

СТУДЕНТЫ

15-55 лет

top

КОМПЬЮТЕРНАЯ
АКАДЕМИЯ



учёба с нуля
и бесплатно



диплом о проф.
переподготовке



трудоустройство
во время обучения



живая помощь
от преподавателей



платформа
с упражнениями



обновленная
программа 2025

Разработка web-проектов

на PYTHON

736 академических часов

368 часов аудиторных занятий

1 год длительность курса

368 часов самостоятельной работы

Наши партнёры:

Yandex Cloud

hh карьерный маркетплейс

1 модуль: Основы Python

Язык/библиотеки:

- Python 3.12+.
- Pytest.
- Pydocstyle.

Концепции:

- ООП (наследование, инкапсуляция, полиморфизм, абстракция).
- Итераторы/генераторы, декораторы, менеджеры контекста.
- Магические методы (`__str__`, `__repr__`).

Чему научишься:

- Программировать на Python как ниндзя: от переменных до сложных структур.
- Понимать ООП (классы, наследование, инкапсуляция) — это основа для Django/FastAPI.
- Работать с данными: списки, словари, файлы (JSON/CSV).
- Использовать «магию» Python: декораторы (как улучшители функций), генераторы (для работы с большими данными без нагрузки на память).
- Тестировать код автоматически (pytest) и писать понятные описания (pydocstyle).

Проще: Освоишь базу Python — научишься создавать программы, которые делают что угодно: от калькулятора до парсера сайтов.

3 модуль: Git, SQL, ООП, API

Системы:

- Git.
- GitHub/GitLab.

Базы данных:

- PostgreSQL.
- SQLite.

Библиотеки:

- SQLAlchemy (ORM), requests, BeautifulSoup/Scrapy.
- Typing (аннотации).

Чему научишься:

- Работать в команде: Git/GitHub (сохранять версии кода, сливать изменения).
- Общаться с базами данных: SQL (писать запросы), ORM (SQLAlchemy — чтобы Python-код сам управлял БД).
- Парсить сайты (BeautifulSoup/Scrapy) и работать с API (requests — например, получать погоду с сервера).
- Типизировать код (чтобы избежать ошибок).

Проще: Научишься создавать «мозги» приложений — работать с данными, внешними сервисами и командой.

5 модуль: JavaScript и TypeScript

Библиотеки:

- ES6+ (Promises, async/await).
- TypeScript (интерфейсы, дженерики).

Инструменты:

- ESLint.
- Webpack.

Чему научишься:

- «Оживлять» сайты: клики, анимации, загрузка данных без перезагрузки страницы.
- Писать надежный код с TypeScript (типы, интерфейсы — как «контроль качества» для JS).
- Настраивать сборку проекта (Webpack) и проверять код (ESLint).

Проще: Добавишь сайтам интерактивность и логику — например, сделаешь динамическую корзину покупок.



2 модуль: Продвинутый Python

Библиотеки:

- Asyncio, threading, multiprocessing.
- Logging, re (регулярные выражения).
- Datetime, collections (deque, Counter).
- Json, csv, unittest/pytest.

Инструменты:

- Pip.
- Venv.

Чему научишься:

- Делать код быстрым: асинхронность (asyncio), потоки (threading), процессы (multiprocessing).
- Обрабатывать сложные данные: регулярные выражения (поиск шаблонов в тексте), коллекции (Counter для подсчета элементов).
- Логировать события (чтобы программа «рассказывала», что происходит).
- Работать с виртуальными окружениями (venv) и пакетами (pip) — чтобы проекты не конфликтовали.

Проще: Ты прокачаешь Python до уровня профессионала — сможешь писать высокоэффективные приложения.

4 модуль: HTML и CSS

Технологии:

- HTML5 (семантика, формы).
- CSS3 (Flexbox, Grid, анимации).
- Препроцессоры: SCSS, Pug.

Инструменты:

- PostgreSQL.
- SQLite.

Чему научишься:

- Верстать страницы с нуля (кнопки, формы, анимации).
- Делать адаптивные сайты (чтобы выглядело идеально на телефоне и ПК).
- Использовать супер-инструменты:**
- Flexbox/Grid — для сложных макетов.
- SCSS/Pug — чтобы писать CSS/HTML быстрее.
- Webpack/Vite — для сборки проекта.

Проще: Сможешь визуализировать любую идею — создавать красивые и удобные интерфейсы.

6 модуль: React

Библиотеки:

- React 18 (хуки, Context API).
- React Router, Zustand, React Query.
- React Intl, Immer.

Инструменты:

- Vite.
- SWC.

Чему научишься:

- Создавать SPA-приложения (как Gmail — без перезагрузок).
- Управлять состоянием (useState, Zustand — например, счетчик товаров в корзине).
- Работать с данными (React Query — загрузка с сервера).
- Строить маршруты (React Router — переходы между страницами).
- Локализовать приложения (React Intl — поддержка языков).

Проще: Научишься делать крутые фронтенд-приложения уровня современных сервисов.

7 модуль: FastAPI + Docker

Фреймворки:

- FastAPI.
- Jinja2.

Базы данных:

- SQLAlchemy.
- Alembic.



Библиотеки:

- Docker, Docker Compose.
- Redis (кеширование).

Чему научишься:

- Создавать API на FastAPI (молниеносные серверы).
- Работать с БД через SQLAlchemy.
- Упаковывать приложения в Docker (чтобы работало где угодно).
- Ускорять сайты кешированием (Redis).

Проще: Построишь бэкэнд для React-приложений — быстрый, надежный и готовый к масштабированию.

8 модуль: Django (продвинутый)

Фреймворки:

- Django 5+.
- DRF (Django REST Framework).

Библиотеки:

- Django ORM, Django Channels.
- Celery (асинхронные задачи).

Библиотеки:

- Pytest-django.

Чему научишься:

- Создавать монолиты на Django (как Instagram).
- Делать REST API (DRF).
- Работать с асинхронными задачами (Celery — например, отправка писем).
- Тестировать всё (pytest-django).

Проще: Освоишь фреймворк для сложных проектов — с админкой, пользователями и фоновыми процессами.

Применение нейросетей в контексте курса:

API Нейросетевых сервисов:

- Интеграция через REST API (OpenAI API, Anthropic Claude API, Google Gemini API, YandexGPT API, GigaChat и т.д.).

Использование в Python/django:

- Генерация текстового контента (описания, комментарии, документация).
- Анализ настроений (Sentiment Analysis) отзывов или комментариев.

Чему научишься:

- Интегрировать ChatGPT/GigaChat в Python/Django через API.
- Генерировать контент: описания товаров, комментарии, документацию.
- Анализировать настройки (например, определять позитивные/негативные отзывы).

Проще: Добавишь «искусственный интеллект» в свои проекты — сделаешь их умными и современными.

Кем может работать выпускник данного курса:

Python Backend Developer (Junior/Middle)

Почему: Глубокое знание Python, Django, FastAPI, ORM, асинхронности, REST API и работы с БД (PostgreSQL/SQLite) — основа для 80% вакансий в backend.

Fullstack Developer (Python + React)

Почему: Полный стек технологий: от верстки (HTML/CSS/SCSS) и React-логики до серверных фреймворков и Docker. Может создавать продукты «под ключ».

React Frontend Developer (TypeScript)

Почему: Продвинутое владение React (хуки, Zustand, React Query), TypeScript и адаптивной версткой. Востребовано в продуктовых командах.

AI Solutions Engineer

Уникальное преимущество: Умение интегрировать нейросети в реальные продукты (чат-боты, генерация контента, оптимизация UI) через OpenAI/ Gemini API. Редкий навык на рынке.

Data Engineer / ETL-разработчик (начальный уровень)

Почему: Навыки парсинга, работы с API, SQL, JSON/CSV и Python-скриптами — база для построения data-пайплайнов.

Почему курс актуален?

Python доминирует в ключевых сферах:

- Web: Django/FastAPI — флагманы для стартапов и enterprise (у 45% вакансий Python-backend).
- AI/Data: Python — язык №1 для нейросетей (TensorFlow/PyTorch) и анализа данных.
- *По данным Stack Overflow 2024: Python в топ-3 востребованных языков.*

Fullstack-универсалы ценятся выше:

- Способность закрыть и фронтенд (React), и бэкэнд (Python) снижает затраты на разработку.

Пример: Стартапы ищут разработчиков, которые могут создать MVP без привлечения 5 специалистов.

Современный инструментарий:

- Frontend: React 18+ (Zustand, React Query), TypeScript, Vite.
- Backend: FastAPI (для микросервисов), Django (для монолитов), асинхронность.
- Infra: Docker, Redis — стандарт для облачных развертываний (AWS/Azure).



17

лет на рынке



4,8

рейтинг
на Я. Картах



500+

филиалов в России



2 800

преподавателей



150 000

студентов



24 000+

ОТЗЫВОВ



Также доступно
онлайн-обучение

*данные действительны
на июль 2025 года