

ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	ИПТИП
КАФЕДРА	Индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Методические указания к практическим занятиям
ПРЕПОДАВАТЕЛЬ	Загородних Николай Анатольевич
СЕМЕСТР	1 семестр, 2025/2026 уч. год

Практика 3. Формы + модалка + базовая валидация

Цель:

Сверстать доступную форму обратной связи, встроить её в модальное окно, подключить минимальную клиентскую валидацию без библиотек. Получить рабочий результат без серверной части.

Выход:

Рабочая форма в модальном окне `<dialog>` (или «ручной» модалке), встроенная валидация, сообщение об успехе. Публикация демо и PR dev → main.

1) Теория

1.1. Базовые элементы формы и семантика

Формы — основной способ ввода данных на сайте. Метки (label) делают поля доступными для клавиатуры и экранных читалок, атрибуты name и autocomplete помогают автозаполнению и отправке.

Ключевые элементы:

- `<form>` — контейнер формы.
- `<input>` — поле ввода (поддерживает типы: text, email, tel, date и др.).
- `<label>` — подпись для поля; связывается через `for="id"` и сам `id` у поля.
- `<button>` — кнопка; по умолчанию не «submit», задавайте тип явно: `type="submit"` или `type="button"`.

- `<select>` — выпадающий список.
- `<textarea>` — многострочное поле.

Правила семантики и доступности:

- Каждому полю — **id**, **name** и **связанный** `<label for="id">`.
- `label` — это **не** placeholder: плейсхолдер — лишь подсказка по формату, он не заменяет подпись.
- Группировку полей делайте через `<fieldset>` + `<legend>` (при необходимости).
- Для вспомогательного текста используйте `aria-describedby`, а ошибки помечайте `aria-invalid="true"`.

Полезные ссылки (RU):

- `<form>`: <https://developer.mozilla.org/ru/docs/Web/HTML/Element/form>
- `<input>`:
<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/input>
- `<label>`:
<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/label>
- `<select>/<option>`:
<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/select>
- `<textarea>`:
<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/textarea>

1.2. Мини-введение в CSS (строго минимум для этой практики)

Подключение файла стилей (во `<head>`):

```
<link rel="stylesheet" href="styles.css">
```

- Начинаем со стилей по **классам**, избегаем `#id` и `!important`.
- Видимые контуры фокуса делайте через `:focus-visible` (это базовая доступность).
- Состояния — аккуратно: `:hover`, `:active`, `:disabled`.

Полная теория по CSS

(каскад/наследование/специфичность/переменные/БЭМ) будет в Практике 4; тут — только минимум для читаемости.

1.3. Модальные окна — что это и когда применять

Модалка блокирует взаимодействие с остальной страницей и удерживает **фокус** внутри окна. Предпочтительно использовать нативный элемент **<dialog>**: он даёт модальный режим `showModal()` и фон-подложку `::backdrop` «из коробки». Альтернатива — «ручная» модалка (контейнер + подложка) с JS для переключения видимости и фокуса.

1.3.1. Вариант А — нативный <dialog> (рекомендуется)

HTML (кнопка + модалка с формой):

```
<!-- Кнопка открытия модалки -->
<button id="openDialog" type="button">Написать нам</button>

<!-- Модалка с формой -->
<dialog id="contactDialog" aria-labelledby="dialogTitle">
  <h2 id="dialogTitle">Обратная связь</h2>

  <form id="contactForm" method="dialog" novalidate>
    <label for="name">Имя</label>
    <input id="name" name="name" type="text" required minlength="2"
autocomplete="given-name">

    <label for="email">E-mail</label>
    <input id="email" name="email" type="email" required
autocomplete="email">

    <label for="phone">Телефон</label>
    <input id="phone" name="phone" type="tel" inputmode="tel"
autocomplete="tel"
      aria-describedby="phoneHelp" placeholder="+7 (____) ____-__-__"
      required pattern="^\+7 \(\d{3}\) \d{3}-\d{2}-\d{2}$">
    <small id="phoneHelp">Формат: +7 (900) 000-00-00</small>

    <label for="date">Дата</label>
    <input id="date" name="date" type="date">

    <label for="topic">Тема</label>
    <select id="topic" name="topic" required>
      <option value="">Выберите тему</option>
      <option>Вопрос</option>
      <option>Заказ</option>
    </select>

    <label for="message">Сообщение</label>
    <textarea id="message" name="message" rows="4" required></textarea>

    <div class="form-actions">
      <button type="button" id="closeDialog">Закрыть</button>
```

```

        <button type="submit">Отправить</button>
      </div>
    </form>
  </dialog>

```

CSS (минимум для понятности):

```

dialog { padding: 1.25rem; border: none; border-radius: 12px; }
dialog::backdrop { background: rgba(0,0,0,.5); }
.form-actions { display:flex; gap:.5rem; margin-top:1rem; }

```

JS (подключение см. 1.4.1):

```

const dlg = document.getElementById('contactDialog');
const openBtn = document.getElementById('openDialog');
const closeBtn = document.getElementById('closeDialog');
const form = document.getElementById('contactForm');
let lastActive = null;

openBtn.addEventListener('click', () => {
  lastActive = document.activeElement;
  dlg.showModal(); // модальный режим +
  затемнение
  dlg.querySelector('input,select,textarea,button')?.focus();
});

closeBtn.addEventListener('click', () => dlg.close('cancel'));

form?.addEventListener('submit', (e) => {
  // валидация см. 1.4.2; при успехе закрываем окно
});

dlg.addEventListener('close', () => { lastActive?.focus(); });
// Esc по умолчанию вызывает событие 'cancel' и закрывает <dialog>

```

Справка:

<dialog>:

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/dialog>

HTMLDialogElement:

<https://developer.mozilla.org/ru/docs/Web/API/HTMLDialogElement>

1.3.2. Вариант В — «ручная» модалка (без <dialog>)

HTML:

```

<button id="openModal" type="button">Написать нам</button>

<div class="modal" id="modal" hidden aria-modal="true" role="dialog" aria-
labelledby="mTitle">

```

```

<div class="modal__backdrop" data-close></div>
<div class="modal__window" role="document">
  <h2 id="mTitle">Обратная связь</h2>
  <form id="contactForm" novalidate>
    <!-- поля формы (как выше) -->
    <div class="form-actions">
      <button type="button" id="closeModal">Закрыть</button>
      <button type="submit">Отправить</button>
    </div>
  </form>
</div>
</div>

```

CSS (минимум):

```

.modal[hidden]{display:none;}
.modal{position:fixed; inset:0; display:grid; place-items:center; z-index:1000;}
.modal__backdrop{position:absolute; inset:0; background:rgba(0,0,0,.5);}
.modal__window{position:relative; z-index:1; background:#fff; border-radius:12px; padding:1.25rem; width:min(92vw,520px);}
.form-actions{display:flex; gap:.5rem; margin-top:1rem;}

```

JS:

```

const modal = document.getElementById('modal');
const open = document.getElementById('openModal');
const close = document.getElementById('closeModal');
let last = null;

function show(){ last=document.activeElement; modal.hidden=false;
modal.querySelector('input,select,textarea,button')?.focus(); }
function hide(){ modal.hidden=true; last?.focus(); }

open.addEventListener('click', show);
close.addEventListener('click', hide);
modal.addEventListener('click', (e)=>{ if (e.target.dataset.close !== undefined) hide(); });

```

В реальном проекте инлайн-стили переносите в styles.css.

1.4. Валидация на клиенте (без библиотек)

Подход: сначала — семантические типы (email, tel, date) и атрибуты (required, при необходимости pattern). Затем — минимальный JS для сообщений и подсветки ошибок.

1.4.1. Подключение JS

- Рекомендуемо в `<head>`:

```
<script src="main.js" defer></script>
```

- Либо перед `</body>`:

```
<script src="main.js"></script>
```

1.4.2. Базовый шаблон валидации (Constraint Validation API)

```
const form = document.getElementById('contactForm');

form?.addEventListener('submit', (e) => {
  // 1) Сброс кастомных сообщений
  [...form.elements].forEach(el => el.setCustomValidity?.(''));

  // 2) Проверка встроенных ограничений
  if (!form.checkValidity()) {
    e.preventDefault();

    // Пример: таргетированное сообщение
    const email = form.elements.email;
    if (email?.validity.typeMismatch) {
      email.setCustomValidity('Введите корректный e-mail, например name@example.com');
    }

    form.reportValidity(); // показать браузерные подсказки

    // Ally: подсветка проблемных полей
    [...form.elements].forEach(el => {
      if (el.willValidate) el.toggleAttribute('aria-invalid',
!el.checkValidity());
    });
    return;
  }

  // 3) Успешная «отправка» (без сервера)
  e.preventDefault();
  // Если форма внутри <dialog>, закрываем окно:
  document.getElementById('contactDialog')?.close('success');
  form.reset();
});
```

Полезно знать:

Constraint Validation API:

https://developer.mozilla.org/ru/docs/Web/HTML/Guides/Constraint_validation

Учебник по валидации:

https://developer.mozilla.org/ru/docs/Learn_web_development/Extensions/Forms/Form_validation

Мобильный UX:

`inputmode="tel|numeric"` — подсказывает виртуальную клавиатуру:

https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Global_attributes/inputmode

1.4.3. Лёгкая маска телефона (доп. задание)

```
const phone = document.getElementById('phone');
```

```
phone?.addEventListener('input', () => {  
  const digits = phone.value.replace(/\D/g, '').slice(0,11); // до 11 цифр  
  const d = digits.replace(/^8/, '7'); // нормализуем 8  
  → 7  
  const parts = [];  
  if (d.length > 0) parts.push('+7');  
  if (d.length > 1) parts.push(' (' + d.slice(1,4));  
  if (d.length >= 4) parts[parts.length - 1] += ')';  
  if (d.length >= 5) parts.push(' ' + d.slice(4,7));  
  if (d.length >= 8) parts.push('-' + d.slice(7,9));  
  if (d.length >= 10) parts.push('-' + d.slice(9,11));  
  phone.value = parts.join('');  
});
```

```
// Строгая проверка (если задаёте pattern из JS):
```

```
phone?.setAttribute('pattern', '^\\+7 \\(\\d{3}\\) \\d{3}-\\d{2}-\\d{2}$');
```

1.5. Источники для самостоятельного чтения (RU)

- **MDN — <form> и элементы форм (база):**

<https://developer.mozilla.org/ru/docs/Web/HTML/Element/form>

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/input>

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/label>

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/select>

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/textarea>

- **MDN — Валидация (Constraint Validation API):**

https://developer.mozilla.org/ru/docs/Web/HTML/Guides/Constraint_validation

https://developer.mozilla.org/ru/docs/Learn_web_development/Extensions/Forms/Form_validation

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Attributes/pattern>

- **MDN — inputmode (мобильная клавиатура):**

https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Global_attributes/inputmode

- **MDN — <dialog> и HTMLDialogElement (модалки):**

<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/dialog>

<https://developer.mozilla.org/ru/docs/Web/API/HTMLDialogElement>

- **MDN — :focus-visible (про фокус; пригодится уже сейчас):**

<https://developer.mozilla.org/ru/docs/Web/CSS/%3Afocus-visible>

1.6. Полезные ссылки

- **HTML Формы — MDN:**

<https://developer.mozilla.org/ru/docs/Web/HTML/Element/form>

- **Формы — HCDev:** <https://hcdev.ru/learn/html5/forms/>

2) Подключение CSS (обязательный минимум перед задачей)

1. Создайте styles.css в корне проекта (или /styles/styles.css).
2. Подключите во <head>:

```
<link rel="stylesheet" href="styles.css">
```

3. Вставьте стартовый набор (разберитесь по комментариям):

```
/* БАЗА: читаемость и предсказуемость размеров */  
* { box-sizing: border-box; }
```



```

html, body { margin: 0; padding: 0; }
body { font-family: system-ui, -apple-system, "Segoe UI", Roboto, Arial,
sans-serif; line-height: 1.5; }

/* Формы: размеры, отступы, визуальный порядок */
label { display: block; margin: 0.5rem 0 0.25rem; }
input, select, textarea, button {
  width: 100%;
  max-width: 480px;
  padding: .5rem .75rem;
  border: 1px solid #c8c8c8;
  border-radius: 8px;
  background: #fff;
}

/* Кнопка по умолчанию */
button, [type="submit"] {
  background: #0a84ff;
  color: #fff;
  border: none;
  cursor: pointer;
}

/* Состояния: клавиатура/мышь/ошибки */
:where(input, select, textarea, button):focus-visible {
  outline: 2px solid #0a84ff; outline-offset: 2px;
}
button:hover { filter: brightness(0.95); }
:disabled, [aria-disabled="true"] { opacity: .6; cursor: not-allowed; }

/* Подсветка ошибочных полей (установите aria-invalid="true") */
[aria-invalid="true"] { border-color: #b00020; }

```

Почему `:focus-visible`: он показывает контур **только** там, где это ожидаемо (клавиатура/доступность), не «мигает» при кликах мышью.

3) Пошаговая работа

Рекомендуемый вариант: форма расположена **внутри** модальки `<dialog>` (см. 1.3.1). Если решили оставить форму на странице — не дублируйте `id`, используйте **один** `id="contactForm"`.

Шаг 1. Реализуйте модальку `<dialog>` с формой (копируйте из 1.3.1).

Шаг 2. Подключите `styles.css` и `main.js` (см. 1.2 и 1.4.1).

Шаг 3. Добавьте минимальную валидацию (см. 1.4.2) и по желанию «лёгкую маску» телефона (1.4.3).

Шаг 4. Проверьте клавиатурную доступность: `Tab/Shift+Tab` по элементам, `Esc` закрывает окно.

Шаг 5. Коммит → PR dev → main → публикация демо.

4) Задания

БАЗА:

1. Сверстайте форму: email, phone, date, select, textarea + label к каждому полю.
2. Встроенная валидация + модальное окно + сообщение об успешной отправке (закрытие модалки).
3. Подключённый styles.css (минимум — раздел 2).

Дополнительно (любой пункт):

- Улучшите pattern и текстовые подсказки для телефона.
- Реализуйте отдельную страницу «Спасибо» (мок-переход).
- Расширьте ARIA-подсказки (aria-describedby для всех сообщений).
- **Проверка номера по «лёгкой маске»** — форматирование ввода по 1.4.3; при несоответствии формату — pattern и сообщение об ошибке.

5) Критерии оценивания (10 б)

- **Поля и семантика (4):** валидные типы, обязательные label, корректные name/id.
- **Валидация (3):** required/type/pattern, корректная работа reportValidity().
- **UX (3):** видимые фокус-контуров :focus-visible, понятные сообщения, модалька доступна с клавиатуры и закрывается корректно.

6) Требования к сдаче

- PR из dev в main, осмысленные сообщения коммитов.
- Ссылка на демо в README.
- Комментарии в коде на ключевых участках.
- Скриншот проверки валидатором W3C (HTML).

7) Контрольные вопросы

1. Чем label отличается от placeholder с точки зрения доступности?
2. Что делает form.checkValidity() и когда нужен form.reportValidity()?
3. Зачем inputmode у телефонного поля и чем он помогает на мобильных?

8) Частые ошибки

- Отсутствуют label/name; вся проверка только в JS; нет видимого фокуса.
- Дублирование id формы (например, на странице и в модалке одновременно).
- Модалка без фокус-ловушки: Tab «уходит» на фон; непонятно, как закрыть с клавиатуры.
- Агрессивная маска телефона, мешающая редактированию и вставке из буфера.