

Формы на Веб-странице

- Понятие формы
- Инициализация формы
- Контекст Использования
- Создание форм
- Текстовые поля
- `<textarea>`
- Поля множественного выбора и меню
- Переключатели
- Флажки
- Выпадающие списки
- Множественный выбор
- Кнопки в форме
- Поле для отправки
- Кнопка для отправки
- Другие поля
- Скрытое поле
- Поле для файла
- Организация элементов формы
- `<label>`
- `<fieldset>`
- `<legend>`
- Атрибуты формы и полей
- `disabled`
- `placeholder`
- `required`
- Дополнительные атрибуты
 - Параметры тега `<form>`
- Создание элементов управления
 - Значения параметра `type` тега `input`
- Создание меню
 - Параметры тега `<select>`
 - Параметры тега `<option>`
- Создание поля для ввода текста
 - Параметры тега `<textarea>`

Понятие формы

Формы являются неотъемлемой частью Интернета, так как они предлагают сайтам метод сбора информации от пользователей и обработки запросов, а также элементы управления практически для любого мыслимого применения. С помощью элементов управления или полей, формы могут запросить небольшой объём информации — часто это поисковый запрос, имя пользователя или пароль. Или большой объём информации — возможно, данные о посылке, платёжная информация или предложение работы.

Мы должны знать, как создавать формы чтобы получить входные данные от пользователя. В этом уроке мы обсудим, как использовать HTML для разметки формы, какие элементы использовать для захвата разных типов данных и как стилизовать формы с помощью CSS. Мы не станем слишком углубляться в то, как ин-

формация из формы обрабатывается на стороне сервера. Обработка форм представляет собой глубокую тему, вне области этого занятия; здесь мы будем придерживаться создания и стилизации форм.

Форма — это средство, позволяющее организовать на Веб-странице диалог с её пользователем. Формы, как правило, предназначены для размещения ответов на адресованные автором страницы посетителям вопросы.

В процессе навигации по сайтам пользователь, в основном, только щёлкает по **ссылкам** чтобы перемещаться по веб-страницам.

Но понятно, что пользователю иногда требуется обеспечить собственные **поля ввода**. Эти виды взаимодействия включают в себя:

- ✓ регистрацию и вход на сайтах;
- ✓ ввод личной информации (имя, адрес, данные кредитной карты и др.);
- ✓ фильтрацию контента (с помощью выпадающих списков, флажков и др.);
- ✓ выполнение поиска;
- ✓ загрузку файлов;
- ✓ выяснения спроса на предлагаемые Вашей фирмой товары или услуги;
- ✓ получения отзывов, например, на вашу Web-страницу;
- ✓ проведения опросов, тестирования.

Чтобы приспособиться к этим потребностям, HTML предлагает интерактивные **элементы управления** формы:

- ✓ текстовые поля (для одной или нескольких строк);
- ✓ переключатели;
- ✓ флажки;
- ✓ выпадающие списки;
- ✓ виджеты для загрузки;
- ✓ кнопки отправки.

Каждый элемент управления создаётся при помощи одного из соответствующих тегов языка HTML с необходимыми параметрами, которые определяют особенности работы той или иной формы.

Инициализация формы

Чтобы добавить форму на страницу мы будем использовать элемент **<form>**. Данный элемент определяет, где на странице появятся элементы управления. Кроме того, элемент **<form>** обёртывает все элементы, включенные в форму, подобно элементу **<div>**:

```
<form action="/login" method="post">
...
</form>
```

К элементу **<form>** может применяться множество различных атрибутов, наиболее распространёнными из которых являются **action** и **method**. Атрибут **action** содержит URL, на который информация в форме будет отправлена для обработки сервером. Атрибут **method** является методом HTTP, который должны использовать браузеры для отправки данных формы. Оба эти атрибута **<form>** имеют отношение к отправке и обработке данных.

Контекст Использования

Категория содержимого	Flow content
Разрешённое содержимое	Flow content, но не содержащий <form> элементов
Пропуск тега	Нет. И открывающий и закрывающий тег должны быть.
Нормативный документ	HTML5, section 4.10.3 (HTML4.01, section 17.3)

Создание форм

Подумайте о форме как о наборе элементов управления, которые работают вместе, чтобы выполнить **одну** операцию. Если вы написали форму входа на сайт, у вас могло быть **три** поля:

- ✓ поле для электронной почты **<input type="email">**
- ✓ поле для пароля **<input type="password">**

✓ кнопка отправки `<input type="submit">`

Эти три элемента HTML будут заключены внутри **одной формы** `<form action="/login" method="POST">`.

Вы можете аналогично добавить форму регистрации на той же HTML-странице, в отдельном элементе `<form>`.

Текстовые поля

Почти всем формам требуется **текстовый** ввод от пользователей, чтобы ввести своё имя, адрес электронной почты, пароль, адрес и др. Текстовые поля формы приходят в разных вариантах.

Одним из основных элементов, используемых для получения текста от пользователей, является элемент `<input>`. Данный элемент включает атрибут **type** для определения, какой тип информации будет получен в элементе управления. Наиболее популярное значение атрибута **type** — это **text**, который обозначает ввод одной строки текста.






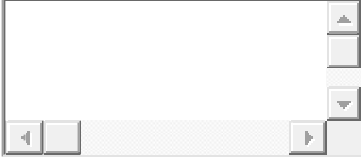
Наряду с установкой атрибута **type**, хорошей практикой будет также дать элементу `<input>` атрибут **name**. Значение атрибута **name** применяется в качестве имени элемента управления и отправляется вместе с входными данными на сервер.

```
<input type="text" name="username">
```

Первоначально было только два текстовых значения атрибута **type** — **text** и **password** (для ввода паролей), однако HTML5 привёз с собой несколько новых значений атрибута **type**.

Эти значения были добавлены, чтобы обеспечить чёткое смысловое значение для полей ввода, а также предоставить лучшее управление пользователям. Если браузер не понимает одно из этих HTML5-значений атрибута **type**, он автоматически вернётся к значению **text**. Ниже приведён список новых типов HTML5.

- | | | |
|------------|----------|--------|
| • color | • month | • tel |
| • date | • number | • time |
| • datetime | • range | • url |
| • email | • search | • week |

Текст	<code><input type="text"></code>		Допускает любой тип символов
Email	<code><input type="email"></code>		Может показывать предупреждение, если введён неверный email
Пароль	<code><input type="password"></code>		Символы показываются как точки
Число	<code><input type="number"></code>		Могут быть использованы клавиши вверх/вниз
Телефон	<code><input type="tel"></code>		Может срабатывать автозаполнение
Много-строчный текст	<code><textarea></textarea></code>		Может быть изменён размер поля

Хотя эти поля очень похожи и позволяют пользователям вводить текст любого рода (даже неправильный), их тип обеспечивает специфическую **семантику** для ввода, определяя, какую информацию поле **должно** содержать.

Браузеры могут впоследствии немного изменить интерфейс элемента управления, чтобы повысить *интерактивность* или подсказать, какое содержимое ожидается.

К примеру, поле для пароля показывает точки вместо символов. А поле для чисел позволяет увеличивать/уменьшать значение с помощью клавиш вверх и вниз.

<textarea>

Ещё одним элементом, используемым для сбора текстовых данных, является элемент **<textarea>**. Он отличается от элемента **<input>** тем, что может принимать большие отрывки текста в несколько строк. Элемент **<textarea>** также содержит начальный и конечный теги, которые могут обернуть простой текст. По-

сколькx <textarea> принимает только один тип значения, атрибут **type** здесь не применим, но атрибут **name** по-прежнему используется.

```
<textarea name="comment">Добавьте сюда свой комментарий</textarea>
```

Поля множественного выбора и меню

Помимо текстовых полей, HTML также позволяет пользователям выбирать данные, используя множественный выбор и раскрывающиеся списки. Есть несколько разных вариантов и полей для этих элементов формы, каждый из которых имеет свои отличительные преимущества.

Переключатели

Переключатели — это простой способ, позволяющий пользователям сделать быстрый выбор из небольшого списка вариантов. Переключатели дают пользователю выбрать только один вариант в отличие от нескольких.

Чтобы создать переключатель, используется элемент **<input>** со значением **radio** у атрибута **type**. Каждый переключатель должен иметь одинаковое значение атрибута **name**, чтобы все они в группе были связаны друг с другом.

С текстовыми полями их значение определяется тем, что пользователь в них набирает; с переключателями пользователь делает множественный выбор. Таким образом, мы должны определить входное значение. Используя атрибут **value** мы можем установить конкретное значение для каждого элемента **<input>**.

Кроме того, для предварительного выбора переключателя мы можем использовать логический атрибут **checked**.

```
<input type="radio" name="day" value="Friday" checked> Пятница  
<input type="radio" name="day" value="Saturday"> Суббота  
<input type="radio" name="day" value="Sunday"> Воскресенье
```

Флажки

Флажки очень похожи на переключатели. Они используют те же атрибуты и шаблоны, за исключением значения атрибута **type**. Разница между ними состоит в том, что флажки позволяют пользователям выбрать несколько значений и связать их все с одним именем, в то время как переключатели ограничивают пользователей одним значением.

```
<input type="checkbox" name="day" value="Friday" checked> Пятница  
<input type="checkbox" name="day" value="Saturday"> Суббота
```

```
<input type="checkbox" name="day" value="Sunday"> Воскресенье
```

Выпадающие списки

Выпадающие списки являются идеальным способом, чтобы практически образом предоставить пользователям длинный список вариантов. Длинный столбец переключателей рядом со списком разных вариантов не только визуально непривлекателен, но кроме того сложен и труден для понимания, особенно на мобильном устройстве. Выпадающие списки, с другой стороны, обеспечивают идеальный формат для длинного списка вариантов.

Для создания выпадающего списка мы будем применять элементы `<select>` и `<option>`. Элемент `<select>` оборачивает все пункты меню, а каждый пункт меню размечен с помощью элемента `<option>`.

Атрибут **name** располагается в элементе `<select>`, а атрибут **value** располагается в элементах `<option>`, вложенных в элемент `<select>`. Таким образом, атрибут **value** в каждом элементе `<option>` связан с атрибутом **name** элемента `<select>`.

Каждый элемент `<option>` оборачивает текст (который виден пользователям) отдельного пункта в списке.

Подобно логическому атрибуту **checked** у переключателей и флажков, для выпадающего меню можно использовать логический атрибут **selected**, чтобы предварительно выделить пункт для пользователей.

```
<select name="day">
  <option value="Friday" selected>Пятница</option>
  <option value="Saturday">Суббота</option>
  <option value="Sunday">Воскресенье</option>
</select>
```

Множественный выбор

Логический атрибут **multiple** при добавлении к элементу `<select>` для стандартного выпадающего списка позволяет пользователю выбрать более одного варианта из списка одновременно. Кроме того, с помощью логического атрибута **selected**, добавленного к более чем одному элементу `<option>`, в меню можно заранее выбрать несколько вариантов.

Размером элемента `<select>` можно управлять с помощью CSS и он должен быть скорректирован соответствующим образом для множественного выбора.

Возможно, есть смысл информировать пользователей, что для выбора нескольких вариантов они должны удерживать клавишу **Shift** во время щелчка, чтобы сделать выбор.

```
<select name="day" multiple>
  <option value="Friday" selected>Пятница</option>
  <option value="Saturday">Суббота</option>
  <option value="Sunday">Воскресенье</option>
</select>
```

Кнопки в форме

После того, как пользователь вводит запрашиваемую информацию, кнопки позволяют пользователю пустить эту информацию в дело. Чаще всего для обработки данных используются поле или кнопка для отправки.

Поле для отправки

Когда пользователь щёлкает по кнопке, данные формы обрабатываются после её заполнения. Кнопка для отправки создаётся с помощью элемента **<input>** со значением **submit** у атрибута **type**. Атрибут **value** применяется для указания текста, который отображается внутри кнопки.

```
<input type="submit" name="submit" value="Отправить">
```

Кнопка для отправки

Кнопка для отправки в виде элемента **<input>** является самодостаточной и не может оборачивать любой другой контент. Если хочется иметь больше контроля над структурой и дизайном поля, наряду с возможностью обернуть другие элементы — тогда может быть использован элемент **<button>**.

Элемент **<button>** выполняет то же самое, что и элемент **<input>** со значением **submit** у атрибута **type**. Однако, он включает в себя открывающий и закрывающий теги, которые могут обернуть другие элементы. По умолчанию, элемент **<button>** действует, как если у атрибута **type** задано значение **submit**, поэтому атрибут **type** и его значение можно по желанию опустить.

Вместо использования атрибута **value** для управления текстом в кнопке для отправки, будет показан текст, который пишется между открывающим и закрывающим тегами элемента **<button>**.

```
<button name="submit">
  <strong>Отправьте нам</strong> сообщение
```



```
</button>
```

Другие поля

Помимо применения, которое мы только что обсудили, элемент **<input>** имеет несколько других вариантов использования. Они включают в себя получение скрытых данных и прикрепление файлов в процессе обработки формы.

Скрытое поле

Скрытые поля предоставляют способ передачи данных на сервер без отображения их пользователям. Скрытые поля обычно используются для отслеживания кодов, ключей или другой информации, которая не имеет отношения к пользователю, но может быть полезна при обработке формы. Эта информация не отображается на странице, однако может быть найдена путём просмотра исходного кода страницы. По этой причине она не должна применяться для уязвимой или защищённой информации.

Чтобы создать скрытое поле используйте значение **hidden** атрибута **type**. Дополнительно включает в себя соответствующие значения атрибутов **name** и **value**.

```
<input type="hidden" name="tracking-code" value="abc-123">
```

Поле для файла

Чтобы позволить пользователям добавить файл в форму, вроде прикрепления файла к письму, используйте значение **file** атрибута **type**.

```
<input type="file" name="file">
```

К сожалению, стилизация с помощью CSS элемента **<input>**, у которого значение атрибута **type** задано как **file**, является трудной задачей. Каждый браузер содержит свой собственный стиль поля по умолчанию и ни один не даёт много свободы, чтобы переопределить этот стиль. JavaScript и другие решения могут быть использованы для этого, но они несколько сложны для построения.

Организация элементов формы

Знать, как получить данные с полей формы, это лишь половина дела. Другая половина — это организация элементов формы и полей в удобном порядке. При

взаимодействии с формами пользователи должны понять что от них требуется и как предоставить запрашиваемую информацию.

С помощью **<label>**, **<fieldset>** и **<legend>** мы можем лучше организовать формы и направлять пользователей правильно их завершать.

<label>

Элемент **<label>** содержит подписи или заголовки для управления формой, однозначно связывая их вместе, создавая тем самым доступную форму для всех пользователей и вспомогательных технологий. **<label>** должны включать в себя текст, описывающий поля к которым они относятся.

<label> могут включать в себя атрибут **for**, его значение должно быть таким же, как значение атрибута **id** у элемента, с которым связан **<label>**. Соответствие значений атрибутов **for** и **id** связывает два элемента вместе, что позволяет пользователям нажать на **<label>** и передать фокус нужному полю формы.

```
<label for="username">Имя пользователя</label>
<input type="text" name="username" id="username">
```

При желании, **<label>** может обернуть поля формы, такие как переключатели или флажки. Это позволяет опустить атрибуты **for** и **id**.

```
<label>
  <input type="radio" name="day" value="Friday" checked> Пятница
</label>
<label>
  <input type="radio" name="day" value="Saturday"> Суббота
</label>
<label>
  <input type="radio" name="day" value="Sunday"> Воскресенье
</label>
```

<fieldset>

<fieldset> группирует поля формы в организованные разделы. Подобно **<section>** или иным структурным элементам, но **<fieldset>** является блочным элементом, который оборачивает связанные элементы, в частности, в **<form>**, для их лучшей организации. **<fieldset>** по умолчанию также включают в себя границы контура, которые могут быть изменены с помощью CSS.

```
<fieldset>
  <label>
    Имя пользователя
    <input type="text" name="username">
```

```
</label>
<label>
  Пароль
  <input type="text" name="password">
</label>
</fieldset>
```

<legend>

Элемент **<legend>** предоставляет подпись или заголовок для элемента **<fieldset>**. Элемент **<legend>** оборачивает текст, описывающий элементы управления формы, которые находятся внутри **<fieldset>**. Разметка должна включать в себя элемент **<legend>** сразу после открывающего тега **<fieldset>**. На странице подпись появится в левом верхнем углу рамки **<fieldset>**.

```
<fieldset>
  <legend>Вход в систему</legend>
  <label>
    Имя пользователя
    <input type="text" name="username">
  </label>
  <label>
    Пароль
    <input type="text" name="password">
  </label>
</fieldset>
```

Атрибуты формы и полей

Для настройки всех различных форм, полей и элементов управления, существует ряд атрибутов и соответствующих значений. Эти атрибуты и значения выполняют несколько разных функций, таких как отключение полей и добавление проверки форм. Далее описаны некоторые часто используемые и полезные атрибуты.

disabled

Логический атрибут **disabled** выключает элемент управления таким образом, что он не доступен для взаимодействия или ввода. Заблокированные элементы не будут отправлять никакого значения на сервер для обработки формы.

Применение атрибута **disabled** к элементу **<fieldset>** отключит все элементы управления формы внутри группы.

```
<label>
  Имя пользователя
  <input type="text" name="username" disabled>
```

```
</label>
```

placeholder

Атрибут **placeholder** в HTML5 предлагает подсказку или совет внутри элемента **<input>** или **<textarea>**, которая исчезает при щелчке по элементу управления или при получении фокуса. Это применяется, чтобы дать пользователям дальнейшую информацию о том, как поле формы должно быть заполнено, к примеру, использовать формат электронной почты.

```
<label>
  Email
  <input type="email" name="email-address" placeholder="name@domain.com">
</label>
```

Основное различие между атрибутами **placeholder** и **value** в том, что текст **value** остаётся на месте, когда элемент управления получает фокус, пока пользователь не удалит его вручную. Это очень удобно для предварительно заполненных данных, таких как личная информация, но не для предоставления указаний.

required

Логический атрибут **required** в HTML5 утверждает, что элемент формы должен содержать значение при отправке на сервер. Если у элемента формы нет значения, будет отображаться сообщение об ошибке с просьбой пользователю заполнить обязательное поле. В настоящее время стили сообщения об ошибке контролируются браузером и не могут быть оформлены в CSS. Некорректные элементы формы, с другой стороны, могут быть стилизованы с помощью псевдоклассов **:optional** и **:required**.

Проверка также специфична для конкретного типа элемента управления. Например, элемент **<input>** со значением **email** у атрибута **type** потребует не только чтобы значение в поле существовало, но также чтобы это был корректный адрес электронной почты.

```
<label>
  Email
  <input type="email" name="email-address" required>
</label>
```

Дополнительные атрибуты

Элементы форм также включают следующие атрибуты, но не ограничиваются ими. Пожалуйста, не стесняйтесь исследовать эти атрибуты при необходимости (подробнее <https://www.w3.org/Submission/web-forms2/>)

- accept
- autocomplete
- autofocus
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- max
- maxlength
- min
- pattern
- readonly
- selectionDirection
- step

Вначале задаётся область формы с помощью парного тега элемента **<form>...</form>**.

Затем между этими тегами размещаются необходимые фрагменты формы с помощью соответствующих элементов, предусмотренных языком HTML. Завершается форма кнопками **submit** и **reset** для отправки значений заполненных полей формы на сервер для обработки и очистки этих полей перед следующим заполнением, соответственно.

Кнопка **submit** отправляет содержимое заполненной формы программе обработки на Веб-сервер. Программы обработки называются CGI-скриптами и разрабатываются на различных языках программирования. Но если создатель сайта не владеет веб-программированием, то тогда можно отправить результаты формы-анкеты на почтовый адрес.

Параметры тега **<form>**

Параметр	Описание	Пример
action	<u>Обязательный параметр</u> . Задёт адрес и имя программы-обработчика формы либо способ пересылки по электронной почте.	<pre><form action="http://Адрес/Имя программы">Элемент формы </form></pre> <pre><form action="mailto:v.fufaev@niks.by">Элемент формы </form></pre>
method	Задаёт метод ввода данных на сервере программы-обработчика формы. Допустимые значения: ✓ GET (по умолчанию)- метод, при котором данные вводятся в командную строку; ✓ POST - метод, при котором данные вводятся через специальный ввод (используется чаще).	<pre><form method=post action=mailto:v.fufaev@niks.by>Элемент формы </form></pre>
name	Задаёт имя формы	<pre><form ac-</pre>

		<pre>tion=mailto:v.fufaev@niks.by NAME="Test_form"> </form></pre>
--	--	---

Создание элементов управления

Специальные теги позволяют создавать различные элементы управления форм, такие как:

- ✓ флажки,
- ✓ переключатели,
- ✓ поля ввода,
- ✓ кнопки.

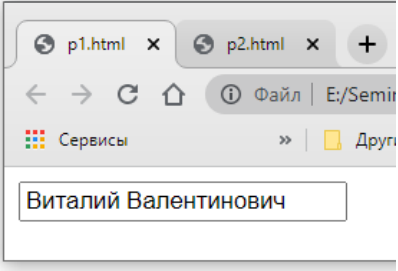
Для этих целей предназначен непарный тег **<input>**. Он должен располагаться внутри тега **<form>**. Многие параметры тега являются обязательными:

- ✓ **name** - параметр обязательный, кроме кнопок подтверждения и сброса. Значение атрибута определяет имя поля формы.
- ✓ **value** - значение параметра задаёт первоначальный текст поля ввода после загрузки страницы, а также определяет надпись на кнопке.
- ✓ **checked** - его наличие показывает, что флажок должен быть установлен по умолчанию.
- ✓ **size** - задаёт видимую длину поля ввода. Длина выражается в символах.
- ✓ **maxlength** - определяет максимальную длину текстовой строки в поле ввода. По умолчанию вводимая текстовая строка может иметь бесконечную длину.
- ✓ **readonly** - закрывает возможность для редактирования элемента.
- ✓ **type** - определяет вид элемента.
- ✓ **src** - задаёт URL-адрес картинки, используемой при создании графической кнопки, инициирующей передачу данных. Используется совместно с параметром `type=image`
- ✓ **align** - определяет вертикальное выравнивание для изображений.

Значения параметра *type* тега *input*

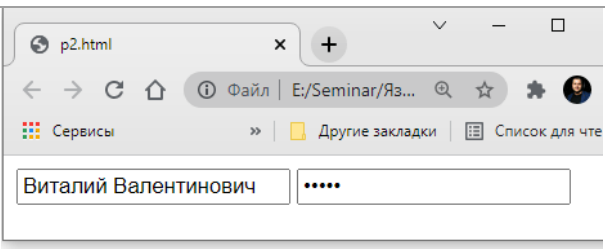
`type=`**text** (по умолчанию)

Создание поля ввода в одну строку, в котором можно сразу после загрузки страницы разместить произвольный текст, используя параметр **value**.

HTML-код	Отображение в браузере
<pre><form action="mailto:v.fufaev@niks.by"> <input name=test1 value="Виталий Валентинович" size="20"> </form></pre>	

`type=password`

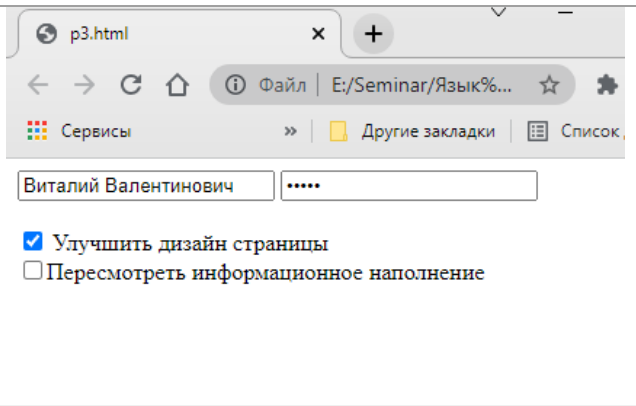
Создание поля для ввода пароля. Введённая информация отображается звездочками (кружочками). Сразу после загрузки страницы значение поля ввода определяется значением количества символов параметра **value** в виде соответствующего количества звёздочек (кружочков).

HTML-код	Отображение в браузере
<pre><form action="mailto:v.fufaev@niks.by"> <input name=test1 value="Виталий Валентинович" size="20"> <input type=password name="p1" value="11111"> </form></pre>	

type=**checkbox**

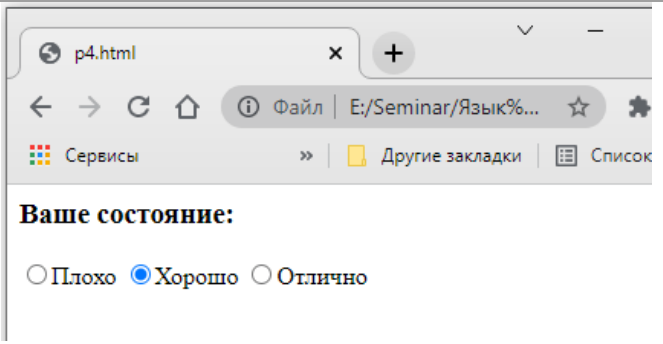
Создание флажка.

HTML-код	Отображение в браузере
<pre><form action="mailto:v.fufaev@niks.by"> <input name="test1" value="Виталий Валентинович" size="20"> <input type="password" name="p1" value="11111">

 <input type="checkbox" name="f1"> Улучшить дизайн страницы
 <input type="checkbox" name="f2"> Пересмотреть информационное наполнение </form></pre>	

type=**radio**

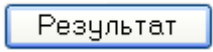
Определение переключателя из группы, может быть нажат только один из них. Если один из переключателей должен быть выбран по умолчанию, то вставляется параметр **checked** без значения.

HTML-код	Отображение в браузере
<pre><form action="mailto:v.fufaev@niks.by"> <h3>Ваше состояние:</h3> <input type="radio" name="s1" value="первый">Плохо <input type="radio" name="s1" value="второй" checked>Хорошо <input type="radio" name="s1" value="третий">Отлично </form></pre>	

*Мы получили 3 кнопки, из них отмечена Хорошо. При нажатии на другую кнопку предыдущая освобождается. Допустим, мы нажали кнопку Отлично. Тогда при нажатии на кнопку **SUBMIT** в программу обработки или по электронной почте будет введено: S1=Третий*

type=**button**

Создание кнопки произвольного назначения. Атрибут **value** используется для определения надписи на кнопке.

HTML-код	Отображение в браузере
<pre><input type="button" name="b1" value=Результат></pre>	

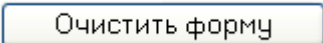
type=submit

Создание кнопки, щелчок на которой подтверждает ввод информации в форму. Атрибут **value** используется для определения надписи на кнопке.

HTML-код	Отображение в браузере
<pre><input type="submit" value="Отправить с приветом!"></pre>	

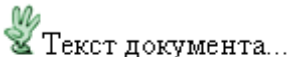

type=reset

Кнопка для отмены ввода данных в форму.

HTML-код	Отображение в браузере
<pre><input type="reset" value="Очистить форму"></pre>	


type=image

Создание кнопки с рисунком для отсылки заполненной формы. Для указания графического файла используется атрибут **src**. Сочетание текста и графики невозможно.

HTML-код	Отображение в браузере
<pre><input type="image" src=hand.gif name="b2" align=bottom alt=подтвердить> Текст докумен- та...</pre>	 

type=file

Средство выбора файла для присоединения к форме. Имя файла записывается в поле ввода. Кроме того, браузер автоматически создает рядом с полем ввода кнопку Обзор, которая позволяет запустить стандартный для операционной системы диалог выбора файлов.

HTML-код	Отображение в браузере
<pre><input type="file" name="F1" size=45></pre>	

Создание меню


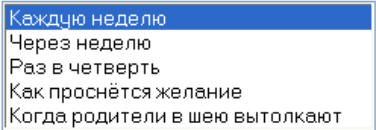
Парный `<select>...</select>` используется для создания в заполняемой форме меню типа «Выбор одного пункта из многих» или «Выбор нескольких пунктов из многих». Количество непарных тегов `<option>` внутри `<select>...</select>` определяет число пунктов списка.

Параметры тега `<select>`

- ✓ **name** - задаёт имя меню;
- ✓ **multiple** - разрешает выбрать сразу несколько пунктов списка. Атрибут не имеет значения;
- ✓ **size** - определяет количество видимых на экране пунктов списка. По умолчанию видна только первая строка, а при щелчке на кнопке со стрелкой раскрывается полный список. При заданном определенном значении список не раскрывается, а прокручивается. Одновременно видно заданное атрибутом количество пунктов.

Параметры тега `<option>`

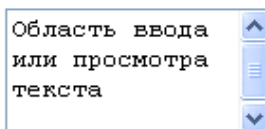
- ✓ **selected** - определяет, какой из пунктов списка должен быть выбран по умолчанию.
- ✓ **value** - значение атрибута в паре с именем меню (значением атрибута **name**) передается на сервер для обработки формы.

HTML-код	Отображение в браузере
<pre><h4>Как часто вы посещаете занятия на упк?</h4> <select name=q1> <option selected value="m1">Каждую неделю <option value="m2">Через неделю </select></pre>	<p>Как часто Вы посещаете занятия на УПК?</p> 
<pre><h4>Как часто вы посещаете занятия на упк?</h4> <select name=q2 size=5> <option selected value="t1">Каждую неделю <option value="t2">Через неделю <option value="t3">Раз в четверть <option value="t4">Как проснётся желание <option value="t5">Когда родители в шею вы- толкают </select></pre>	<p>Как часто Вы посещаете занятия на УПК?</p> 

Создание поля для ввода текста

Парный тег `<textarea>...</textarea>` используется для создания области фиксированного размера для ввода или просмотра текста неограниченного размера.

Пример области ввода:



Параметры тега `<textarea>`

- ✓ **name** - обязательный параметр. Определяет название, которое будет использоваться при идентификации заполненного поля сервером.
- ✓ **rows** - определяет количество строк текста, видимых на экране.
- ✓ **cols** - определяет ширину текстового поля в печатных символах.
- ✓ **wrap** - определяет способ переноса слов в заполняемой данной заполняемой форме. Возможные значения:

- **off** - перенос слов не происходит (значение по умолчанию)
- **virtual** - перенос слов только отображается, на сервер же поступает неделимая строка.
- **physical** - перенос слов будет происходить во всех точках переноса.

HTML-код	Отображение в браузере
<pre><p>Выскажите своё мнение о качестве преподавания в УПК:</p> <textarea name="test" rows=15 cols=35> </textarea></pre>	