

LOG IN

SIGN UP

# Forms & Lists

Your username



mysuperusername690

Your email



mysupermail@mail.com

Your password



**Формы**

# Примеры форм



Visual Idiot

last online: 12:49AM

Email address

Password

Proceed

## login or register

hello@johnappleseed.co.uk



F\*@%!



take me there..

[ah, forgetful me!](#)



Username

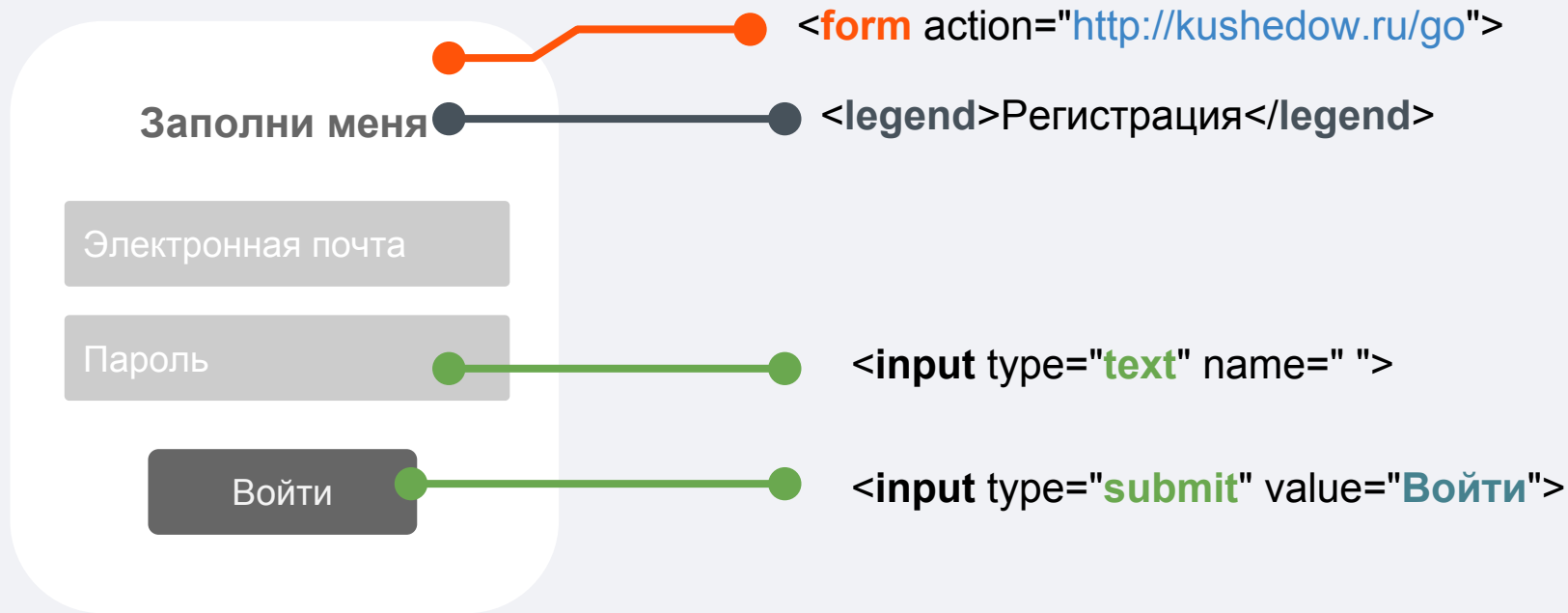


Password

SIGN IN

Not a member? [Sign up now →](#)

# Из чего состоит форма



# Из чего состоит форма

form

Заказать консультацию

Имя

Телефон

Заказать

h2

P

input

P

input

input:submit

Имя

Ваше имя

`<fieldset>`

`<legend> Имя </legend>`

`</fieldset>`

Данные

Электронная почта

Пароль

Войти

`<form action="http://kushedow.ru/go">`

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

`<input type="submit" value="Войти">`

# Из чего состоит форма

Выбери пунктик

☐ Не хочу

☐ Не умею

☐ Не люблю

Или.. ▼

`<form action="http://kushedow.ru/go">`

`<input type="checkbox" id="want" name=" ">`

`<label for="want">Не хочу</label>`

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

`<select>`

`<option>Или да</option>`

`<option>Или нет</option>`

`</select>`

# FORM

---

устанавливает форму на веб-странице для обмена данными между пользователем и сервером

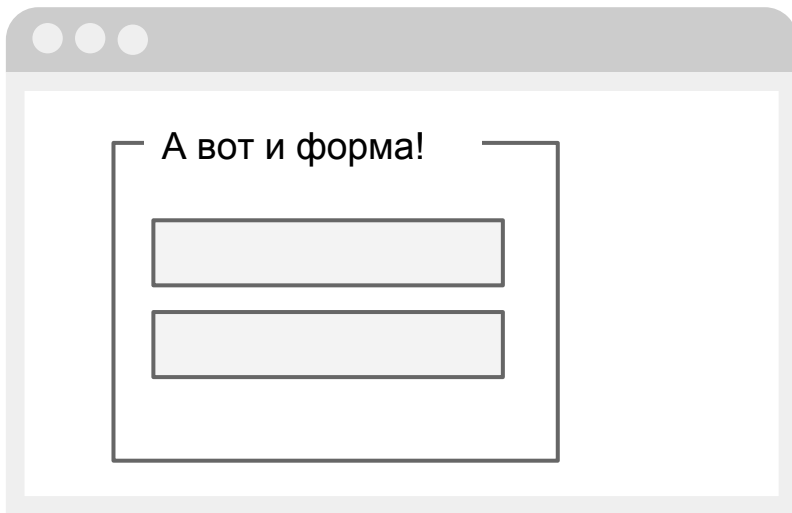
## Атрибуты:

- **name** - имя формы
- **method** - метод протокола HTTP
- **action** - адрес программы или документа, который обрабатывает данные формы



# FIELDSET

предназначен для группирования элементов формы

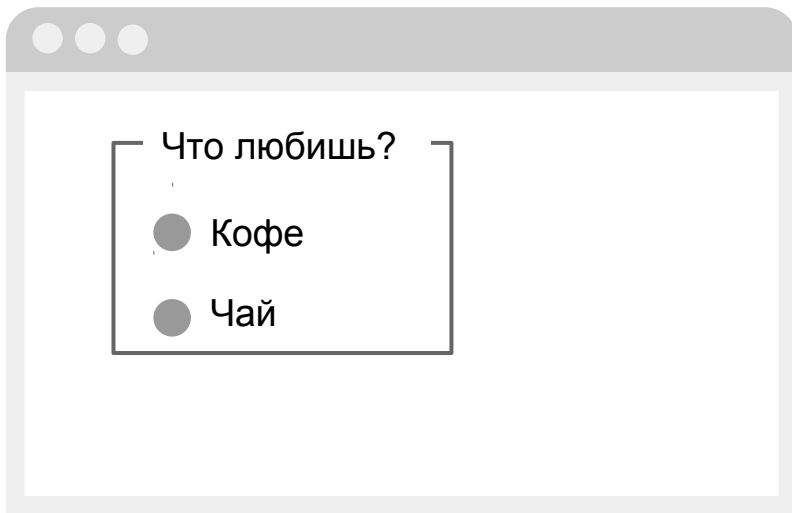


`form>fieldset>legend+input*2`

```
<form action="">  
  <fieldset>  
    <legend>А вот и форма!</legend>  
    <input type="text">  
    <input type="text">  
  </fieldset>  
</form>
```

# LEGEND

применяется для создания заголовка группы элементов формы, которая определяется с помощью тега `<fieldset>`



What you like?

☐ Coffee

☐ Tea

`form>fieldset>legend+input*2`

```
<form action="">
```

```
<fieldset>
```

```
<legend>Что любишь?</legend>
```

```
<input type="radio"><label>Кофе</label>
```

```
<input type="radio"><label>Чай</label>
```

```
</fieldset>
```

```
</form>
```

Теперь рассмотрим **элементы** и  
**типы** форм

# Поехали!



# Элемент **INPUT**

- создает поле формы (кнопку, поле ввода, чекбокс и т.п.)
- должен располагаться внутри элемента **FORM**
- является одиночным тегом
- содержание может быть изменено или активизировано пользователем

# INPUT TYPE="TEXT"

однострочное поле для ввода текста

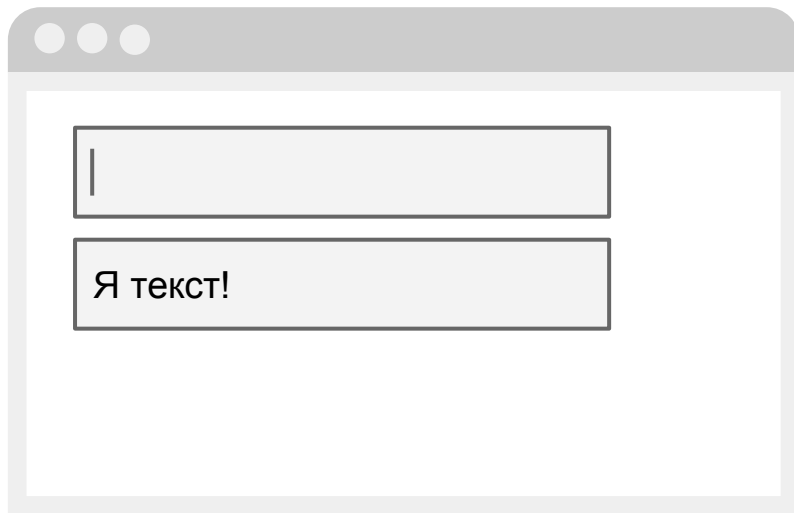


Illustration of a web browser window showing two text input fields. The top field is empty with a vertical cursor. The bottom field contains the text "Я текст!".

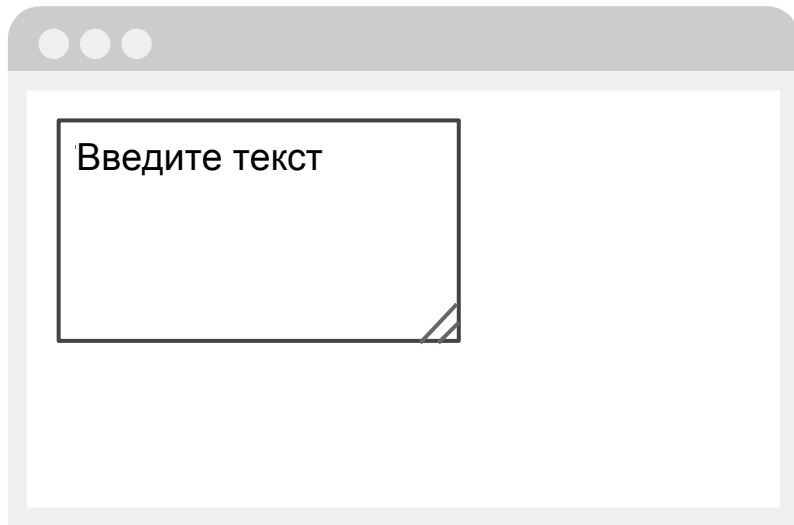
input+input[value="Я текст"]

`<input type="text">`

`<input type="text" value="Я текст">`

# TEXTAREA

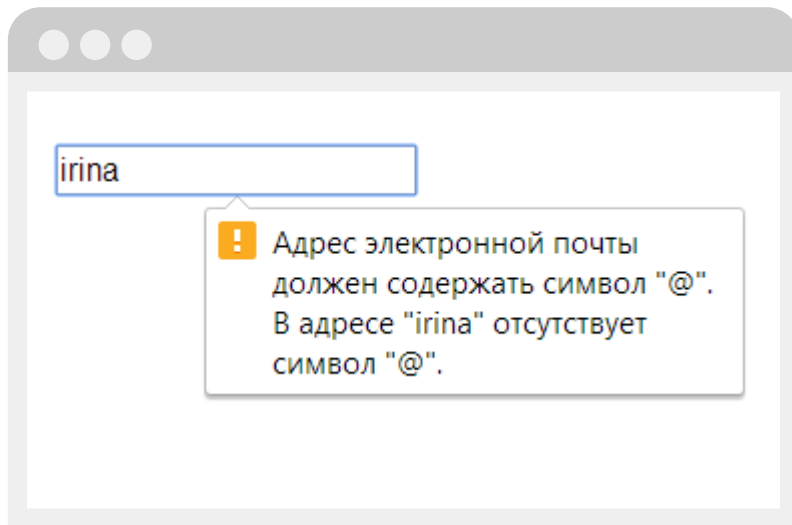
представляет собой элемент формы для создания области, в которую можно вводить несколько строк текста



**<textarea>Введите текст</textarea>**

# INPUT TYPE="EMAIL"

подобен типу **TEXT**, только автоматически проверяет правильность ввода электронной почты



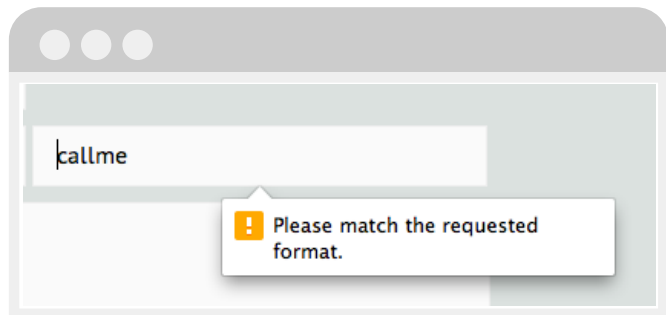
```
input[type="email"]
```

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



# INPUT TYPE="TEL"

подобен типу TEXT, только автоматически проверяет правильность ввода номера телефона



`input[type="tel"]`

`<input type="tel">`

Сделаем проверку на введенные данные

`<p>Введите телефон в формате 344xxxx или 8911xxxxxxxx:</p>`

`<input type="tel" pattern="[0-9]{7,11}">`

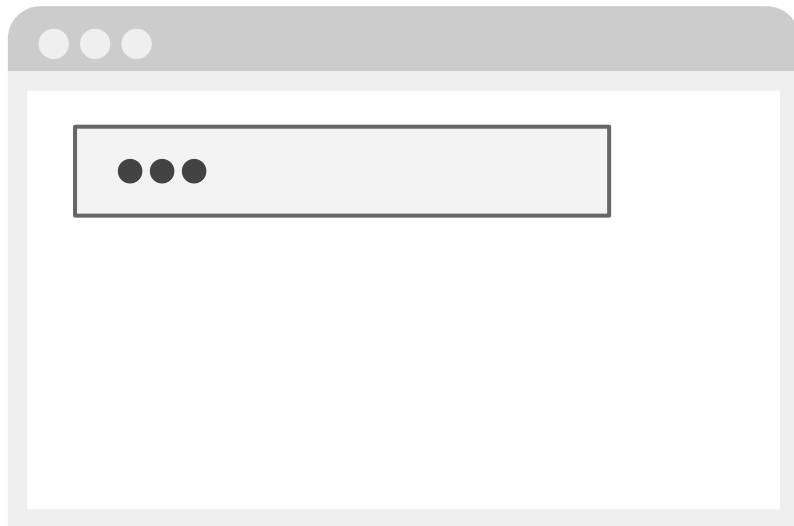


только цифры

кол-во цифр, от 7 до 11

# INPUT TYPE="PASSWORD"

подобен типу **TEXT**, однако все вводимые символы представляются в виде \*, чтобы скрыть текст от подсматривающих глаз, когда вводится пароль

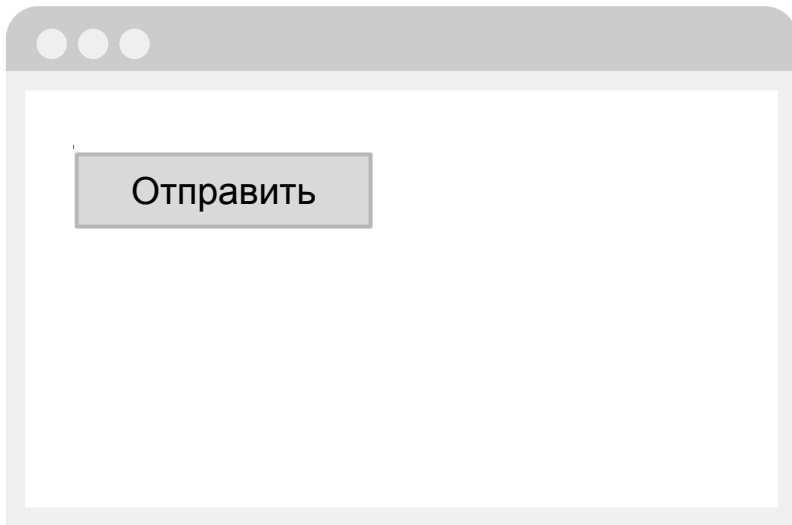


```
input[type="password"]
```

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

# INPUT TYPE="SUBMIT"

создает кнопку, при нажатии которой заполненная форма посылается на сервер  
атрибут **VALUE** в данном случае изменяет надпись на кнопке



```
input[type="submit"]
```

```
<input type="submit"  
value="Отправить">
```

# INPUT TYPE="CHECKBOX"

создает поле ввода для атрибутов типа Boolean (**да/нет**) или для атрибутов, которые могут одновременно принимать несколько значений



```
input[type="checkbox"]*3
```

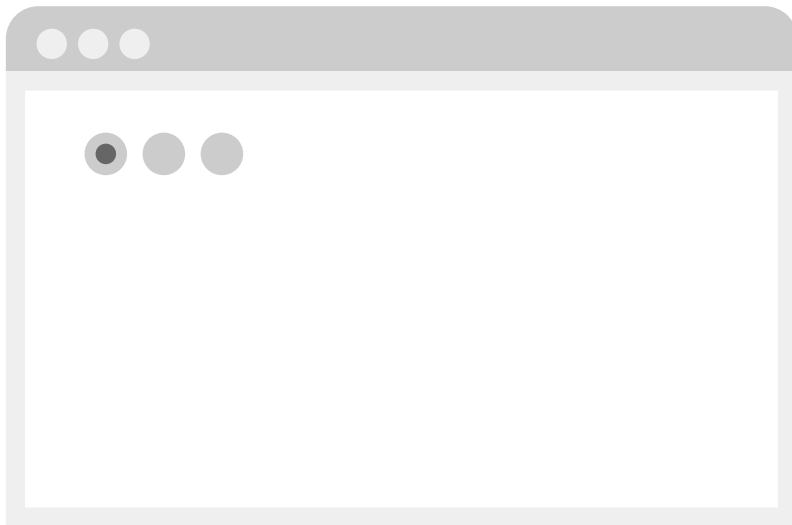
```
<input type="checkbox">
```

```
<input type="checkbox">
```

```
<input type="checkbox">
```

# INPUT TYPE="RADIO"

создает поле ввода для атрибутов, которые принимают одно значение из нескольких  
ВОЗМОЖНЫХ



```
input[type="radio"]*3
```

```
<input type="radio">
```

```
<input type="radio">
```

```
<input type="radio">
```

# INPUT TYPE="RANGE"

позволяет сделать плавающий ползунок со значением

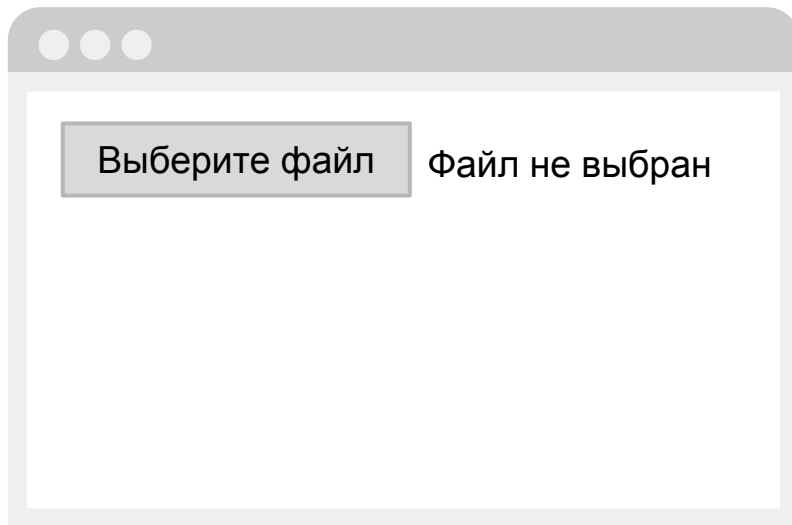


```
input[type="range"]
```

```
<input type="range">
```

# INPUT TYPE="FILE"

дает возможность пользователям прикрепить файл к содержимому формы

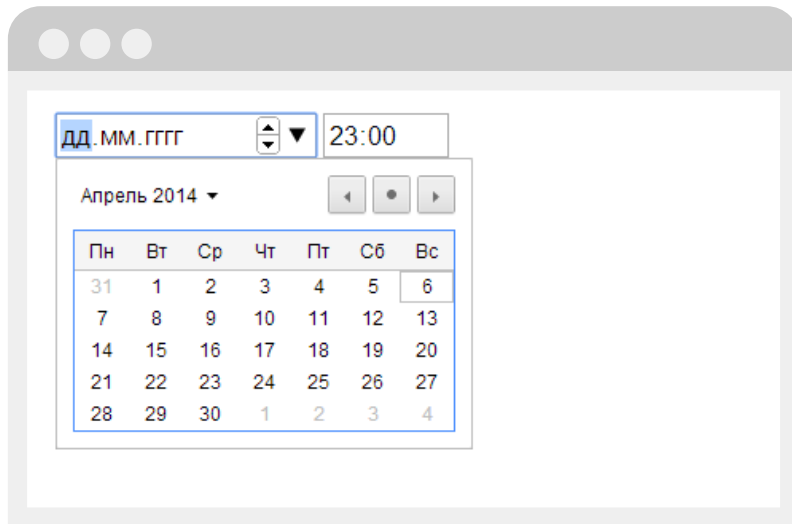


```
input[type="file"]
```

```
<input type="file">
```

# INPUT TYPE="DATETIME"

создает календарь, предназначенный для указания даты, месяца или недели, и возможность вводить ещё и время



dd.mm.yyyy 23:00

Апрель 2014

Пн	Вт	Ср	Чт	Пт	Сб	Вс
31	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

`input[type="datetime"]`

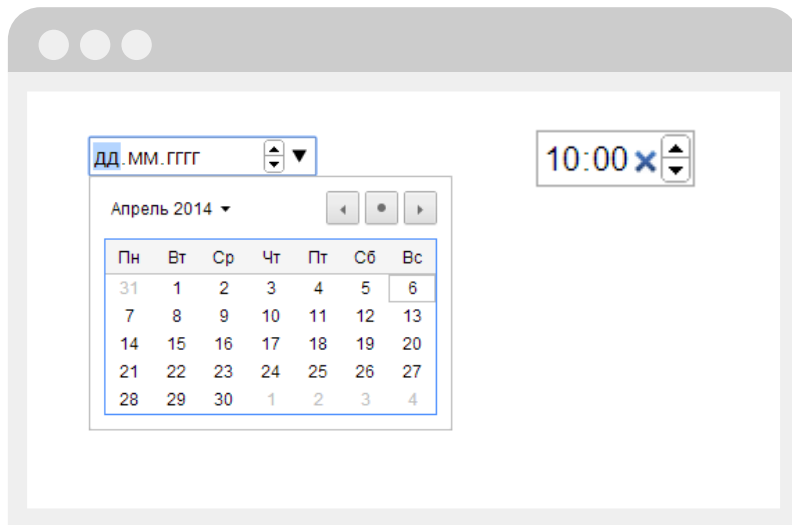
`<input type="datetime">`



# INPUT TYPE="DATE"

# INPUT TYPE="TIME"

также есть возможность использовать по отдельности - календарь и время



The screenshot shows a web browser window with two input fields. The first field is a date picker with a dropdown menu showing 'ДД.ММ.ГГГГ' and a calendar for April 2014. The second field is a time picker showing '10:00' with a blue 'x' and a dropdown arrow.

Пн	Вт	Ср	Чт	Пт	Сб	Вс
31	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

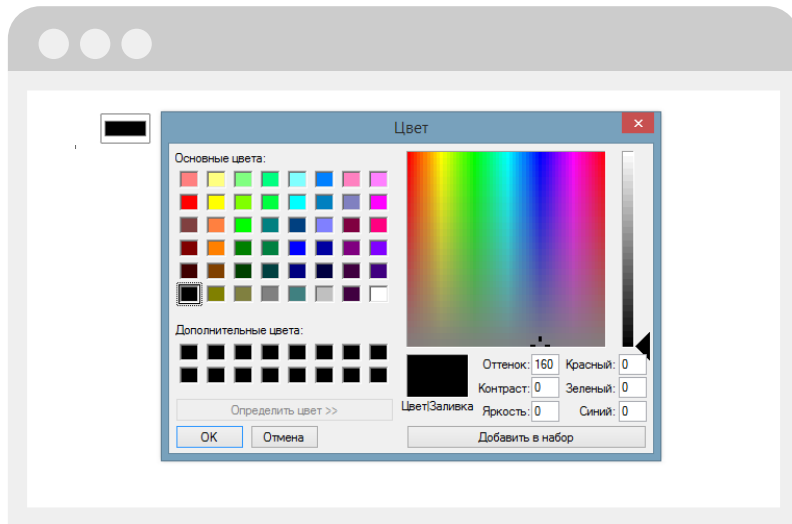
`input[type="date"] + input  
[type="time"]`

`<input type="date">`

`<input type="time">`

# INPUT TYPE="COLOR"

отображает виджет для выбора цвета

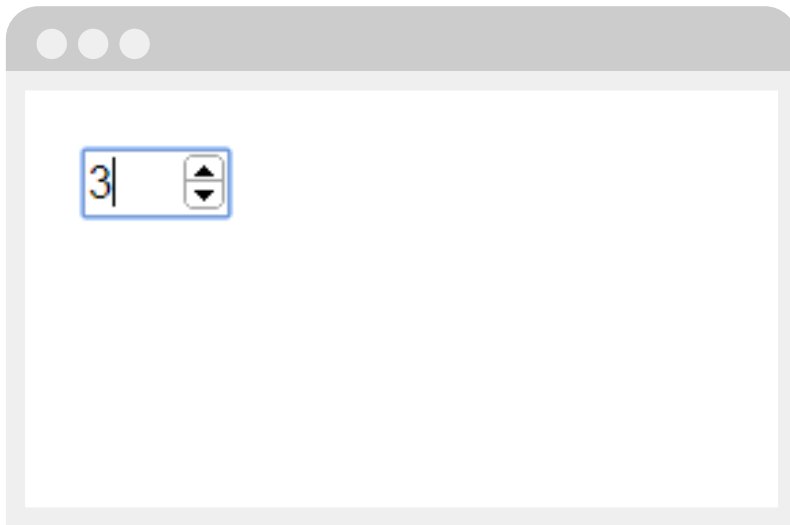


```
input[type="color"]
```

```
<input type="color">
```

# INPUT TYPE="NUMBER"

поле для ввода числовых значений

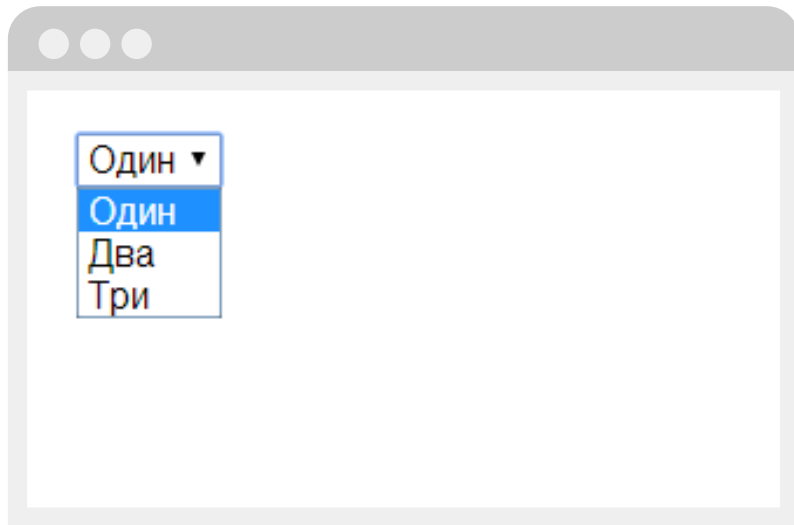


```
input[type="number"]
```

```
<input type="number">
```

# SELECT

позволяет создать элемент интерфейса в виде раскрывающегося списка, а также список с одним или множественным выбором



`select>option*3`

```
<select>  
  <option>Один</option>  
  <option>Два</option>  
  <option>Три</option>  
</select>
```

# LABEL

устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы



Да

Нет

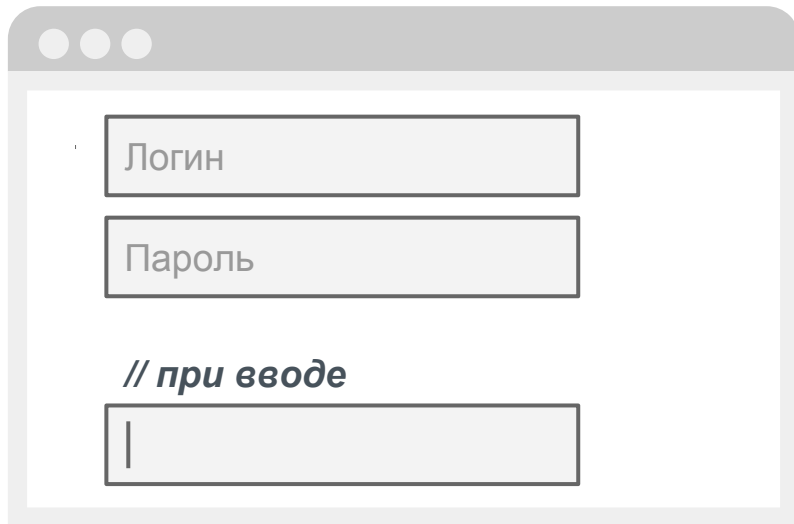
```
input[type="radio"]*2 + label*2
```

```
<input type="radio" id="yes" name=" ">  
<label for="yes">Да</label>
```

```
<input type="radio" id="no" name=" ">  
<label for="no">Нет</label>
```

# PLACEHOLDER

выводит текст внутри поля формы, который исчезает при вводе первого символа



The diagram shows a web browser window with three input fields. The first field contains the text 'Логин' (Login). The second field contains the text 'Пароль' (Password). The third field is empty and has the text '// при вводе' (// on input) written above it, indicating the placeholder text disappears when the user starts typing.

`input[placeholder]*2`

`<input type="text" placeholder="Логин">`

`<input type="password" placeholder="Пароль">`

**Advanced:** исчезновение сразу при вводе курсора

`<input onfocus="this.value = '';return false;" value="ваш e-mail">`

# Вставка удобного поля для ввода номера телефона



При этом поле называем так:

```
<input type="text" id="phone">
```

1. Нам понадобится [скрипт](#) «Masked Input Plugin», написанный на jQuery, и сама [библиотека](#)

2. Подключаем

```
<script src="jquery.js" type="text/javascript"></script>
```

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

3. Активируем

```
<script type="text/javascript">
```

```
jQuery(function($){
```

```
    $("#phone").mask("+7 (999) 999-9999");
```

```
});
```

```
</script>
```

# Совет

Браузеры **по умолчанию** имеют предустановленные стили для элементов форм (отличающиеся от браузера к браузеру).

В некоторых случаях стилизация элементов может “**поломаться**” в браузере.

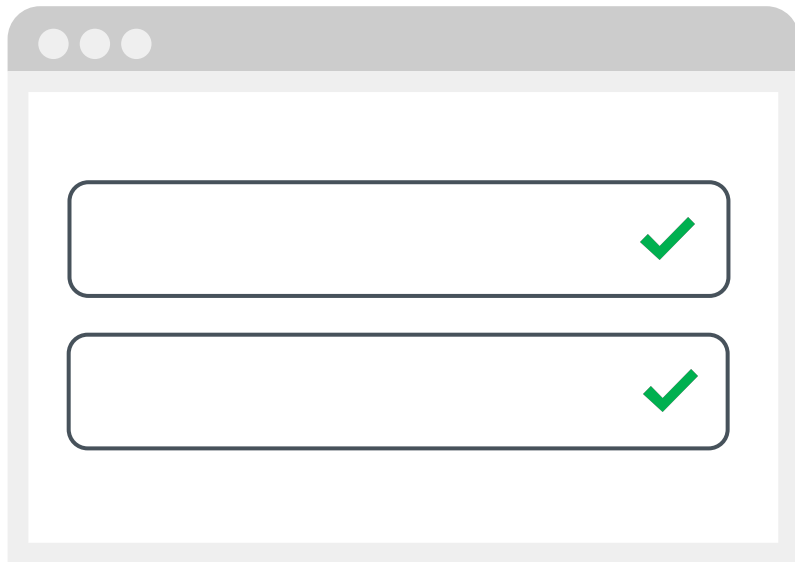
Поэтому лучше **не стилизовать чекбоксы, радиокнопки, загрузку файлов и селект.**

Мы предупредили.



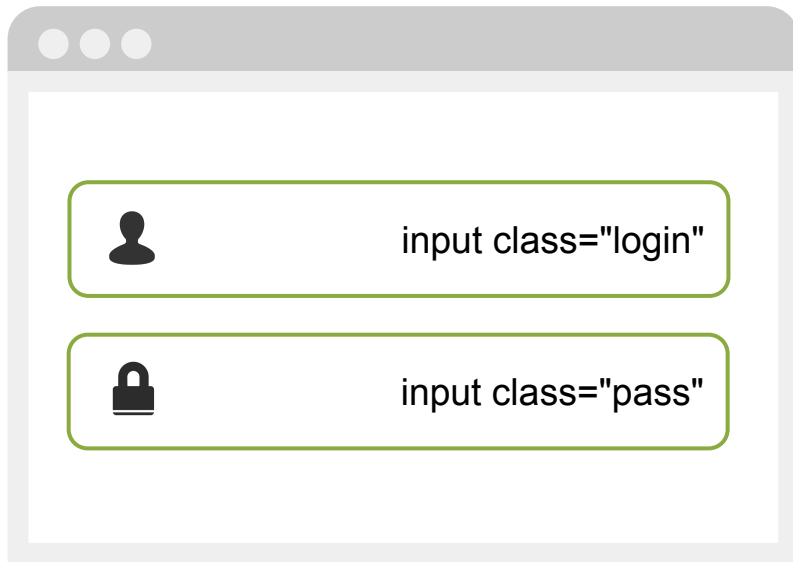


# Фон для элемента формы



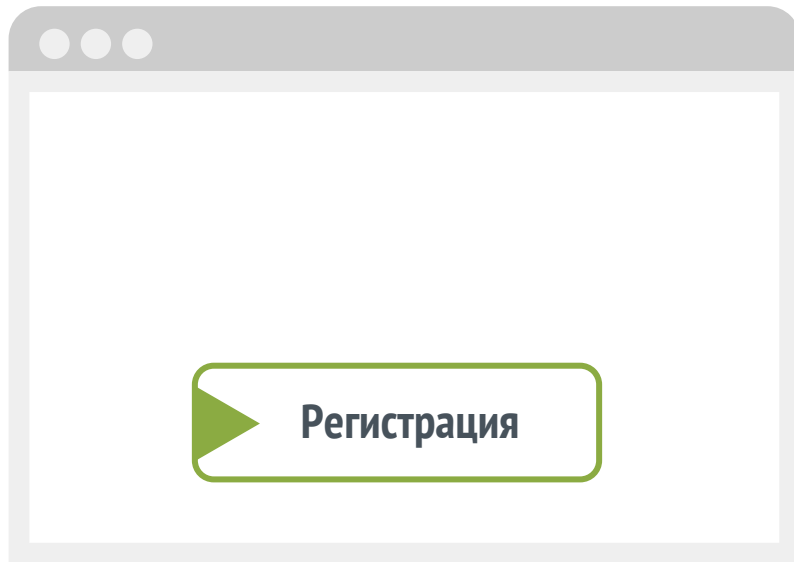
```
input {  
    background-image: url(check.png);  
    background-position: 92% 50%;  
    background-size: 30px;  
    background-repeat: no-repeat;  
}
```

# Фон для элемента формы



```
input {  
    background-image: url(lock.png);  
    background-position: 5px 5px;  
    background-size: 16px;  
    background-repeat: no-repeat;  
}
```

# Фон для элемента формы



```
input {  
    background-image: url(check.png);  
    background-position: left center;  
    background-size: 30px;  
    background-repeat: no-repeat;  
}
```



# Списки

# Виды списков

---

● Яблоко

● Груша

● Банан

1. Задача 1

2. Задача 2

3. Задача 3

● Пункт 1

● Подпункт 1

● Подпункт 2

● Пункт 2

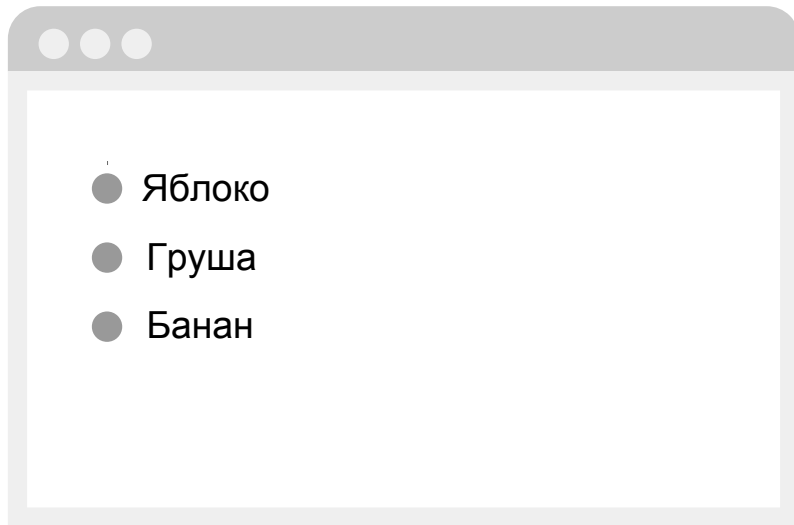
Термин 1

Аннотация1 к термину 1

Аннотация2 к термину 1

# Неупорядоченный список

определяется тем, что перед каждым элементом списка добавляется небольшой маркер, обычно в виде закрашенного кружка



`ul>li*3`

`<ul>`

`<li> Яблоко </li>`

`<li> Груша </li>`

`<li> Банан </li>`

`</ul>`

# Стили маркеров списка

маркеры могут принимать один из трех видов:  
**круг (по умолчанию), окружность и квадрат**

**type="disk"**

● Яблоко

● Груша

● Банан

**type="circle"**

○ Яблоко

○ Груша

○ Банан

**type="square"**

■ Яблоко

■ Груша

■ Банан



# Упорядоченный список

предназначен для перечисления элементов, следующих в строго определенном порядке, по умолчанию элементы списка нумеруются по порядку



```
ol>li{Задача$}*3
```

```
<ol>
```

```
<li> Задача 1 </li>
```

```
<li> Задача 2 </li>
```

```
<li> Задача 3 </li>
```

```
</ol>
```

# Стили нумерации списка

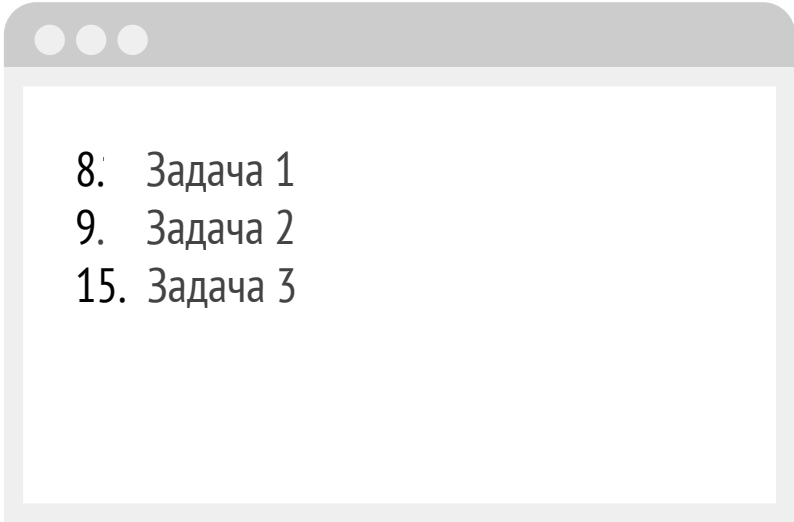
---

атрибут **TYPE** позволяет изменять стиль нумерации:

- A** - заглавные буквы
- a** - строчные буквы
- I** - заглавные римские цифры
- i** - строчные римские цифры
- 1** - арабские цифры (по умолчанию)

# Цифра старта

есть ещё один атрибут **start** - его числовое значение служит для задания номера списка, отличного от 1, и **value** - дает возможность назначить произвольный номер любому элементу списка

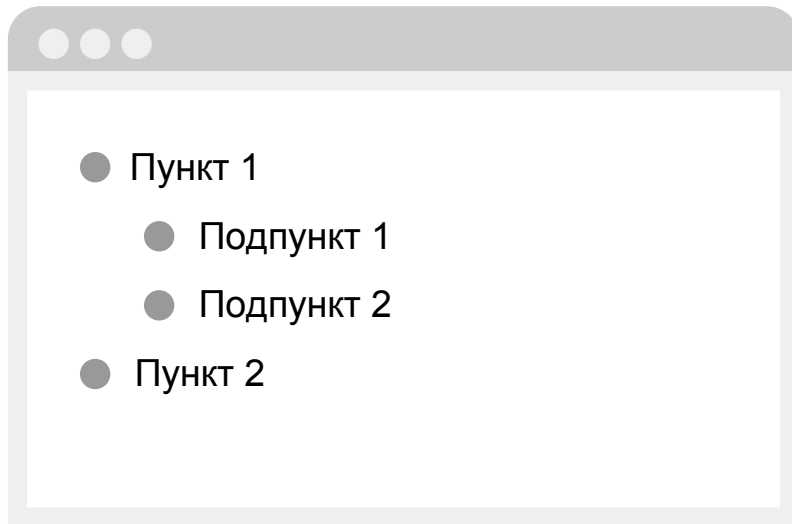


8. Задача 1  
9. Задача 2  
15. Задача 3

```
<ol start = "8">  
  <li> Задача 1 </li>  
  <li> Задача 2 </li>  
  <li value="15"> Задача 3</li>  
</ol>
```

# Вложенные списки

вложенные или многоуровневые списки определяются порядком размещения тегов:



`ul>(li>ul>li*2)+li`

```
<ul>
  <li> Пункт 1
    <ul>
      <li> Подпункт 1 </li>
      <li> Подпункт 2 </li>
    </ul>
  </li>
  <li> Пункт 2</li>
</ul>
```

# Списки определений

---

для создания списков определений используют три тэга:

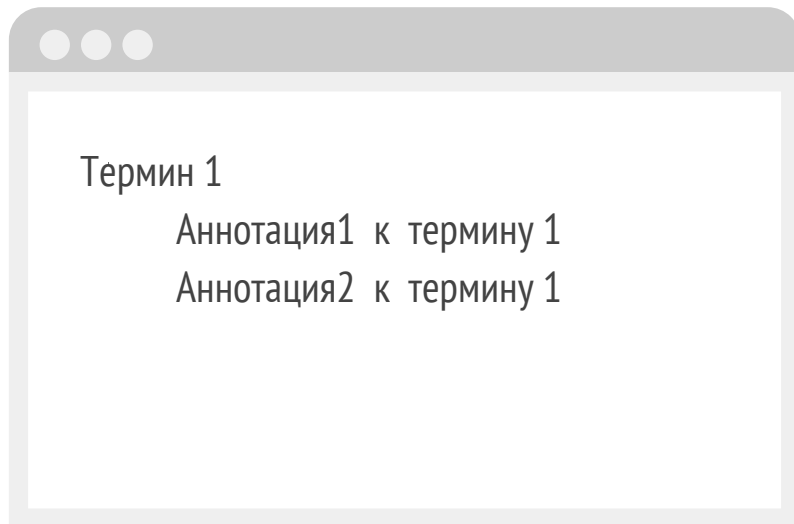
**<DL>** - начало/конец списка.

**<DT>** - начало/конец конкретного термина.

**<DD>** - начало/конец поясняющей статьи термина.

Тэги <DT> и <DD> не обязательно чередовать, можно "привязать" к одному определению несколько терминов, и, наоборот.

# Списки определений



**dl>dt+dd\*2**

**<dl>**

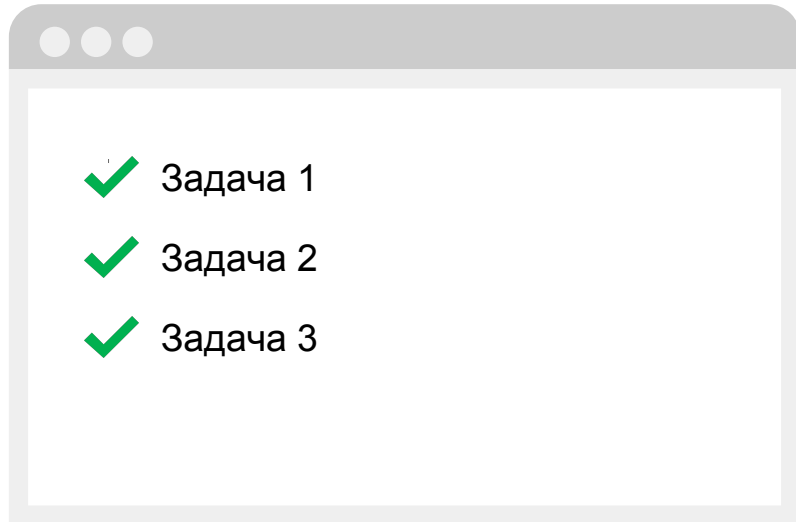
**<dt>**Термин 1**</dt>**

**<dd>**Аннотация1 к термину 1**</dd>**

**<dd>**Аннотация2 к термину 1**</dd>**

**</dl>**

# Свойство list-style-image



```
ul li {  
    list-style-image: url(check.png);  
}
```

# Картинки для списков через background



```
ul li {  
    list-style:none;  
    background: url(check.png) no-repeat;  
    background-position: 0% 50%;  
    background-size: 40px;  
}
```



**Напоминалка: рассказать про что-то  
очень важное**