Государственное общеобразовательное учреждение

«Забайкальский краевой лицей-интернат»

ИТОГОВЫЙ

ИНДИВИДУАЛЬНЫЙ

ПРОЕКТ

Разработка сайта по изучению языка программирования python

Выполнил ученик 11 «Б»

Быков Иван Владимирович

Научный руководитель

Хаустова Мария Вадимовна

учитель информатики

Чита

2024

**Содержание проекта**

Титульный лист

**Содержание проекта**

[Паспорт проекта 2](#_Toc175997947)

[Введение 4](#_Toc175997948)

[Глава I. Основы разработки сайтов 6](#_Toc175997949)

[1.1 История развития python и его особенности 6](#_Toc175997950)

[1.2 Основа разработки сайта (HTML) 8](#_Toc175997951)

[1.3 JavaScript, CSS 9](#_Toc175997952)

[1.4 Базы данных, Bootstrap, Django 11](#_Toc175997953)

[2.5 Хостинг 15](#_Toc175997954)

[Глава II. Создание сайта 17](#_Toc175997955)

[2.1 Выбор основных инструментов для создания сайта и их подключение 17](#_Toc175997956)

[2.2 Разработка структуры, макета и дизайна сайта 19](#_Toc175997957)

[2.3 Разработка внутренних механизмов сайта, наполнение его контентом 20](#_Toc175997958)

[2.4 Разработка курса 21](#_Toc175997959)

[2.5 Размещение сайта в сети интернет 22](#_Toc175997960)

[Заключение 23](#_Toc175997961)

[Список интернет-ресурсов 24](#_Toc175997962)

[Приложение 1. Макет сайта 25](#_Toc175997963)

[Приложение 2. Дизайн сайта 27](#_Toc175997964)

# **Паспорт проекта**

|  |  |  |
| --- | --- | --- |
| № П/П | Наименование | Описание |
| 1 | Название проекта | Разработка сайта по изучению языка программирования python |
| 2 | Автор проекта | Быков Иван Владимирович |
| 3 | Аннотация проекта | Проект направлен на создание сайта, который будет помогать людям начать изучать программирование на основе языка Python |
| 4 | Проблема, на решение которой направлен проект | Незнание школьниками языка программирования python, который является основным для решения задач по информатике школьного курса |
| 5 | Цели проекта | Создание обучающего сайта для любых желающих с материалами для изучения программирования |
| 6 | Задачи проекта | Разработать макет и структуру сайта  Разработать внутренние механизмы сайта  Разработать программу для обучения программированию  Загрузить сайт в информационную сеть интернет |
| 7 | Сроки реализации проекта | |  |  | | --- | --- | | Сроки | Мероприятия | | Подготовительный этап | | | 01.11.2023-01.02.2024 | Изучение теоретического материала | | Основной этап | | | 01.03.2023-01.08.2024 | Разработка сайта  Разработка обучающих материалов | | Аналитический этап | | | 01.08.2023-01.09.2024 | Размещение сайта в сети интернет проверка его работоспособности | |
| 8 | Ресурсное обеспечение проекта | ресурсы для реализации проекта: персональный компьютер, электронные документации, находящиеся в открытом доступе |
| 9 | Источники и объем финансирования | отсутствуют |
| 10 | Ожидаемые результаты проекта | Функционирующий сайт с обучающими материалами |

Перспективы развития проекта

Перспектива заключается в добавление новых функций на сайт таких как форум, статьи от пользователей, сохранение прогресса пользователей. Так же перспектива заключается в расширении образовательной информации на сайте и добавление материалов по другим языкам программирования.

# **Введение**

Язык программирования python является одним из самых популярных языков программирования (по данным Tiobe). Это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

*Актуальность* выбора данной темы заключается в том, что разработка данного сайта станет отличным помощником в изучении языка программирования Python.

*Проблема*: незнание школьниками языка программирования python, который является основным для решения задач по информатике школьного курса

*Цель*: создание обучающего сайта «LPIE» (learning python is easy) по изучении языка программирования Python.

*Задачи*:

1. Изучение теоретического материала по разделу информатики «разработка сайтов»

2. Определение среды для разработки по критериям практичности и удобства освоения

3. Разработка шаблонов сайта и его дизайна

4. Разработка внутренней системы сайта, наполнение его пассивным и интерактивным контентом

5. Размещение сайта в сети интернет.

*Объект*: разработка сайтов

*Предмет*: язык программирования Python.

*Методы*: Анализ и синтез, моделирование

*Целевая аудитория*: учащиеся 11 «Б» класса ГОУ«ЗабКЛИ»

*Итоговый продукт*: обучающий сайт «LPIE»

# **Глава I. Основы разработки сайтов**

## 1.1 История развития python и его особенности

Задумка по реализации языка появилась в конце 1980-х годов, а разработка его реализации началась в 1989 году сотрудником голландского института CWI Гвидо ван Россумом. Для распределённой операционной системы Amoeba требовался расширяемый скриптовый язык, и Гвидо начал разрабатывать Python на досуге, позаимствовав некоторые наработки для языка ABC (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале 1991 года Гвидо опубликовал исходный текст в группе новостей alt.sources. С самого начала Python проектировался как объектно-ориентированный язык.

Гвидо ван Россум назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона», поскольку автор был поклонником этого телешоу, как и многие другие разработчики того времени, а в самом шоу прослеживалась некая параллель с миром компьютерной техники.

Наличие дружелюбного, отзывчивого сообщества пользователей считается, наряду с дизайнерской интуицией Гвидо, одним из факторов успеха Python. Развитие языка происходит согласно чётко регламентированному процессу создания, обсуждения, отбора и реализации документов PEP (англ. *Python Enhancement Proposal*) — предложений по развитию Python.

3 декабря 2008 года, после длительного тестирования, вышла первая версия Python 3000 (или Python 3.0, также используется сокращение Py3k). В Python 3 устранены многие недостатки архитектуры с максимально возможным (но не полным) сохранением совместимости со старыми версиями Python.

Язык использует динамическую типизацию вместе с подсчётом ссылок и циклический сборщик мусора для менеджмента памяти. Также есть динамические разрешения имён (динамическое связывание), которые связывают имена методов и переменных во время выполнения программы.

Python предлагает поддержку функционального программирования в традициях Лиспа. Так, в Python есть функции filter, map и reduce. Так же были заимствованы понятия характеристик списков, ассоциативных массивов (словарей), множеств и генераторов списков.

Вместо того, чтобы встроить в ядро Python всю функциональность языка, он был спроектирован таким образом, чтобы быть легко расширяемым. Это сделало язык популярным средством добавления программируемых интерфейсов к существующим приложениям. Видение Гвидо ван Россума маленького ядра с большой стандартной библиотекой и легко расширяемым интерпретатором проистекало из негативного опыта разработки языка ABC, который придерживался противоположного подхода.

Python стремится к более простому, менее громоздкому синтаксису и грамматике, предоставляя разработчикам выбор в их методологии кодирования. Python придерживается философии «должен существовать один — и, желательно, только один — очевидный способ сделать это».

Важная цель разработчиков Python — делать его забавным для использования. Это было отражено в названии языка, данном в честь Монти Пайтона. Также это отражено в иногда игривом подходе к обучающим программам и справочным материалам, таким как примеры программ из документаций, которые используют названия spam и eggs вместо использующихся в документации множества других языков foo и bar.

## 1.2 Основа разработки сайта (HTML)

Разработка сайта (Веб-разработка) — процесс создания веб-сайта или веб-приложения. Основными этапами процесса являются веб-дизайн, вёрстка страниц, программирование на стороне клиента и сервера, а также конфигурирование веб-сервера.

Язык гипертекстовой разметки HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки.

Первым общедоступным описанием HTML был документ «Теги HTML», впервые упомянутый в Интернете Тимом Бернерсом-Ли в конце 1991 года. В нём описываются 18 элементов, составляющих первоначальный, относительно простой дизайн HTML.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью. Однако современное применение HTML очень далеко от его изначальной задачи. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

Текстовые документы, содержащие разметку на языке HTML, обрабатываются браузерами, которые отображают документ в его форматированном виде. Такие приложения, обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер.

## 1.3 JavaScript, CSS

JavaScript, часто сокращенно JS, — это язык программирования, который является одной из основных технологий Всемирной паутины, наряду с HTML и CSS. По состоянию на 2023 год 98,7% веб-сайтов используют JavaScript на стороне клиента для управления поведением веб-страницы, часто включая сторонние библиотеки. Все основные веб-браузеры имеют специальный движок JavaScript для выполнения кода на устройствах пользователей.

JavaScript — это высокоуровневый, соответствующий стандарту ECMAScript. Он имеет динамическую типизацию, объектную ориентацию на основе прототипов и первоклассные функции. Он мультипарадигмальный, поддерживает событийно-ориентированный, функциональный и императивный стили программирования. Он имеет интерфейсы прикладного программирования (API) для работы с текстом, датами, регулярными выражениями, стандартными структурами данных и объектной моделью документа (DOM).

Стандарт ECMAScript не включает в себя какие-либо средства ввода-вывода (I/O), такие как сетевые средства, хранилища или графические средства. На практике веб-браузер или другая система выполнения предоставляет API-интерфейсы JavaScript для ввода-вывода.

CSS (Cascading Style Sheets «каскадные таблицы стилей») — формальный язык декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки. Также может применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана) или при выводе устройствами, использующими шрифт Брайля.

## 1.4 Базы данных, Bootstrap, Django

В вычислительной технике база данных — это организованный набор данных или тип хранилища данных, основанный на использовании системы управления базами данных (СУБД), программного обеспечения, которое взаимодействует с конечными пользователями, приложениями и самой базой данных для сбора и анализа данных. СУБД дополнительно включает в себя основные средства, предназначенные для администрирования базы данных. Совокупность базы данных, СУБД и связанных с ней приложений можно назвать системой баз данных. Часто термин «база данных» также широко используется для обозначения любой СУБД, системы баз данных или приложения, связанного с базой данных.

Небольшие базы данных могут храниться в файловой системе, тогда как большие базы данных размещаются в компьютерных кластерах или облачных хранилищах. Проектирование баз данных охватывает формальные методы и практические соображения, включая моделирование данных, эффективное представление и хранение данных, языки запросов, безопасность и конфиденциальность конфиденциальных данных, а также вопросы распределенных вычислений, включая поддержку одновременного доступа и отказоустойчивости.

Реляционные базы данных стали доминирующими в 1980-х годах. Они моделируют данные в виде строк и столбцов в ряде таблиц, и подавляющее большинство из них используют SQL для записи и запроса данных. В 2000-х годах стали популярны нереляционные базы данных, называемые NoSQL, поскольку они используют разные языки запросов.

Bootstrap — свободный набор инструментов для создания [сайтов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) и [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Включает в себя [HTML](https://ru.wikipedia.org/wiki/HTML) и [CSS](https://ru.wikipedia.org/wiki/CSS)-шаблоны оформления для [типографики](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)-расширения.

Bootstrap начала разрабатываться как внутренняя библиотека компании [Twitter](https://ru.wikipedia.org/wiki/Twitter) под названием Twitter Blueprint. После нескольких месяцев разработки он был открыт под названием Bootstrap [19 августа](https://ru.wikipedia.org/wiki/19_%D0%B0%D0%B2%D0%B3%D1%83%D1%81%D1%82%D0%B0) [2011 года](https://ru.wikipedia.org/wiki/2011_%D0%B3%D0%BE%D0%B4).

Основными нововведениями второй версии, появившейся 31 января 2012 года, стали 12-колоночная сетка и поддержка [адаптивности](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%B0%D0%BF%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD). Третья версия была выпущена 19 августа 2013 года. В ней адаптивность получила дальнейшее развитие, был осуществлён переход к концепции [mobile first](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%B0%D0%BF%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD#Основные_принципы), оптимизации прежде всего под мобильные устройства. Дизайн по умолчанию теперь стал [плоским](https://ru.wikipedia.org/wiki/Flat_Design). Работа над четвёртой версией начата 29 октября 2014 года. 18 января 2018 года выпущена первая стабильная версия Bootstrap 4. 5 мая 2021 года выход Bootstrap 5.

Основными инструментами Bootstrap являются:

* Сетки — заранее заданные размеры колонок, которые можно сразу использовать.
* Шаблоны — фиксированный или резиновый шаблон документа.
* Типографика — инструмент описания [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), определение некоторых классов для [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), таких как код, цитаты и т. п.
* Медиа — инструмент, предоставляющий некоторое управление изображениями и видео.
* Таблицы — средства оформления таблиц, вплоть до добавления функциональности сортировки.
* Формы — некоторые классы для оформления форм и некоторых событий, происходящих с ними.
* Навигация — некоторые классы оформления для панелей, вкладок, перехода по страницам, меню и панели инструментов.

**Django** — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. *Don't repeat yourself*)

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Архитектура Django похожа на «Модель-Представление-Контроллер» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление (англ. *View*), а презентационная логика Представления реализуется в Django уровнем Шаблонов (англ. *Template*). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).

Первоначальная разработка Django как средства для работы новостных ресурсов достаточно сильно отразилась на его архитектуре: он предоставляет ряд средств, которые помогают в быстрой разработке веб-сайтов информационного характера. Так, например, разработчику не требуется создавать контроллеры и страницы для административной части сайта, в Django есть встроенное приложение для управления содержимым, которое можно включить в любой сайт, сделанный на Django, и которое может управлять сразу несколькими сайтами на одном сервере. Административное приложение позволяет создавать, изменять и удалять любые объекты наполнения сайта, протоколируя все совершённые действия, и предоставляет интерфейс для управления пользователями и группами (с пообъектным назначением прав).

В дистрибутив Django также включены приложения для системы комментариев, синдикацииiubtuyyugk RSS и Atom «статических страниц» (которыми можно управлять без необходимости писать контроллеры и представления), перенаправления URL и другое.

Некоторые возможности Django:

1. ORM, API доступа к БД с поддержкой транзакций
2. встроенный интерфейс администратора, с уже имеющимися переводами на многие языки
3. диспетчер URL на основе регулярных выражений
4. расширяемая система шаблонов с тегами и наследованием
5. подключаемая архитектура приложений, которые можно устанавливать на любые Django-сайты
6. «generic views» — шаблоны функций контроллеров
7. библиотека для работы с формами (наследование, построение форм по существующей модели БД)

## 2.5 Хостинг

**Хо́стинг** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *hosting*) — [услуга](https://ru.wikipedia.org/wiki/%D0%A3%D1%81%D0%BB%D1%83%D0%B3%D0%B0) по предоставлению [ресурсов](https://ru.wikipedia.org/w/index.php?title=%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%80%D0%B5%D1%81%D1%83%D1%80%D1%81%D1%8B&action=edit&redlink=1) для размещения [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) на [сервере](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)), постоянно имеющем доступ к сети (обычно [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)).

Обычно услуга хостинга входит в пакет по обслуживанию сайта и подразумевает, как минимум, размещение файлов сайта на [сервере](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)), на котором запущено [ПО](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), необходимое для обработки запросов к этим файлам ([веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)). Как правило, в обслуживание уже входит предоставление места для [почтовой корреспонденции](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0), [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [DNS](https://ru.wikipedia.org/wiki/DNS), файлового хранилища на специально выделенном [файл-сервере](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и т. п., а также поддержка функционирования соответствующих сервисов.

Хостинг [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), размещение файлов, хостинг [электронной почты](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0), услуги [DNS](https://ru.wikipedia.org/wiki/DNS) могут предоставляться отдельно, как самостоятельные услуги либо входить в комплексную услугу.

Одним из критериев выбора хостинга является используемая [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), поскольку от неё зависит [программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), которое будет поддерживать функциональность сервисов в дальнейшем. Важным аспектом описания хостинга является наличие таких служб и возможностей: поддержка [CGI](https://ru.wikipedia.org/wiki/CGI): [Perl](https://ru.wikipedia.org/wiki/Perl), [PHP](https://ru.wikipedia.org/wiki/PHP), [Python](https://ru.wikipedia.org/wiki/Python), [ASP](https://ru.wikipedia.org/wiki/Active_Server_Pages), [Ruby](https://ru.wikipedia.org/wiki/Ruby), [JSP](https://ru.wikipedia.org/wiki/JSP), [Java](https://ru.wikipedia.org/wiki/Java); поддержка [баз данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), а также установленные модули и [фреймворки](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) для каждой из возможностей.

Хостинг как услугу оценивают по таким ограничениям:

* размер: [дискового пространства](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4) под файлы [пользователя](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C);
* количество: месячного [трафика](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA); [сайтов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), которые можно разместить в рамках одной [учётной записи](https://ru.wikipedia.org/wiki/%D0%A3%D1%87%D1%91%D1%82%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C); [FTP](https://ru.wikipedia.org/wiki/FTP) пользователей; [E-Mail](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0) ящиков и объём дискового пространства, предназначенного для почты; [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и размер дискового пространства под базы данных; одновременных процессов на пользователя; [ОЗУ](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C), и максимальное время исполнения, выделяемое каждому процессу пользователя.

Качественные ограничения:

* свободные ресурсы [CPU](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), [оперативной памяти](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C), которые влияют на быстродействие сервера;
* пропускная способность каналов, от которой зависит загрузка информации;
* удалённость оборудования хостера от [целевой аудитории](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0) сайта, которая влияет на загрузку информации.

Виды хостинга:

1. [Виртуальный хостинг](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) — сервер с множеством сайтов, владельцы которых имеют одинаковые права и обязанности.
2. Виртуальный выделенный сервер ([VPS](https://ru.wikipedia.org/wiki/VPS)/VDS) — автономная (выделенная) часть дискового пространства на сервере и фиксированные ресурсы. Владелец получает права администратора, и самостоятельно может настраивать и устанавливать программы.
3. [Выделенный сервер](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) — полное владение сервером с отдельной [ОС](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), [ПО](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).
4. [Colocation](https://ru.wikipedia.org/wiki/Colocation) — размещение сервера, которым владеет отдельный человек, [предприниматель](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BF%D1%80%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D1%82%D0%B5%D0%BB%D1%8C), компания, в дата-центре хостинговой компании.

# **Глава II. Создание сайта**

## 2.1 Выбор основных инструментов для создания сайта и их подключение

В качестве редактора кода была выбрана программа PyCharm Community Edition, так как она обладает многими технологиям, одни из которых: подсветка синтаксиса, что намного облегчает и ускоряет работу с кодом.

Для создания основы сайта будет использоваться язык гипертекстовой разметки HTML. Для того чтобы придать сайту приятный удобный внешний вид будет использоваться Фреймворк Bootstrap, позволяющий легко создать практичный красочный дизайн.

Для разработки внутреннего строения сайта был выбран Веб-фреймворк Django работающий на языке программирования python. Django является одним из лучших фреймворков для создания сайтов, так как он обладает многими готовыми модулями для разработки. В фреймворке предусмотрены встроенная система аутентификации; ORM, API доступа к БД с поддержкой транзакций и другие.

В качестве базы данных выбрана реляционная MySQL. Эта база данных поддерживается фреймворком Django. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц.

Установка

Основные программы были скачены с официальных сайтов

Установка фреймворка Django производится в терминале компьютера командой pip install django.

Далее для создания проекта на основе Django в терминале нужно обратиться к команде django-admin startproject lpie (название проекта). После выполнения команды создастся папка проекта с файлами, с которыми можно работать на языке программирования python.

После этого можно запустить локальный сервер через терминал. Для этого нужно перейти в папку проекта с помощью команды cd lpie (название папки). Потом нужно написать команду python manage.py runserver или python3 manage.py runserver. После выполнения команды запуститься локальный сервер и появиться ссылка на его страницу. Для остановки локального сервера в терминале нужно нажать на комбинацию клавиш Ctrl + C. После чего произойдёт остановка локального сервера.

Последующая настройка проекта будет проводиться в подпункте 2.4 Разработка внутренних механизмов сайта, наполнение его контентом.

Фреймворк bootstrap подключается непосредственно в html файлах в теге <link>. Его можно подключить как онлайн, так и скачать необходимые данные с официального сайта для большего удобства. Для его подключения нужно в теге <link> прописать атрибут rel=”stylesheet”, чтобы указать, что подключённый файл следует читать как файл стиля. Так же нужно прописать атрибут href=”css/bootstrap.min.css”, который указывает путь до необходимых файлов.

Так же для полноценной работы фреймворка нужно подключить java script файлы, которые подключаются в теге <script> с помощью атрибута src=” js/bootstrap.bundle.min.js ”.

СУБД MySQL будет внедрена в проект на этапе подключения сайта к хостингу, а во время разработки будет использоваться встроенная в Django СУБД SQlite.

## 2.2 Разработка структуры, макета и дизайна сайта

Структура сайта состоит из двух основных страниц: главной и курсов. Эти страницы располагаются в навигационной панели. Главная страница содержит в себе основную информацию о проекте. Страница с курсами содержит все доступные курсы для обучения.

Из страницы с курсами можно перейти на страницу с отдельным курсом, которая состоит из модулей, которые включают в себя уроки, состоящие из теоретического материала и заданий для проверки знаний. В конце каждого модуля содержится итоговая проверка знаний пользователя за модуль. В конце курса содержится итоговая проверка знаний.

На каждой странице вверху находится навигационная панель, для удобного перемещения по сайту, а внизу футер(footer) с основной информацией о проекте

Пользователи могут предлагать изменения для сайта или сообщать об ошибках на сайте через специальную форму, располагающуюся на сайте.

Макет сайта был разработан с учетом удобства и понятности для обычного пользователя. Интерфейс спроектирован таким образом, чтобы навигация по сайту была интуитивно понятной и простой. Расположение основных элементов продумано для эффективного взаимодействия пользователя с сайтом (см. приложение 1).

Для дизайна сайта были выбраны светлые, приятные для восприятия тона. Шрифт подобраны с учетом читабельности и визуальной привлекательности. Графические элементы, такие как иконки и иллюстрации, дополняют общую стилистику проекта (см. приложение 2).

Дизайн и макет адаптированы под различные устройства и экраны. Сайт хорошо выглядит и функционирует как на настольных компьютерах, так и на мобильных девайсах. Это обеспечивает удобство использования проекта для пользователей, независимо от их предпочтений и технических возможностей.

## 2.3 Разработка внутренних механизмов сайта, наполнение его контентом

Создание приложений. Сайт состоит из двух приложений main и courses. Main отвечает за главные страницы и информацию о проекте. Courses отвечает за работу курсов.

Разработка представлений (views) и маршрутизация. Для каждой ссылки была создана функция для обработки HTTP-запросов. Так же для функций были созданы URL-маршруты для связи URL-адресов с соответствующими представлениями. В функциях была прописана логика для загрузки курсов, модулей, уроков и проверки знаний.

Разработка шаблонов (templates). Созданные HTML-шаблоны были перенесены в папку templates и подключены к настройкам проекта. Так же в шаблоны был добавлен шаблонизатор Jinja для облегчения разработки. Так же были использованы Django-специфичные теги и фильтры для динамического генерирования контента.

Настройка статических файлов. Была создана папка static для хранения статических файлов сайта (CSS стили, JavaScript, изображения). Папка статик была подключена в настройках проекта и HTML-шаблонах.

Проектирование моделей данных. В файле models.py приложения courses были разработаны четыре модели данных: Course, Module, Lesson, Practice и Answer. Были прописаны взаимосвязи между таблицами с помощью поля отношений ForeignKey (отношение многие к одному). Для удобства заполнения и использования все модели данных были подключены в админ-панель.

Добавление папки медиа(media). Была добавлена и подключена в настройках папка медиа для нестатических изображений.

## 2.4 Разработка курса

Для создания курса было решено использовать официальную документацию Python Структура и содержание курса. Структуру курса было решено сделать состоящей из модулей и уроков, которые будут включать в себя теоретический и практический материалы. По документации были разработаны следующие темы и подтемы курса:

1) Введение в Python: Что такое Python; Установка Python; Использование среды разработки Python.

2) Основы Python: Переменные и типы данных: Операторы и выражения; Управляющие конструкции.

3) Коллекции в Python: Списки; Кортежи; Словари; Множества.

4)Функции в Python: Определение и вызов функций; Аргументы и возвращаемые значения.

5) Модули и пакеты: Импорт модулей; Создание собственных модулей

6) Ввод/вывод в Python: Чтение и запись файлов; Форматирование вывода.

7) Обработка ошибок: Исключения и их обработка.

Было решено преподносить учебный материал в виде текста и проверять его с помощью заданий, которые включают в себя выбор ответов и ввод ответов на практические задания.

Для сбора данных от пользователей в курс добавлена форма обратной связи. Полученные данные будут применены для внесения дальнейших доработок и улучшений курса

Планируется расширять курс дополнительными модулями и уроками.

Материалы курса будут перенесены в базу данных после размещения сайта на хостинге

## 2.5 Размещение сайта в сети интернет

Для размещения сайта была выбрана платформа PythonAnywhere, так как она идеально подходит для размещения маленьких проектов на Django

Процесс размещения сайта на PythonAnywhere включает три основных этапа: загрузка кода в PythonAnywhere, настройка virtualenv (виртуального окружения), настройка приложения с использованием WSGI

В PythonAnywhere предустановлено много полезных модулей, но они могут быть не лучшими и для этого нужно установить виртуальное окружение. Для его установки нужно запустить консоль Bash и прописать следующую команду: $ mkvirtualenv myvirtualenv --python = /usr/bin/python3.10

Далее устанавливаются необходимые пакеты (Django и дополнительные библиотеки для его работы) в виртуальную среду, используя pip. Далее для подключения виртуальной среды на хостинге нужно перейти во вкладку Wed и в разделе Virtualenv ввести путь: /home/myusername/.virtualenvs/myvirtualenv

Чтобы приложение заработало, нужно настроить файл WSGI, который импортирует приложение в качестве переменной Python. Для настройки нужно перейти на **Web** нажать **Add new web app** и выбрать **Manual configuration**. После завершения будет показана страница с параметрами конфигурации вашего веб-сайта. На этой странице нужно ввести путь к virtualenv. Далее нужно отредактировать файл WSGI, который был создан. Главный шаги: Добавление пути, по которому находится приложение, в sys.path Импорт приложения WSGI и его вызов application.

Чтобы начать использовать MySQL нужно перейти на вкладку MySQL и установить пароль. Чтобы настроить Django для доступа к базе данных MySQL на PythonAnywhere, необходимо сделать это в файле настроек:

DATABASES = {'default': {'ENGINE': 'django.db.backends.mysql', 'NAME': 'имя\_пользователя$имя\_базы\_данных','USER':т'<имя\_пользователя>','PASWORD’: 'mysql\_password ‘, 'HOST’: 'mysql\_hostname ‘,}}

После запуска сервера материалы сайта были перенесены в базы данных.

# **Заключение**

В ходе выполнения проекта был изучен материал по теме разработка сайтов, после чего был разработан макет сайта, которому был предан красивый внешний вид. После макета был прописан внутренний код сайта и добавлены обучающие материалы по теме «Python». После завершения разработки сайта, он был размещён в информационной сети интернет и был проверен на работоспособность. Цель и задачи проекта были достигнуты.

# **Список интернет-ресурсов**

<https://ru.wikipedia.org/> (информация о инструментах разработки)

<https://www.mysql.com/> (установка и документация СУБД MySQL)

<https://www.jetbrains.com/> (установка PyCharm Community Edition)

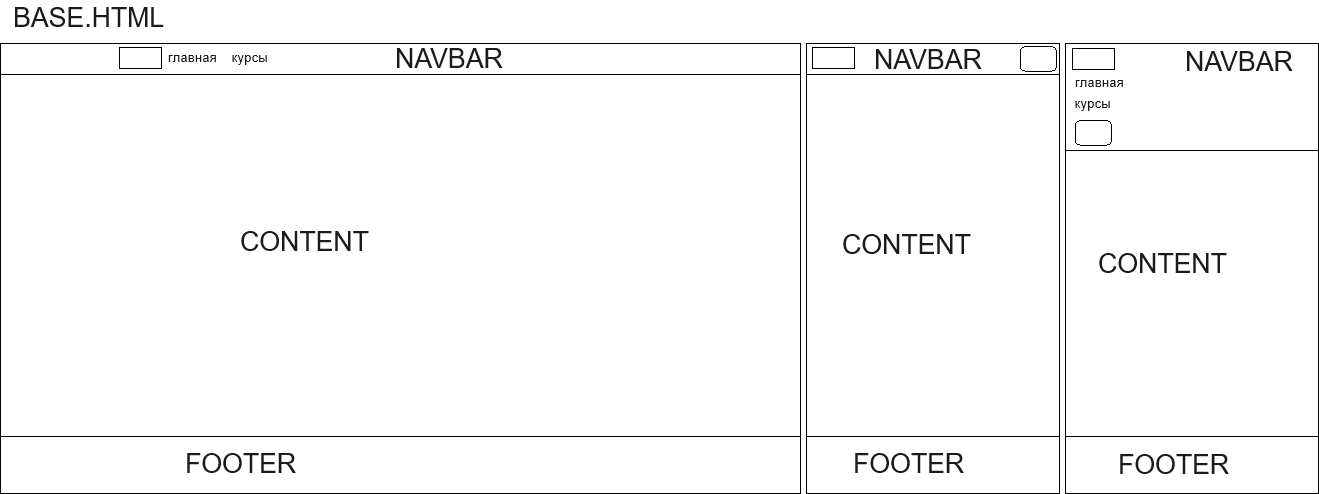
<https://getbootstrap.com/> (документация и установка bootstrap)

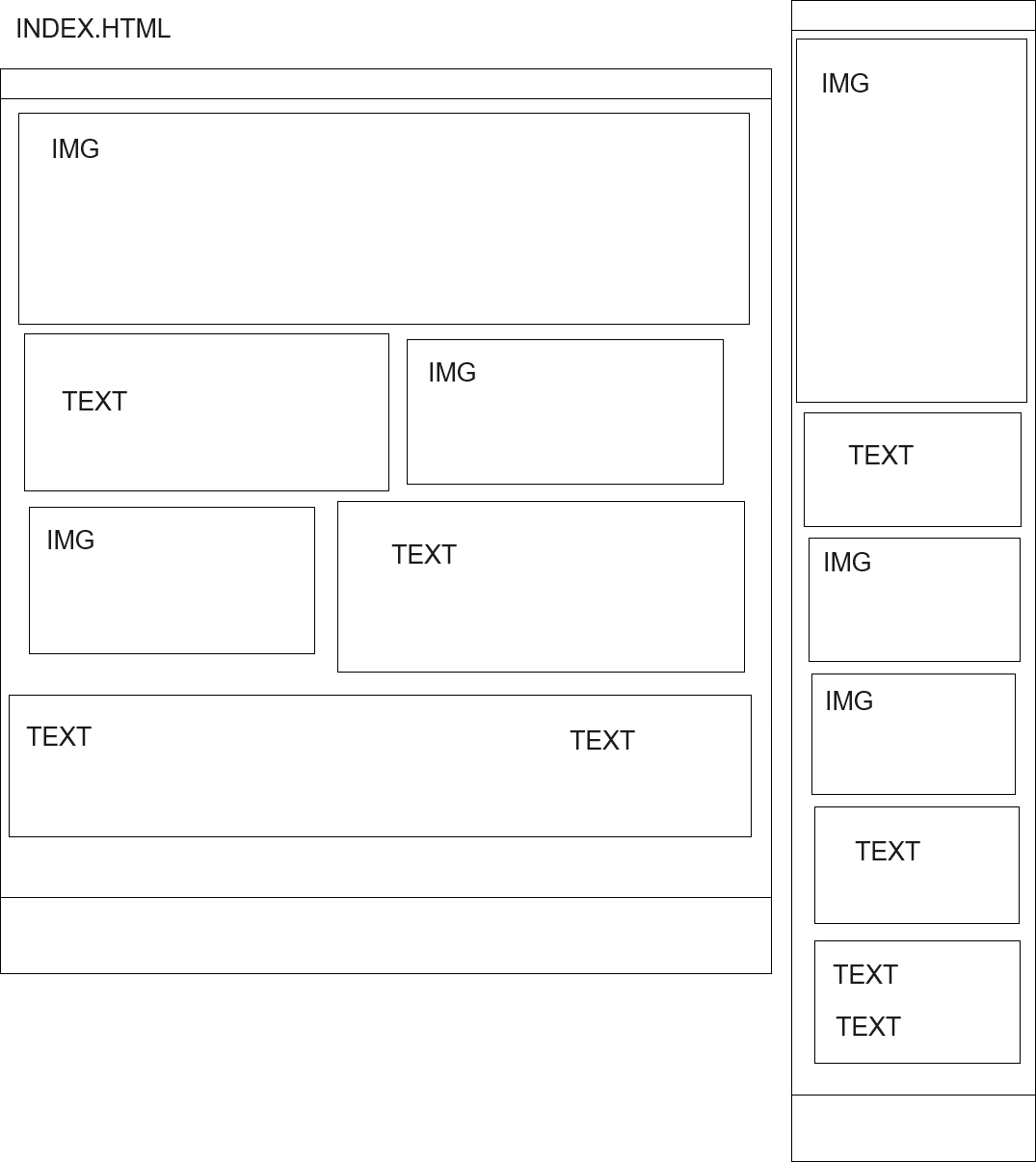
<https://www.djangoproject.com/> (документация Django)

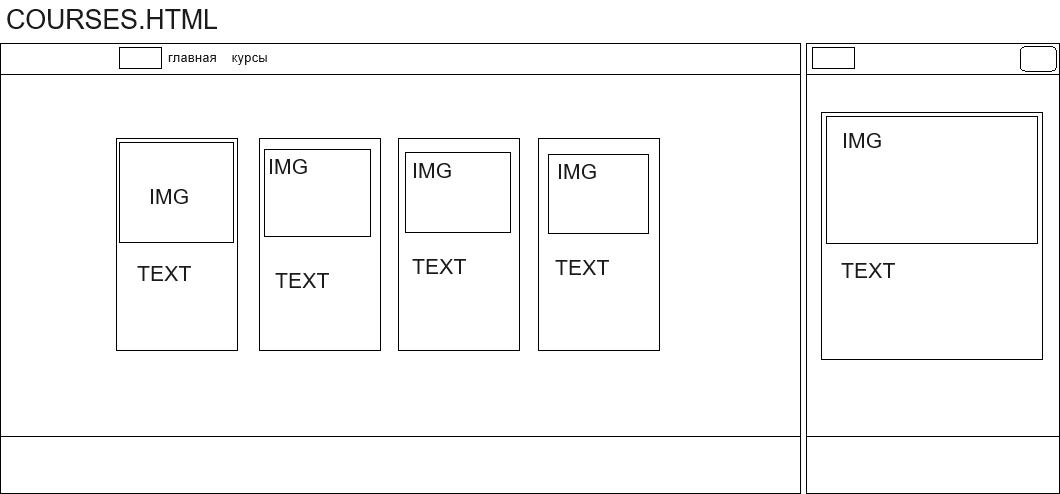
<https://www.python.org/> (установка python и его документация)

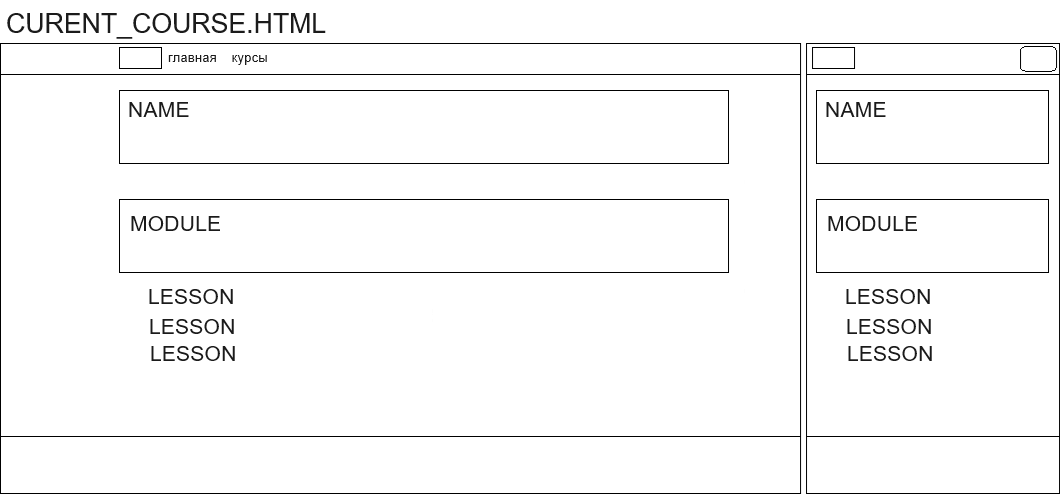
<https://www.pythonanywhere.com/> (хостинг для сайта и его документация)

# **Приложение 1. Макет сайта**











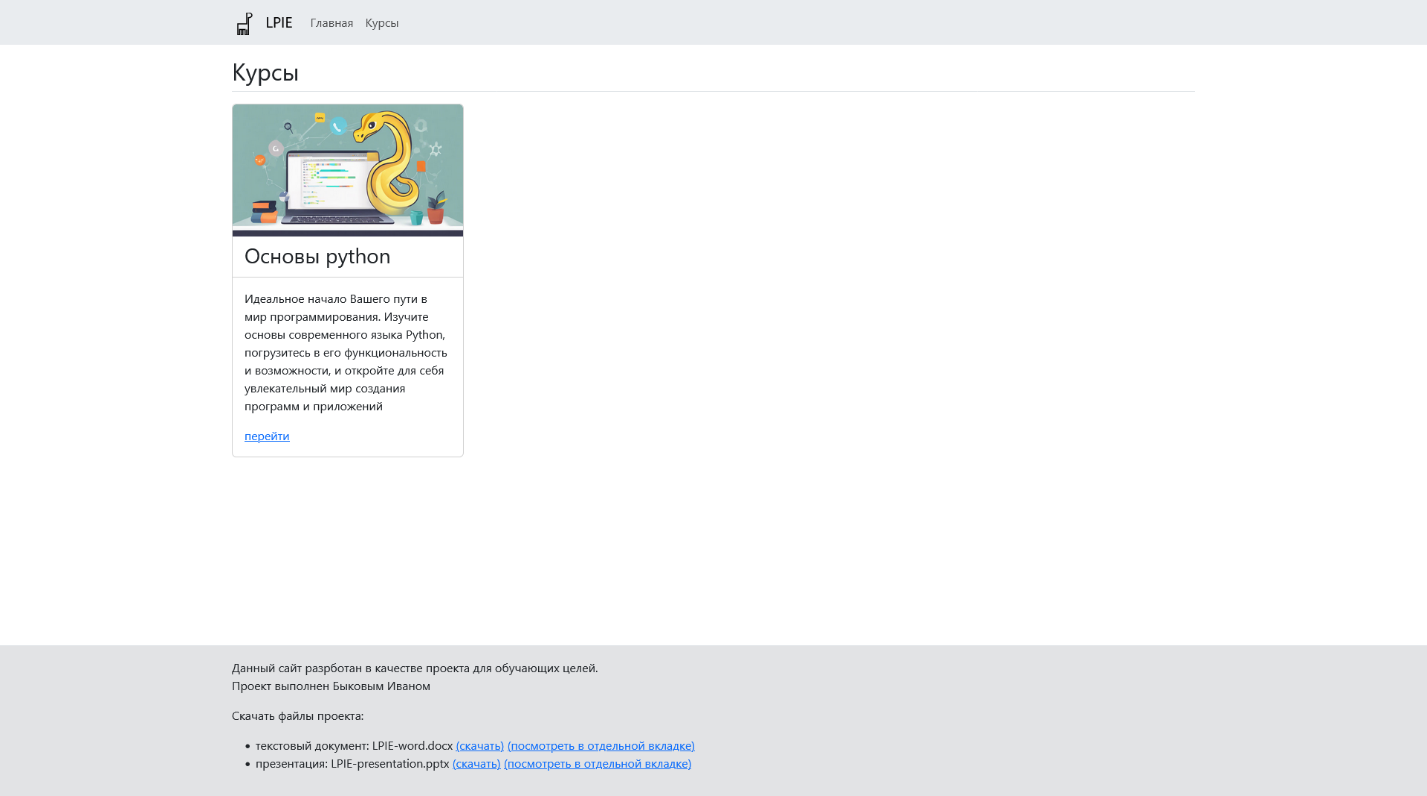
# **Приложение 2. Дизайн сайта**

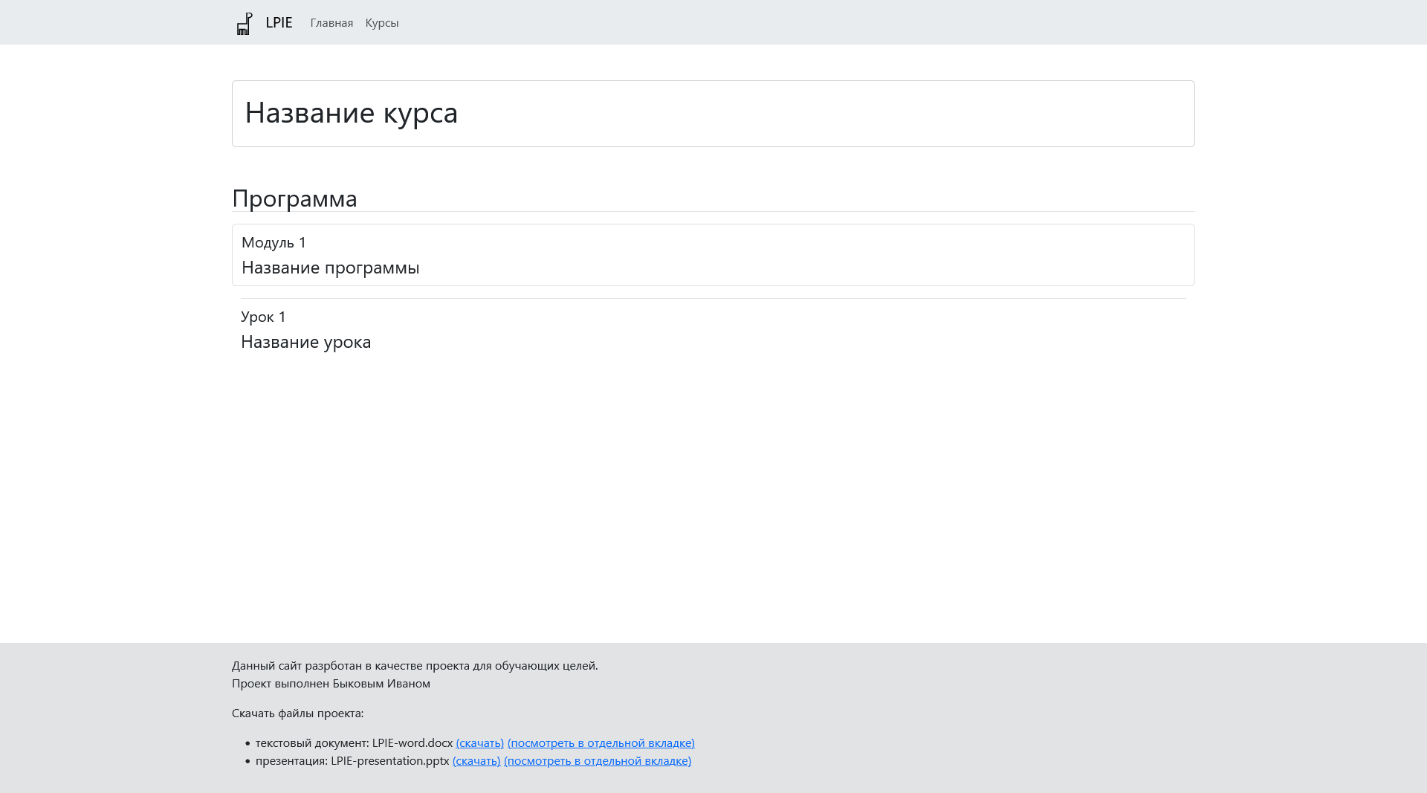
Desktop

Главная

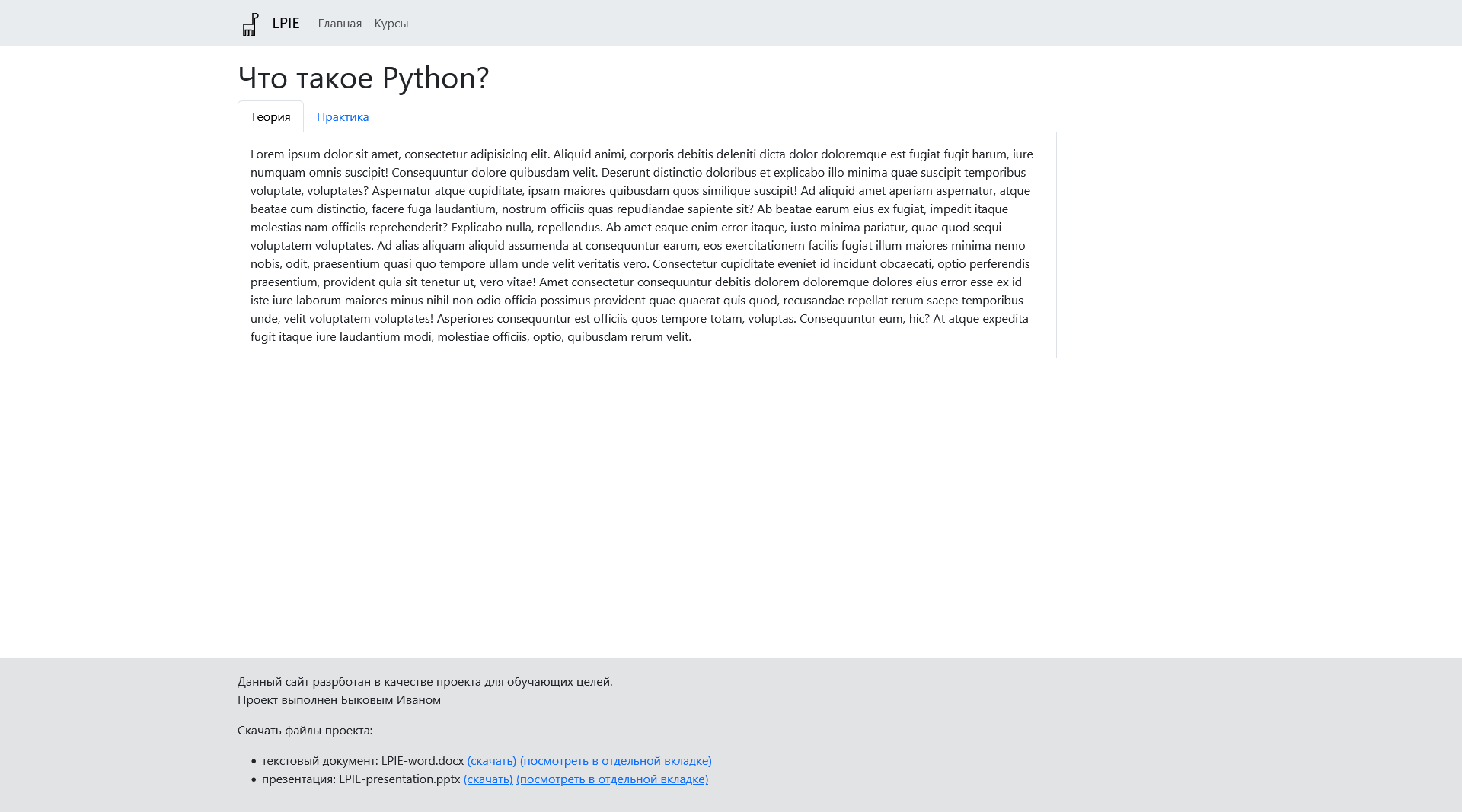


Курсы

Курс

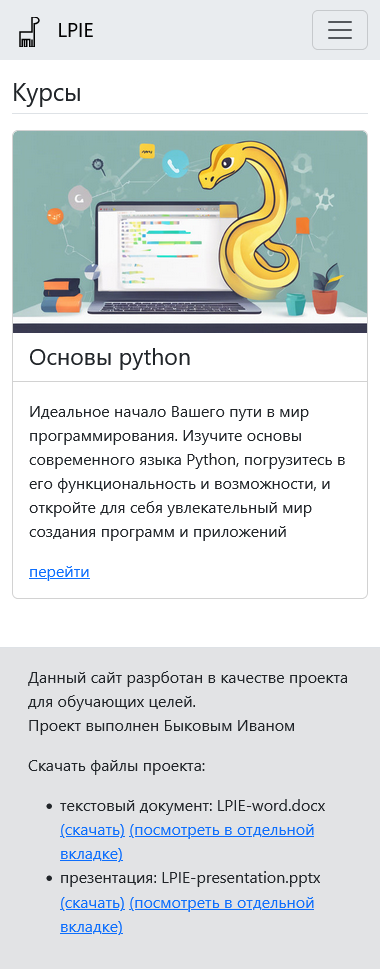
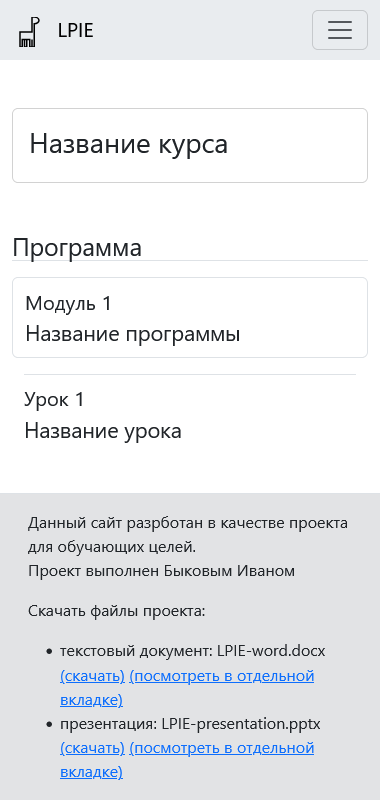


Урок



Mobile

Курсы Курс

Главная Урок

