

Лекция 10

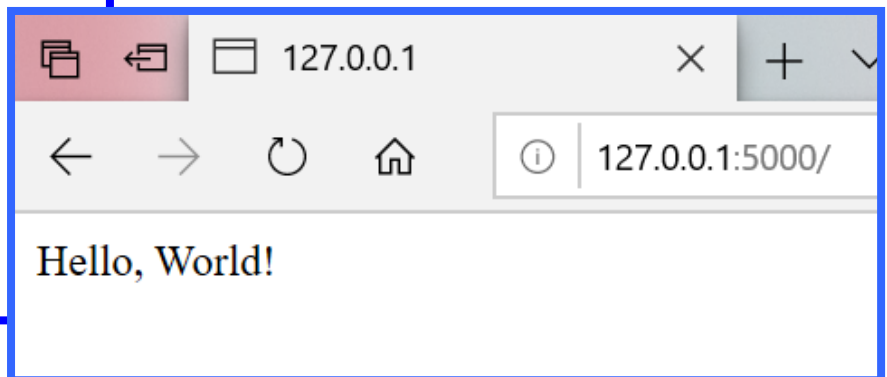
Разработка web-приложений

1 Фреймворк Flask

```
1 # импорт из пакета flask класса Flask
2 from flask import Flask
3
4 # создание объекта класса Flask
5 app = Flask(__name__)
6
7 # связать URL адрес с функцией
8 @app.route('/')
9 def index():
10     return 'Hello, World!'
11
12 # если файл "web.py" запущен
13 # как основная программа
14 if __name__ == "__main__":
15     # запускаем сервер
16     app.run()
```

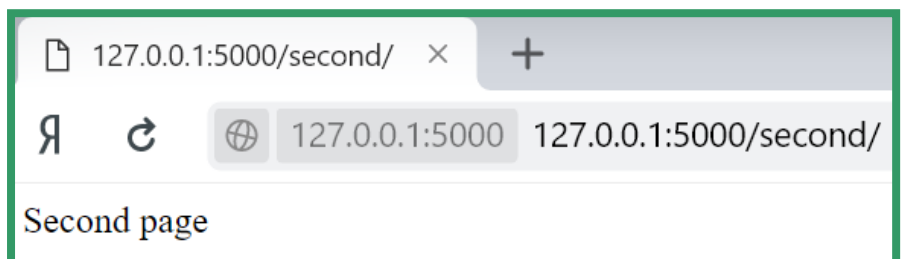
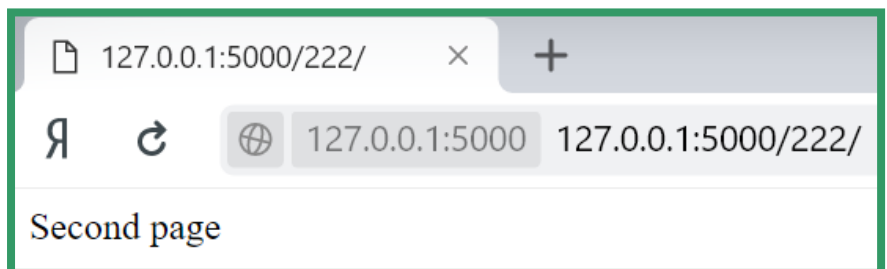
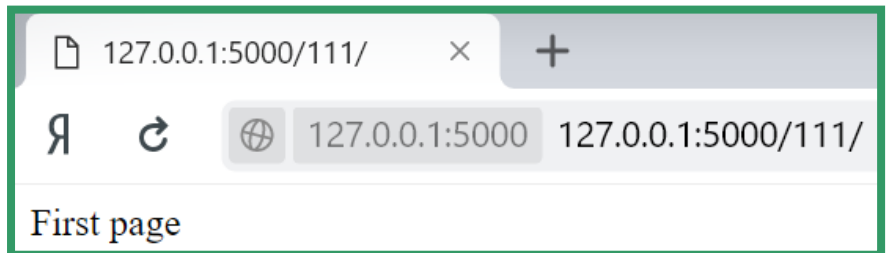
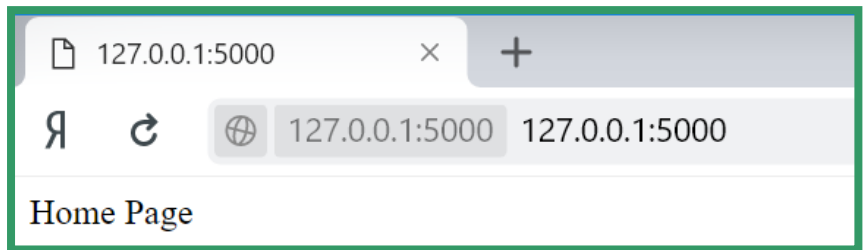
Фреймворки:

- Flask
- Django
- Pyramid



Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def index():
7     return 'Home Page'
8
9 @app.route('/111/')
10 def first():
11     return 'First page'
12
13 @app.route('/222/')
14 @app.route('/second/')
15 def second():
16     return 'Second page'
17
18 if __name__ == "__main__":
19     app.run()
```



Динамические URL

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/user/<id>/')
6 def user_profile(id):
7     return "Profile page of user {}".format(id)
8
9 if __name__ == "__main__":
10     app.run()
```

127.0.0.1:5000/user/100/ × +

Я ↻ 127.0.0.1:5000 127.0.0.1:5000/user/100/

Profile page of user #100

127.0.0.1:5000/user/777/ × +

Я ↻ 127.0.0.1:5000 127.0.0.1:5000/user/777/

Profile page of user #777

127.0.0.1:5000/user/Alex/ × +

Я ↻ 127.0.0.1:5000 127.0.0.1:5000/user/Alex/

Profile page of user #Alex

2 Шаблоны. Формы

Шаблон - текстовый файл с HTML-кодом.

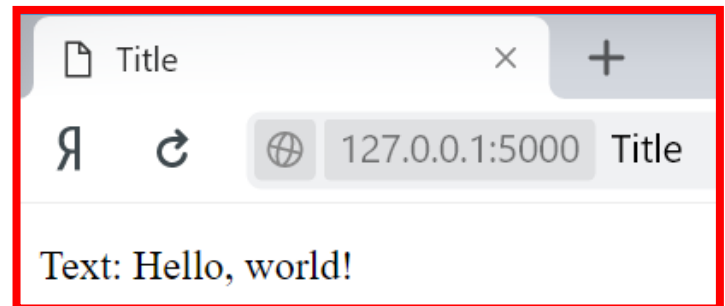
```
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def index():
7     return render_template('index.html', txt='Hello, world!')
8
9 if __name__ == "__main__":
10     app.run()
```

Функцию `render_template` для отрисовки шаблонов

\templates\index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8
9     <p>Text: {{ txt }}</p>
10
11 </body>
12 </html>
```

По умолчанию Flask ищет шаблоны в подкаталоге `templates` внутри папки приложения.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   {% if message %}
9     <p>{{ message }}</p>
10  {% endif %}
11
12  <form action="" method="post">
13    <p>
14      <label for="username">Username</label>
15      <input type="text" name="username">
16    </p>
17    <p>
18      <label for="password">Password</label>
19      <input type="password" name="password">
20    </p>
21    <p>
22      <input type="submit">
23    </p>
24  </form>
25
26 </body>
27 </html>

```

```

1 from flask import Flask, render_template, request
2
3 app = Flask(__name__)
4
5 @app.route('/', methods=['POST', 'GET'])
6 def login():
7     message = ''
8     if request.method == 'POST':
9         username = request.form.get('username')
10        password = request.form.get('password')
11
12        if username == 'root' and password == 'pass':
13            message = "Correct username and password"
14        else:
15            message = "Wrong username or password"
16        return render_template('index.html', message=message)
17
18 if __name__ == "__main__":
19     app.run()

```

**Библиотека Wtforms -
обработка html-форм**

Wrong username or password

Username

Password

Correct username and password

Username

Password

3 Работа с базой данных

Добавление информации:

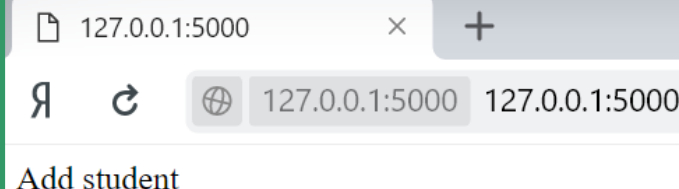
```
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3
4 app = Flask(__name__)
5 app.debug = True
6 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:secret@localhost/university'
7
8 db = SQLAlchemy(app)
9
10 class Student111(db.Model):
11     __tablename__ = 'student'
12     id = db.Column(db.Integer(), primary_key=True)
13     name = db.Column(db.String(25), nullable=False)
14
15 @app.route('/', methods=['POST', 'GET'])
16 def login():
17     s1 = Student111(id = 777777, name = 'Sidorov S.S.')
18     db.session.add(s1)
19     db.session.commit()
20     return 'Add student'
21
22 if __name__ == "__main__":
23     app.run()
```

SQLAlchemy – фреймворк для работы с реляционными базами данных в Python.

Установка: `pip install flask-sqlalchemy`

```
mysql> select* from student;
+-----+-----+
| ID    | NAME          |
+-----+-----+
| 111222 | Ivanov I.I.   |
| 333444 | Petrov P.P.   |
| 777888 | Mishin M.M.   |
+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select* from student;
+-----+-----+
| ID    | NAME          |
+-----+-----+
| 111222 | Ivanov I.I.   |
| 333444 | Petrov P.P.   |
| 777777 | Sidorov S.S.  |
| 777888 | Mishin M.M.   |
+-----+-----+
4 rows in set (0.00 sec)
```



Местоположение базы данных в виде URI:

dialect+driver://username:password@host:port/database

База данных (dialect)	Драйвер DBAPI (driver)
MySQL	PyMysql
PostgreSQL	Psycopg 2
MS-SQL	pyodbc
Oracle	cx_Oracle

Примеры:

URL базы данных для MySQL с использованием драйвера PyMysql
'mysql+pymysql://root:pass@localhost/my_db'

URL базы данных для PostgreSQL с использованием psycopg2
'postgresql+psycopg2://root:pass@localhost/my_db'

URL базы данных для MS-SQL с использованием драйвера pyodbc
'mssql+pyodbc://root:pass@localhost/my_db'


```

1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3
4 app = Flask(__name__)
5 app.debug = True
6 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:secret@localhost/university'
7
8 db = SQLAlchemy(app)
9
10 class Student111(db.Model):
11     __tablename__ = 'student'
12     id = db.Column(db.Integer(), primary_key=True)
13     name = db.Column(db.String(25), nullable=False)
14
15     def __repr__(self):
16         return "{} : {}".format(self.id, self.name)
17
18 @app.route('/', methods=['POST', 'GET'])
19 def login():
20     s = db.session.query(Student111).all()
21     n = db.session.query(Student111).count()
22     return str(n)+' - ' + str(list(s))
23
24 if __name__ == "__main__":
25     app.run()

```

Получение количества записей в таблице и содержимого таблицы

127.0.0.1:5000 × +

Я ↻ 🌐 127.0.0.1:5000 127.0.0.1:5000

4 - [111222 : Ivanov I.I., 333444 : Petrov P.P., 777777 : Sidorov S.S., 777888 : Mishin M.M.]

Некоторые методы для выполнения запросов

Метод	Описание
<code>all()</code>	Возвращает результат запроса в виде списка.
<code>count()</code>	Возвращает количество записей в запросе.
<code>first()</code>	Возвращает первый результат запроса или <code>None</code> , если в нем нет строк.
<code>first_or_404()</code>	Возвращает первый результат запроса или ошибку 404, если в нем нет строк.
<code>get(pk)</code>	Возвращает объект, который соответствует данному первичному ключу или <code>None</code> , если объект не найден. <i>Пример:</i> <code>db.session.query(Student111).get(111222)</code>
<code>get_or_404(pk)</code>	Возвращает объект, который соответствует данному первичному ключу или ошибку 404, если объект не найден.
<code>filter(*criterion)</code>	Возвращает новый экземпляр <code>flask_sqlalchemy.BaseQuery</code> с оператором <code>WHERE</code> . <i>Пример:</i> <code>db.session.query(Student111).filter(Student111.id == 777777).all()</code>