

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ  
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«УЧЕБНЫЙ ЦЕНТР СКБ КОНТУР»

Утверждаю  
Директор АНО ДПО  
«Учебный центр СКБ Контур»

 Т.В. Рубан  
1 сентября 2023 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**

дополнительная общеразвивающая программа

**DOCKER ДЛЯ ЗАПУСКА ПРИЛОЖЕНИЙ**

**Документ:** Свидетельство

**Форма обучения:** заочная с использованием дистанционных образовательных технологий, электронного обучения

**Срок обучения:** 2 недели

**Объем программы:** 20 ак. ч.

Москва, 2023 г.

## Оглавление

1. ОБЩИЕ ПОЛОЖЕНИЯ .....	3
1.1. Назначение программы.....	3
1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы .....	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы .....	4
3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА.....	5
3.1. Учебный план .....	5
3.2. Содержание.....	6
3.3 Календарный учебный график.....	8
4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ .....	9
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса .....	9
4.2. Требования к материально-техническим условиям.....	9
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы .....	11
4.4. Общие требования к организации образовательного процесса.....	12
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	12
5.1. Формы контроля знаний и требования к его проведению .....	12
5.2. Критерии оценки знаний слушателей .....	13
5.3. Фонд оценочных средств.....	14

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

### 1.1. Назначение программы

Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Docker для запуска приложений» направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет освоить навыки по основам программирования с использованием инструмента Docker.

Программа регламентирует цели, планируемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки слушателей и включает в себя: учебный план, фонды оценочных средств, программу итоговой аттестации, календарный учебный график и другие материалы, обеспечивающие качество подготовки слушателей.

### 1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

1. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ.
2. Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
3. Приказ Минпросвещения РФ от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

## 2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

### 2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Основная цель программы – дать слушателям знания и навыки по основам программирования с использованием инструмента Docker.

#### Задачи дисциплины

В процессе изучения курса «Docker для запуска приложений» необходимо решить следующие задачи и рассказать слушателям:

- о работе с контейнерами программ в Docker;
- об особенностях запуска и развертывания инфраструктуры проекта;
- о создании и оптимизации docker-образов;
- о работе с сетью;
- о применении Docker Compose;
- о запуске веб-приложений.

### 2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

К освоению дополнительной общеобразовательной программы – дополнительной общеразвивающей программы допускаются любые лица без предъявления требований к уровню образования.

Возраст слушателей: 18 лет и старше.

### 2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Нормативный срок освоения программы – 2 недели.

### 2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 20 академических часов. Из них 9 академических часов - самостоятельная работа, 11 академических часов – работа на образовательной онлайн-платформе.

## **2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы**

В результате обучения слушатели должны будут овладеть следующими знаниями, умениями и навыками:

### Знать:

- особенности применения инструмента Docker;
- основы работы с контейнерами и образами инструмента Docker;
- основы создания сервисов с Docker Compose;
- хранение данных с помощью Docker Compose.

### Уметь:

- запускать приложения в контейнере;
- выполнять оптимизацию образов;
- подключать локальные каталоги и Dev Container;
- выполнять сборку и запуск веб-приложений.

### Владеть:

- навыками применения Docker Compose для запуска и управления мультиконтейнерными приложениями;
- навыками применения инструмента Docker для запуска приложений.

### 3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

#### 3.1. Учебный план для заочной формы обучения с ДОТ и ЭО

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельн ая работа	Работа на образователь ной онлайн- платформе	
<b>1</b>	<b>Docker</b>	<b>10</b>	<b>5</b>	<b>5</b>	<b>Зачет</b>
1.1	Контейнеры	2	1	1	Тестирование
1.2	Образы	2	1	1	Тестирование
1.3	Файлы	2	1	1	Тестирование
1.4	Сеть	2	1	1	Тестирование
1.5	Оптимизация	2	1	1	Тестирование
<b>2</b>	<b>Docker Compose</b>	<b>8</b>	<b>4</b>	<b>4</b>	<b>Зачет</b>
2.1	Compose: сервисы	2	1	1	Тестирование
2.2	Compose: сеть и данные	2	1	1	Тестирование
2.3	Compose: тонкая настройка	2	1	1	Тестирование
2.4	Запуск веб-приложения	2	1	1	Тестирование
<b>ИТОГОВАЯ АТТЕСТАЦИЯ</b>		<b>2</b>	<b>–</b>	<b>2</b>	<b>Зачет</b>
<b>Всего:</b>		<b>20</b>	<b>9</b>	<b>11</b>	<b>–</b>

## 3.2. Содержание

### Тема 1. Docker

#### Урок 1.1. Контейнеры

- Знакомство с контейнерами
- Установка Docker
- Запуск приложения в контейнере
- Практическое задание: установка и удаление Docker любым удобным способом, остановка и удаление контейнера.

#### Урок 1.2. Образы

- Что такое образ
- Использование готовых образов
- Создание собственного образа
- Практическое задание: создать Dockerfile из образа ubuntu.

#### Урок 1.3. Файлы

- Способы хранения файлов
- Использование Volume
- Подключение локальных каталогов и Dev Container
- Практическое задание: создать контейнер, который сможет вывести список файлов в произвольном локальном каталоге основной операционной системы.

#### Урок 1.4. Сеть

- Сеть в Docker
- Способы организации сети
- Управление портами
- Практическое задание: используя сетевые возможности Docker, подключить веб-интерфейс к СУБД и проверить его работоспособность в браузере.

#### Урок 1.5. Оптимизация

- Выбор образа
- Используем layer caching
- Используем multi-stage builds
- Практическое задание: веб-приложение отображает демонстрационный веб-интерфейс на React, оптимизировать итоговый образ.

### Тема 2. Docker Compose

#### Урок 2.1. Compose: сервисы

- Знакомство с Docker Compose
- Создание сервисов
- Переменные окружения
- Практическая работа: создать файл compose.yaml с environment чтобы команда “docker compose up -d” запускала аналогичный сервис.

#### Урок 2.2. Compose: сеть и данные

- Файлы конфигурации и секреты
- Настройка сети
- Хранение данных
- Практическая работа: имеются два сервиса, взаимодействующих по сети. Однако база данных не сохраняется и инициализируется заново при каждом запуске сервиса. Необходимо подключить к ней том для постоянного хранения данных с помощью Docker Compose.

### Урок 2.3. Compose: тонкая настройка

- Зависимости и профили
- Переиспользование конфигурации
- Несколько файлов конфигурации
- Практическая работа: внести изменения в конфигурацию, не меняя содержимое `command`, чтобы один из сервисов вывел в терминал слово **secret**.

### Урок 2.4. Запуск веб-приложения

- Подготовка конфигурации
- Сборка и запуск
- Отладка приложения
- Практическая работа: создать `Dockerfile`, который поместит внутрь образа код и все его зависимости.

### 3.3 Календарный учебный график

Календарный график обучения является примерным, составляется и утверждается для каждой группы.

Срок освоения программы — 2 недели. Начало обучения — по мере набора группы. Примерный режим занятий: не более 8 академических часов в день, до 40 часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

Примерный график освоения программы:

№	Темы / дни	ВР	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Docker	РП	2	2	1											
		СР				2	1	1	1							
2.	Docker Compose	РП								2	1	1				
		СР											2	1	1	
<b>Итоговая аттестация</b>		РП														<b>2</b>



## **4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ**

### **4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса**

**Требования к образованию и обучению лица, занимающего должность преподавателя:** высшее образование — специалитет или магистратура, направленность (профиль) которого, как правило, соответствует преподаваемому учебному курсу, дисциплине (модулю).

**Дополнительное профессиональное образование** — профессиональная переподготовка, направленность (профиль) которой соответствует преподаваемому учебному курсу, дисциплине (модулю).

Педагогические работники обязаны проходить в установленном законодательством Российской Федерации порядке обучение и проверку знаний и навыков в области охраны труда.

Рекомендуется обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже чем один раз в три года.

**Требования к опыту практической работы:** при несоответствии направленности (профиля) образования преподаваемому учебному курсу, дисциплине (модулю) — опыт работы в области профессиональной деятельности, осваиваемой слушателями или соответствующей преподаваемому учебному курсу, дисциплине (модулю).

**Преподаватель:** стаж работы в образовательной организации не менее одного года; при наличии ученой степени (звания) — без предъявления требований к стажу работы.

**Особые условия допуска к работе:** отсутствие ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации

Прохождение обязательных предварительных (при поступлении на работу) и периодических медицинских осмотров (обследований), а также внеочередных медицинских осмотров (обследований) в порядке, установленном законодательством Российской Федерации

Прохождение в установленном законодательством Российской Федерации порядке аттестации на соответствие занимаемой должности.

### **4.2. Требования к материально-техническим условиям**

Организация проводит занятия по адресу: г. Москва, ул. Суцевский Вал, д. 18. Аудитории для занятий расположены на 11-м этаже здания.

Все занимаемые помещения соответствуют обязательным нормам пожарной безопасности и требованиям санитарно-эпидемиологических служб. Помещения имеют централизованные системы водоснабжения, отопления и канализации. Воздухообмен помещений обеспечивается современными системами кондиционирования, за счет приточно-вытяжной вентиляционной системы.

Учебным центром СКБ Контур заключен договор с организацией общественного питания о возможности обеспечения слушателей питанием.

В учебной аудитории проводятся лекции и практические занятия. Аудитория оснащена столами и стульями, в составе учебного оснащения маркерная доска и флипчарт, в случае необходимости подключается мультимедийный проектор, слушателям предоставляются компьютеры.

Компьютерная сеть учебного центра оснащена необходимым оборудованием для доступа в интернет по выделенному каналу. На каждом компьютере обеспечен постоянный доступ к компьютерной программе «Контур.Школа».

Для проведения вебинаров и онлайн-трансляций используется оснащенная современным оборудованием видеостудия:

- помещение оборудовано посадочными местами для спикера(ов);
- спикеру предоставляется персональный компьютер с соответствующими мультимедийными характеристиками (Intel Core i3 либо идентичные по характеристикам, оперативная память: от 4 Гб и выше для всех ОС), со стабильным соединением с сетью Интернет на скорости не менее 1 Мбит/с;
- видеочасть (максимальное разрешение видео — не менее 3840 x 2160).

Размещение материалов вебинаров и доступ к ним участников обеспечивает техническая платформа (сайт, система управления сайтом, другие технические средства):

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением участникам возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание участникам о предстоящем вебинаре за 1 час до начала мероприятия.
5. Использование защищенных соединений, передача и прием видео и звука по протоколам RTMP(S) или аналогичным.
6. Управление качеством и разрешением передаваемого/принимаемого видео вплоть до разрешения HD 720p на каждого участника мероприятия (адаптивный стриминг).
7. Обмен короткими текстовыми сообщениями (чат).
8. Осуществление записи мероприятий в формате, не требующем конвертации для проигрывания (mp4, AVI, WMA и т.д.).
9. Система регистрации на вебинар.
10. Техническое сопровождение проведения вебинара.
11. Отображение числа участников.
12. Техническая доступность услуги не менее 99,8% времени.
13. Устойчивость при проведении вебинара при одновременном подключении до 3000 участников.
14. Возможность участия пользователей на вебинарах в браузерах Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari с установленным плагином Adobe Flash Player.
15. Передача аудио- и видеосообщения на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

1. Размещение расписания и описания учебных программ и условий обучения.
2. Онлайн-трансляция учебных занятий с возможностью обратной связи.
3. Размещение тестов и проведение онлайн-тестирования.
4. Размещение и выбор образовательного контента и заданий для слушателей.
5. Хранение учебно-методических материалов.
6. Обратная связь слушателей к организаторам и преподавателям.
7. Автоматическая фиксация хода учебного процесса, промежуточных и итоговых результатов слушателей.
8. Хранение информации о ходе учебного процесса и результатов обучения в течение периода обучения.
9. Сбор и хранение заявок на обучение и сведений о слушателях.
10. Создание и актуализация контента и учебно-методических материалов.
11. Информационно-консультационное обслуживание слушателей.

#### **4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы**

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа обеспечивается учебно-методическими материалами по всем модулям образовательной программы.

Фонд учебно-научной библиотеки содержит основную и дополнительную учебную, учебно-методическую, научную литературу, справочно-библиографические и периодические издания (в том числе и на электронных носителях) по всем темам и дисциплинам реализуемой программы.

#### **Нормативно-правовая база**

1. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 №149-ФЗ
2. ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» (утвержден приказом Росстандарта от 30 ноября 2010 г. N 631-ст);
3. ГОСТ Р ИСО/МЭК 14764-2002 «Информационная технология. Сопровождение программных средств» (принят постановлением Госстандарта РФ от 25 июня 2002 г. N 248-ст);
4. ГОСТ Р ИСО/МЭК 90003-2014 «Разработка программных продуктов. Руководящие указания по применению ИСО 9001:2008 при разработке программных продуктов» (утвержден приказом Росстандарта от 23 октября 2014 г. N 1405-ст).

#### **Список литературы**

1. Docker на практике / [Иан Милл, Эйдан Хобсон Сейерс]; М.: ДМК Пресс, 2020 г. - 516 с.
2. Программирование Cloud Native. Микросервисы, Docker и Kubernetes / [И. Портянкин]; «Издательские решения», 2022 г. – 45 с.
3. Разработка требований к программному обеспечению. 3-е изд., дополненное / [Вигерс К., Битти Дж.]; БХВ; 2023 г. - 736 с.

#### **Периодические издания**

1. Журнал «Вестник компьютерных и информационных технологий», №2, 2023г.  
<http://www.vkit.ru/index.php/archive-rus/1228-02-february>

2. Научно-практический журнал «Программные продукты и системы» №1, 2023г.  
<http://www.swsys.ru/index.php>

#### **Интернет-ресурсы**

1. Docker: трюки и хитрости // Дзен, 2023. – URL: <https://dzen.ru/a/ZEzeo1wvp0QIK29e>
2. Основы контейнеризации (обзор Docker и Podman) // Хабр, 2023. – URL: <https://habr.com/ru/articles/659049/>
3. Dockerize Python: создаём образ Docker из приложения на Python// Tproger.ru, 2022. – URL: <https://tproger.ru/articles/dockerize-python-sozdajom-obraz-docker-iz-prilozhenija-na-python>

#### **4.4. Общие требования к организации образовательного процесса.**

При реализации дополнительной общеобразовательной программы – дополнительной общеразвивающей программы используется лекционно-семинарская система, практические занятия, индивидуальные и групповые консультации, мастер-классы и другие интерактивные формы обучения. Интегративно-дифференцированная организация занятий в процессе подготовки слушателей позволяет не только адаптировать образовательный процесс к индивидуальным особенностям и интересам слушателей, но и обеспечить свободу выбора и вариативность образования, сформировать у слушателей стремление к самообразованию, способствовать решению индивидуальных задач развития личности.

Особую значимость на уровне технологии обучения имеет организационно-педагогическое условие, предполагающее организацию самостоятельной работы слушателей как средство формирования профессиональных компетенций. Самостоятельная работа слушателей является обязательной составляющей образовательного процесса подготовки слушателей дополнительной общеобразовательной программы – дополнительной общеразвивающей программы.

#### **Применение дистанционных образовательных технологий и электронного обучения**

В учебном процессе могут использоваться следующие организационные формы учебной деятельности:

- обзорные (установочные) лекции с использованием дистанционных образовательных технологий;
- самостоятельная работа с СДО: работа с электронным учебником, просмотр видеолекций и др.;
- самостоятельная работа с программами контроля знаний (тестами);
- консультации (индивидуальные с применением электронных средств, групповые и предэкзаменационные);
- семинары с использованием дистанционных образовательных технологий;
- коллоквиумы;
- итоговые аттестационные работы.

### **5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

#### **5.1. Формы контроля знаний и требования к его проведению**

Для проведения промежуточной и итоговой аттестации программы разработан фонд оценочных средств по программе, являющийся неотъемлемой частью учебно-методического комплекса.

**Объектами оценивания выступают:**

- степень освоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы, активность на занятиях.

**Текущий контроль знаний** слушателей проводится преподавателем, ведущим занятия в учебной группе, на протяжении всего обучения по программе.

Текущий контроль знаний включает в себя наблюдение преподавателя за учебной работой слушателей и проверку качества знаний, умений и навыков, которыми они овладели на определенном этапе обучения посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем.

**Промежуточная аттестация** — оценка качества усвоения слушателями содержания учебных блоков непосредственно по завершении их освоения, проводимая в форме зачета посредством тестирования или в иных формах, в соответствии с учебным планом и учебно-тематическим планом.

**Итоговая аттестация** — процедура, проводимая с целью установления уровня знаний, слушателей с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы. Итоговая аттестация слушателей осуществляется в форме зачета посредством тестирования.

Слушатель допускается к итоговой аттестации после изучения тем образовательной программы в объеме, предусмотренном для лекционных и практических занятий.

Слушателям, освоившим дополнительную общеобразовательную программу - дополнительную общеразвивающую программу по теме «Docker для запуска приложений», и успешно прошедшим итоговую аттестацию, выдается свидетельство установленного образца с указанием названия программы, календарного периода обучения, длительности обучения в академических часах.

Для аттестации слушателей на соответствие их персональных достижений требованиям соответствующей ОП созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

Фонды оценочных средств соответствуют целям и задачам программы подготовки специалиста, учебному плану и обеспечивают оценку качества общепрофессиональных и профессиональных компетенций, приобретаемых слушателями.

## 5.2. Критерии оценки знаний слушателей

Предмет оценивания (компетенции)	Объект оценивания (навыки)	Показатель оценки (знания, умения)
Основы работы с инструментом Docker	Использование инструмента Docker для запуска программ	<p><b>Знания:</b></p> <ul style="list-style-type: none"> <li>• Запуск приложения в контейнере</li> <li>• Использование готовых образов</li> <li>• Создание собственного образа</li> <li>• Способы хранения файлов</li> <li>• Способы организации сети</li> <li>• Особенности управления портами</li> </ul> <p><b>Умения:</b></p>

		<ul style="list-style-type: none"> <li>• Производить установку и удаление Docker любым удобным способом</li> <li>• Запускать приложения в контейнере</li> <li>• Выполнять оптимизацию образов</li> <li>• Подключать локальные каталоги и Dev Container</li> </ul>
Основы работы с инструментом Docker Compose	Применение Docker Compose для запуска и управления мультиконтейнерными приложениями	<p><b>Знания:</b></p> <ul style="list-style-type: none"> <li>• Создание сервисов с Docker Compose</li> <li>• Настройка сети</li> <li>• Хранение данных с помощью Docker Compose</li> <li>• Переиспользование конфигурации, изменения в конфигурации</li> <li>• Сборка и запуск веб-приложения</li> </ul> <p><b>Умения:</b></p> <ul style="list-style-type: none"> <li>• Применять Docker Compose для запуска и управления мультиконтейнерными приложениями</li> <li>• Проводить настройку сервиса базы данных при помощи переменных окружения</li> <li>• Вносить изменения в конфигурацию</li> <li>• Выполнять сборку и запуск веб-приложения</li> </ul>

### Критерии оценки аттестации – зачета:

1. Оценка «**Зачтено**» выставляется слушателю, продемонстрировавшему твердое и всесторонние знания материалы, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации демонстрировали отличный уровень знаний и умений слушателей. Не менее 80% правильных ответов при решении тестов.

2. Оценка «**Не зачтено**» выставляется слушателю, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации демонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

### 5.3. Фонд оценочных средств

Тест к уроку 1.1.

1. Какой из перечисленных компонентов всегда будет общим для контейнеров, запущенных на одной операционной системе?
  - a. Виртуальная память (user space)
  - b. Сеть
  - c. Ядро
  - d. Дисковое пространство
2. Потребление какого ресурса можно ограничить в контейнерах?
  - a. Виртуальная память
  - b. CPU

- c. Дисковое пространство
  - d. Всех перечисленных
3. Какой из перечисленных инструментов не является реализацией контейнеров?
    - a. LXC
    - b. Docker
    - c. Hyperkit
    - d. OVZ
  4. Какой способ установки Docker работает на операционных системах GNU/Linux, Windows, macOS?
    - a. Docker Desktop
    - b. Docker Engine
    - c. Colima
    - d. Все перечисленные
  5. Какая операционная система позволяет запустить Docker на текущем ядре?
    - a. Windows
    - b. GNU/Linux
    - c. macOS
    - d. Все перечисленные
  6. Как создать и запустить новый контейнер из образа?
    - a. docker container
    - b. docker exec
    - c. docker run
    - d. docker start
  7. Как выполнить произвольную команду в запущенном контейнере?
    - a. docker container
    - b. docker exec
    - c. docker run
    - d. docker start

Тест к уроку 1.2.

1. Что такое образ?
  - a. Дополнительный экземпляр пользовательского пространства, программно изолированный от основной операционной системы и других образов
  - b. Специальный шаблон, который содержит набор файлов и инструкций для создания контейнера
  - c. Текстовый файл с набором последовательных инструкций по созданию контейнера
  - d. Текстовая метка, которая идентифицирует версию или какой-то специальный вариант контейнера
2. Можно ли изменить содержимое слоя?
  - a. Да
  - b. Можно только удалять файлы
  - c. Нет
  - d. Нет, но можно добавить новый слой
3. Что такое Docker Registry?
  - a. Специальный сервис, который хранит готовые образы Docker
  - b. Специальный шаблон, который содержит набор файлов и инструкций для создания контейнера
  - c. Дополнительный экземпляр пользовательского пространства, программно изолированный от основной операционной системы и других Docker Registry

- d. Текстовый файл с набором последовательных инструкций по созданию контейнера
4. С помощью какой команды можно скачать образ?
    - a. docker image pull
    - b. docker image download
    - c. docker image copy
    - d. docker image save
  5. Что такое Dockerfile?
    - a. Файл образа Docker
    - b. Текстовый файл с набором последовательных инструкций по созданию образа Docker
    - c. Специальный модуль для управления файлами внутри контейнера
    - d. Специальный шаблон, который содержит набор файлов и инструкций для создания контейнера
  6. С какой инструкции начинается создание образа?
    - a. RUN
    - b. CMD
    - c. FROM
    - d. ENTRYPOINT
  7. С помощью какой команды можно опубликовать образ?
    - a. docker image publish
    - b. docker image deploy
    - c. docker image push
    - d. docker image release

Тест к уроку 1.3.

1. Можно ли сохранить файл в образе?
  - a. Да, при запуске контейнера
  - b. Да, при создании контейнера
  - c. Да, при создании образа
  - d. Нет
2. Чем отличаются инструкции COPY и ADD?
  - a. Инструкция COPY работает только с простыми локальными файлами, ADD — с URL и архивами
  - b. Ничем не отличаются
  - c. Инструкция COPY копирует локальные файлы в образ, ADD — создает новые
  - d. Инструкция COPY копирует файлы, уже существующие внутри образа, ADD — добавляет локальные файлы в образ
3. Чем отличаются опции --volume и --mount?
  - a. Опция --volume используется для подключения volumes, опция --mount — для bind mounts
  - b. Ничем не отличаются
  - c. Опция --mount позволяет более детально настроить параметры монтирования
  - d. Опция --volume создает том, опция --mount подключает его к контейнеру
4. Где физически расположены файлы volumes?
  - a. В текущем каталоге
  - b. В оперативной памяти
  - c. Зависит от операционной системы и версии Docker
  - d. Пользователь указывает расположение при монтировании
5. Что будет, если подключить том к нескольким контейнерам?
  - a. Том будет подключен только к первому контейнеру



- b. Том будет подключен только к последнему контейнеру
  - c. Первый контейнер сможет менять содержимое тома, последующие — только читать
  - d. Контейнеры получают совместный доступ к файлам
6. Где физически расположены файлы bind mounts?
- a. В текущем каталоге
  - b. В оперативной памяти
  - c. Зависит от операционной системы и версии Docker
  - d. Пользователь указывает расположение при монтировании
7. Что такое dev-контейнер?
- a. Подход для запуска экспериментального кода, который пишется строго внутри и будет автоматически удален после остановки контейнера
  - b. Среда разработки (IDE), запущенная внутри контейнера
  - c. Контейнер с заранее подготовленным окружением для запуска исходного кода
  - d. Контейнер Docker, находящийся в процессе разработки и не готовый к запуску

Тест к уроку 1.4.

1. Какой инструмент Docker использует для управления сетями?
  - a. VPNKit
  - b. gVisor
  - c. Network namespaces
  - d. Все перечисленные в зависимости от способа установки и используемой операционной системы
2. Какая сеть по умолчанию доступна после установки Docker?
  - a. Bridge
  - b. Host
  - c. None
  - d. Доступны все перечисленные
3. Какой сетевой драйвер не поддерживается в Windows?
  - a. Bridge
  - b. Host
  - c. None
  - d. Поддерживаются все перечисленные
4. Какая сеть поддерживает обращение по именам контейнеров?
  - a. Стандартная сеть bridge
  - b. Пользовательская сеть с драйвером bridge
  - c. Любая сеть с драйвером bridge
  - d. Стандартная сеть None
5. Какая сеть поддерживает обращение контейнера к самому себе по имени localhost?
  - a. Стандартная сеть bridge
  - b. Пользовательская сеть с драйвером bridge
  - c. Стандартная сеть None
  - d. Поддерживают все перечисленные
6. Что делает инструкция EXPOSE?
  - a. Назначает на указанные порты внешние порты со случайными номерами
  - b. Назначает на указанные порты внешние порты с теми же номерами
  - c. Указывает, какие порты необходимо пробросить. Может быть использована с опцией -P для назначения внешних портов со случайными номерами
  - d. Указывает, какие порты необходимо пробросить. Может быть использована с опцией -P для назначения внешних портов с теми же номерами

7. С помощью какой команды можно узнать список опубликованных портов контейнера?
  - a. docker port
  - b. docker ports
  - c. docker expose
  - d. docker publish

Тест к уроку 1.5.

1. Как называется стабильная версия программного продукта с увеличенным сроком поддержки?
  - a. TLS
  - b. SSL
  - c. LTS
  - d. TDD
2. Какой из перечисленных образов Python будет самым легковесным?
  - a. python:3.12-bookworm
  - b. python:3.12-slim
  - c. python:3.12-slim-bullseye
  - d. python:3.12-alpine
3. Механизм layer caching позволяет...
  - a. повторно использовать слои при сборке того же образа
  - b. повторно использовать слои при сборке на той же системе
  - c. хранить временные файлы в слоях образа
  - d. повторно использовать слои при сборке на разных системах
4. Можно ли использовать кэш между разными экземплярами Docker?
  - a. Нет
  - b. Да, при помощи Docker Registry
  - c. Да, при помощи layers cache
  - d. Да, при помощи overlay network
5. В каком файле можно указать список шаблонов для пропуска файлов при копировании в образ?
  - a. .dockernocopy
  - b. .dockerskip
  - c. .dockerblacklist
  - d. .dockerignore
6. Удаление файла в инструкции RUN уменьшит размер образа?
  - a. Да
  - b. Нет
  - c. Да, если файл будет удален в той же инструкции, в которой был создан
  - d. Нет, если файл будет удален в той же инструкции, в которой был создан
7. В следующем этапе multi-stage builds в качестве базового образа можно использовать...
  - a. только образ из Docker Registry
  - b. только образ из предыдущего этапа
  - c. образ из любого предшествующего этапа
  - d. образ из Docker Registry или из любого предшествующего этапа

Тест к теме 1. «Docker»

1. Какой из перечисленных инструментов не является реализацией контейнеров?
  - a. LXC
  - b. Docker
  - c. Hyperkit

- d. OVZ
- 2. Как выполнить произвольную команду в запущенном контейнере?
  - a. docker container
  - b. docker exec
  - c. docker run
  - d. docker start
- 3. Что такое образ?
  - a. Дополнительный экземпляр пользовательского пространства, программно изолированный от основной операционной системы и других образов
  - b. Специальный шаблон, который содержит набор файлов и инструкций для создания контейнера
  - c. Текстовый файл с набором последовательных инструкций по созданию контейнера
  - d. Текстовая метка, которая идентифицирует версию или какой-то специальный вариант контейнера
- 4. Что такое Dockerfile?
  - a. Файл образа Docker
  - b. Текстовый файл с набором последовательных инструкций по созданию образа Docker
  - c. Специальный модуль для управления файлами внутри контейнера
  - d. Специальный шаблон, который содержит набор файлов и инструкций для создания контейнера
- 5. С помощью какой команды можно опубликовать образ?
  - a. docker image publish
  - b. docker image deploy
  - c. docker image push
  - d. docker image release
- 6. Чем отличаются инструкции COPY и ADD?
  - a. Инструкция COPY работает только с простыми локальными файлами, ADD — с URL и архивами
  - b. Ничем не отличаются
  - c. Инструкция COPY копирует локальные файлы в образ, ADD — создает новые
  - d. Инструкция COPY копирует файлы, уже существующие внутри образа, ADD — добавляет локальные файлы в образ
- 7. Что будет, если подключить том к нескольким контейнерам?
  - a. Том будет подключен только к первому контейнеру
  - b. Том будет подключен только к последнему контейнеру
  - c. Первый контейнер сможет менять содержимое тома, последующие — только читать
  - d. Контейнеры получают совместный доступ к файлам
- 8. Какой инструмент Docker использует для управления сетями?
  - a. VPNKit
  - b. gVisor
  - c. Network namespaces
  - d. Все перечисленные в зависимости от способа установки и используемой операционной системы
- 9. Что делает инструкция EXPOSE?
  - a. Назначает на указанные порты внешние порты со случайными номерами
  - b. Назначает на указанные порты внешние порты с теми же номерами

- c. Указывает, какие порты необходимо пробросить. Может быть использована с опцией -P для назначения внешних портов со случайными номерами
  - d. Указывает, какие порты необходимо пробросить. Может быть использована с опцией -P для назначения внешних портов с теми же номерами
10. С помощью какой команды можно узнать список опубликованных портов контейнера?
- a. docker port
  - b. docker ports
  - c. docker expose
  - d. docker publish
11. Как называется стабильная версия программного продукта с увеличенным сроком поддержки?
- a. TLS
  - b. SSL
  - c. LTS
  - d. TDD
12. Удаление файла в инструкции RUN уменьшит размер образа?
- a. Да
  - b. Нет
  - c. Да, если файл будет удален в той же инструкции, в которой был создан
  - d. Нет, если файл будет удален в той же инструкции, в которой был создан

Тест к уроку 2.1.

1. Какая новая сущность появляется в Docker Compose?
  - a. Image
  - b. Service
  - c. Volume
  - d. Network
2. Какой язык используется для файла конфигурации Compose?
  - a. XML
  - b. JSON
  - c. YAML
  - d. TOML
3. Что из перечисленного входит в возможности Docker Compose?
  - a. Создание изолированных окружений
  - b. Переносимая конфигурация
  - c. Поддержка переменных окружения
  - d. Compose обладает всеми перечисленными возможностями
4. Какая команда запустит все сервисы, перечисленные в файле Compose?
  - a. docker compose exec
  - b. docker compose up
  - c. docker compose create
  - d. docker compose push
5. При каких значениях "pull\_policy" образ может быть скачан из Docker Registry?
  - a. always
  - b. missing
  - c. if\_not\_present
  - d. При всех перечисленных
6. Какой из перечисленных способов задания переменных окружения обладает наивысшим приоритетом?
  - a. Инструкции ENV или ARG в Dockerfile

- b. Файл “.env”
  - c. `docker compose run -e VARIABLE=value`
  - d. Поле “environment” в файле Compose
7. В каких случаях в переменных окружения работают управляющие символы включающие в себя “\n”, “\r”, “\t” и “\”?
- a. Внутри двойных кавычек
  - b. Внутри одиночных кавычек
  - c. Без кавычек
  - d. Во всех перечисленных случаях

Тест к уроку 2.2.

1. Для чего нужен атрибут `configs` в `service`?
  - a. Для настройки параметров сервиса
  - b. Для переиспользования параметров между сервисами
  - c. Для подключения файлов конфигурации к сервису
  - d. В `service` нет такого атрибута
2. В какой каталог по умолчанию монтируются секреты?
  - a. `/secrets`
  - b. `/run/secrets`
  - c. `/var/secrets`
  - d. `/usr/share/secrets`
3. Как разделить сервисы на несколько изолированных подсетей?
  - a. Поместить сервисы в разные проекты Compose.
  - b. Создать несколько сетей с помощью “networks” и подключить к ним конкретные сервисы
  - c. Оба способа будут работать
  - d. Сервисы Docker Compose подключенные к сети нельзя изолировать друг от друга
4. К какой сети по-умолчанию подключаются сервисы Docker Compose?
  - a. К сети `bridge`, которая создается при запуске Docker Engine
  - b. К сети `host`, которая создается при запуске Docker Engine
  - c. К сети `bridge` с названием “projectname\_default”, которая создается при запуске Docker Compose
  - d. Сервисы Docker Compose по-умолчанию изолированы друг от друга
5. Как изолировать от сети отдельные сервисы?
  - a. Создать и подключить к сервисам сеть с драйвером `none`
  - b. Использовать атрибут “network\_mode” со значением “none”
  - c. Использовать пустой атрибут “networks”
  - d. Просто не добавлять к этим сервисам атрибут “networks”
6. Для чего нужен атрибут `external`?
  - a. Для отключения изоляции ресурсов Docker (`networks`, `volumes`)
  - b. Для подключения уже существующих ресурсов, которые не управляются Compose
  - c. Для создания общих ресурсов между сервисами одной конфигурации Compose
  - d. Для создания общих ресурсов между разными серверами Docker
7. Как подключить локальный каталог к сервису?
  - a. С помощью атрибута `mounts`
  - b. С помощью атрибута `binds`
  - c. С помощью атрибута `volumes`
  - d. К сервису Docker Compose нельзя подключить локальный каталог

Тест к уроку 2.3.

1. Сервис “first” зависит (depends\_on) от “second”. Сервис “third” зависит от “first”. В каком порядке будут запущены сервисы?
  - a. first, second, third
  - b. second, first, third
  - c. third, second, first
  - d. third, first, second
2. Сервис “first” зависит (depends\_on) от “second” и “third”. Что произойдет при запуске команды “docker compose up first”?
  - a. Запустится только сервис “first”
  - b. Запустятся все три сервиса. Сначала “first”, затем “second” и “third”
  - c. Запустятся все три сервиса. Сначала “second” и “third”, затем “first”
  - d. Compose вернет ошибку
3. Как дождаться полной готовности сервиса используя depends\_on?
  - a. Использовать service\_healthy из расширенной конфигурации
  - b. Использовать service\_completed\_successfully из расширенной конфигурации
  - c. Использовать service\_started из расширенной конфигурации
  - d. Compose всегда дожидается полной готовности сервиса указанного в depends\_on
4. Что произойдет при выполнении команды “docker compose up”, если всем сервисам заданы профили?
  - a. Все сервисы будут запущены
  - b. Не запустится ни один сервис
  - c. Запустятся сервисы с самым первым профилем встречающимся в файле Compose
  - d. Запустятся сервисы, для которых указан только один профиль
5. К каким типам записей можно применить YAML merge (<<:)?
  - a. Только к объектам
  - b. Только к спискам
  - c. К объектам и спискам
  - d. К любым
6. Какие сервисы можно указывать в “extends”?
  - a. Любые из текущего файла
  - b. Любые из внешнего файла
  - c. Любые из текущего и внешнего файлов
  - d. Сервисы с таким же именем из внешнего файла
7. Что произойдет, если файл подключенный при помощи “include” содержит сервис с таким же именем, как основной файл?
  - a. Будет использован сервис из подключенного файла
  - b. Будет использован сервис из текущего файла
  - c. Будет произведено слияние двух сервисов в один
  - d. Compose вернет ошибку

Тест к уроку 2.4.

1. Какую версию компонентов лучше всего использовать при создании приложений?
  - a. Последнюю доступную
  - b. Предыдущую доступную
  - c. Актуальную стабильную или LTS
  - d. Предыдущую стабильную или LTS
2. Какой подход следует использовать при контейнеризации приложений?
  - a. Один сервис — один контейнер
  - b. Один контейнер — один процесс

- c. Один контейнер — один volume
  - d. Одна сеть — один контейнер
3. Почему для работы с PostgreSQL лучше выбрать бинарную (прекомпиленную) версию psycopg2-binary?
    - a. Обычная версия не устанавливается в Docker
    - b. Для нее доступны более свежие версии
    - c. Она быстрее работает
    - d. Чтобы не тратить время на сборку из исходного кода
  4. Нужно ли создавать виртуальное окружение Python внутри контейнера Docker, например, при использовании Poetry?
    - a. Нет, так как виртуальное окружение Python не работает внутри контейнера
    - b. Да, так как Python не работает в Docker без виртуального окружения
    - c. Нет, так как виртуальное окружение Python служит для изоляции и внутри контейнера не имеет смысла
    - d. Да, чтобы использовать определенную версию Python внутри контейнера
  5. Для чего нужен пакет django-environ?
    - a. Для работы с виртуальным окружением Python
    - b. Для управления переменными окружения из приложения Django
    - c. Чтобы вынести отдельные настройки проекта Django в переменные окружения
    - d. Для установки готовых приложений Django
  6. Как запустить произвольную команду внутри сервиса <service>?
    - a. docker compose <service> run <command>
    - b. docker compose run <service> <command>
    - c. docker compose <service> <command>
    - d. docker compose up <service> <command>
  7. Можно ли в PyCharm использовать для отладки точки останова в Python-коде запущенном с помощью Docker Compose?
    - a. Можно
    - b. Нельзя
    - c. Можно, если приложение запущено из PyCharm
    - d. Можно, если приложение запущено из PyCharm в режиме отладки

#### Тест к теме 2. «Docker Compose»

1. Что из перечисленного входит в возможности Docker Compose?
  - a. Создание изолированных окружений
  - b. Переносимая конфигурация
  - c. Поддержка переменных окружения
  - d. Compose обладает всеми перечисленными возможностями
2. При каких значениях “pull\_policy” образ может быть скачан из Docker Registry?
  - a. always
  - b. missing
  - c. if\_not\_present
  - d. При всех перечисленных
3. В какой каталог по умолчанию монтируются секреты?
  - a. /secrets
  - b. /run/secrets
  - c. /var/secrets
  - d. /usr/share/secrets
4. Как изолировать от сети отдельные сервисы?
  - a. Создать и подключить к сервисам сеть с драйвером none
  - b. Использовать атрибут “network\_mode” со значением “none”

- c. Использовать пустой атрибут “networks”
  - d. Просто не добавлять к этим сервисам атрибут “networks”
5. Сервис “first” зависит (depends\_on) от “second”. Сервис “third” зависит от “first”. В каком порядке будут запущены сервисы?
    - a. first, second, third
    - b. second, first, third
    - c. third, second, first
    - d. third, first, second
  6. Какие сервисы можно указывать в “extends”?
    - a. Любые из текущего файла
    - b. Любые из внешнего файла
    - c. Любые из текущего и внешнего файлов
    - d. Сервисы с таким же именем из внешнего файла
  7. Какой подход следует использовать при контейнеризации приложений?
    - a. Один сервис — один контейнер
    - b. Один контейнер — один процесс
    - c. Один контейнер — один volume
    - d. Одна сеть — один контейнер
  8. Как запустить произвольную команду внутри сервиса <service>?
    - a. docker compose <service> run <command>
    - b. docker compose run <service> <command>
    - c. docker compose <service> <command>
    - d. docker compose up <service> <command>

#### Итоговое тестирование

1. Какой из перечисленных инструментов не является реализацией контейнеров?
  - a. LXC
  - b. Docker
  - c. Hyperkit
  - d. OVZ
2. Как создать и запустить новый контейнер из образа?
  - a. docker container
  - b. docker exec
  - c. docker run
  - d. docker start
3. Можно ли изменить содержимое слоя?
  - a. Да
  - b. Можно только удалять файлы
  - c. Нет
  - d. Нет, но можно добавить новый слой
4. С какой инструкции начинается создание образа?
  - a. RUN
  - b. CMD
  - c. FROM
  - d. ENTRYPOINT
5. Можно ли сохранить файл в образе?
  - a. Да, при запуске контейнера
  - b. Да, при создании контейнера
  - c. Да, при создании образа
  - d. Нет
6. Чем отличаются опции --volume и --mount?



- a. Опция `--volume` используется для подключения `volumes`, опция `--mount` — для `bind mounts`
  - b. Ничем не отличаются
  - c. Опция `--mount` позволяет более детально настроить параметры монтирования
  - d. Опция `--volume` создает том, опция `--mount` подключает его к контейнеру
7. Какой сетевой драйвер не поддерживается в Windows?
- a. Bridge
  - b. Host
  - c. None
  - d. Поддерживаются все перечисленные
8. Какая сеть поддерживает обращение по именам контейнеров?
- a. Стандартная сеть `bridge`
  - b. Пользовательская сеть с драйвером `bridge`
  - c. Любая сеть с драйвером `bridge`
  - d. Стандартная сеть `None`
9. Механизм `layer caching` позволяет...
- a. повторно использовать слои при сборке того же образа
  - b. повторно использовать слои при сборке на той же системе
  - c. хранить временные файлы в слоях образа
  - d. повторно использовать слои при сборке на разных системах
10. В следующем этапе `multi-stage builds` в качестве базового образа можно использовать...
- a. только образ из `Docker Registry`
  - b. только образ из предыдущего этапа
  - c. образ из любого предшествующего этапа
  - d. образ из `Docker Registry` или из любого предшествующего этапа
11. Какая новая сущность появляется в `Docker Compose`?
- a. Image
  - b. Service
  - c. Volume
  - d. Network
12. Какая команда запустит все сервисы, перечисленные в файле `Compose`?
- a. `docker compose exec`
  - b. `docker compose up`
  - c. `docker compose create`
  - d. `docker compose push`
13. Для чего нужен атрибут `configs` в `service`?
- a. Для настройки параметров сервиса
  - b. Для переиспользования параметров между сервисами
  - c. Для подключения файлов конфигурации к сервису
  - d. В `service` нет такого атрибута
14. Как подключить локальный каталог к сервису?
- a. С помощью атрибута `mounts`
  - b. С помощью атрибута `binds`
  - c. С помощью атрибута `volumes`
  - d. К сервису `Docker Compose` нельзя подключить локальный каталог
15. Что произойдет при выполнении команды `“docker compose up”`, если всем сервисам заданы профили?
- a. Все сервисы будут запущены
  - b. Не запустится ни один сервис

- c. Запустятся сервисы с самым первым профилем встречающимся в файле Compose
  - d. Запустятся сервисы, для которых указан только один профиль
16. К каким типам записей можно применить YAML merge (<<:)?
- a. Только к объектам
  - b. Только к спискам
  - c. К объектам и спискам
  - d. К любым
17. Какую версию компонентов лучше всего использовать при создании приложений?
- a. Последнюю доступную
  - b. Предыдущую доступную
  - c. Актуальную стабильную или LTS
  - d. Предыдущую стабильную или LTS
18. Для чего нужен пакет django-environ?
- a. Для работы с виртуальным окружением Python
  - b. Для управления переменными окружения из приложения Django
  - c. Чтобы вынести отдельные настройки проекта Django в переменные окружения
  - d. Для установки готовых приложений Django