

HTML формы

Данильченко Анна Александровна

Преподаватель кафедры программного
обеспечения систем ЖГТУ

Форма заявки на участие в конференции

Имя	<input type="text"/>	
Отчество	<input type="text"/>	
Фамилия	<input type="text"/>	
Контактный телефон	<input type="text"/>	
Пол	<input type="radio"/> М <input type="radio"/> Ж	
Секции, в которых примите участие	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4	
Город	Москва <input type="button" value="v"/>	
Тема доклада и аннотация		
Тема:		<input type="text"/>
Вы можете отправить доклад		<input type="text"/>
		<input type="button" value="Обзор..."/>
<input type="button" value="Послать"/> <input type="button" value="Сброс"/>		

Форма – это набор элементов, таких как поля ввода, поля выбора, переключатели. Форма позволяет создавать интерактивные страницы, т.е. организовывать диалог с пользователем.

Форма создается с помощью тега:

<FORM>

</FORM>

HTML Форма

Позволяет пользователю вводить информацию для последующей ее обработки. Задается с помощью элемента `<form>`

```
01. <form action="http://example.com/login" method="post">  
02.     <input name="login" type="text"/>  
03.     <input name="password" type="password"/>  
04.     <input value="Login" type="submit"/>  
05. </form>
```

Атрибуты формы

method - сообщает серверу о методе запроса. Может быть "GET" и "POST"

action - url для обработки данных формы. Может быть абсолютным или относительным

enctype - способ кодирования данных формы. По умолчанию значени атрибута "application/x-www-form-urlencoded". Для отправки файлов значени должно быть "multipart/form-data"

GET - POST

Главное отличие методов POST и GET заключается в способе передачи информации. В методе GET параметры передаются через адресную строку, т.е. по сути в HTTP-заголовке запроса, в то время как в методе POST параметры передаются через тело HTTP-запроса и никак не отражаются на виде адресной строки.

<http://www.htmlbook.ru/cgi-bin/handler.cgi?nick=%C2%E0%ED%FF+%D8%E0%EF%EE%F7%EA%E8%ED&page=5>

Атрибуты формы

accept-charset - Устанавливает кодировку, в которой сервер может принимать и обрабатывать данные формы

```
<form action="handler.php" accept-charset="windows-1251">
```

Autocomplete - Включает автозаполнение полей формы.

Name - Имя формы.

Novalidate - Отменяет встроенную проверку данных формы на корректность ввода.

Target - Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.

Атрибут type

Таблица 1. Атрибут type

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
color	Виджет для выбора цвета.
date	Поле для выбора календарной даты.
datetime	Указание даты и времени.
datetime-local	Указание местной даты и времени.
email	Для адресов электронной почты.
number	Ввод чисел.

range	Ползунок для выбора чисел в указанном диапазоне.
search	Поле для поиска.
tel	Для телефонных номеров.
time	Для времени.
url	Для веб-адресов.
month	Выбор месяца.
week	Выбор недели.

Поддержка этих значений браузерами

Значение	 Internet Explorer	 Chrome	 Opera	 Safari	 Firefox	 Android	 iOS
color	✗	✓ 21.0+	✓ 11.01+	✗	✗	✗	✗
date	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+
datetime	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+
datetime-local	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+
email	✓ 10.0	✓ 5.0+	✓ 10.62+	✓ 5.0+	✓ 4.0+	✓ 2.3+	✓ 3.1+
month	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+
number	✓ 10.0	✓ 6.0+	✓ 10.62+	✓ 5.0+	✗	✓ 2.3+	✓ 4.0+
range	✓ 10.0	✓ 5.0+	✓ 10.62+	✓ 5.0+	✓ 23.0+	✓	✓ 5.0+
search	✓ 10.0	✓ 5.0+	✓ 11.01+	✓ 5.0+	✓ 4.0+	✓	✓ 4.0+
tel	✓ 10.0	✓ 5.0+	✓ 11.01+	✓ 5.0+	✓ 4.0+	✓	✓ 3.1+
time	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+
url	✓ 10.0	✓ 5.0+	✓ 10.62+	✓ 5.0+	✓ 4.0+	✓ 2.3+	✓ 3.1+
week	✗	✓ 5.0+	✓ 10.62+	✓ 5.0+	✗	✗	✓ 5.0+

Элемент `<input>`

Самый разносторонний элемент. Позволяет создавать разные элементы формы

Основные

атрибуты: **name, value, type, readonly, disabled, placeholder**

Элементы формы

1. **Текстовые поля** - для ввода текстовой информации

```
<input type=text name=имя_параметра [value=значение]  
[size=размер_поля] [maxlength=длина_поля]>
```

maxlength - максимально допустимой длиной текста

size - количество знакомест

Value - в поле будет изначально отображаться значение данного атрибута

`<input>` text

Текстовое поле

Ø1. `<input type="text" name="name" value="Peter"/>`

Ø2. `<input type="text" name="surname" value="Jackson"/>`



Peter Jackson

`<input>` password

Поле с паролем

```
<input type="password"/>
```

Password

.....

`<input>` radio

Переключатель

Ø1. `<input type="radio" name="iso_code" value="uk"/>`

Ø2. `<input type="radio" name="iso_code" value="en"/>`

Ø3. `<input type="radio" name="iso_code" value="de"/>`

UK EN DE

`<input>` checkbox

Флажок

Ø1. `<input type="checkbox" name="ppt" value="ppt" />`

Ø2. `<input type="checkbox" name="pdf" value="pdf" />`

Ø3. `<input type="checkbox" name="txt" value="txt" />`

PPT PDF TXT

`<input>` hidden

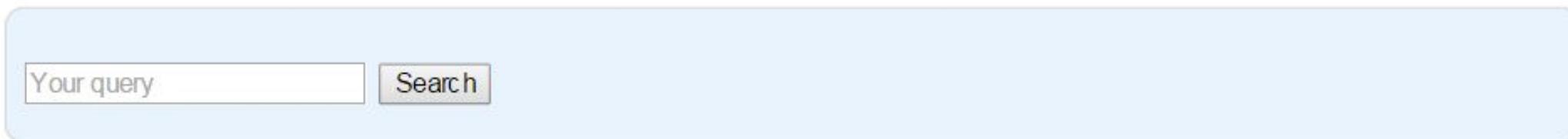
Скрытое поле

```
<input type="hidden" value="User_Identifier"/>
```

`<input>` submit

Кнопка для отправки формы

```
<input type="submit" value="Search"/>
```

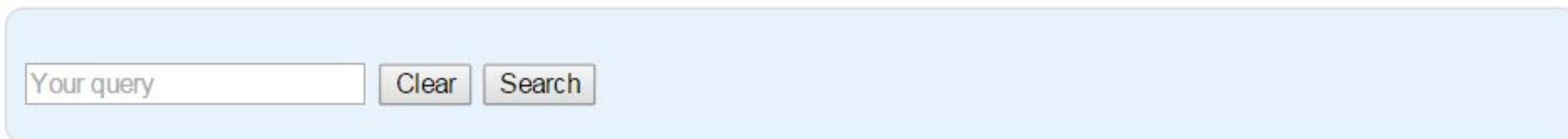


Your query

```
<input> reset
```

Кнопка для очистки формы

```
<input type="reset" value="Clear"/>
```

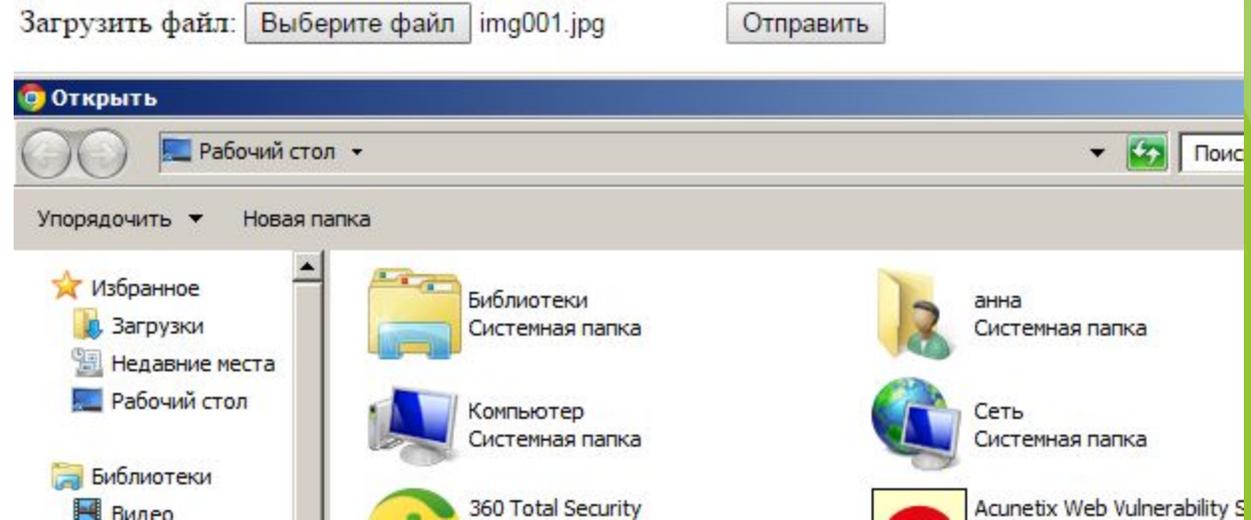


Your query

`<input type="file">`

Поле для отправки файла. Атрибут enctype формы должен иметь значение multipart/form-data

```
<form enctype="multipart/form-data" action="upload.php" method="post">  
  Загрузить файл: <input name="my_file" type="file">  
  <input type="submit" value="Отправить">  
</form>
```



`<input>` image

Кнопка с изображением

```
<input type="image" src="path/to/image"/>
```

HTML



Элемент `<select>`

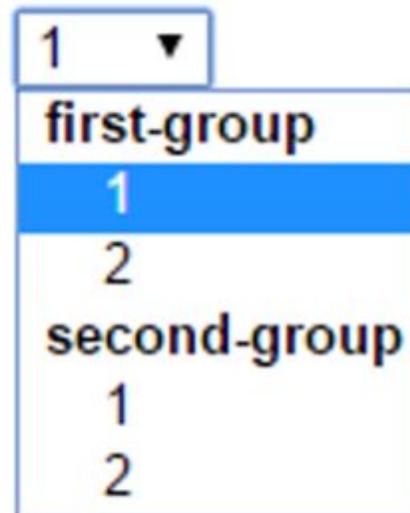
Раскрывающийся список, список с множественным выбором

```
<select name="students" multiple>
  <option value="Tom">Tom</option>
  <option value="Michel">Michel</option>
  <option value="Helen" selected>Helen</option>
</select>
```



Можна групувати елементи у випадаючому списку:

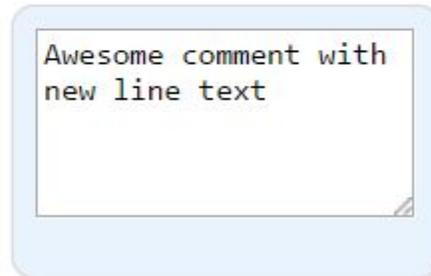
```
<select name="groups">  
  <optgroup label="first-group">  
    <option value="one">1</option>  
    <option value="two">2</option>  
  </optgroup>  
  <optgroup label="second-group">  
    <option value="one">1</option>  
    <option value="two">2</option>  
  </optgroup>  
</select>
```



Элемент `<textarea>`

Поле для ввода многострочного текста

```
. <textarea cols="30" rows="5">  
.     Awesome comment  
.     with new line text  
. </textarea/>
```



Теги формы: `<label>`

Задаёт связь между элементом формы и его меткой (текстом-подписью)

```
<label for="city"> City </label>
```

```
<input id="city" name="city" type="text" value="New York"/>
```

City

Теги формы: `<fieldset>`

Предназначен для группирования элементов формы

```
<fieldset>  
    // group of elements  
</fieldset>
```



A visual representation of a form fieldset. It is a light blue rounded rectangle containing three input elements: a text input labeled "Name", a text input labeled "Surname", and a text area labeled "Some text".

```
<fieldset>
  <legend>Group</legend>
  Login: <input type="text" name="login" />
  Passwors: <input type="password"
              name="password" />
</fieldset>
```

Group	
Login: <input type="text"/>	Passwors: <input type="password"/>

Тег <datalist>

Создает список вариантов, которые можно выбирать при наборе в текстовом поле. Изначально этот список скрыт и становится доступным при получении полем фокуса или наборе текста.

```
<input list="<идентификатор>">
<datalist id="<идентификатор>">
  <option value="Текст1">
  <option value="Текст2">
</datalist>
```

```
<p>Выберите любимого персонажа:</p>
<p><input list="character">
  <datalist id="character">
    <option value="Чебурашка"></option>
    <option value="Крокодил Гена"></option>
    <option value="Шапокляк"></option>
  </datalist></p>
```

Выберите любимого персонажа:

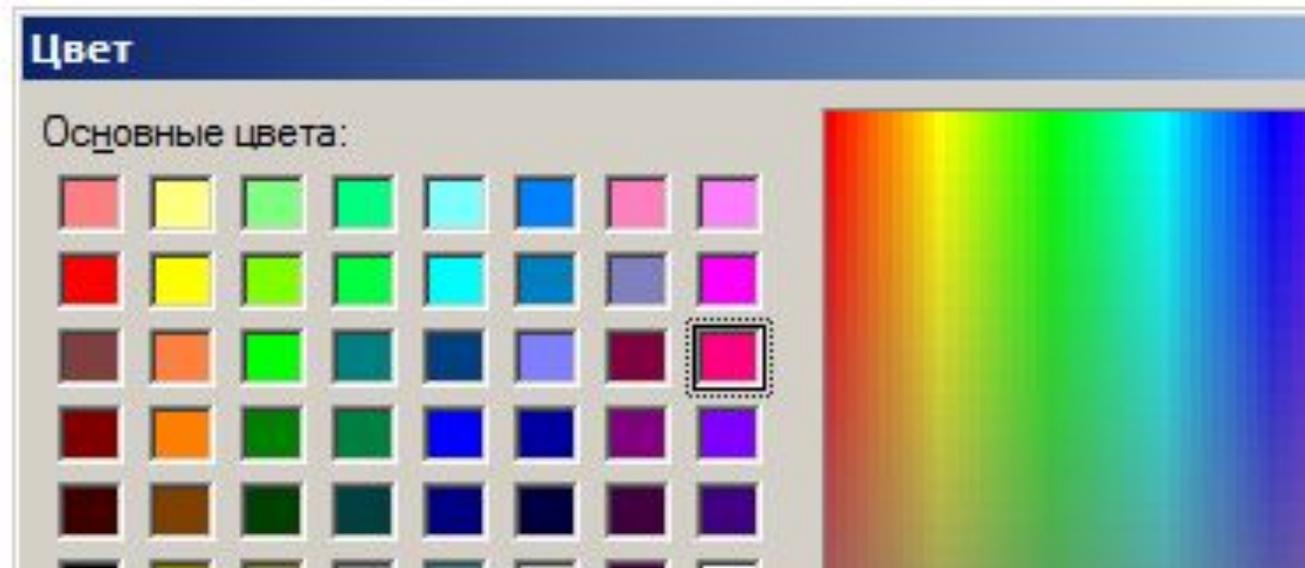
Че
Чебурашка
Крокодил Гена
Шапокляк

color

Позволяет выбрать цвет из палитры цветов.

<p>Выберите цвет <input type="color" name="selectcolor"></p>

Выберите цвет



date

Выбор даты в календаре.

<p>Выберите дату <input type="date" name="selectdate" ></p>

Выберите дату

Июнь 2015

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

datetime-local

Выбор даты и с локальным временем в календаре.

<p>Выберите дату <input type="datetime-local" name="selectdatetimelocal" ></p>

Выберите дату

Июнь 2015

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

month

Выбор месяца и года в календаре

<p>Выберите месяц <input type="month" name="selectmonth" ></p>

Выберите месяц

Июнь 2015 ▾

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

email

Текстовое поле для ввода адресов электронной почты. По умолчанию элемент принимает только один адрес, но указав булев атрибут `multiple="multiple"` можно разрешить пользователю ввод нескольких адресов через запятую.

```
<input type="email" placeholder="Ваша почта"/>
```



Ваша почта

number

Поле числового ввода. Визуально похоже на текстовое, но с добавлением кнопок-стрелок, позволяющих увеличивать и уменьшать значение.

```
<input type="number" step=2 />
```



Добавлен тег текстового уровня `<output></output>` для выведения результатов

```
<form oninput="result.value = parseInt(a.value)
                + parseInt(b.value)">
  <input type="number" id="a">
  +
  <input type="number" id="b">
  =
  <output name="result"></output>
</form>
```

+ = 10

range

определяет поле, которые может содержать значения в определенном интервале. Отображается как ползунок, который можно перетаскивать мышкой. Вы можете ограничивать диапазон принимаемых чисел с помощью атрибутов `min` (минимальное допустимое число) и `max` (максимальное допустимое число).



```
<input name='range' type='range' min='1' max='5' />
```

Атрибуты input

1. **Autofocus** - Автоматически устанавливает фокус в поле формы. В таком поле можно сразу набирать текст без явного щелчка по нему курсором мыши.
2. **Disabled** - Блокирует доступ и изменение поля формы. Оно в таком случае отображается серым и недоступным для активации пользователем. Заблокированное в поле значение не передается на сервер.
3. **Placeholder** - Выводит подсказывающий текст.
4. **Readonly** - Устанавливает, что поле не может изменяться пользователем.
5. **Step** - Шаг приращения для числовых полей.
6. **Value** - Значение элемента.
7. **Multiple** - указывает, что данное поле может принимать несколько значений одновременно

Валидация форм: `required`, `pattern`

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

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

```
<input type="url" name="url" required/>
```

```
<input type="text" name="price" pattern="\d+(\.\d{2})?" />
```

Required - указывает, что данное поле должно быть обязательно заполнено перед отправкой.

Pattern - Указывает регулярное выражение, согласно которому требуется вводить и проверять данные в поле формы. Если присутствует атрибут `pattern`, то форма не будет отправляться, пока поле не будет заполнено правильно.

CSS+формы

▶ Псевдокласс :focus

- Псевдокласс :focus определяет стиль для элемента получающего фокус. Например, им может быть текстовое поле формы, в которое устанавливается курсор. Такими элементами могут быть теги `<a>` `<input>` `<select>` и `<textarea>`

HTML

```
<span class="field">
```

```
<input type="password" name="pass" class="form-text">
```

```
</span>
```

CSS

```
.form-text:focus {  
background: #ffe; /* Цвет фона */ border: 1px solid  
#29B0D9; /* Параметры рамки */  
}
```

Псевдокласс :required

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

▶ HTML

```
<p>
```

```
<label for="email">E-mail</label>
```

```
<input id="email" type="text" required>
```

```
</p>
```

CSS

```
input:required {
```

```
    border: 1px solid red; /* Красная рамка для обязательных полей */
```

```
}
```

Псевдокласс :optional

- ▶ Применяет стилевые правила к тегу `<input>`, у которого не задан атрибут `required`. Он позволяет выделять поля обязательные к заполнению перед отправкой формы. Таким образом `:optional` применяется к необязательным полям формы.
- ▶ `<p>Имя: <input name="name"></p>`
- ▶ `<p>Возраст: <input type="number" min="18" required name="old"></p>`
`<p><input type="submit" value="Отправить"></p>`
- ▶ `<style>`
- ▶ `input:optional { opacity: 0.5; /* Полупрозрачность */ }`
- ▶ `</style>`

Псевдокласс :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. Проверяет на соответствие регулярному выражению в атрибуте `pattern`
- ▶ **HTML**
- ▶ `<p><input type="email" required></p>`
- ▶ **CSS**
- ▶ `input:valid { background: #dfd; /* Зеленый цвет фона */ }`

Псевдокласс :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.
- ▶ HTML
- ▶ `<p><input type="email" required></p>`
- ▶ CSS
- ▶ `input:invalid { background: #fdd; /* Красный цвет фона */ }`

Язык текста.

- ▶ Псевдокласс `lang` определяет язык текста того или иного элемента или документа в целом.
 - `ru` - Русский
 - `en` - Английский
 - `de` - Немецкий
 - `fr` - Французский
 - `it` - Итальянский

HTML

```
<p lang="ru">Русский текст выделен синим</p>
```

```
<p lang="en">English text is chosen red</p>
```

CSS

```
p:lang(ru) {color: #00f;}
```

```
p:lang(en) {color: #f00;}
```

Псевдокласс :checked

- ▶ Псевдокласс :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.

```
.checkbox:checked  
{  
  box-shadow: 0 0 0 3px rgba(255,255,0,.5);  
}
```

- ▶ Псевдокласс :indeterminate
- ▶ Псевдокласс :indeterminate задает стиль для элементов форм, таким как флажки и переключатели, когда они находятся в неопределенном состоянии. К примеру, если из группы флажков ни один не помечен, то флажки находятся в указанном состоянии.

Регулярные выражения

Это шаблон (строка), которая состоит из символов и метасимволов (спецсимволов) и который применяется для манипуляции над строками: поиск, замена, валидация

```
var reg = /ab+c/;
```

<https://regex101.com>

Якоря

^	Начало строки +
\A	Начало текста +
\$	Конец строки +
\Z	Конец текста +
\b	Граница слова +
\B	Не граница слова +
\<	Начало слова
\>	Конец слова

\c	Управляющий символ
\s	Пробел
\S	Не пробел
\d	Цифра
\D	Не цифра
\w	Слово
\W	Не слово
\xhh	Шестнадцатиричный символ hh
\Oxxx	Восьмиричный символ xxx

Специальные символы

\	Экранирующий символ +
\n	Новая строка +

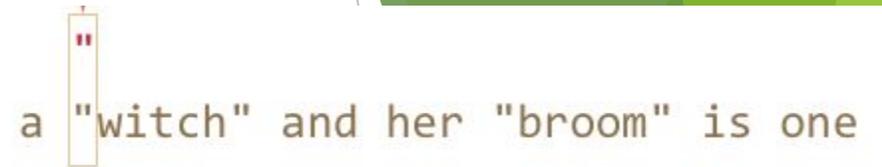
. – любой символ

*	0 или больше +
*?	0 или больше, нежадный +
+	1 или больше +
+	1 или больше +
+	1 или больше, нежадный +
?	0 или 1 +
??	0 или 1, нежадный +
{3}	Ровно 3 +
{3,}	3 или больше +
{3,5}	3, 4 или 5 +
{3,5}?	3, 4 или 5, нежадный +

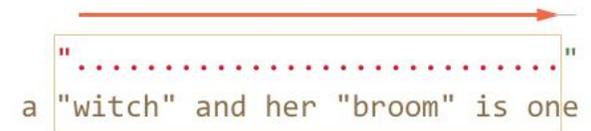
.	Любой символ, кроме переноса строки (\n) +
(a b)	a или b +
(...)	Группа +
(?:...)	Пассивная группа +
[abc]	Диапазон (a или b или c) +
[^abc]	Не a, не b и не c +
[a-q]	Буква между a и q +
[A-Q]	Буква в верхнем регистре между A и Q +
[0-7]	Цифра между 0 и 7 +
\n	n-ая группа/подшаблон +

Жадный поиск берем выражение `".+"`. И строку

- ▶ 1. Первый символ шаблона - это кавычка `"`.
- ▶ 2. Кавычка найдена, далее движок проверяет, есть ли соответствие для остальной части паттерна. (точка). Она обозначает «любой символ», так что следующая буква строки `'w'` вполне подходит
- ▶ 3. Далее «любой символ» повторяется, так как стоит квантификатор `.+`. Движок регулярных выражений берёт один символ за другим, до тех пор, пока у него это получается. В данном случае это означает «до конца строки»:
- ▶ 4. Итак, текст закончился, движок регулярных выражений больше не может найти «любой символ», он закончил повторения для `.+` и переходит к следующему символу шаблона. Следующий символ шаблона - это кавычка. Её тоже необходимо найти, чтобы соответствие было полным.
- ▶ 5. Иными словами, он сокращает текущее совпадение на один символ:
- ▶ 6. Движок продолжает отступать, он уменьшает количество повторений точки `'.'` до тех пор, пока остаток паттерна, то есть в данном случае кавычка `"`, не совпадёт:



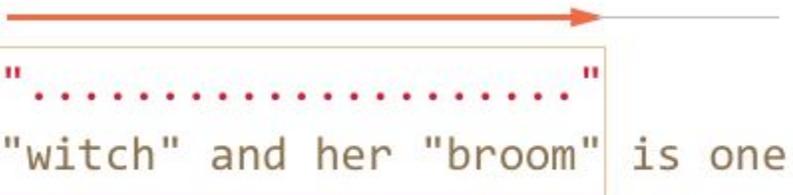
a "witch" and her "broom" is one



" "

a "witch" and her "broom" is one

Результат

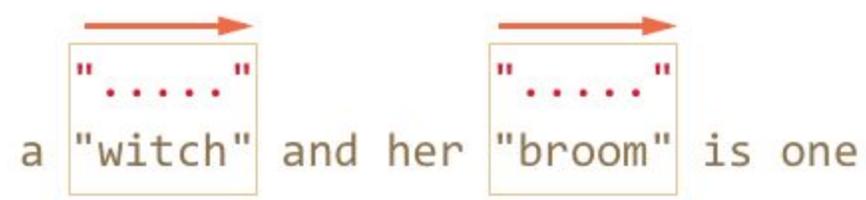
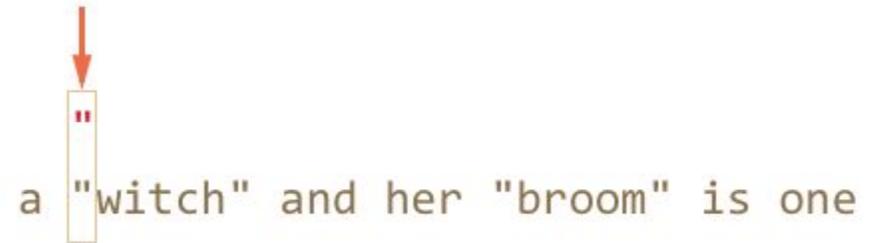


a "witch" and her "broom" is one

В жадном режиме (по умолчанию) регэксп повторяет квантификатор настолько много раз, насколько это возможно, чтобы найти соответствие.

Ленивый режим

- ▶ Ленивый режим работы квантификаторов - противоположность жадному, он означает «повторять минимальное количество раз».
- ▶ 1. Первый шаг - тот же, кавычка "" найдена на 3-й позиции:
- ▶ 2. Второй шаг - тот же, находим произвольный символ '.':
- ▶ 3. А вот дальше - так как стоит ленивый режим работы +, то ... повторит точку (произвольный символ) ещё раз, а останавливается на достигнутом и пытается проверить, есть ли соответствие остальной части шаблона, то есть ""':Если бы остальная часть шаблона на данной позиции совпала, то совпадение было бы найдено. Но в данном случае - нет, символ 'i' не равен ','.
- ▶ 4. Движок регулярных выражений увеличивает количество повторений точки на одно и пытается найти соответствие остатку шаблона ещё раз:



Ленивость распространяется только на тот квантификатор, после которого стоит ?.

Ленивый режим без необходимости лишний раз квантификатор не повторяет.

Модификаторы шаблонов

g	Глобальный поиск
i	Регистронезависимый шаблон
m	Многострочный текст
s	Считать текст одной строкой

Задание: Составить регулярное выражение для:

- 1.Дата (напр., 21/3/2006)
- 2.Адрес email
- 3.Тег HTML
- 4.url
- 5.Трехзначное число с 2 знаками после точки (цена)

Маска ввода в полях формы на jQuery

masked input – очередной плагин предназначенный для javascript фреймворка jQuery, с его помощью можно установить строгий формат ввода (маску ввода) в текстовые поля формы что хорошо подойдет для ввода времени, даты, телефона и т.п.

Плагин корректно работает во всех браузерах начиная с IE6 +, FF 1.5 +, Opera, Safari, Chrome.

Подключение:

-
- 1 `<script src="jquery.js" type="text/javascript"></script>`
 - 2 `<script src="jquery.maskedinput.js" type="text/javascript"></script>`

Настройка

Далее нужно вызвать функцию плагина для всех элементов формы для которых надо создать маску, и передать в нее необходимые параметры:

```
jQuery(function($){  
    $("#date").mask("99/99/9999");  
    $("#phone").mask("(999) 999-9999");  
    $("#tin").mask("99-99999999");  
    $("#ssn").mask("999-99-9999");  
});
```

Функция плагина принимает 2 параметра первый — сама маска для ввода, второй — объект дополнительных параметров.

a — трактуется как буквенный символ из диапазона (A-Z,a-z)

9 — трактуется как числовой символ (0-9)

* — трактуется как алфавитно цифровой символ (A-Z,a-z,0-9)

Пример

В объекте дополнительных параметров можно указать заполнитель маски ввода, по умолчанию он равен `_`, к примеру вы захотели изменить его на пробел

```
jQuery(function($){
  $("#product").mask("99/99/9999",{
    placeholder:" "});
});
```

Так же в качестве дополнительного параметра можно указать функцию обратного вызова, которая будет вызвана после того как все элементы маски будут заполнены, например:

```
1 | jQuery(function($){
2 |     $("#product").mask("99/99/9999",{completed : function(){
3 |         alert("Вы ввели: "+this.val());}
4 |     });
5 | });
```

Иногда бывают такие ситуации, что возникает потребность не заполнять пользователем некоторую часть маски ввода, для этого вы можете использовать спецсимвол `?` — все что идет после этого символа является не обязательным для ввода с стороны пользователя, например:

```
1 | jQuery(function($){
2 |     $("#phone").mask("(999) 999-9999? x99999");
3 | });
```

Задание

Анкета

Фамилия

Имя

Email

Телефон

Дата рождения

Как с вами связаться?

- Email
- Телефон

Какую температуру воздуха предпочитаете? -10C 50C

Ваш любимый цвет

Откуда вы о нас узнали?

- <http://www.techcrunch.com/> TechCrunch
- <http://www.readwriteweb.com/> ReadWrite Web
- <http://www.engadget.com/> Engadget
- <http://www.ajaxian.com/> Ajaxian