

# Формы обратной связи

Работа с формами на фронтенде

# Введение в формы

- С помощью HTML и CSS мы можем создать внешний вид формы
- Чтобы форма работала, нам нужен сервер (можно сделать рабочую форму, используя PHP)
- Сейчас мы будем работать над внешним видом

# Тэг form

- Атрибут action позволяет настроить, на какую страницу будет происходить отправка формы
- Атрибут method определяет, с помощью какого запроса будет происходить отправка формы (GET, POST)
- POST – более безопасный метод отправки
- GET – менее безопасный, т.к. вся информация будет отображаться в URL
- Пример GET запроса в поиске

# Label и input

- Атрибут type определяет тип ввода (есть много разных типов)
- Атрибут name нужен, чтобы работать с информацией, пришедшей с формы на сервере (это уникальное имя поля ввода)
- Label нужен чтобы добавить подпись к полю ввода
- Атрибут for нужен, чтобы ассоциировать label с полем ввода
- По умолчанию input и label – строчные

# Атрибуты placeholder, value, required, disabled

- Value – позволяет задать значение любого поля ввода
- Placeholder – подсказка для пользователя о том, что должно быть введено в поле
- Required – делает поле обязательным, добавляя встроенную в HTML5 валидацию
- Disabled – делает поле ввода невозможным для использования

# Типы полей ввода

Некоторые типы полей позволяют добавить встроенную валидацию, другие – сделать более сложные типы ввода (радиокнопки, чекбоксы)

- Text
- Email
- Textarea
- Select + option
- Number
- Date

# Чекбоксы и радиокнопки

- Для того чтобы сгруппировать поля ввода, нужно задать им одинаковое имя
- Чтобы задать чекбокс, нужно использовать `type="checkbox"`
- Чтобы задать радиокнопку, нужно использовать `type="radio"`

# Тип submit и кнопки

- `<input type="submit" name="submit" value="submit">` создает кнопку, с помощью которой можно будет отправить форму
- Элемент `button`



# Немного о подходах к дизайну форм

- Чем форма короче, тем лучше
- Лучше сделать несколько этапов чем одну длинную форму
- Лишние поля лучше убрать
- Лейблы лучше плейсхолдеров
- Пользователь должен понимать, в какой форме должен быть ввод

# Добавление стилей формам

- Использование имени класса form-group для группировки поля ввода и лейбла
- Базовая стилизация полей ввода и лейблов

# Стилизация кнопок

- <https://css-tricks.com/overriding-default-button-styles/>
- <https://github.com/necolas/normalize.css/blob/master/normalize.css>

# Псевдо-классы

- :hover – наведение
- :focus – поле ввода, получившее фокус (например переключением табуляции или наведением)
- :active – активизация (нажатие) на элемент
- :required – обязательное поле
- :valid – валидное поле
- :invalid – не валидное поле
- :disabled – задизейбленное поле
- :checked – выбранное поле для переключателей
- 

[https://developer.mozilla.org/en-US/docs/Learn/Forms/UI\\_pseudo-classes](https://developer.mozilla.org/en-US/docs/Learn/Forms/UI_pseudo-classes)

# Кастомные переключатели

- Appearance: none – чтобы убрать стандартные стили браузера
- Использование псевдо-элемента ::before и позиционирования для создания кастомного переключателя
- [https://developer.mozilla.org/en-US/docs/Learn/Forms/Advanced\\_form\\_styling](https://developer.mozilla.org/en-US/docs/Learn/Forms/Advanced_form_styling)
- <https://css-tricks.com/custom-styling-form-inputs-with-modern-css-features/>

# Проект

Создать форму, похожую на этот codepen  
<https://codepen.io/freeCodeCamp/full/VPaoNP>

- Требования:
- Сделать заголовок и подзаголовок
- Настроить фон
- Сделать полупрозрачный фон для формы
- Использовать поля ввода типов text, email, number, select, textarea, radio, checkbox, кнопку
- Для textarea добавить плейсхолдер
- По желанию – сделать кастомные переключатели