

HTML&CSS: Forms

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ПОИГРАЕМ ;)

■ Всегда ли основная ось горизонтальная?

■ Что делает свойство `justify-content`?

■ Что делает свойство `align-items`?

■ Что делает свойство `align-content`?

ЦЕЛЬ

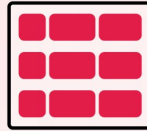
Изучить HTML Form, какие существуют поля ввода для разных типов информации

ПЛАН ЗАНЯТИЯ

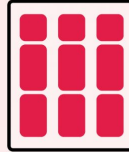
- 1. Свойства для flex элементов
- 2. Принципы работы форм
- 3. Теги и атрибуты форм

CSS Flexbox

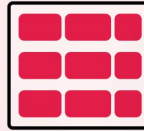
flex-direction



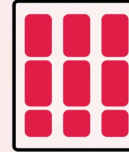
row



column



row-reverse

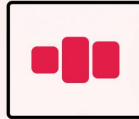


column-reverse

align-items



flex-start



center



flex-end



stretch

justify-content



flex-start



center



flex-end



space-between

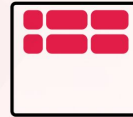


space-around

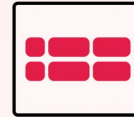


space-evenly

align-content



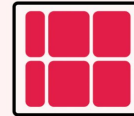
flex-start



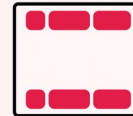
center



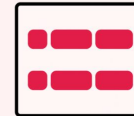
flex-end



stretch

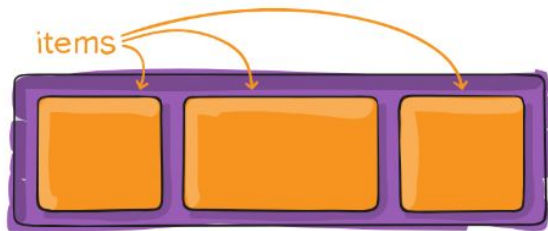


space-between



space-around

Свойства для дочерних элементов(flex элементы)



order

По умолчанию flex элементы располагаются в исходном порядке. Однако свойство `order` управляет порядком их появления в контейнере flex

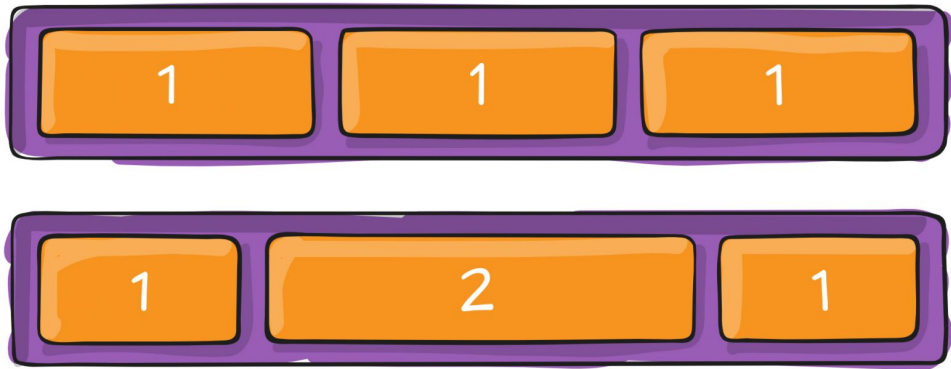
```
.item {  
  order: <integer>; /* default is 0 */  
}
```



Свойства для дочерних элементов(flex элементы)

flex-grow

Это свойство определяет способность flex элемента растягиваться в случае необходимости. Оно принимает значение от нуля, которое служит пропорцией



```
.item {  
  flex-grow: <number>; /* default 0 */  
}
```

Отрицательные числа не поддерживаются.

Свойства для дочерних элементов(flex элементы)

flex-shrink

Это свойство определяет способность гибкого элемента сжиматься при необходимости.

```
.item {  
    flex-shrink: <number>; /* default 1 */  
}
```

flex-basis

Задаёт базовый размер флекс элемента по основной оси. Это свойство определяет размер контент-бокса

```
.item {  
    flex-basis: <length> | auto; /* default auto */  
}
```

Свойства для дочерних элементов(flex элементы)

flex

Это сокращение для использования **flex-grow**, **flex-shrink** и **flex-basis** вместе. Второй и третий параметры (flex-shrink и flex-basis) являются необязательными. По умолчанию это 0 1 auto.

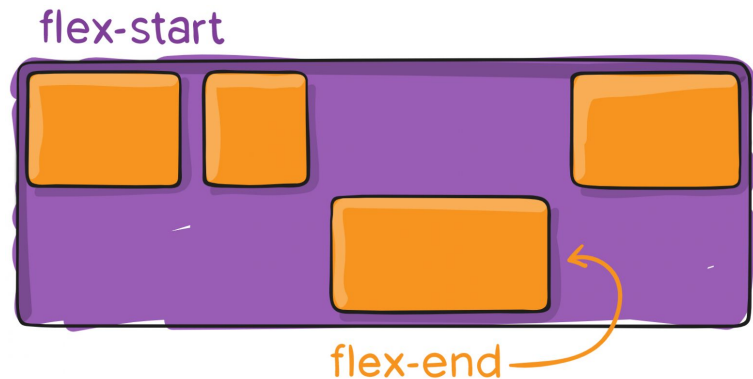
```
.item {  
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]  
}
```

Свойства для дочерних элементов(flex элементы)

align-self

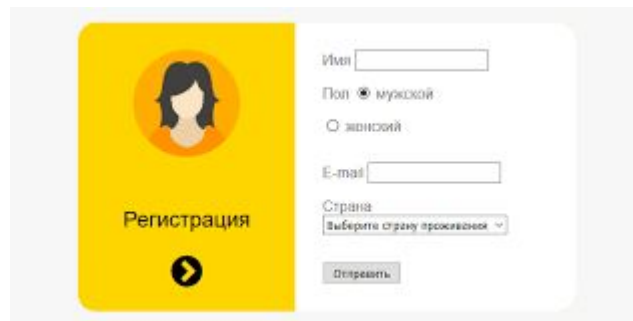
Это свойство позволяет переопределить выравнивание по умолчанию (или указанное с помощью align-items) для отдельных элементов flex.

(см. align-items свойство, чтобы понять доступные значения)



```
.item {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```

Принципы работы форм

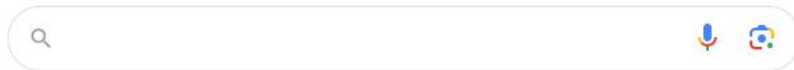


The image shows a registration form with a yellow sidebar on the left and a white main area on the right. The sidebar contains a circular profile picture placeholder with a woman's silhouette, the word "Регистрация" (Registration), and a right-pointing arrow icon. The main area contains the following fields and controls:

- Имя (Name): A text input field.
- Пол (Gender): Radio buttons for "мужской" (male) and "женский" (female).
- E-mail: A text input field.
- Страна (Country): A dropdown menu with the text "выберите страну проживания" (select country of residence).
- Отправить (Send): A button at the bottom right.

Форма

Веб-форма — это набор текстовых полей, списков, кнопок и других активизируемых щелчком мыши элементов управления, посредством которых посетитель страницы может предоставить ей тот или иной вид информации. Формы в Интернете повсюду — благодаря формам мы можем создавать учетные записи электронной почты, просматривать и покупать товары в интернет-магазинах, осуществлять финансовые транзакции и многое другое.

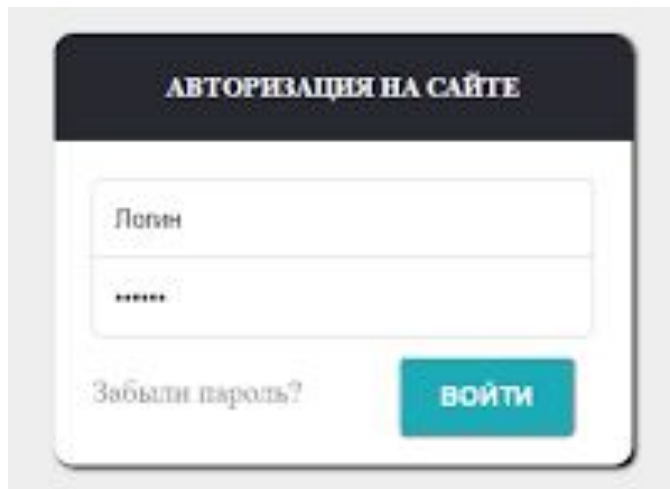


Google Search

I'm Feeling Lucky

Виды взаимодействия с помощью форм 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.



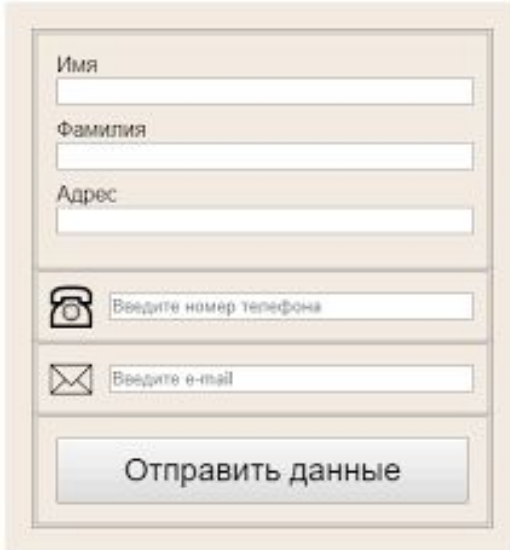
The image shows a login form with a dark header containing the text "АВТОРИЗАЦИЯ НА САЙТЕ". Below the header, there are two input fields: the first is labeled "Логин" and the second is for a password, indicated by dots. Below the password field, there is a link "Забыли пароль?" and a teal button labeled "ВОЙТИ".

Принцип работы

Форма предназначена для обмена данными между пользователем и сервером.

Процесс отправки данных формы на сервер заключается в том, что при нажатии на кнопку submit, данные формы отправляются на указанный адрес.

Думайте о форме как о наборе элементов управления, которые работают вместе, чтобы выполнить одну операцию.



The image shows a web form with a light beige background and a thin border. It contains the following elements:

- Имя** (Name): A text input field.
- Фамилия** (Surname): A text input field.
- Адрес** (Address): A text input field.
- Phone**: A section with a telephone icon and a text input field labeled "Введите номер телефона".
- Email**: A section with an envelope icon and a text input field labeled "Введите e-mail".
- Submit Button**: A large button at the bottom labeled "Отправить данные".

Теги и атрибуты форм



Элемент **<form>**

Элемент `<form>` определяет форму и атрибуты, которые определяют поведение этой формы. Элемент **`<form>`** является блочным элементом

Все элементы управления (**`<input>`**, **`<textarea>`** ,**`<button>`** и т.п.) должны находиться внутри элемента **`<form>`**.

`<form>`

...

элементы формы

....

`</form>`

Атрибуты элемента **<form>**

1. Атрибут **action** определяет действие, которое должно быть выполнено при отправке формы. Его значением является URL, который указывает путь к обработчику формы. Обычно данные формы отправляются в файл на сервере, когда пользователь нажимает на кнопку отправки.

<form **action="./action.php">**

...

</form>

В данном случае данные отправляются в файл с именем **"action.php"**. Этот файл содержит серверный скрипт, который обрабатывает данные формы

В этом примере данные отправляются на абсолютный URL

```
<form action="http://page.com">
```

Если атрибуты не указаны, как показано ниже, данные из формы **<form>** отправляются на ту же страницу, на которой размещается данная форма

Элемент `<input>`

Одним из наиболее часто используемых элементов формы является элемент `<input>`. Он позволяет создать различные типы полей. Рассмотрим какие типы полей можно создать при помощи него и его атрибута `type`.

1.

2. ☒

3. ☐

4.

5.

6.

7. Файл не выбран

8.

9.

10.

11.

12.

13. ☐

14.

15.

16.

17.

18.

19.

20.

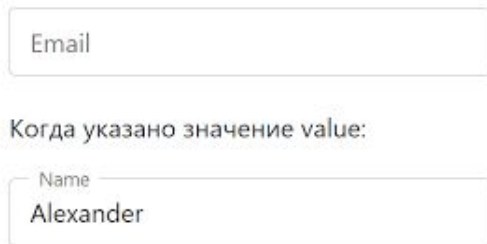
21.

22.

1. **button** - создает кнопку,
2. **checkbox** - создает флажок,
3. **color** - поле для ввода цвета, генерирует палитры цветов,
4. **date** - позволяет вводить дату в формате дд.мм.гггг,
5. **datetime-local** - позволяет вводить дату и время,
6. **email** - поле для ввода почтового адреса,
7. **file** - позволяет загружать файлы с компьютера пользователя,
8. **hidden** - скрывает элемент управления,
9. **image** - создает кнопку, на кнопке можно вставить изображение,
10. **month** - позволяет ввести месяц и год,
11. **number** - предназначено для ввода целочисленных значений. Атрибуты min, max и step задают верхний, нижний пределы и шаг между значениями,
12. **password** - поле для ввода пароля,
13. **radio** - создает переключатель, который можно включить или выключить,
14. **range** - создает ползунок, min и max позволяют установить диапазон,
15. **reset** - кнопка для очистки формы,
16. **search** - обозначает поле поиска,
17. **submit** - кнопка для отправки данных формы на сервер,
18. **tel** - поле для ввода номера телефона,
19. **text** - поле для ввода текста,
20. **time** - позволяет вводить время по шаблону чч:мм,
21. **url** - поле предназначено для указания URL-адресов,
22. **week** - позволяет пользователю выбрать одну неделю в году.

Основные атрибуты элемента `<input>`

- `name` - имя элемента
- `type` - задаёт тип тега `input` (мы рассматривали его на предыдущих слайдах). По умолчанию значение атрибута - `"text"`
- `autocomplete` - автозаполнение для данного поля. Значения `off` и `on`
- `disabled` - делает элемент недоступным. Недоступные элементы не передаются на сервер.
- `value` - значение поля по умолчанию или надпись на кнопке.
- `placeholder` - текст-подсказка в поле формы, который пропадает автоматически при получении поле фокуса.



Email

Когда указано значение value:

Name
Alexander

Основные атрибуты элемента `<input>`

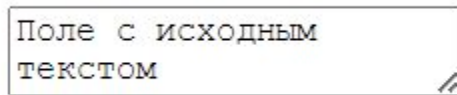
- `autofocus` - автоматическое получение фокуса при загрузке страницы.
- `min` - минимальное значение в полях типа `date`, `number` и `range`.
- `max` - максимальное значение в полях типа `date`, `number` и `range`.
- `size` - размер поля типа `text`.
- `maxlength` -ограничение на количество символов, которые можно ввести в поле типа `text`
- `required` - автоматически проверяет, заполнено ли поле.



Элемент `<textarea>`

Поле `<textarea>` представляет собой элемент формы для создания области, в которую можно вводить несколько строк текста. В отличие от тега `<input>` в текстовом поле допустимо делать переносы строк, они сохраняются при отправке данных на сервер.

Между тегами `<textarea>` и `</textarea>` можно поместить любой текст, который будет отображаться внутри поля.

An empty rectangular text area with a thin border and a small cursor icon at the bottom right corner.A rectangular text area with a thin border, containing the text "Поле с исходным текстом" (Field with original text) and a small cursor icon at the bottom right corner.

Атрибуты элемента `<textarea>`

- `cols` - ширина поля в символах.
- `rows` - высота поля в строках текста.

Примечание: для данного элемента доступны общие атрибуты, которые были рассмотрены ранее (напр, **name**, **placeholder**, **form**, **required**, **disabled**).

Элемент `<select>`

Тег **`<select>`** позволяет создать элемент интерфейса в виде раскрывающегося списка, а также список с одним или множественным выбором.

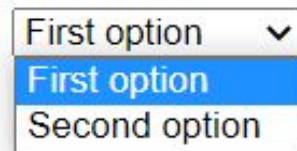
Ширина списка определяется самым широким текстом, указанным в теге **`<option>`**, а также может изменяться с помощью стилей. Каждый пункт создается с помощью тега **`<option>`**, который должен быть вложен в контейнер **`<select>`**.

Элемент `<select>`

Код

```
<select>  
  <option>First option</option>  
  <option>Second option</option>  
</select>
```

Результат



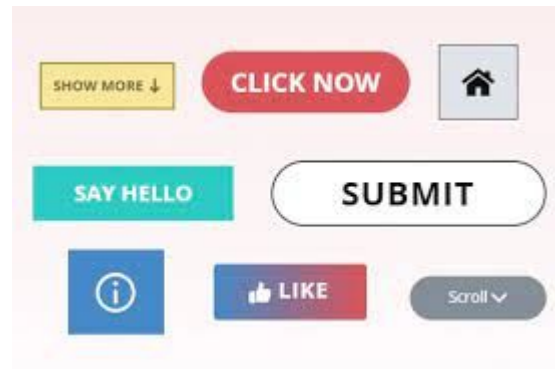
Основные атрибуты элемента `<select>`

- `disabled` - блокирует доступ и изменение элемента.
- `name` - имя элемента для отправки на сервер или обращения через скрипты.
- `required` - список обязателен для выбора перед отправкой формы.
- `size` - количество отображаемых строк списка.

Элемент <button>

Тег **<button>** создает на веб-странице кнопки и по своему действию напоминает результат, получаемый с помощью тега **<input>** (с атрибутом `type="button | reset | submit"`).

В отличие от этого тега **<input>**, **<button>** предлагает расширенные возможности по созданию кнопок. Например, на подобной кнопке можно размещать любые элементы HTML, в том числе изображения. Используя стили можно определить вид кнопки путем изменения шрифта, цвета фона, размеров и других параметров.



Атрибуты элемента `<button>`

Элемент `<button>`, как и другие элементы форм поддерживает атрибуты `disabled`, `form`, `name`.

Рассмотрим подробнее атрибут `type`:

Доступные значения:

- `submit`: Кнопка отправляет данные формы на сервер. Это значение по умолчанию, если атрибут не указан или если атрибут динамически изменен на пустое или недопустимое значение.
- `reset`: Кнопка сбрасывает все элементы управления к их начальным значениям. Удаляет данные, введенные в форму.
- `button`: Кнопка не имеет поведения по умолчанию. При этом на странице могут быть скрипты, активируемые при возникновении определённых событий на кнопке.

Элемент `<label>`

Так как элементы формы сами по себе не очень описательны, им, как правило, предшествует текстовая метка - label.

`<label>Email</label>`

`<input type="email">`

HTML label Tag

Student ☒

Business ☐

Other ☐

В то время как placeholder уже обеспечивает некоторую подсказку о том, какое ожидается содержание, метки имеют преимущество оставаясь видимыми в любое время и могут использоваться наряду с другими типами элементов формы, таких как флажки или переключатели.

Элемент `<label>`

Вы можете применять короткие абзацы для описания элементов формы, использование `<label>` является семантически более правильным, потому что они существуют только в формах. `<label>` также может быть связан с определённым элементом формы с помощью атрибута `for`, соответствующему значению `id` у поля.

```
<label for="first_name">Имя</label>
```

```
<input id="first_name">
```

При щелчке по метке фокус переходит к текстовому полю и помещает курсор внутрь него

Click me



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH