

Лабораторная работа №3

Фреймы. Формы ввода данных

Теоретические сведения

1. Фреймы

Фреймы используются для деления экрана браузера на независимые части. Содержимое *фрейма* — это отдельная HTML-страница, которая может находиться как, на локальном сервере так и на удаленном узле. Фреймовая *структура* определяет только способ организации экрана и задаёт, где находится начальное содержимое каждого *фрейма*. Описание фреймовой структуры строится в отдельном файле. Поэтому для создания, например, документа, который отображается в двух фреймах, необходимо три файла. Наиболее частое применение фреймов - это отделение меню от основного информационного наполнения. Хотя фреймы достаточно удобная вещь, использование их не рекомендуется.

Файл, который содержит описание структуры фреймов, не может содержать элемент `<body>`. Такой файл имеет основной элемент `<html>`, в который включаются разделы `<head>` и `<frameset>`.

Фреймы задаются с помощью тегов `<FRAMESET>` и `<FRAME>`. Тег `<frameset>` содержит описание распределения окон с указанием размера каждого фрейма, а `<frame>` задает адреса страниц.

Тег `<frameset>` допускает вложение. Закрывающий тег обязателен. Основные атрибуты:

- **cols** - задает число вертикальных панелей.
- **rows** - задает число горизонтальных панелей.

Значение указываются либо в пикселях, либо в относительных единицах - процентах по отношению к доступной части окна.

Отдельные браузеры поддерживают также атрибуты:

- **border** - указывает толщину рамки в пикселях.
- **bordercolor** - указывает цвет рамки.
- **frameborder** - указывает вид рамки (плоский или объемный): "1" - рисуется объемная рамка, "0" - рисуется плоская рамка.
- **framespacing** - создает пустое поле заданной толщины между фреймами, **frameborder** при этом должен принимать значение "0".

Пример: следующая запись формирует экран, состоящий из трех строк: высота верхней — 20 пикселей, средней — 80 пикселей, нижней — 20 пикселей:

```
<FRAMESET ROWS="20, 80, 20">
```

следующая запись формирует экран, на котором верхняя строка занимает 10% высоты экрана, средняя — 60%, а нижняя — оставшиеся 30%:

```
<FRAMESET ROWS="10%, 60%, 30%">
```

Можно использовать комбинации значений. Следующая запись формирует экран, на котором верхняя строка имеет высоту 20 %, средняя — 80 пикселей, а нижняя занимает все оставшееся место:

<FRAMESET ROWS="20%, 80, *">

Тег **<FRAME>** определяет внешний вид и поведение *фрейма*. Не имеет закрывающего тега. Вся суть тега **<FRAME>** в его атрибутах.

Синтаксис: одиночный тег:

<frame attribute1="значение1" attribute2=" значение2">

1. **NAME** – задает имя фрейма, которое нужно для организации переходов между фреймами и использования в: программах при организации ссылки. При этом атрибут **TARGET** тега гиперссылки должен иметь тоже значение. Удобно использовать для задания ссылок в одном *фрейме*, и открытия их в другом фрейме с заданным именем, при этом первое остается неизменным.
2. **MARGINWIDTH** - задает горизонтальный отступ между содержимым кадра и его границами. Наименьшее значение этого атрибута равно 1. Нельзя указать 0. По умолчанию атрибут равен 6.
3. **MARGINHEIGHT** - задает поля в верхней и нижней частях фрейма.
4. **CROLLING** = **yes** | **no** | **auto** - задает возможность пользоваться полосами прокрутки во фрейме. По умолчанию **SCROLLING=auto**.
5. **NORESIZE**= **NORESIZE**| **RESIZE** – определяют возможность управления размерами фрейма мышью.
6. **SRC** – задает адрес загружаемого во фрейм ресурса.

Тег **<NOFRAMES>** задает контейнер для браузеров не поддерживающих фреймы. Содержимое контейнера выводится, только если фреймы не поддерживаются.

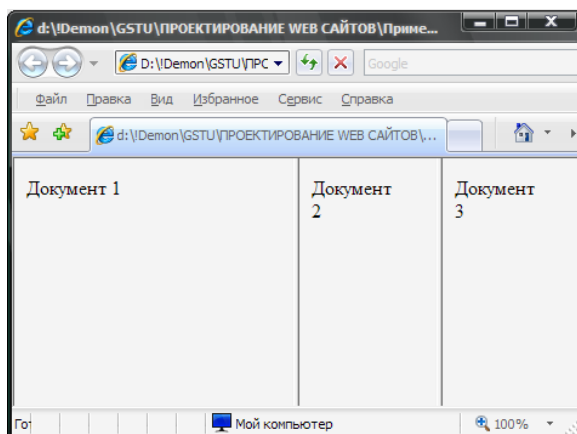
Рассмотрим примеры задания фреймов.

Пример 1. страница с фреймовой структурой, набор вертикальных фреймов.

<html>

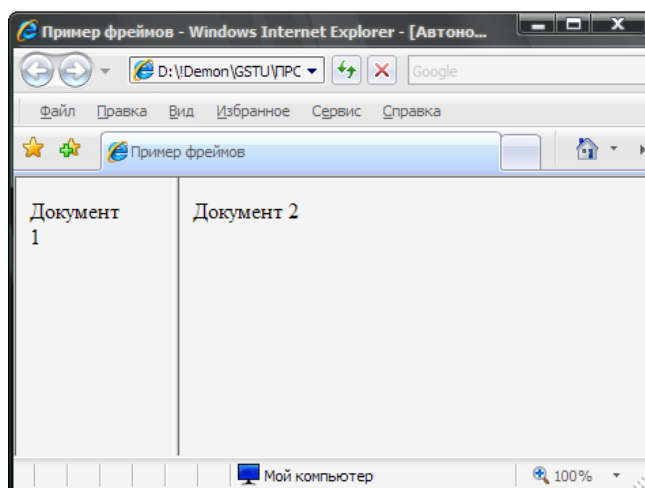
```
<frameset cols="50%,25%,25%">  
  <frame src="frame_1.htm">  
  <frame src="frame_2.htm">  
  <frame src="frame_3.htm">  
</frameset>
```

</html>



Пример 2. страница с фреймовой структурой, два вертикальных фрейма, с поддержкой браузеров не обрабатывающих фреймы.

```
<HTML>
<HEAD>
<TITLE>Пример фреймов</TITLE>
</HEAD>
<FRAMESET COLS="25%, 75%">
  <FRAME SRC="menu.html" noresize="noresize">
  <FRAME SRC="main.html">
</NOFRAMES>
  Вы просматриваете эту страницу с помощью браузера, не
  поддерживающего фреймы.
</NOFRAMES>
</FRAMESET>
</HTML>
```



2. Формы

2.1 Задание формы

Форма может располагаться в любом месте тела документа, между тегом **<form>** и его закрывающим тегом **</form>**. Внутри формы располагаются теги задающие элементы интерфейса **input**, **select**, **textarea**.

Атрибуты тега:

атрибут action - задает URL приложения, которое должно получить и обработать данные формы.

ПРИМЕРЫ:

```
<form action="http://www.mysite/cgi-bin/update">
```

```
...
```

```
</form>
```

Передаёт значения формы приложению с названием **update**, расположенному в каталоге **cgi-bin**.

```
<form action="http://mysite.com/scripts/myprogram.pl">
```

- URL-адрес файла программы, написанной на языке Perl;

```
<form action="www.anyserver.ru/mypage.asp">
```

- URL-адрес на asp-файл (Active Server Page);

```
<form action="www.anyserver.ru/mypage.php">
```

- URL-адрес на PHP " файл;

```
<form action="mailto:my@server.ru">
```

- для отправки данных по электронной почте по указанному адресу.

атрибут method - устанавливает метод, посредством которого браузер передает серверу для обработки данные формы. Существуют два метода **POST** и метод **GET** (по умолчанию).

Передаваемые данные в запросе представляются последовательностью вида:

значение_атрибута_name=значение_атрибута_value

атрибут enctype – метод кодирования данных передаваемых на сервер. Возможные значения:

- **application/x-www-form-urlencoded**

Например, вот что браузер посылает серверу после того, как пользователь заполнил форму с двумя полями ввода, имена которых **name** и **address**. В первом поле только одна строка текста, тогда как второе содержит несколько строк ввода:

name=Lena+Ivanova&address=Gomel+Sovetskaya+187

- **multipart/form_data** - использовать с формами, содержащими поле выбора файла, который загружает пользователь. Применяется, только когда атрибуту **method = post**.

```
-----146931364513459
```

```
Content Disposition: form data; name="name"
```

Lena Ivanova

```
-----146931364513459
```

```
Content Disposition: form data; name="address"
```

Gomel Sovetskaya 187

```
-----146931364513459
```

- **text/plain** - применять при посылке форм по адресу электронной почты. Закодированными в этом формате остаются только символы возврата каретки и перевода строки в многострочных полях ввода.

атрибут name – имя формы;

2.2 Элементы пользовательского интерфейса

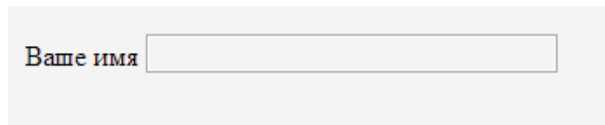
Определение элементов управления формы — тег <INPUT>

С помощью тега **<input>** можно создавать элементы различных типов: поля ввода данных, кнопки и переключатели и тд. Тип элемента определяется атрибутом **type**. Атрибут **type** может принимать следующие значения:

text—текстовое поле ввода данных; дополнительные атрибуты:

- **maxlength** — максимальное количество вводимых символов;
- **size** — ширина поля, выраженная в количестве одновременно видимых символов;
- **value** — значение (содержимое поля);

Ваше имя <INPUT TYPE=text NAME=Name SIZE=35>

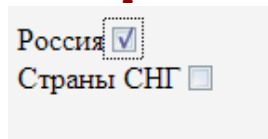


checkbox — переключатель типа флажок; дополнительные атрибуты:

- **value** — значение, ассоциированное с флажком;
- **checked** — если указан, то флажок является установленным;
-

**
Россия<INPUT NAME="Страна" TYPE=checkbox VALUE="Россия">**

**
Страны СНГ<INPUT NAME="Страна" TYPE=checkbox VALUE="СНГ">**

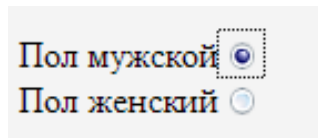


radio — переключатель типа радиокнопка (селекторный переключатель);

- **value** — значение, ассоциированное с радиокнопкой;
- **checked** — если указан, то флажок является установленным;

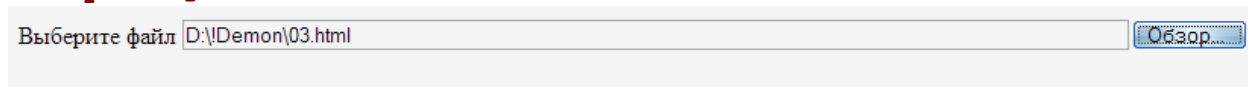
**
Пол мужской<INPUT NAME="Пол" TYPE=radio VALUE="Мужской">**

**
Пол женский<INPUT NAME="Пол" TYPE=radio VALUE="Женский">**



file — текстовое поле ввода данных с кнопкой **Обзор**, щелчок на которой открывает диалоговое окно «Выбор файла»; полное имя выбранного файла отображается в поле ввода. Атрибуты такие как у type="text";

Выберите файл <INPUT TYPE=file NAME=Name SIZE=100>



password — текстовое поле ввода данных, в котором все вводимые символы отображаются как символ *. Атрибуты такие как у `type="text"`;

Введите пароль (макс 6 символов) `<INPUT TYPE=password NAME=Name SIZE=35 maxlength=6>`

Введите пароль (макс 6 символов)

button — кнопка;

- **value** строка, отображаемая на кнопке. По умолчанию на кнопке нет надписи;

`<INPUT TYPE=button VALUE="Нажми меня">`

Нажми меня

submit — кнопка отправки данных формы, на сервер;

- **value** строка, отображаемая на кнопке. По умолчанию на кнопке находится надпись «Подача запроса» или **Submit**.

`<INPUT TYPE=submit>`

`<INPUT TYPE=submit VALUE="Нажми меня">`

Подача запроса Нажми меня

image — графическое изображение, по действию аналогично кнопке **submit**; Когда пользователь щелкает мышью по изображению, браузер сохраняет координаты соответствующей точки экрана. Далее он "обрабатывает" введенную в *форму* информацию и передает на сервер.

- **src** - адрес графического файла;

`<INPUT TYPE=image src="light.png" >`



reset — кнопка Сброс, щелчок на которой отменяет все изменения в элементах, содержащихся в форме (восстанавливает исходные значения).

- **value** строка, отображаемая на кнопке. По умолчанию на кнопке находится надпись «Сбросить» или **Reset**.

`<INPUT TYPE=reset >`

`<INPUT TYPE=reset VALUE="Отмена">`

Сбросить Отмена

hidden — скрытое, невидимое поле. Позволяет включить в отправляемую *форму* значения атрибутов **NAME** и **VALUE**, которые пользователь изменить не может. Такие метки полезны при наличии нескольких *форм* для дальнейшей обработки данных.

Кроме рассмотренных ранее, тег `<input>` имеет следующие атрибуты:

- **name** — имя (идентификатор) элемента. Обязательно если предполагается использовать элемент в составе формы для передачи значения атрибута **value** на сервер;
- **disabled** — делает элемент недоступным пользователю;
- **accesskey** — определяет клавишу быстрого доступа к элементу (в сочетании с клавишей `<Alt>`); Например:

`<input type= "button" value="OK" accesskey="Y">;`

- **tabindex** — целое число, определяющее порядок перехода к элементу по клавиши <TAB>.

Атрибуты элементов формы

Атрибуты (× = обязательный; • = необязательный;
пусто = не поддерживается)

Тег формы или тип поля ввода <input>	Accept	accesskey	align	alt	border	cols	checked	disabled	maxlength	multiple	name	notab	onBlur	onChange	onClick	onFocus	onSelect	readonly	rows	size	src	tabindex	tborder	usemap	value	wrap
button		•						•			×	•	•		•	•						•	•		×	
checkbox		•					•	•			×	•			•			•				•	•		×	
file	•	•						•	•		×	•	•	•	•	•		•		•		•	•		•	•
hidden											×														×	
image		•	•	•	•			•			•	•			•						×	•	•	•	•	
password		•						•	•	•	×	•	•	•	•	•	•	•		•		•	•		×	
radio		•					•	•			×	•			•			•				•	•		×	
reset		•						•				•		•	•	•						•	•		•	•
submit		•						•			•	•		•	•	•						•	•		•	•
text		•						•	•		×	•	•	•	•	•	•	•		•		•	•		•	•
<button>		•						•			×		•		•	•						•			•	
<select>								•		•	×		•	•	•	•				•		•				
<textarea>		•				•		•			×		•	•	•	•	•	•	•			•				•

2.3 Задание кнопок, тег <button>

Тег <button> имеет следующие атрибуты: **accesskey**, **disabled**.

Кнопка, заданная тегом <button>, может использоваться в формах, однако, поскольку тег не имеет атрибутов **value** и **name** его содержимое не может быть отправлено на сервер с помощью кнопки **submit**. Используется совместно со скриптами.

ПРИМЕР.

`<button>Кликай ТыТ! </button>`

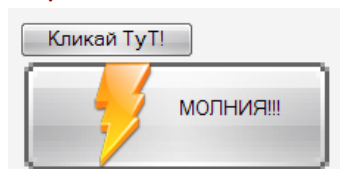
`
`

`<button>`

``

`МОЛНИЯ!!!`

`</button>`



2.4 Раскрывающийся список: тег <SELECT>

Тег <SELECT> используется для создания раскрывающегося списка. Это контейнерный тег, внутри которого используются теги <option>, предназначенные для задания элементов списка. Тег имеет следующие атрибуты:

align — горизонтальное выравнивание списка; возможны значения:

- **absbottom** — выравнивание нижней границы списка по нижней границе текущей строки;
- **absmiddle** — выравнивание середины списка по середине текущей строки;
- **baseline** — выравнивание нижней границы списка по базовой линии текущей строки;
- **bottom** — то же, что и **baseline**;
- **top** — верхняя граница списка выравнивается по самому высокому элементу текущей строки;
- **texttop** — верхняя граница списка выравнивается по самому высокому текстовому элементу текущей строки;
- **left** — список располагается у левого края окна; текст и другие элементы обтекают его справа;
- **right** — список располагается у правого края окна; текст и другие элементы обтекает его слева;

accesskey — определяет клавишу быстрого доступа к раскрывающемуся списку; Например, **<select accesskey="S">**

name — имя (идентификатор) списка;

disabled — делает список недоступным пользователю;

multiple — возможность выбора из списка одновременно нескольких элементов;

size — количество одновременно видимых элементов списка; по умолчанию — 1, Если это число больше 1, то список снабжается полосой прокрутки;

tabindex — целое число, определяющее порядок перехода к элементу с помощью клавиши <TAB>;

Тег **<option>** имеет следующие атрибуты:

selected — обозначает выбранный (выделенный) элемент списка;

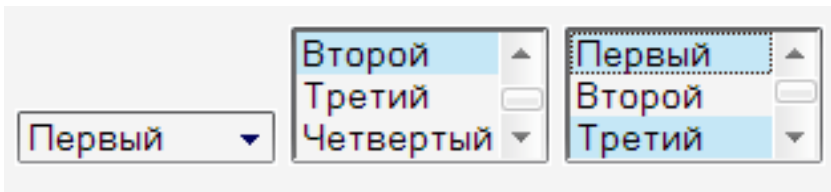
value — значение, ассоциированное с элементом списка.

ПРИМЕР:

```
<select name="List1">
  <option value=1>Первый
  <option value=2>Второй
  <option value=3>Третий
  <option value=4>Четвертый
</select>
```

```
<select name="List2" size="3">
  <option value=1>Первый
  <option value=2 selected>Второй
  <option value=3>Третий
  <option value=4>Четвертый
</select>
```

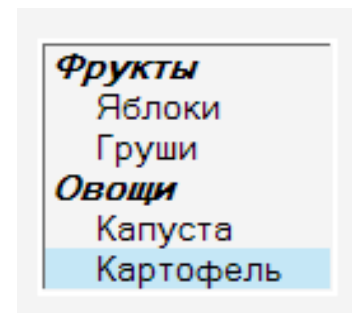
```
<select name="List2" size="3" multiple>
  <option value=1>Первый
  <option value=2 selected>Второй
  <option value=3>Третий
  <option value=4>Четвертый
</select>
```

Для создания иерархических списков внутри тега **<SELECT>** можно использовать контейнерный тег **<optgroup>**, в который заключаются теги **<option>** объединяемых в одну группу. Атрибут **label** тега **<optgroup>** - это невыбираемый элемент списка, выполняющий роль заголовка группы элементов. При этом выбираемые элементы группы, заданные посредством тега **<option>**, отображаются с левым отступом.

ПРИМЕР:

```
<select name="List1" size=6>
<optgroup label="Фрукты">
  <option value=1>Яблоки
  <option value=2>Груши
</optgroup>
<optgroup label="Овощи">
  <option value=3>Капуста
  <option value=4 selected>Картофель
</optgroup>
</select>
```



2.5 Текстовая область: тег **<textarea>**

Контейнерный **<textarea>** позволяет создать текстовую область.

Тег **<textarea>** имеет следующие атрибуты:

align — горизонтальное выравнивание текстовой области (значения как у **SELECT**);

cols — количество символьных позиций по горизонтали;

rows — количество строк, в окне текстовой области;

readonly — устанавливает текстовую область в режим просмотра;

name — имя (идентификатор) текстовой области;

tabindex — целое число, определяющее порядок перехода к элементу с помощью клавиши **<TAB>**;

disabled — делает список недоступным пользователю;

wrap — определяет режим автоматического переноса символов на другую строку; возможные значения: **physical**, **virtual** и **off** (выключено). По умолчанию включен автоматический перенос посимвольно.

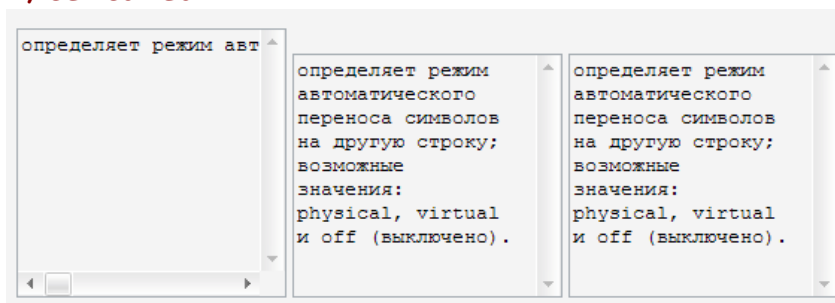
ПРИМЕР:

```
<textarea rows=10 cols=20 name=text1 wrap=off>
определяет режим автоматического переноса символов на другую
строку; возможные значения: physical, virtual и off (выключено).
</textarea>
```

```
<textarea rows=10 cols=20 name=text1 wrap=virtual >
определяет режим автоматического переноса символов на другую
строку; возможные значения: physical, virtual и off (выключено).
</textarea>
```

```
<textarea rows=10 cols=20 name=text1 wrap=physical>
```

определяет режим автоматического переноса символов на другую строку; возможные значения: `physical`, `virtual` и `off` (выключено).



2.6 Теги `<fieldset>`, `<legend>` и `<label>`

Для создания выделенного рамкой блока элементов пользовательского интерфейса служит контейнерный тег `<fieldset>`. В нем может находиться контейнерный тег `<legend>`, содержимое которого отображается в верхней части блока.

Тег `<fieldset>` не имеет атрибутов.

Тег `<legend>` имеет атрибуты:

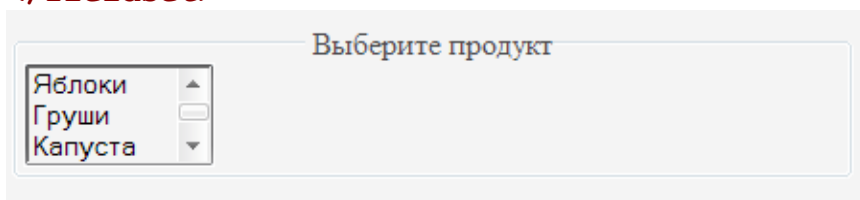
align — горизонтальное выравнивание содержимого (**left**, **right**, **center**);

accesskey — определяет клавишу быстрого доступа к первому элементу пользовательского интерфейса;

tabindex — целое число, определяющее порядок перехода с помощью клавиши `<Тав>` к первому элементу пользовательского интерфейса. Установить размеры блока по своему усмотрению можно с помощью таблиц стилей.

Пример:

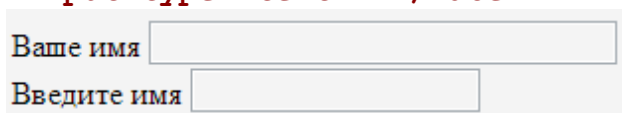
```
<fieldset>
  <legend align=center>Выберите продукт</legend>
  <select name="List1" size=3>
    <option value=1>Яблоки
    <option value=2>Груши
    <option value=3>Капуста
    <option value=4>Картофель
  </select>
</fieldset>
```



Обычно элементы пользовательского интерфейса сопровождаются поясняющими надписями. Это можно сделать, расположив текст надписи или с помощью тега `<label>` задающего расположение надписи рядом с элементом. Например:

```
Ваше имя <INPUT NAME=Name SIZE=35>
<br>
```

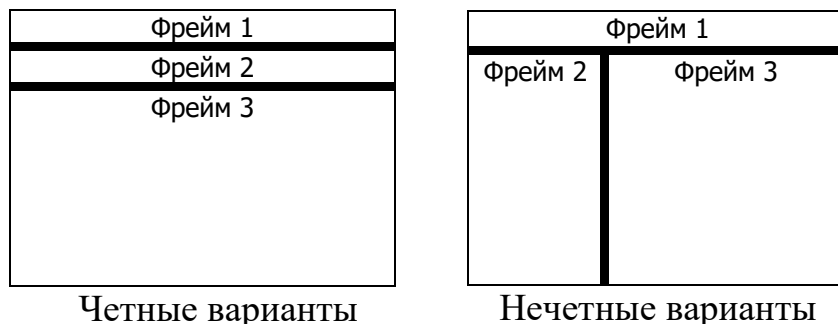
```
<label>Введите имя
<input type="text"> </label>
```



Практическое задание

Задание 1.

На рабочем диске создать папку «**Laba03**». В блокноте (или аналогичном текстовом редакторе – «**Notepad++**», создать **WEB** сайт заданного содержания. Структура сайта задается **фреймами и таблицам** (рисунок 1.).



Четные варианты

Нечетные варианты

Рисунок 1. - Структура сайта

Содержимое фреймов (рисунок 2.).

Фрейм	Описание	Чем задается	Вид		
Фрейм 1	Заголовок	Таблица	Логотип	Название	
Фрейм 2	Панель навигации	Таблица	Главная	Форма	О Авторе
Фрейм 3	Вид окна для ссылок – «Главная», «О Авторе».	произвольно			
Фрейм 3	Вид окна для ссылки – «Форма».	форма	Фрейм 4 Форма ввода данных		
			Фрейм 5 Информационная панель		

Сайт должен содержать :

1. **Логотип** – картинка;
2. **Заголовок** – текст по смыслу формы;
3. **Панель навигации** – для ссылок на внутренне страницы сайта;
4. **Главная** – описание назначения формы ввода данных;
5. **О Авторе** – информация о авторе сайте, фото, контактная информация;
6. **Форма ввода данных** – форма для ввода данных в зависимости от варианта (таблица 1.), и содержать не менее 3 – различных элементов ввода данных, кнопки «**отправить**» и «**очистить**»;
7. **Информационная панель** – краткая информация описательного характера для каждого поля формы. Ссылка располагается рядом с каждым информационным полем и открывается в фрейме №5.

Дополнительные требования к оформлению страниц:

1. Дизайн продумать самостоятельно. Сайт должен содержать разнообразные элементы форматирования, изображения, шрифты и т.д.;
2. Ширина, высота всех фреймов задается жестко. Возможно изменение высоты **только** для 5 фрейма;
3. Метод передачи – **GET**, метод кодирования данных – **application/x-www-form-urlencoded**.

Требования к оформлению отчета:

Полученный сайт продемонстрировать преподавателю и только после этого оформлять отчет. В отчете привести скриншоты главной страницы и страницы формы, а также привести их **html код и код страницы задающей структуры фреймов**. Описать пояснения к фрагментам кода задающим структуру фреймов и форму ввода данных. Привести пример строки запроса на сервер.

Пример описания кода для фреймов:

```
<FRAMESET ROWS="55, *">
```

```
    <FRAME src="head.html" noresize="noresize" >
```

- код задает 2 фрейма, первый с высотой 55px второй все оставшееся место экрана. Первый фрейм ссылается на страницу **head.html**, граница фрейма – не изменяемая.

```
<FRAMESET COLS="25%, *">
```

```
    <FRAME SRC="menu.html" noresize="noresize" >
```

```
    <FRAME SRC="frame_1.htm" name="main">
```

Во второй фрейм вставлен фрейм с двумя колонками, шириной 25% от ширины экрана, и оставшимся свободным местом. Первый фрейм ссылается на страницу **menu.html**, граница фрейма – не изменяемая. Второму фрейму задано имя **name="main">**, ссылается на страницу **frame_1.htm**.

Таблица 1. – Варианты макетов страниц

Вариант	Описание
1	Цех, участок, Ф.И.О. Объем выполненной работы
2	УДК, Ф.И.О. автора, Наименование, Количество
3	Номер поезда, Наименование, Место отправления, Место прибытия, Категория (скорый.)
4	Организация, Ф.И.О., Год рождения, Занятое место
5	Страна, Площадь, Население, Континент, Столица
6	Марка автомобиля, Номер, Цвет, Год выпуска, Пробег
7	Порода собаки, Кличка, Отец, Мать, Дата рождения, ФИО хозяина
8	Вид овощей, Название сорта, Дата посадки, Дата уборки, Урожай
9	Дисциплина, Объем лекций, Объем лабораторных, Вид контроля, Группа
10	Дата, Температура, Давление, Облачность, Направление ветра
11	Наименование, Фирма, Стоимость, Количество, Дата
12	Название группы, Страна, Альбом, Дата выпуска, Число продаж
13	Название вершины, Высота, Страна, Год покорения, Количество восхождений
14	Ф.И.О., Год рождения, Рост, Вес, Группа крови

Вопросы к защите

1. Назначение фреймов.
2. Задание структуры фреймов.
3. задание внешнего вида фреймов.
4. Формы. Назначение. Задание формы.
5. Определение элементов управления формы — тег <INPUT>.
6. Задание кнопок.
7. Задание списков.
8. Задание текстовых областей.