



# Программа курса **Python Advanced**

**Версия 2.0.0**

**Продолжительность курса: 100 пар**

## **Тематический план**

- 1. Разработка веб-страниц на языке разметки HTML5  
с использованием каскадных таблиц стилей CSS3 . . . . . 14 пар**
- 2. Разработка клиентских сценариев  
с использованием JavaScript . . . . . 24 пары**
- 3. Python Advanced . . . . . 62 пары**

# Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3

**Версия 3.1.0**

**Продолжительность курса: 14 пар**

## Цель курса

Обучить слушателя созданию и верстке статических web-страниц с использованием технологий HTML5, CSS3. Сложить для слушателя целостное представление о технологической цепочке создания web-сайтов и сформировать понимание актуальных тенденций развития web-технологий. Научить слушателя выбирать наиболее подходящий способ для создания web-страниц. Научить тестировать и проверять код web-страниц.

## По окончании курса слушатель будет:

- знать и уметь применять основы HTML – теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов;
- знать и уметь применять основы CSS – значения, списки, цвета, шрифты и другие метрики форматирования;
- владеть навыками проверки и отладки кода web-документов;
- владеть навыками формирования содержимого web-документов для различных экранов – от стандартных браузеров до мобильных устройств;
- владеть навыками быстрого и качественного форматирования сложных web-документов;
- знать основы HTML5 и CSS3;
- создавать и использовать формы.

В качестве редактора для создания можно использовать любой бесплатный продукт. Например, Notepad++, Microsoft Visual Studio Community, Visual Studio Code.

По окончании данного курса студент сдает все практические задания курса. На основании всех сданных заданий выставляется оценка по предмету.

Обязательное практическое задание в конце курса, выполняемое студентом дома: «Создание web-сайта с последующим размещением в Internet».

Основные требования: блочная верстка, валидный код.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Introduction to Developing Websites Using HTML and CSS (этот курс является подготовкой к экзамену Microsoft Technology Associate (MTA) 98-383);
- HTML5 Application Development Fundamentals (этот курс является подготовкой к экзамену Microsoft Technical Associate exam 98-375).

## Тематический план

- Модуль 1.** Введение в Web-технологии. Структура HTML.  
Форматирование текста при помощи HTML..... **2 пары**
- Модуль 2.** Форматирование при помощи CSS. Списки.  
CSS отступы и поля ..... **3 пары**
- Модуль 3.** Графика в web-дизайне. Оптимизация графики.  
Гиперссылки. Принципы навигации web-сайта ..... **3 пары**
- Модуль 4.** Таблицы..... **2 пары**
- Модуль 5.** Позиционирование.  
Верстка web-страниц блоками..... **2 пары**
- Модуль 6.** Формы..... **2 пары**

# Модуль 1

## Введение в Web-технологии. Структура HTML. Форматирование текста при помощи HTML

1. Введение в предмет.
2. Введение в языки разметки. Язык разметки гипертекста HTML.
  - Internet.
  - Протокол HTTP.
  - Развитие HTML, версии. Версия HTML5.
  - Вопросы межбраузерной совместимости. Война браузеров.
  - W3C.
3. Теги – основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте HTML5. Синтаксические отличия HTML4, XHTML, HTML5.
4. Основные ошибки в записях тегов.
  - Спецификации <!DOCTYPE HTML>.
  - Валидация документа при помощи FireFox – дополнение HTML Validator.
  - Понятие well-formed.
  - Прародители HTML5: SGML и XML.
5. Структура HTML5 документа.
  - Основные элементы и их назначения.
  - Новые теги задания структуры: <header>, <nav>, <section>, <article>, <aside>, <footer>. Доступность новых тегов в современных браузерах. Отображение новых тегов в устаревших браузерах.
6. Кодировки страницы и теги <meta>.
  - Применение тега <meta> – задание информации о странице (expires, refresh, autor, copyright, keywords, description).
  - Задание кодировки страницы при помощи тега <meta>.
  - Символьные подстановки и кодировки.
7. Классификация тегов: линейные и блочные.
  - Линейные.
  - Блочные.
8. Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>...<h6>. Выравнивание текста в блочных элементах: атрибут align.
9. Классификация тегов: логическое и физическое форматирования.

- Теги физического форматирования.
- Теги логического форматирования.
- Краткий обзор основных тегов логического форматирования: `<abbr>`, `<acronym>`, `<cite>`, `<code>`, `<del>`, `<dfn>`, `<ins>`.

## 10. Практика: создание простейшей web-страницы.

# Модуль 2

## Форматирование при помощи CSS. Списки. CSS отступы и поля

### 1. CSS – каскадные таблицы стилей.

- Введение. Обзор версий. Назначение: HTML служит для задания структуры, CSS – для форматирования.
- Встраивание CSS в HTML при помощи атрибута `style`.  
Правила записи CSS свойств.

### 2. Теги без форматирования `<div>` – блочный, `<span>` – линейный.

### 3. Аналогия HTML и CSS на примере линейных и блочных тегов.

### 4. Дополнительные свойства CSS для форматирования текста: `letter-spacing`, `line-height`, `text-indent`, `text-transform`, `white-space`, `word-spacing`.

### 5. Использование атрибутов `class` и `id` для задания стилей.

- Создание стилей для тегов, классов, идентификаторов внутри тега `<style>`. Понятие селекторов. Правило записи селекторов: селектор тегов, селектор классов, селектор идентификаторов, универсальный селектор `*`.
- Приоритет использования стилей (`tag / class / id / style`). Повышение приоритета правилом `!important`.
- Наследуемость стилей. Стандартные значения свойств.
- Отслеживание стилей при помощи средства разработки `firebug` (дополнение для Firefox).

### 6. Использование внешних CSS файлов стилей.

- Подключение CSS файлов при помощи тега `<link>` и инструкции `@import`.
- CSS файлы и кэш браузера.

### 7. Практика: форматирование текста при помощи CSS.

### 8. Создание списков.

- Неупорядоченные списки: элементы `<ul>`, `<li>`.
- Упорядоченные списки: элементы `<ol>`, `<li>`.

- Атрибуты type, value, start.

## 9. Создание вложенных списков.

## 10. Форматирование списков при помощи CSS.

- Свойства list-style-type, list-style-image, list-style-position.
- Сокращенная запись свойства list-style.
- Оформление многоуровневых списков. Вложенные селекторы.

## 11. Списки определений: элементы <dl>, <dd>, <dt>.

## 12. Управление отступами и полями.

- Свойство margin и его потомки margin-left, margin-top, margin-right, margin-bottom.
- Свойство padding и его потомки padding-left, padding-top, padding-right, padding-bottom.
- Отличие padding от margin и их назначения.
- Отмена отступов по умолчанию у некоторых тегов: <body>, <h1>...<h6>, <p>.

## 13. Практика: создание списков.

# Модуль 3

## Графика в web-дизайне. Оптимизация графики. Гиперссылки. Принципы навигации web-сайта

### 1. Форматы графических файлов в Web.

### 2. Тег <img /> и его атрибуты (src, alt, width, height, border).

- Свойство border – аналог атрибута border.
- Задание свойств margin, padding, border для изображения.
- Выравнивание изображений на странице при помощи атрибута align. Аналог атрибута align – свойство float.

### 3. Фон страницы – свойство background.

- Задание фона в виде цвета: background-color. Обязательное задание фона для элемента <body>.
- Задание фона в виде изображения: background-image, background-repeat, background-position, background-attachment.
- Изображения и кэш браузера.

### 4. Общие сведения о гиперссылках.

- Тег <a> и его атрибуты (href, target).
- Эргономика, удобство навигации.

## 5. Абсолютная и относительная адресация.

- Организация внешних ссылок.
- Организация внутренних ссылок с помощью элемента `<a>`. Атрибуты `id` и `name`.
- Организация «смешанного» перехода (на указанный элемент во внешнем HTML-документе).
- Графические ссылки. Отмена границ у ссылок.

## 6. Создание меню при помощи структуры списков (`<ul>`, `<li>`), его форматирование. Свойство `display`. Преобразование ссылки в блочный элемент.

## 7. Псевдоклассы.

- Псевдоклассы ссылок: `active`, `hover`, `link`, `visited`.
- Псевдоклассы для обычных элементов: `first-child`, `first-line`, `first-letter`.

## 8. CSS свойство `cursor`.

## 9. Практика: работа по разработке галереи изображений.

## 10. Свойства из CSS3.

- Работа с фоном: создание градиентов, изменение размеров фона – свойства `background` и `background-size`.
- Работа с границами: скругленные края у блоков – свойства `border-radius`.
- Задание полупрозрачности элементам страниц – свойство `opacity`.
- Полная поддержка селекторов CSS 2.1.

## 11. Работа с мультимедиа.

- Вставка видео на странице посредством тега `<video>`.
- Вставка аудио на странице посредством тега `<audio>`.
- Создание изображений и анимации посредством тега `<canvas>`.
- Использование SVG формата.

# Модуль 4

## Таблицы

### 1. Создание простейшей таблицы. Теги `<table>`, `<tr>` и `<td>`.

- Атрибуты `border`, `cellspacing`, `cellpadding`. Их возможные аналоги CSS: `border`, `padding`.
- Указание ширины и высоты ячейки: атрибуты `width`, `height`. Правила задания ширины и высоты. Аналоги CSS: свойства `width`, `height`.
- Выравнивание данных в таблице: атрибуты `align` и `valign`. Аналоги CSS: свойства `text-align`, `vertical-align`.

- Управление цветом фона и цветом рамок таблицы (отдельной строки, отдельной ячейки).
  - Использование изображений в качестве фона таблицы (отдельной строки, отдельной ячейки).
2. Объединение ячеек: атрибуты `colspan`, `rowspan`.
  3. Теги логического структурирования таблиц: `<thead>`, `<tbody>`, `<tfoot>`. Теги логического группирования столбцов: `<colgroup>`, `<col>`.
  4. Управление рамками таблицы: атрибуты `frame`, `rules`.
  5. Практика: создание сложных таблиц.
  6. Основы табличной верстки. Пример табличной верстки: ее минусы.

## Модуль 5

### Позиционирование. Верстка web-страниц блоками

1. Свойство `position`.
  - Рассмотрение позиционирования: `relative` и `absolute`.
  - Свойства `top`, `left`, `bottom`, `right`.
2. Свойства `visibility`, `overflow`.
3. Практика.
4. Основы верстки блоками. Правила верстки.
  - Вложение блоков.
  - Задание ширины и высоты блокам при помощи свойства `width` и `height`.
  - Обтекание блоков. Отмена обтекания блоков. Свойства `float` и `clear`.
  - Правила задания отступов и полей.
  - Задание минимальной высоты и ширины блока: свойства `min-height`, `min-width`. Задание этих свойств в браузере IE6.
  - Выравнивание внутри блоков (`margin`, `text-align`, `line-height`, `position`). Кроссбраузерность выравниваний.
5. Рассмотрение простейших структур страниц и элементов.
  - Структура сайта фиксированного размера.
6. Резиновая структура. Блоки с отрицательными `margin`.



# Модуль 6

## Формы

### 1. Введение в формы.

### 2. Управляющие элементы форм.

- Кнопки (отправки, сброса, пр.).
- Флажки.
- Кнопки с зависимой фиксацией (радиокнопки).
- Всплывающие списки.
- Текстовый ввод.
- Выбор файлов.
- Скрытые управляющие элементы.

### 3. Создание форм при помощи HTML.

- Элемент `<form>`.
- Элемент `<input>`.
- Элемент `<button>`.
- Элементы `<select>`, `<optgroup>` и `<option>`.
- Элемент `<textarea>`.
- Метки `<label>`.
- Структура форм: `<fieldset>` и `<legend>`.

### 4. Элементы форм из HTML5.

### 5. Валидация форм при помощи HTML5.

### 6. Форматирование элементов форм при помощи CSS.

# Разработка клиентских сценариев с использованием JavaScript

**Версия 2.5.0**

**Продолжительность курса: 24 пары**

## Цель курса

Обучить слушателя разработке клиентских сценариев с использованием JavaScript. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

## По окончании курса слушатель будет:

- владеть базовыми конструкциями языка JavaScript, такими как переменные, условия, циклы, строки, массивы функции и т. д.
- знаком с ООП и его основными понятиями;
- уметь обрабатывать возникающие ошибки;
- разбираться в понятиях «событие», «обработчик события».
- создавать функции-обработчики различных событий;
- понимать отличия BOM и DOM;
- уметь взаимодействовать с объектами из BOM и DOM;
- разбираться в тонкостях реализации клиентских сценариев под разные браузеры;
- владеть принципами создания форм и анализа данных пользователя с использованием регулярных выражений;
- уметь сохранять пользовательские данные с помощью механизма cookie;
- понимать особенности применения HTML5 по отношению к JavaScript;
- уметь сериализовать и парсить данные, используя JSON;
- владеть принципами создания асинхронных запросов при помощи AJAX.

По окончании данного курса студент сдает все практические задания курса. На основании всех сданных заданий выставляется оценка по предмету.

## Тематический план

<b>Модуль 1.</b>	Введение в JavaScript .....	<b>2 пары</b>
<b>Модуль 2.</b>	Объект. Массивы. Объект Array. Строки. Объект String. Объект Date. Объект Math. Введение в ООП .....	<b>6 пар</b>
<b>Модуль 3.</b>	Обработка событий .....	<b>4 пары</b>
<b>Модуль 4.</b>	Browser Object Model. Document Object Model .....	<b>4 пары</b>
<b>Модуль 5.</b>	Формы .....	<b>2 пары</b>
<b>Модуль 6.</b>	Проверка достоверности форм. Использование Cookie .....	<b>2 пары</b>
<b>Модуль 7.</b>	JSON, AJAX.....	<b>4 пары</b>

# Модуль 1

## Введение в JavaScript

1. Сценарии, выполняемые на стороне клиента.
2. Что такое JavaScript?
3. История создания JavaScript.
4. Различия между JavaScript и Java, JScript, ECMAScript.
5. Версии JavaScript.
6. Понятие Document Object Model.
7. Понятие Browser Object Model.
8. Внедрение в HTML документы. Редакторы кода JavaScript.
9. Тег `<noscript>`.
10. Основы синтаксиса.
  - Регистрозависимость.
  - Комментарии.
  - Ключевые и зарезервированные слова.
11. Переменные. Правила именования переменных.
12. Типы данных.
13. Операторы.
  - Арифметические операторы.
  - Операторы отношений.
  - Логические операторы.
  - Оператор присваивания.
  - Битовые операторы.
  - Приоритет операторов.
  - Оператор `typeof`.
14. Ввод/вывод данных. Диалоговые окна.
15. Условия.
  - Что такое условие?
  - `if`.
  - `if else`.
  - Тернарный оператор `?:`.
  - `switch`.

## 16. Циклы.

- Что такое цикл?
- while.
- do while.
- for.
- break.
- continue.
- Понятие метки.

## 17. Что такое функция?

- Синтаксис объявления функции.
- Параметры функции.
- Возвращаемое значение функции. Ключевое слово return.

## 18. Объект arguments.

- Цель и задачи объекта.
- Свойство length.

## 19. Область видимости переменной. Ключевое this.

## 20. Рекурсия.

# Модуль 2

## Объект. Массивы. Объект Array. Строки. Объект String. Объект Date. Объект Math. Введение в ООП

### 1. Объекты.

- Что такое объект?
- Введение в объектный тип данных.
- Объект Object.
- Ключевое слово new.
- Понятие свойства.
- Добавление свойств. Синтаксис добавления свойств.
- Синтаксис обращения к свойствам.

### 2. Массивы.

- Что такое массив?
- Объект Array.

- Создание массива.
  - Обращение к элементам массива.
  - Свойства и методы Array.
3. **Строки.**
    - Объект String.
    - Свойства и методы String.
  4. **Задержки и интервалы. Периодический вызов функций.**
  5. **Объект Date. Обработка даты и времени.**
  6. **Объект Math. Свойства и методы. Случайные числа.**
  7. **Что такое ООП?**
  8. **Три фундаментальных принципа ООП.**
    - Инкапсуляция.
    - Наследование.
    - Полиморфизм.
  9. **Понятие класса и объекта в терминах JavaScript.**
  10. **Свойства.**
  11. **Методы.**
  12. **Свойства-аксессоры.**
    - get-свойства (геттеры).
    - set-свойства (сеттеры).
  13. **Конструктор.**
  14. **Понятие prototype.**
    - Что такое prototype.
    - Цели и задачи prototype.
  15. **Наследование.**

## Модуль 3

### Обработка событий

1. **Что такое событие?**
2. **Что такое обработчик события?**
3. **Обработка событий в сценариях.**
4. **Управление стилями элементов web-страницы.**

5. Объект `event` и его свойства.
6. Обработчики событий по умолчанию (стандартные обработчики), запрет вызова стандартного обработчика.
7. Объект `Image`. Управление рисунками и ролловерами.

## Модуль 4

### Browser Object Model. Document Object Model

1. Что такое Browser Object Model?
2. Объекты Browser Object Model.
  - Объект `Window`. Открытие, перемещение и изменение размера окон.
  - Объект `Navigator`. Управление браузером.
  - Объект `Screen`. Свойства экрана.
  - Объекты `Location` и `History`. Перемещение по страницам.
  - Коллекция `Frames`. Управление фреймами.
3. Что такое Document Object Model?
4. Отличия DOM от BOM.
5. Представление HTML-документа в виде дерева.
6. Объекты модели DOM. Иерархия узлов.
7. Свойства и методы модели DOM. Модель событий DOM.
8. Изменение дерева DOM.
9. Знакомство с объектами `Document` и `Link`.
10. Управление выделением и текстовым диапазоном: объекты `Selection` и `TextRange`.
11. Особенности DOM в HTML5.

## Модуль 5

### Формы

1. Применение форм. Размещение элементов формы в HTML.
2. Коллекция `Forms`.  
Создание и программирование элементов формы.
  - Кнопки: элементы `Button`, `Submit`, `Reset`.

- Текстовые поля: элементы Text, Password, File Upload, Textarea.
- Скрытое поле формы: общее понятие об элементе Hidden.
- Флажок: элемент Checkbox.
- Переключатель: элемент Radio.
- Список: элементы Select, Option.

## Модуль 6

### Проверка достоверности форм. Использование Cookie

1. Объект RegExpr. Правила записи регулярных выражений.
2. Методы объектов String и RegExpr для работы с регулярными выражениями.
3. Проверка достоверности данных формы.
4. Что такое cookie?
5. Преимущества и недостатки cookie.
6. Создание, использование и удаление cookie.

## Модуль 7

### JSON, AJAX

1. Что такое JSON?
2. Цели и задачи JSON.
3. Синтаксис JSON.
  - Переменные.
  - Объекты.
  - Массивы.
4. Объект JSON.
  - Что такое сериализация?
  - Что такое парсинг?
  - Методы stringify и parse.
5. Настройка пользовательской сериализации в JSON. Метод toJSON.
6. Синхронные и асинхронные запросы.



**7. Что такое AJAX?**

**8. Объект XMLHttpRequest.**

- Создание через ActiveX объект.
- Создание через объект XMLHttpRequest.

**9. Методы и свойства XMLHttpRequest.**

**10. Понятие HTTP заголовка.**

**11. Использование метода GET. URL кодирование.**

**12. Использование метода POST.**

# Python Advanced

**Версия 2.0.0**

**Продолжительность курса: 62 пары**

## Цель курса

Обучить слушателя продвинутым приемам программирования на Python, а также веб-программированию на Python.

## По окончании курса слушатель будет:

- разбираться в тонкостях построения веб-приложений с использованием Python;
- обрабатывать и анализировать данные форм;
- применять регулярные выражения;
- сохранять данные пользователя в файлах cookies;
- работать с механизмом сессий;
- понимать принципы сетевого взаимодействия;
- взаимодействовать с источниками данных;
- внедрять AJAX в веб-приложения;
- понимать и разбираться в тонкостях паттерна MVC;
- создавать веб-проекты с использованием Python и паттерна MVC;
- использовать Flask/Bottle;
- разрабатывать web-приложения с помощью фреймворка Django.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

## Тематический план

<b>Модуль 1.</b>	Упаковка данных.....	<b>4 пары</b>
<b>Модуль 2.</b>	Параллельное, многопоточное, сетевое программирование .....	<b>6 пар</b>
<b>Модуль 3.</b>	Введение в работу с базами данных.....	<b>10 пар</b>
<b>Модуль 4.</b>	Использование баз данных в Python.....	<b>6 пар</b>
<b>Модуль 5.</b>	Паттерн MVC/MVT .....	<b>4 пары</b>
<b>Модуль 6.</b>	Фреймворки.....	<b>24 пары</b>
<b>Модуль 7.</b>	Создание чат-ботов с помощью Python.....	<b>6 пар</b>
<b>Модуль 8.</b>	Экзамен .....	<b>2 пары</b>

# Модуль 1

## Упаковка данных

### 1. Сериализация и десериализация.

- Что такое сериализация?
- Что такое десериализация?
- Цели и задачи сериализации и десериализации.
- Практические примеры использования.

### 2. Модуль pickle.

### 3. Модуль json.

### 4. Сторонние модули сериализации.

# Модуль 2

## Параллельное, многопоточное, сетевое программирование

### 1. Параллельное и многопоточное программирование.

- Создание потоков.
- Синхронизация потоков.
- Очереди задач.
- GIL и особенности реализации многопоточности в Python.
- Процессы и передача данных между процессами.

### 2. Сетевое программирование.

- Модель OSI, tcp/udp.
- Протокол HTTP/HTTPS.
- Клиент-серверная модель.
- Понятие Request и Response.
- Реализация эхо-сервера простейшего, многопоточного и асинхронного.
- Apache и Nginx.

# Модуль 3

## Введение в работу с базами данных

### 1. Введение в теорию баз данных.

- История и этапы развития.

- Понятия база данных и система управления базами данных.
- Сравнение существующих моделей баз данных:
  - файловая модель;
  - сетевая модель;
  - иерархическая модель;
  - реляционная модель;
  - объектно-ориентированная модель.
- Понятие реляционной модели баз данных.
- Двенадцать правил Кодда.
- СУБД MySQL:
  - что такое MySQL;
  - история развития MySQL;
  - версии MySQL;
  - установка MySQL.
- Таблицы:
  - первичный ключ;
  - значение по умолчанию;
  - уникальность.
- Типы данных.
- Индексы.
- Запросы:
  - введение в язык структурированных запросов SQL;
  - язык SQL. Стандарты языка SQL;
  - понятия DDL, DML, DCL.

## 2. Запросы SELECT, INSERT, UPDATE, DELETE.

- Оператор SELECT:
  - предложение SELECT;
  - предложение FROM;
  - предложение WHERE;
  - предложение ORDER BY.
- Ключевые слова IN, BETWEEN, LIKE.
- Оператор INSERT.
- Оператор UPDATE.
- Оператор DELETE.

## 3. Многотабличные базы данных.

- Аномалии взаимодействия с однотабличной базой данных:
  - аномалии обновления;

- аномалии вставки;
  - аномалии обновления;
  - аномалии удаления.
- Принципы создания многотабличной базы данных:
    - причины создания многотабличной базы данных;
    - внешний ключ;
    - связи. Типы связей;
    - целостность данных.

#### **4. Нормализация.**

- Необходимость нормализации.
- Понятие нормальной формы.
- Первая нормальная форма.
- Вторая нормальная форма.
- Третья нормальная форма.
- Нормальная форма Бойса-Кодда.

#### **5. Многотабличные запросы.**

- Принципы создания многотабличного запроса.
- Декартовое произведение.

#### **6. Функции агрегирования.**

- Функция COUNT.
- Функция AVG.
- Функция SUM.
- Функция MIN.
- Функция MAX.

#### **7. Понятие группировки. Ключевое слово GROUP BY.**

#### **8. Ключевое слово HAVING. Сравнительный анализ HAVING и WHERE.**

#### **9. Подзапросы.**

- Необходимость создания и использования подзапросов.
- Сравнение подзапросов и многотабличных запросов.
- Принцип работы подзапросов.

#### **10. Операторы для использования в подзапросах, объединения.**

- Операторы для использования в подзапросах:
  - оператор EXISTS;
  - операторы ANY/SOME;
  - оператор ALL.

- Объединение результатов запроса:
  - принципы объединения;
  - ключевое слово UNION;
  - ключевое слово UNION ALL.
- Объединения JOIN:
  - понятие inner join;
  - понятие left join;
  - понятие right join;
  - понятие full join.

**11. План выполнения запроса.**

**12. Оптимизация запросов.**

**13. Понятие транзакции. Использование транзакций.**

**14. Представления.**

- Создание представлений.
- Модификация представлений.
- Удаление представлений.
- Изменения данных через представления.

**15. Хранимые процедуры.**

**16. Триггеры.**

## Модуль 4

### Использование баз данных в Python

**1. Использование баз данных.**

**2. ORM системы.**

- Теория и практика использования.
- SQLAlchemy.
- PonyORM и другие.

**3. NoSQL базы данных.**

- Основы NoSQL, теорема CAP.
- Основные типы NoSQL баз данных.
- СУБД Redis.
- СУБД MongoDB.

**4. Работа с базами данных, сериализация данных.**

- Запись и чтение в формате XML DOM.
- StAX и SAX-парсеры.

## Модуль 5

### Паттерн MVC/MVT

1. Что такое паттерн MVC?
2. Цели и задачи паттерна Model-View-Controller.
3. Model.
  - Что такое Model?
  - Цели и задачи Model.
4. View.
  - Что такое View?
  - Цели и задачи View.
5. Controller.
  - Что такое Controller?
  - Цели и задачи Controller.
6. Примеры использования паттерна MVC.

## Модуль 6

### Фреймворки

1. Классификация web-фреймворков.
  - Web-фреймворк Flask.
  - Механизм шаблонов и язык шаблонов Jinja2.
  - Сессии и формы.
  - WEB-фреймворк Bottle.
2. Асинхронные web-приложения.
  - Фреймворк Tornado.
  - Библиотека Twisted.
3. Django – фреймворк для создания web-приложений.
  - Установка Django.
  - Создание проекта Django.
  - Структура Django-проекта (url-view-model-template).
  - Паттерн MVC/MVT.
4. Модели и ORM.
  - Модели и поля.



- Связи между таблицами, проектирование и реализация БД благодаря механизму ORM.
- Миграции.
- Менеджеры модели.
- Административная часть.

## **5. Работа с административной частью, настройка отображения.**

- Настройка интерфейса администратора.
- Сортировка, фильтры, редактирование полей.
- Вывод данных.

## **6. Язык шаблонов и создание web-форм.**

- Шаблоны.
- Роутинг, представления.
- Синтаксис, логические конструкции.
- Контекстный процессор.

## **7. Создание форм и страниц.**

- Добавление страниц.
- GET/POST запросы.
- Статическое содержимое и динамическое.
- Авторизация.
- Ограничения прав доступа.
- Валидация.
- AJAX.

## **8. Стандартные задачи.**

- Встроенные class-based views.
- Использование форм с Django CBV.

## **9. Стандартные задачи.**

- Аутентификация.
- Пагинация.

## **10. Погружение в Django.**

- Middleware.
- Сигналы.
- Сообщения.
- Сессии.

## **11. Погружение в Django. Продолжение.**

- Древовидные структуры в Django.

- Миксины.
- Дебаггер Django.
- Логирование, отправка почты.

## Модуль 7

### Создание чат-ботов с помощью Python

1. Что такое чат-бот?
2. Цели и задачи чат-ботов.
3. Архитектура чат-бота.
4. Практический пример создания чат-бота.

## Модуль 8

### Экзамен