

# Веб-программирование магистрам. Задания

## Оглавление

HTML.....	2
CSS.....	4
JavaScript.....	7
docker.....	9
nginx.....	10
Исследование протокола OAuth2.....	11
Логирование.....	13
XML.....	14
s3.....	15

## Задание 1

# HTML

Необходимо создать файлы следующих «типов»:

1. Текстовый файл
2. Текстовый файл на хостинге
3. html-файл
4. html-файл на хостинге
5. Подключение css
6. php-программа локально (по возможности)
7. php-программа на хостинге

Содержимое файлов — по вариантам.

Доступ к хостингу:

```
ssh -p 20022 stud20@kdenisb.ru  
пароль Zz8Gws4Vv
```

Размещение «сайта»:

```
/var/www/html/web
```

URL «сайта»:

```
http://kdenisb.ru:20080/web/
```

Создавайте для себя отдельные директории (прост. - «папка»), дублируйте все файлы себе локально для отчетности.

## Варианты

Предметная область	Разметка		
	Списки <a href="http://htmlbook.ru/html/ul">http://htmlbook.ru/html/ul</a>	Таблицы <a href="http://htmlbook.ru/html/table">http://htmlbook.ru/html/table</a>	Список определений <a href="http://htmlbook.ru/html/dt">http://htmlbook.ru/html/dt</a>
меню	0	1	2
телефонный справочник	3	4	5
список студентов	6	7	8
список преподавателей	9	10	11
дисциплины	12	13	14
адреса	15	16	17
расписание	18	19	20
оборудование	21	22	23
библиотека	24	25	26
товары	27	28	29

todo	30	31	32
зоопарк	33	34	35
фильмотека	36	37	38
версии ПО			
страны			
депутаты			

## Задание 2

# CSS

Необходимо построить семантическую сеть для html/css верстки.

Порядок работы:

1. Сделать html-файл (3-5 тэгов с контентом)
2. Сделать и подключить к html-файлу css-файл (4-5 селекторов, 5-6 свойств)
3. Построить семантическую сеть контента, тэгов, классов, идентификаторов и свойств
4. Построить на семантической сети пути наследования/каскадирования css-свойств (3 штуки)

Варианты контента

Предметная область	Варианты	
меню	1	17
телефонный справочник	2	18
список студентов	3	19
список преподавателей	4	20
дисциплины	5	21
адреса	6	22
расписание	7	23
оборудование	8	24
библиотека	9	25
товары	10	26
todo	11	27
зоопарк	12	28
фильмотека	13	29
версии ПО	14	30
страны	15	31
депутаты	16	32

Пример выполнения

html

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <title></title>
```

```

    <link href="s.css" rel="stylesheet"/>
</head>
<body>
  <div>Взросление</div>
  <div class="edu">Школьник</div>
  <div class="edu hse" id="stud">Студент
    <div id="mag">Магистрант</div>
  </div>

</body>
</html>

```

s.css

```

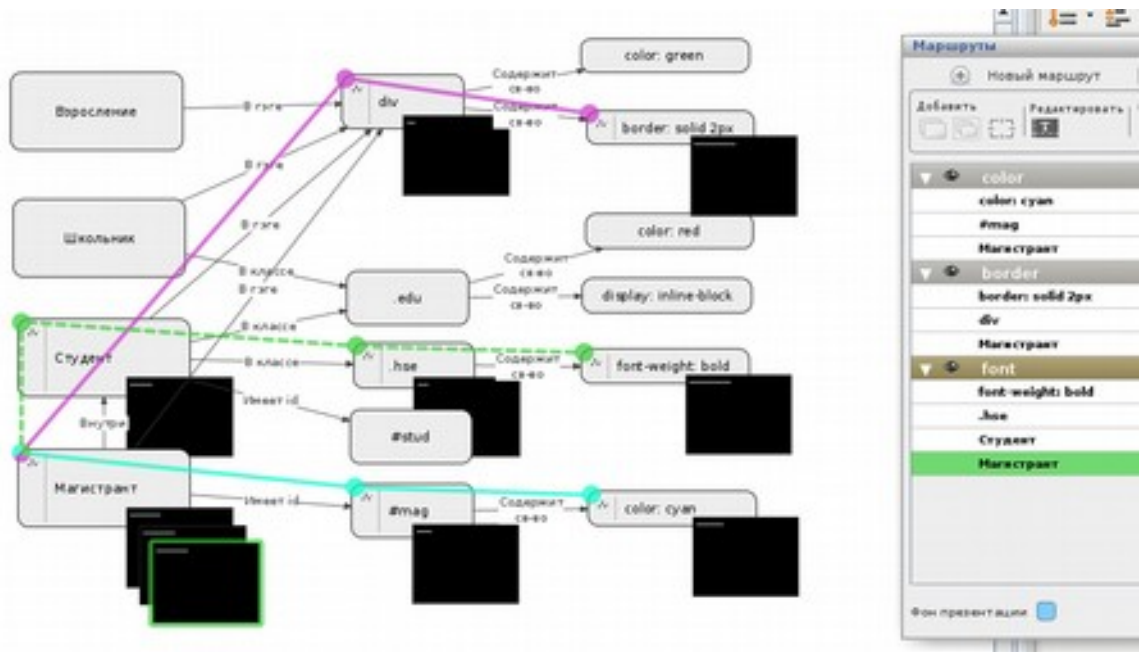
#mag {
  color: cyan;
}

.edu {
  color: red;
  display: inline-block;
}

div {
  color: green;
  border: solid 2px;
}

.hse {
  font-weight: bold;
}

```



### Задание 3

# JavaScript

Необходимо написать и сделать вызов функций в javascript.

Виды функций:

1. Обычная
2. Обычная без имени
3. Стрелочная
4. Стрелочная без имени
5. Замыкание

Дополнительные указания

Должны быть созданы и задействованы локальные переменные в функции.

Контролировать область видимости переменных в функции, дополнительно создавая одноименные глобальные переменные.

Варианты функций

Предметная область	Варианты	
Сумма двух чисел	1	17
Разность двух чисел	2	18
Сравнение двух чисел	3	19
Произведение двух чисел	4	20
Максимум из двух чисел	5	21
Возведение первого числа в степень равную второму числу	6	22
Частное двух чисел	7	23
Конкатенация двух строк	8	24
Сравнение двух строк	9	25
Сумма элементов массива	10	26
Поиск минимума в массиве	11	27
Упорядочивание по возрастанию двух чисел	12	28
Произведение элементов массива	13	29
Поиск максимума в массиве	14	30
Минимум из двух чисел	15	31
Определение числа элементов в массиве	16	32

Пример выполнения (для замыканий)

```
<body>
  <script>
    var ret=-99;

    function f1() {
      ret++;
      let r=ret;
      return function() {
        let ret;
        if (r>=0) ret=r;
        else if (r<=0) ret=-r;
        else ret = -1;
        return ret;
      }
    }
    ret++;
    x=f1();
    ret++;
    ret++;
    ret++;
    xx=f1();
    y=x();
    yy=xx();

    document.write("modul ot " + ret + " = "
      + y + " = " + yy);

  </script>
```

## Задание 4

# docker

Необходимо создать докер nginx

Порядок действий:

- 1 Установить утилиту «docker»
- 2 Создать докер-контейнер nginx
  - 2.1 Скачать образ
  - 2.2 Пробросить порт 80 «наружу»
  - 2.3 Пробросить каталог с «приложением»
  - 2.4 Пробросить конфигурационный файл nginx в контейнер
- 3 Сделать текстовый файл в качестве «приложения»
- 4 Проверить функционирование «приложения»

Полезные команды:

- Запустить команду bash в контейнере: «docker exec -it имяКонтейнера bash»
- Создать и запустить контейнер: «docker run --name имяКонтейнера -v каталогХоста:каталогДокера -p портХоста:портДокера названиеОбразаДокера:версия»
- Список запущенных докеров: «docker ps»
- Запустить ранее созданного докера: «docker start имяКонтейнера»

Полезные ссылки:

<https://docs.docker.com>

<https://docs.microsoft.com/ru-ru/virtualization/windowscontainers/quick-start/set-up-environment?tabs=Windows-10-Client>



## Задание 5

# nginx

Видеоурок <https://youtu.be/n6TQ1JdVeeo>

Необходимо разобраться с настройками nginx

Порядок действий:

- 1 Исследовать настройку nginx согласно варианту
- 2 Создать файл так, чтобы он был доступен из Интернет, по путям согласно настройкам nginx
- 3 Открыть файл через браузер
- 4 Создать приложение nodejs, работающее на порту и обрабатывающее запросы согласно настройкам nginx
- 5 Открыть приложение из браузера
- 6 Исследовать протоколы работы nginx

Доступ к хостингу:

```
ssh -p 20022 stud20@kdenisb.ru
```

пароль Zz8Gws4Vv

Размещение конфига nginx:

```
/etc/nginx/sites-available/default
```

Пример выполнения для варианта 0:

```
cd /var/www/html/web/nginx
```

```
mkdir -p v0/gg/toto/
```

```
echo "kdb" > v0/gg/toto/aaaa.html
```

в браузере <http://kdenisb.ru:20088/v0/gg/zzz/aaaa>

```
node app.js
```

в браузере <http://kdenisb.ru:20088/v0/app>

```
tail /var/log/nginx/access.log
```

Содержимое протокола работы nginx:

```
89.179.74.3 - - [16/Oct/2020:21:54:45 +0000] "GET /v0/gg/zzz/aaaa HTTP/1.1" 200 4 "-"
```

```
"Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0"
```

```
89.179.74.3 - - [16/Oct/2020:22:35:55 +0000] "GET /v0/app HTTP/1.1" 502 182 "-" "Mozilla/5.0
```

```
(X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0"
```

Содержимое файла app.js:

```
var express = require('express');
```

```
var app = express();
```

```
app.get('/v0/app', function (req, res) {
```

```
  res.send('Hello World!');
```

```
});
```

```
app.listen(3100, function () {
```

```
  console.log('Example app listening on port 3100!');
```

```
});
```

Установка expressjs:

```
npm install express
```

## Задание 6

# Исследование протокола OAuth2

Выполнить настройку системы Keycloak, создать пользователей, зарегистрировать клиентское приложение, провести тестирование системы.

## Порядок работы

1. Исследовать настройки системы (realm oat)
2. Сделать свой realm,
3. Создать клиента,
4. Создать пользователей,
5. Определить ссылки для аутентификации и авторизации,
6. Выполнить запросы для получения данных пользователя.

При возникновении проблем с установкой/настройкой можно использовать установленный для образца realm «oat».

## Рекомендации

Создавая realm, используйте маленькие латинские буквы, например, инициалы.

Видеоурок: <https://youtu.be/vnH7xP-4UVg>

Полезные программы

- «postman» - среда конструирования запросов
- «curl» - команда отправки запроса

Адреса ресурсов

- Система Keycloak - <http://kdenisb.ru:58080> имя:admin пароль: admin

## Примеры запросов из видеоурока

```
http://kdenisb.ru:58080/auth/realms/oat/protocol/openid-connect/auth?
response_type=code&client_id=kctst_1&client_secret=be483feb-029a-4ddb-a638-
73ab5ddc55d1&redirect_uri=http://kdenisb.ru:20080/web/kdb/auth/
```

```
http://kdenisb.ru:20080/web/kdb/auth/?session_state=25771115-01f4-4942-b2fe-
da77843e4f58&code=b0f1204a-06f0-48cc-82e1-51a7bd6d1a6c.25771115-01f4-4942-b2fe-
da77843e4f58.bb60a134-b562-44b6-bdf6-9bcd268d39a5
```

```
curl -X POST -d "client_id=kctst_1&client_secret=be483feb-029a-4ddb-a638-
73ab5ddc55d1&session_state=71eaae45-75a2-45b6-bf6a-5984fd1dbc17&code=8363ca71-8d6a-
443e-afc2-1cbdd2397354.71eaae45-75a2-45b6-bf6a-5984fd1dbc17.bb60a134-b562-44b6-bdf6-
9bcd268d39a5&grant_type=authorization_code&redirect_uri=http://kdenisb.ru:20080/web/kdb/
auth/" "http://kdenisb.ru:58080/auth/realms/oat/protocol/openid-connect/token"
```



## Задание 7

# Логирование

Провести анализ логов в системе elasticsearch.

## Порядок работы

1. Создать тестовую веб-страницу на сервере <http://kdenisb.ru:20080> ,
2. Исследовать веб-интерфейс среды elasticsearch (Kibana) <http://kdenisb.ru:5601> ,
3. Сделать тестовые запросы к веб-странице,
4. Сделать запросы для формирования выборки по вариантам в системе elasticsearch.

## Варианты

Вариант	Строка в url http-запроса	Число результатов с кодом завершения 200	Число результатов с кодом завершения 404	Временной промежуток, минут
1	world	5	3	10
2	hello	3	4	15
3	peace	4	6	20
4	make	6	5	10
5	take	5	3	15
6	first	3	4	20
7	tkd	4	6	10
8	fast	6	5	15
9	penta	5	3	20
10	channel	3	4	10
11	price	4	6	15
12	about	6	5	20
13	disk	5	3	10
14	contact	3	4	15
15	sell	4	6	20
16	film	6	5	10
17	news	5	3	15
18	for	3	4	20
19	newyear	4	6	11
20	client	6	4	14

# Рекомендации

Видеоурок: <https://youtu.be/2II764kbjvU>

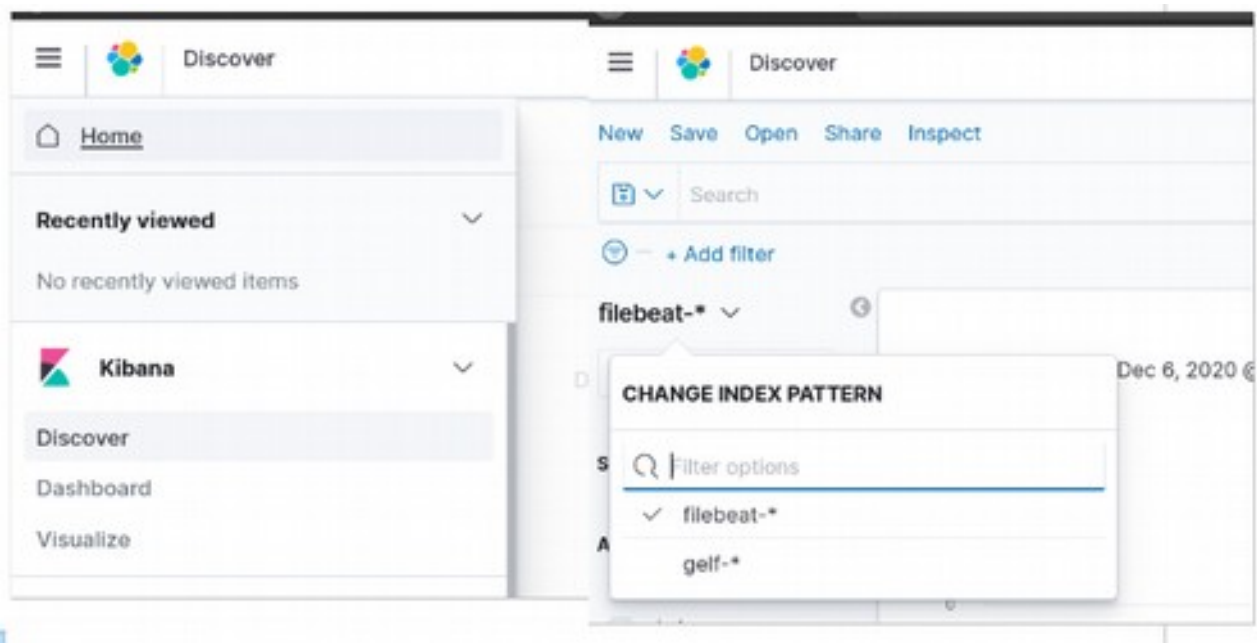
Примеры запросов

- Все логи с сервера <http://kdenisb.ru:20080> «beat.hostname: lamp»

Адреса ресурсов

- Веб-интерфейс Elasticsearch - <http://kdenisb.ru:5601>
- Доступ к хостингу сервера <http://kdenisb.ru:20080> - ssh -p 20022 stud20@kdenisb.ru  
пароль Zz8Gws4Vv (каталог /app/web)
- Документация по языку запросов (KQL):  
<https://www.elastic.co/guide/en/kibana/7.9/kuery-query.html>

Как найти страницу просмотра логов



## Задание 8

# XML

Составить XML-файл и стиль трансляции.

## Порядок работы

1. Составить XML-по варианту,
2. Составить XSLT-программу трансляции,
3. Выполнить трансляцию с командной строки,
4. Выполнить трансляцию через браузер.

## Варианты

Предметная область	Разметка после трансляции		
	Списки	Таблицы	Список определений
меню	0	1	2
телефонный справочник	3	4	5
список студентов	6	7	8
список преподавателей	9	10	11
дисциплины	12	13	14
адреса	15	16	17
расписание	18	19	20
оборудование	21	22	23
библиотека	24	25	26
товары	27	28	29
todo	30	31	32
зоопарк	33	34	35
фильмотека	36	37	38
версии ПО	39	40	41
страны	42	43	44
депутаты	45	46	47

## Рекомендации

Видеоурок: <https://youtu.be/EfQNgCxJe5g>

- «xsltproc» - транслятор XML, интерпретатор XSLT

Адреса ресурсов

- Доступ к хостингу сервера <http://kdenisb.ru:20080> - ssh -p 20022 stud20@kdenisb.ru  
пароль Zz8Gws4Vv (каталог /app/web)

## Пример выполнения

Исходные файлы:

.....

prez\_ex.xml

.....

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="get_price.xslt"?>
<ProductList>
  <Title>Запчасти</Title>
  <Product isaction="yes">
    <ProductName>Колесо</ProductName>
    <UnitPrice>700</UnitPrice>
  </Product>
  <Product isaction="yes">
    <ProductName>Дворники</ProductName>
    <UnitPrice>50</UnitPrice>
  </Product>
  <Product isaction="no">
    <ProductName>Кордан</ProductName>
    <UnitPrice>300</UnitPrice>
  </Product>
</ProductList>
```

.....

get\_price.xslt

.....

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>

  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="ProductList">
    <table border="1">
      <xsl:apply-templates select="Product[@isaction='yes']"/>
    </table>
```

```

</xsl:template>

<xsl:template match="Product">
  <tr>
    <xsl:apply-templates/>
  </tr>
</xsl:template>

<xsl:template match="ProductName">
  <td>
    <xsl:value-of select="text()"/>
  </td>
</xsl:template>

<xsl:template match="UnitPrice">
  <td>
    <xsl:value-of select="text()"/>
  </td>
</xsl:template>

</xsl:stylesheet>

```

#### ВЫЗОВ:

```
xsltproc get_price.xslt prez_ex.xml
```

#### Результат выполнения

```

<html><body><table border="1">
<tr>
  <td>
    <td>Колесо</td>
    <td>700</td>
  </tr>
<tr>
  <td>
    <td>Дворники</td>
    <td>50</td>
  </tr>
</table></body></html>

```



## Задание 9

# s3

Размещение и обработка данных в объектном хранилище

## Порядок работы

1. Создать «ведро» и директорию,
2. Разместить файл через веб-интерфейс,
3. Разместить файл через интерфейс командной строки,
4. Сформировать и разместить структурированный файл,
5. Выполнить произвольные запросы к данным структурированного файла.

## Варианты

\*Предметную область можно взять не по варианту, а использовать из предыдущих заданий.

Предметная область	Структура файла	
	csv	json
меню	1	17
телефонный справочник	2	18
список студентов	3	19
список преподавателей	4	20
дисциплины	5	21
адреса	6	22
расписание	7	23
оборудование	8	24
библиотека	9	25
товары	10	26
todo	11	27
зоопарк	12	28
фильмотека	13	29
версии ПО	14	30
страны	15	31
депутаты	16	32

## Рекомендации

Видеоурок:

- «mc» - утилита командной строки для работы с s3-хранилищами

Адреса ресурсов

- s3-хранилище minio `http://kdenisb.ru:9000`  
`MINIO_ACCESS_KEY=AKIAWOSFODNN9EXAMPLE`  
`MINIO_SECRET_KEY=wJqlrXUtnFEMI/K7MBENG/bPxRfjCYEXAMPLEKEY`

## Команды из видеоурока

```
wget https://dl.min.io/client/mc/release/linux-amd64/mc
chmod 755 mc
```

```
mc alias set minio http://kdenisb.ru:9000 AKIAWOSFODNN9EXAMPLE
wJqlrXUtnFEMI/K7MBENG/bPxRfjCYEXAMPLEKEY --api S3v4
mc ls minio
mc cp ukaz.pdf minio:/btst/
mc cat minio/btst/c.json
{"xxx": [ {"id": "1", "a": 5}, {"id": "2", "a": 9} ]}
mc sql -e "SELECT id FROM S3Object[*] where a>4" minio/btst/id_a.csv
```