

«Информационные и программные системы»

ТЕМА 2 «Основы языка PHP»

Лекция № 3 «Взаимодействие PHP и html-формы»



Учебные вопросы:

1. Создание html-документов
2. Создание формы средствами языка HTML
3. Методы передачи данных GET и POST.

1. Создание html-документов

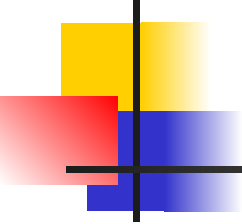


Способы создания HTML-документов:

- Первый способ состоит в разметке создаваемого документа вручную.
- Создание с использованием редактора HTML-кода (программы создания Web - страниц).
- Использование среды программирования.

Язык разметки гипертекста

HTML



HTML (от англ. HyperText Markup Language – язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине.

Три главных свойства.

- Принципиальная невозможность достижения абсолютной точности воспроизведения исходного документа.
- Машинно-независимый язык.
- Не является языком программирования.

Основные конструкции языка

Язык HTML состоит из специальных управляющих конструкций – тэгов.

Например `<html>`

Выделяют *парные* и *непарные* тэги:

Непарные (*одинарные*) тэги оказывают воздействие на весь документ или определяют разовый эффект в месте своего появления. Непарные тэги **
** или **
**

Парные (*двойные*) тэги оказывают влияние на часть документа, заключенного между ними. Парные тэги состоят из двух частей: открывающей метки вида **<P>** и закрывающего тэга вида **</P>** (с тем же именем, но с добавлением символа «/»)

Атрибуты

Дополнением к тэгам являются **атрибуты** – это свойства тега.

```
<a href="https://vk.com/itc.digital" target="_blank">ITC  
Вконтакте</a>
```

атрибуты

Правила написания разметки HTML

- Использование элементов !
doctype, html, head, body
- Строчные буквы в тегах
- Кавычки при указании значений атрибутов



Структура HTML- документа

`<!DOCTYPE html>` - так обозначается HTML5
`<HTML>`

- раздел заголовка (ограничивается парным тегом `<HEAD>`) – содержит служебную информацию
- тело документа (ограничивается парным тэгом `<BODY>`) – определяет видимую часть документа

- НИКУЛИН
 - ИПС 20
 - ИС 201
 - Практи
 - Atom
 - PascalA
 - 1.html
 - Безымя
 - Лб_Усл
 - Пример
 - Пример
 - Пример
 - Пример
 - Пример
 - Denwer
 - fon1.jpс
 - fon2.jpс
 - index1.l
 - Projecti
 - Projecti
 - Projecti
 - Projecti
 - st.html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Первый</title>
6  </head>
7  <body>
8      Привет
9  </body>
10 </html>
11

```

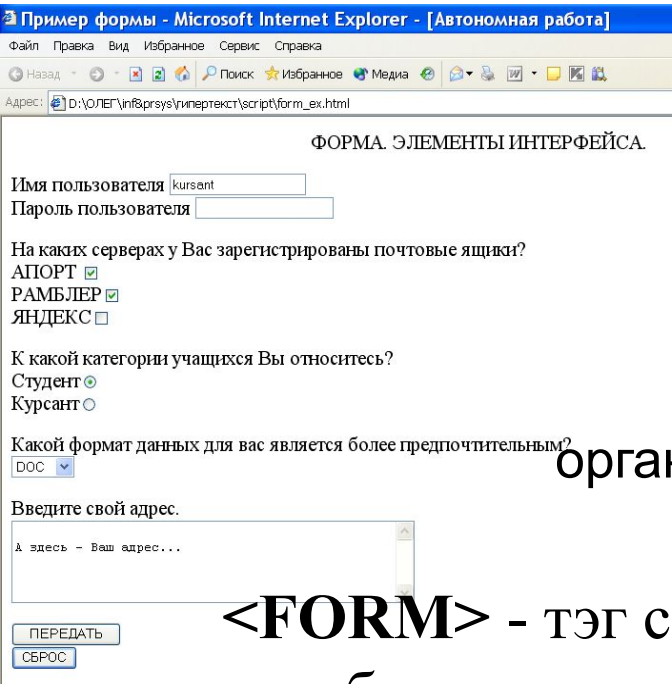
Ознакомиться с настройкой редактора Atom
 \\DATA\ИС\Настройка редактора Atom.docx



Visual Studio Code

```
JS app.js x
6 const session = require('express-session');
7 const bodyParser = require('body-parser');
8 const path = require('path');
9
10 const app = express();
11 app.set('views', path.join(__dirname, 'client/views'));
12 app.set('view engine', 'pug');
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(express.static(path.join(__dirname, 'client'), { maxAge: 31557600000 }));
16
17 app.get('/', async function (req, res) {
18   const data = await sentimentService();
19   let sentimentWithLevel = [];
20   Amanda Silver
21   for (let s in data.tweets) {
22     let newTweet = {
23       sentiment: s.sentiment,
24       level: util.getHappinessLevel(s.sentiment)
25     };
26     sentimentWithLevel.push(newTweet);
27   }
28
29   res.render('index', {
30     tweets: sentimentWithLevel,
31     counts: data.counts
32   });
33 });
34
35 const port = process.env.PORT || 3000;
36 app.listen(port);
37 console.log("app listening on port: " + port);
```

Ln 21, Col 31 Spaces: 2 UTF-8 LF JavaScript



2. Создание формы средствами языка HTML

Форма — набор элементов графического интерфейса, которые используются для организации диалога пользователя с web-сервером

<FORM> - тэг создания экранной формы, используется с атрибутами:

- ACTION - определяет, где находится обработчик формы
- METHOD - определяет, каким образом данные из формы будут переданы обработчику
- ENCTYPE - определяет, каким образом данные из формы будут закодированы для передачи обработчику

Например,

`<FORM action=1.php method=GET>` элементы `</FORM>`

```
2 <html>
3 <head>
4 <title>Пример формы</title>
5 </head>
6 <body bgcolor=#ffffff>
7 <center>ФОРМА. ЭЛЕМЕНТЫ ИНТЕРФЕЙСА.</center>
8 <p>
9 <form >
10 Имя пользователя
11 <INPUT TYPE="text" NAME="user" VALUE="kursant"><br>
12 Пароль пользователя
13 <INPUT TYPE="password" NAME="access"><br>
14 <p>
15 На каких серверах у Вас зарегистрированы почтовые ящики?<br>
16 АПОРТ <INPUT TYPE="checkbox" NAME="aport" VALUE="aport" CHECKED> <br>
17 РАМБЛЕР<INPUT TYPE="checkbox" NAME="rambler" VAL-UE="rambler" CHECKED><br>
18 ЯНДЕКС<INPUT TYPE="checkbox" NAME="yadex" VAL-UE="yandex"><br>
19 <p>
20 К какой категории учащихся Вы относитесь?<br>
21 Студент<INPUT TYPE="radio" NAME="category" VAL-UE="student" CHECKED><br>
22 Курсант<INPUT TYPE="radio" NAME="category" VAL-UE="kursant"><br>
23 <p>
24 Какой формат данных для вас является более предпочтитель-ным?<br>
25 <SELECT NAME="format" SIZE=1>
26 <OPTION VALUE="doc" >DOC
27 <OPTION VALUE="xls" > XLS
28 <OPTION VALUE="html" > HTML
29 </SELECT>
30 <p>
31 Введите свой адрес.<br>
32 <TEXTAREA NAME=address ROWS=5 COLS=50>
33 А здесь - Ваш адрес...
34 </TEXTAREA>
35 <p>
36 <INPUT TYPE=button NAME=myButton VALUE=ПЕРЕДАТЬ><br>
37 <INPUT TYPE=reset VALUE=СБРОС>
38 </form>
39 <p>
40 </body>
41 </html>
```

Имя пользователя

Пароль пользователя

АПОРТ

РАМБЛЕР

ЯНДЕКС

Студент

Курсант

<INPUT>— наиболее часто используемый тэг для вставки графических элементов.

Атрибутами тэга являются:

- **TYPE**= тип элемента (text – текстовое поле, date – поле для ввода даты, password – поле для ввода пароля, checkbox – кнопка-флажок, radio – радио-кнопка, submit - кнопка отправки формы, button – кнопка, reset - кнопка сброса, hidden - скрытый элемент формы)
- **NAME**=имя_элемента
- **VALUE**=значение элемента, которое будет передано обработчику под заданным именем.
- **MAXLENGTH**=N (максимальное количество символов, которое можно ввести)
- **SIZE**=N (количество символов выводимых полем - ширина поля)

<SELECT> - тэг создания ниспадающего меню
или списка

<OPTION> определяет элемент

Атрибутами тэга <SELECT> являются:

- SIZE=k - высота окна в строках
- **NAME** - имя элемента
- MULTIPLE - выбор нескольких элементов

Атрибутами тэга <OPTION> являются:

- **VALUE** - значение элемента, которое будет передано обработчику под заданным именем.
- CHECKED или SELECTED (без значения) - элемент выбран по умолчанию.

<TEXTAREA> - тэг создания поля для ввода многострочного текста



Атрибутами тэга <TEXTAREA> являются:

- COLS - устанавливает ширину окна в символах
- ROWS - устанавливает высоту окна в строках
- **NAME** - имя элемента

Пример

```
<TEXTAREA NAME=address ROWS=5 COLS=50>
```

А здесь - Ваш адрес...

```
</TEXTAREA>
```

Поле для ввода ФИО

```
<INPUT TYPE=text NAME=user VALUE=введите_ФИО>
```

Создание поля для ввода пароля

(по умолчанию введено 12345)

```
<INPUT TYPE=password NAME=access VALUE=kursant>
```

Создание 3-х элементов графического интерфейса

- кнопка-флажок (по умолчанию выбрана первая кнопка-флажок)

- первая
- вторая
- третья

```
<INPUT TYPE=checkbox NAME=server VALUE=aport CHECKED> первая
```

```
<br><INPUT TYPE=checkbox NAME=server VALUE=rambler> вторая
```

Создание кнопок-переключателей

(по умолчанию выбрана первая кнопка-переключатель)

- ```
<INPUT TYPE=radio NAME= category VALUE=student CHECKED >
```
- ```
<INPUT TYPE=radio NAME=category VALUE=kursant>
```

Создание кнопки

```
<INPUT TYPE=submit NAME=myButton VALUE=Отправить>
```

Пример 7 Создание списка дисциплин

```
<SELECT NAME="format" SIZE=2>
<OPTION VALUE="doc" > ИиПС
<OPTION VALUE="xls"> ИТ
<OPTION VALUE="html"> ИС
</SELECT>
```

Создание области для ввода персональных данных:

ФИО
 Адрес
 Дата рождения

```
<TEXTAREA NAME=address ROWS=5 COLS=50>
ФИО Адрес Дата рождения
</TEXTAREA>
```

3. Методы передачи данных GET и POST

Метод **GET** - данные из формы присоединяются в конце URL страницы, на которую посылаются данные (URL в атрибуте action)

Метод **POST** - посылает данные формы на страницу action как отдельный поток данных, который не появляется в поле адреса браузера, кроме того, можно посылать любое требуемое для обработки формы количество символов

<FORM **method**= "get" **action**= " test.php " >

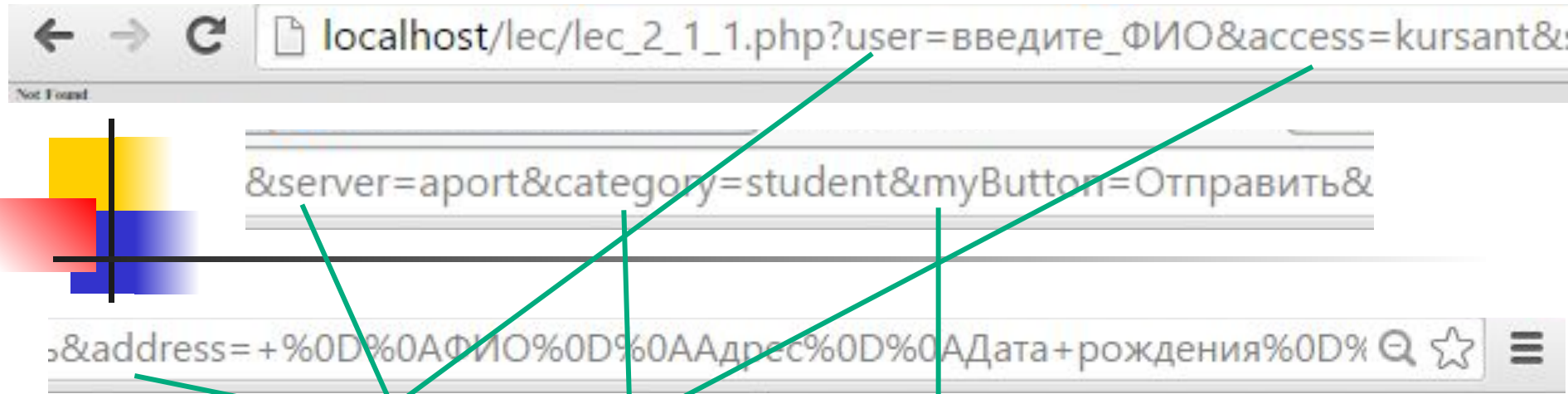




Использование HTML-форм для передачи данных на сервер

PHP имеет три типа переменных для работы с формами:

- `$_GET[]` — ассоциативный массив, который содержит все значения, передаваемые в сценарий с помощью метода формы GET.
- `$_POST[]` – ассоциативный массив, который содержит все значения, передаваемые в сценарий с помощью метода формы POST.
- `$_REQUEST[]` – ассоциативный массив, который содержит все значения, передаваемые в сценарий с помощью методов POST и GET.



user	access	server	category	myButton	adress
введите_ФИО	kursant	aport	student	Отправить	ФИО Адрес Дата рождения

`$_GET[' ']`

ИЛИ

ИЛИ

`$_REQUEST[' ']`

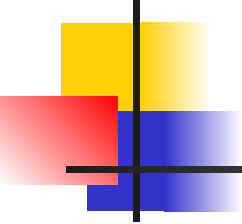
`$_POST[' ']`

`$_POST [" индекс "]`

<code>user</code>	<code>access</code>	<code>server</code>	<code>category</code>	<code>myButton</code>	<code>adress</code>
введите_ФИО	kursant	aport	student	Отправить	ФИО Адрес Дата рождения

PHP-скрипт для вывода введенных данных

```
<?php
echo "Your Name is: " . $_POST["user"];
echo "Your password is: " . $_POST["access"];
echo "You chose: " . $_POST["server"];
echo "You are: " . $_POST["category"];
echo "Your address is: " . $_POST["adress"];
?>
```



Задания для
самостоятельной
подготовки к лабораторному
занятию:

Создать php-скрипт, выводящий на экран введенные через форму ваши персональные данные