib: визуализация данных



Линейные графики

Идеальны для отображения трендов и динамики показателей во времени.

Примеры использования:

- Динамика роста подписчиков
- Изменение CTR кампании по дням
- Сезонные колебания продаж



Столбчатые диаграммы

Отлично подходят для сравнения категорий или каналов между собой.

Примеры использования:

- Сравнение эффективности каналов
- Распределение бюджета по кампаниям
- Конверсия по разным целевым страницам



Круговые диаграммы

Наглядно показывают доли в общей структуре или распределение.

Примеры использования:

- Источники трафика на сайт
- Возрастная структура аудитории
- Распределение бюджета по каналам

```
import matplotlib.pyplot as plt

# Данные для графика
months = ['Янв', 'Фев', 'Март', 'Апр', 'Май']
sales = [120, 135, 170, 190, 210]

# Создание линейного графика
plt.figure(figsize=(10, 6))
plt.plot(months, sales, marker='o', linewidth=2)
plt.title('Динамика продаж')
plt.xlabel('Месяц')
plt.ylabel('Продажи, тыс. руб.')
plt.grid(True)
plt.savefig('sales_chart.png')
plt.show()
```

Другие полезные библиотеки



NumPy

Библиотека для научных вычислений и работы с многомерными массивами данных.

Применение в маркетинге:

- Расчет сложных метрик и KPI
- Статистический анализ кампаний
- Быстрая обработка больших наборов данных



Seaborn

Библиотека для создания статистических графиков на основе Matplotlib с более красивым дизайном.

Применение в маркетинге:

- Создание профессиональных отчетов
- Визуализация распределений и корреляций
- Построение тепловых карт взаимосвязей



Requests

Библиотека для отправки HTTP-запросов и работы с API различных сервисов.

Применение в маркетинге:

- Получение данных из Facebook Ads API
- Интеграция с аналитическими сервисами
- Парсинг данных с веб-сайтов



Streamlit

Библиотека для создания интерактивных веб-приложений и дашбордов с минимальным кодом.

Применение в маркетинге:

- Создание маркетинговых дашбордов
- Интерактивные отчеты для клиентов
- Инструменты для анализа данных без навыков фронтенд-разработки

Как устроен процесс работы с данными?

Сбор данных

Получение информации из различных источников:

- Парсинг веб-сайтов
- API социальных сетей и рекламных кабинетов
- Опросы и формы
- CRM-системы и базы данных

Отчетность

Формирование выводов и рекомендаций:

- Выявление инсайтов
- Формулирование гипотез
- Разработка стратегии
- Принятие решений на основе данных



Очистка данных

Подготовка данных к анализу:

- Обработка пропущенных значений
- Удаление дубликатов
- Исправление ошибок и аномалий
- Стандартизация форматов

Анализ

Извлечение полезной информации:

- Расчет ключевых метрик
- Поиск закономерностей
- Сегментация данных
- Статистические тесты

Визуализация

Наглядное представление результатов:

- Построение графиков и диаграмм
- Создание дашбордов
- Интерактивные отчеты
- Инфографика

Python с его библиотеками предоставляет инструменты для каждого этапа этого процесса, позволяя маркетологам работать с данными на профессиональном уровне.

Как эффективно учиться?

→ Экспериментируйте с кодом

Не бойтесь изменять примеры, играть с параметрами и даже ломать программы — это лучший способ понять, как они работают

→ Используйте официальную документацию

Документация библиотек — ценный ресурс с подробными описаниями и примерами использования всех функций

→ Разбирайте чужой код

Анализ готовых решений помогает понять лучшие практики и расширить свой арсенал приемов программирования

→ Применяйте знания на практике

Решайте маленькие, но реальные задачи из вашей профессиональной деятельности — это закрепит навыки и принесет пользу

→ Понимайте логику, а не заучивайте код

Важнее понять принципы работы программы, чем запомнить конкретный синтаксис — его всегда можно найти в справочнике



«Программирование — это не про запоминание, а про решение проблем и логическое мышление. Не бойтесь ошибок — они ваши лучшие учителя»

Помните: каждый программист, независимо от уровня, постоянно ищет информацию и решения в интернете. Это нормальная практика!



Что ждет вас в основной программе?

После завершения этого вводного курса вы продолжите свое путешествие в мир Python для маркетинга и аналитики. Вас ждет:

1

Безопасный парсинг данных

Вы научитесь получать информацию из различных источников с соблюдением правовых и этических норм, узнаете, как обходить ограничения и избегать блокировок.

2

Создание сложных программ

Освоите функции, модули и объектно-ориентированное программирование для разработки масштабируемых решений для маркетинговых задач.

3

Интерактивные дашборды

Научитесь создавать красивые и функциональные веб-интерфейсы для визуализации данных, которые можно показать руководству или клиентам.

4

Автоматизация рутины

Сможете настроить автоматический сбор данных, генерацию отчетов и выполнение других повторяющихся задач без вашего участия.

Главное — вы уже сделали первый и самый важный шаг на этом пути!

Материалы для самостоятельного изучения

Справочные материалы (англ.)

- [Официальная документация Python](#) — полное руководство по языку
- [Документация Pandas](#) — подробное описание функций для анализа данных
- [Документация Matplotlib](#) — справочник по созданию визуализаций

Русскоязычные ресурсы

- [Python World](#) — обучающие материалы на русском языке
- [Хабр: Python](#) — статьи и tutorиалы от сообщества
- [Stepik: Программирование на Python](#) — популярный бесплатный курс



Сообщества для вопросов

- [Stack Overflow](#) — крупнейший ресурс с вопросами и ответами по программированию
- [Reddit: Learn Python](#) — форум для начинающих
- Телеграм-каналы и чаты по Python и анализу данных

Практика

- [Kaggle](#) — платформа с наборами данных и соревнованиями
- [Google Colab](#) — бесплатная среда для экспериментов с Python