



**ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКИЙ МОДУЛЬ
БФУ имени И. Канта**

ИСПОЛНЯЕМЫЕ ТЭГИ. ПСЕВДОКЛАССЫ

ДОПОЛНЕНИЕ

`<form>...</form>` устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению. Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Синтаксис

```
<form action="URL">...</form>
```



Атрибуты

accept-charset устанавливает кодировку, в которой сервер может принимать и обрабатывать данные.

action адрес программы или документа, который обрабатывает данные формы.

autocomplete включает автозаполнение полей формы.

enctype способ кодирования данных формы.

method метод протокола HTTP.

name имя формы.

novalidate отменяет встроенную проверку данных формы на корректность ввода.

target имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.



ИСПОЛНЯЕМЫЕ ТЭГИ

`<a>...` зависит от присутствия атрибутов `name` или `href`, тем самым устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Синтаксис

```
<a href="URL">...</a>
```

```
<a name="идентификатор">...</a>
```



Атрибуты

accesskey активация ссылки с помощью комбинации клавиш.

coords устанавливает координаты активной области.

download предлагает скачать указанный по ссылке файл.

href задает адрес документа, на который следует перейти.

hreflang идентифицирует язык текста по ссылке.

name устанавливает имя якоря внутри документа.

rel отношения между ссылаемым и текущим документами.

rev отношения между текущим и ссылаемым документами.

shape задает форму активной области ссылки для изображений.

tabindex определяет последовательность перехода между ссылками при нажатии на кнопку Tab.

target имя окна или фрейма, куда браузер будет загружать документ.

title добавляет всплывающую подсказку к тексту ссылки.

type указывает MIME-тип документа, на который ведёт ссылка.



`<input type=“...”>` является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем. Главным образом `<input>` предназначен для создания текстовых полей, различных кнопок, переключателей и флажков. Хотя элемент `<input>` не требуется помещать внутрь контейнера `<form>`, определяющего форму, но если введенные пользователем данные должны быть отправлены на сервер, где их обрабатывает серверная программа, то указывать `<form>` обязательно.

Основной атрибут тега `<input>`, определяющий вид элемента — `type`. Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики. Кроме того, в HTML5 добавлено еще более десятка новых элементов.

Синтаксис

`<input атрибуты>`

Тип	Описание	Вид
button	Кнопка.	<input type="button" value="Кнопка"/>
checkbox	Флажки. Позволяют выбрать более одного варианта из предложенных.	<input type="checkbox"/> Пиво <input type="checkbox"/> Чай <input type="checkbox"/> Кофе
file	Поле для ввода имени файла, который пересылается на сервер.	<input type="file" value="Выберите файл"/> Файл не выбран
hidden	Скрытое поле. Оно никак не отображается на веб-странице.	
image	Поле с изображением. При нажатии на рисунок данные формы отправляются на сервер.	<input alt="Отправить" type="image"/>
password	Обычное текстовое поле, но отличается от него тем, что все символы показываются звездочками. Предназначено для того, чтобы никто не подглядел вводимый пароль.	<input type="password"/>
radio	Переключатели. Используются, когда следует выбрать один вариант из нескольких предложенных.	<input type="radio"/> Пиво <input type="radio"/> Чай <input type="radio"/> Кофе
reset	Кнопка для возвращения данных формы в первоначальное значение.	<input type="reset" value="Сбросить"/>
submit	Кнопка для отправки данных формы на сервер.	<input type="submit" value="Отправить"/>
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.	<input type="text"/>

Тип	Описание	Вид
color	Виджет для выбора цвета.	
date	Поле для выбора календарной даты.	<input type="text" value="дд.мм.гггг"/>
datetime	Указание даты и времени.	<input type="text"/>
datetime-local	Указание местной даты и времени.	<input type="text" value="дд.мм.гггг --:--"/>
email	Для адресов электронной почты.	<input type="text"/>
number	Ввод чисел.	<input type="text" value=""/>
range	Ползунок для выбора чисел в указанном диапазоне.	
search	Поле для поиска.	<input type="text"/>
tel	Для телефонных номеров.	<input type="text"/>
time	Для времени.	<input type="text" value="--:--"/>
url	Для веб-адресов.	<input type="text"/>
month	Выбор месяца.	<input type="text" value="-----"/>
week	Выбор недели.	<input type="text" value="Неделя --, ----"/>

Атрибуты

accept устанавливает фильтр на типы файлов, которые вы можете отправить через поле загрузки файлов.

accesskey переход к элементу с помощью комбинации клавиш.

align определяет выравнивание изображения.

alt альтернативный текст для кнопки с изображением.

autocomplete включает или отключает автозаполнение.

autofocus устанавливает фокус в поле формы.

border толщина рамки вокруг изображения.

checked предварительно активированный переключатель или флажок.

disabled блокирует доступ и изменение элемента.

form связывает поле с формой по её идентификатору.

formaction определяет адрес обработчика формы.

formenctype устанавливает способ кодирования данных формы при их отправке на сервер.

formmethod сообщает браузеру каким методом следует передавать данные формы на сервер.

formnovalidate отменяет встроенную проверку данных на корректность.

formtarget определяет окно или фрейм в которое будет загружаться результат, возвращаемый обработчиком формы.

list указывает на список вариантов, которые можно выбирать при вводе текста.

max верхнее значение для ввода числа или даты.

maxlength максимальное количество символов разрешенных в тексте.

min нижнее значение для ввода числа или даты.

multiple позволяет загрузить несколько файлов одновременно.

name имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.

pattern устанавливает шаблон ввода.

placeholder выводит подсказывающий текст.

readonly устанавливает, что поле не может изменяться пользователем.

required обязательное для заполнения поле.

size ширина текстового поля.

src адрес графического файла для поля с изображением.

step шаг приращения для числовых полей.

tabindex определяет порядок перехода между элементами с помощью клавиши Tab.

type сообщает браузеру, к какому типу относится элемент формы.

value значение элемента.



`<select><option>Пункт 1</option></select>` позволяет создать элемент интерфейса в виде раскрывающегося списка, а также список с одним или множественным выбором, как показано далее. Конечный вид зависит от использования атрибута `size` тега `<select>`, который устанавливает высоту списка. 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>`. Если планируется отправлять данные списка на сервер, то требуется поместить элемент `<select>` внутрь формы. Это также необходимо, когда к данным списка идет обращение через скрипты.

Синтаксис

```
<select>  
  <option>Пункт 1</option>  
  <option>Пункт 2</option>  
</select>
```

Атрибуты

accesskey позволяет перейти к списку с помощью некоторого сочетания клавиш.

autofocus устанавливает, что список получает фокус после загрузки страницы.

disabled блокирует доступ и изменение элемента.

form связывает список с формой.

multiple позволяет одновременно выбирать сразу несколько элементов списка.

name имя элемента для отправки на сервер или обращения через скрипты.

required список обязателен для выбора перед отправкой формы.

size количество отображаемых строк списка.

tabindex определяет последовательность перехода между элементами при нажатии на клавишу Tab.

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

Синтаксис

```
<textarea атрибуты>
```

```
    текст
```

```
</textarea>
```

Атрибуты

accesskey переход к полю с помощью сочетания клавиш.

autofocus автоматическое получение фокуса.

cols ширина поля в символах.

disabled блокирует доступ и изменение элемента.

form связывает текстовое поле с формой по её идентификатору.

maxlength максимальное число введенных символов.

name имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.

placeholder указывает замещающийся текст.

readonly устанавливает, что поле не может изменяться пользователем.

required обязательное для заполнения поле.

rows высота поля в строках текста.

tabindex порядок перехода между элементами при нажатии на клавишу Tab.

wrap параметры переноса строк.



`<canvas id="идентификатор"></canvas>` Создает область, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства. При помощи тега `<canvas>` можно создавать рисунки, анимацию, игры и др. (HTML5).

Синтаксис

```
<canvas id="идентификатор"></canvas>
```

Атрибуты

height задает высоту холста. По умолчанию 300 пикселей.

width задает ширину холста. По умолчанию 150 пикселей.

ПСЕВДОКЛАССЫ

Псевдокласс в CSS — это ключевое слово, добавленное к селектору, которое определяет его особое состояние

```
Селектор::псевдокласс {  
    свойство: значение;  
    -----  
    свойство: значение;  
}
```

```
Селектор дополнительные теги::псевдокласс {  
    свойство: значение;  
    -----  
    свойство: значение;  
}
```

:invalid применяется к полям формы, содержимое которых не соответствует указанному типу. Например, для `type="number"` должно вводиться число, а не буквы, для `type="email"` корректный адрес электронной почты.

Синтаксис

```
input:invalid { ... }
```

:valid применяется к полям формы, содержимое которых проходит проверку в браузере на соответствие указанному типу. Например, для `type="number"` вводится число, а не буквы, для `type="email"` корректный адрес электронной почты.

Синтаксис

```
input:valid { ... }
```

:link применяется к ссылкам, которые еще не посещались пользователем, и задает для них стилевое оформление.

Синтаксис

```
a:link { ... }
```

:active определяет стиль для активной ссылки. Активной ссылка становится при нажатии на нее.

Синтаксис

```
a:active { ... }
```

:visited применяется к ссылкам, уже посещённым пользователем, и задаёт для них стилевое оформление.

Синтаксис

```
a:visited { ... }
```

:disabled используется для применения стиля к заблокированным элементам форм. Такие элементы не могут получить фокус, быть нажатыми или активированы, в текстовых полях нельзя набирать текст.

Синтаксис

```
селектор(тег):disabled { ... }
```

:enabled используется для применения стиля к доступным (не заблокированным) элементам форм. По умолчанию, все элементы форм являются доступными.

Синтаксис

```
селектор(тег):enabled { ... }
```

:checked применяется к элементам интерфейса, таким как переключатели (checkbox) и флажки (radio), когда они находятся в положение «включено». Переключение элементов в такое состояние происходит с помощью атрибута checked тега <input> или пользователем.

Синтаксис

```
checkbox(radio):checked { ... }
```

:lang(<язык>) Определяет язык, который используется в документе или его фрагменте. С помощью псевдокласса :lang можно задавать определённые настройки, характерные для разных языков, например, вид кавычек в цитатах.

Синтаксис

```
селектор(тег):lang(<язык>) { ... }
```

Значения

В качестве языка могут выступать следующие значения: ru — русский; en — английский ; de — немецкий; fr — французский; it — итальянский и др.



:focus определяет стиль для элемента получающего фокус. Например, им может быть текстовое поле формы, в которое устанавливается курсор.

Синтаксис

```
селектор(тег):focus { ... }
```

:hover определяет стиль элемента при наведении на него курсора мыши, но при этом элемент еще не активирован, иными словами кнопка мыши не нажата.

Синтаксис

```
селектор(тег):hover { ... }
```

:first-child и **:last-child** псевдоклассы связаны с иерархией в HTML. Они нацеливаются на элементы HTML в зависимости от порядка, в котором они появляются в коде.

Синтаксис

```
селектор(тег):first-child { ... }
```

```
селектор(тег):last-child { ... }
```

:nth-child данный псевдокласс является глобальной версией **:first-child** и **:last-child**. С помощью **:nth-child** вы можете вычислить конкретный целевой тег.

Синтаксис

```
селектор(тег):nth-child(значение) { ... }
```

Значения: число тега, **odd** (нечетный), **even** (четный),

число * **n** + число (умножения на n + смещение, где n = от 0 до max количество тегов).

:nth-last-child используется для добавления стиля к элементам на основе нумерации в дереве элементов. В отличие от псевдокласса **:nth-child** отсчёт ведётся не от первого элемента, а от последнего.

Синтаксис

```
селектор(тег):nth-last-child(значение) { ... }
```

:not задаёт правила стилей для элементов, которые не содержат указанный селектор.

Синтаксис

```
селектор(тег):not(селектор(тег)) { ... }
```


:required применяет стилевые правила к тегу `<input>`, у которого установлен атрибут `required`. Он позволяет выделять поля обязательные к заполнению перед отправкой формы.

Синтаксис

```
input:required { ... }
```

:optional применяет стилевые правила к тегу `<input>`, у которого не задан атрибут `required`. Он позволяет выделять поля необязательные к заполнению перед отправкой формы. Таким образом `:optional` применяется к необязательным полям формы.

Синтаксис

```
input:optional { ... }
```

:empty представляет пустые элементы, иными словами такие, которые не содержат дочерних элементов, текста, пробелов или переносов строк. К примеру, `<p></p>` является пустым элементом, а `<p> </p>`, `<p> </p>` или `<p>эге</p>` уже нет.

Синтаксис

```
input:empty { ... }
```

:blank представляет пустые элементы, иными словами такие, которые не содержат дочерних элементов или текста. По своему действию похож на `:empty`, но `:blank`, в отличие от него, считает пустыми элементы, содержащие пробел или перенос строк.

Синтаксис

```
input:blank { ... }
```

:target для перехода к выбранному фрагменту документа, в адресе пишется символ # и указывается имя идентификатора. К примеру, в адресе <http://www.w3.org/TR/css3-selectors/#target-pseudo> происходит переход к тегу, атрибут id которого задан как target-pseudo. Такая запись адреса называется «целевой элемент». Псевдокласс :target применяется к целевому тегу, иными словами, к идентификатору, который указан в адресной строке браузера.

Синтаксис

```
селектор(тег):target { ... }
```



ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКИЙ МОДУЛЬ
БФУ имени И. Канта

Спасибо за внимание!