

Яндекс

Яндекс

<form> как таковые

рассказ про HTML формы

Артём Пендюрин, разработчик интерфейсов

<form>

Элемент HTML form (<form>) представляет секцию документа, содержащую интерактивные элементы управления, которые позволяют пользователю отправлять информацию на сервер.

```
<form
  method="GET"
  action="https://yandex.ru/search/"
  target="_blank"
  autocomplete="off"
  novalidate
></form>
```

Что внутри?

- › Поля `<input>`, `<select>`, `<textarea>`
- › Кнопки `<button>`, `<input type="submit">`
- › Разметка форм `<label>`, `<fieldset>`, `<legend>`
- › Остальная вёрстка [👉]

[👉] Кроме `<form>`

Поля

text	<input type="text"/>	color	<input type="color"/>
checkbox	<input type="checkbox"/>	file	<input type="file"/>
radio	<input type="radio"/>	date	<input type="date"/>
datetime	<input type="datetime"/>	datetime-local	<input type="datetime-local"/>
email	<input type="email"/>	month	<input type="month"/>
number	<input type="number"/>	password	<input type="password"/>
range	<input type="range"/>	search	<input type="search"/>
tel	<input type="tel"/>	time	<input type="time"/>
url	<input type="url"/>	week	<input type="week"/>
select	<input type="text" value="option 1"/>	textarea	<input type="text"/>

disabled

text	<input type="text"/>	color	<input type="color"/>
checkbox	<input type="checkbox"/>	file	<input type="file"/>
radio	<input type="radio"/>	date	<input type="date"/>
datetime	<input type="datetime"/>	datetime-local	<input type="datetime-local"/>
email	<input type="email"/>	month	<input type="month"/>
number	<input type="number"/>	password	<input type="password"/>
range	<input type="range"/>	search	<input type="search"/>
tel	<input type="tel"/>	time	<input type="time"/>
url	<input type="url"/>	week	<input type="week"/>
select	<input type="select"/>	textarea	<input type="text"/>

<label>

Нужен для связи поля с его подписью

```
<label>  
    Имя <input name="name">  
</label>
```

```
<label for="name">Имя</label>  
<input name="name" id="name">
```

<fieldset> и <legend>

- › <fieldset> — для группировки полей
- › <legend> — для заголовка

```
<fieldset>
  <legend>ФИО</legend>

  <input type="text" name="first_name">
  <input type="text" name="middle_name">
  <input type="text" name="last_name">
</fieldset>
```


Submit



Как сохранить данные формы?

- › Вызвать метод `HTMLFormElement.submit()`
- › Нажать на `<input type="submit">` внутри формы
- › Нажать на `<button type="submit">` внутри формы
- › Нажать Enter в поле формы, у которой есть кнопка submit

Что происходит по submit?

- › Валидация [👉]
- › Событие `submit` [👉]
- › Useragent выполняет запрос базирясь на `action` и `method`
- › Результат отобразится там, куда указывает `target`

[👉] *Если действие инициировал пользователь*

Простой пример

Сверстаем форму, которая сабмитится в iframe.



erblack.me/html-form/types/



```
<form action="/get" method="GET" target="form-result">
```

```
  <!-- Поля формы -->
```

```
    <button type="submit">Сохранить</button>
```

```
</form>
```

```
<iframe src="/get" name="form-result"></iframe>
```

Пример посложнее



erblack.me/html-form/colors/



RGB

R

G

B

CMYK

C

M

Y

K

HSL

H

S

L



Reset



Как вызвать reset?

- › Вызвать метод `HTMLFormElement.reset()`
- › Нажать на `<input type="reset">` внутри формы
- › Нажать на `<button type="reset">` внутри формы

Как работает reset?

Reset сбрасывает значения полей формы до значения по умолчанию

И что это за значения?

Значения, которые были у полей при отрисовке страницы

`HTMLInputElement.defaultValue` // За исключением radio, checkbox и file

`HTMLInputElement.defaultChecked` // Для radio и checkbox

`HTMLTextAreaElement.defaultValue`

`HTMLOptionElement.defaultSelected`

Проверка наличия изменений

Напишем функцию, которая будет определять есть ли изменения в элементах формы

```
function isChanged(element) {  
    switch (element.nodeName) {  
        default:  
            return false;  
    }  
}
```

TEXTAREA, SELECT

```
case 'TEXTAREA':  
    return element.value !== element.defaultValue;  
  
case 'SELECT':  
    return Array.from(element.options)  
        .some(  
            (option) => option.selected !== option.defaultSelected  
        );
```

INPUT

```
case 'INPUT':  
    switch (element.type) {  
        case 'reset':  
        case 'submit':  
            return false;  
        case 'checkbox':  
        case 'radio':  
            return element.checked !== element.defaultChecked;  
        default:  
            return element.value !== element.defaultValue;  
    }
```

FORM, FIELDSET

Для формы и филдсета будем проверять, есть ли изменения в полях внутри них

```
case 'FORM':  
case 'FIELDSET':  
    return Array.from(element.elements).some( isChanged );
```

Пример проверки изменений



 erblack.me/html-form/changed/

Валидация



Что умеет HTML валидация?

- › Не позволять неправильный ввод
- › Показывать ошибки при неправильном вводе
- › Не даёт сохранить форму

Чего не умеет HTML валидация?

Гарантировать валидность [👉]

Валидируйте на сервере, даже если валидируете на клиенте

Как и любая клиентская валидация [👉]

required


Обязательное

! Вы пропустили это поле.

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

minlength, maxlength

От 5 до 10
СИМВОЛОВ

 Минимальное количество символов: 5. Длина текста сейчас: 4.

```
<input type="text" name="length" minlength="5" maxlength="10">
```

min, max

От 5 до 10

! Значение должно быть больше или равно 5.

```
<input type="number" name="number" min="5" max="10">
```

pattern

\D



Введите данные в указанном формате.
Никаких цифр

```
<input type="text" name="pattern" pattern="\D" title="Никаких цифр">
```

По типу поля

Почта



Адрес электронной почты должен содержать символ «@». Похоже, вы его пропустили: erblacklive.ru.

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

Кастомная валидация

Делится на 7



8 на 7 не делится

```
function customValidation({target}) {  
    if (Number(target.value) % 7 !== 0) {  
        target.setCustomValidity(`${target.value} на 7 не делится`);  
    } else {  
        target.setCustomValidity('');  
    }  
}
```

Validity State

```
HTMLInputElement.validity → ValidityState {  
  badInput: false,  
  patternMismatch: false,  
  rangeOverflow: false,  
  rangeUnderflow: false,  
  stepMismatch: false,  
  tooLong: false,  
  tooShort: false,  
  typeMismatch: false,  
  valueMissing: false,  
  customError: false,  
  valid: true  
}
```


Пример html валидации



 erblack.me/html-form/changed/


Что в итоге?



Выводы

- › html формы и поля умеют всякое
- › Все поля в формах помнят свои дефолтные значения
- › html валидация гибкая, удобная, но ничего не гарантирует
- › Надеюсь вы узнали сегодня что-то новое

Все примеры

 github.com/ErBlack/html-form

 erblack.me/html-form



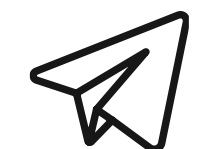
Спасибо за внимание

Артём Пендюрин

разработчик интерфейсов



ertema@yandex-team.ru



@ErBlack



@ErBlack



erblack.me

Вопросы