

# Элементы ввода, формы, валидация данных



Web.Dev.  
Courses

[sp.courses.dp.ua](http://sp.courses.dp.ua)

Подробнее о DOM

# DOM – Document Object Model

*(объектная модель  
документа)*

*Стандарт который определяет из  
каких объектов браузер собирает  
дерево документа, и какие свойства  
есть у этих объектов у этих.*

<https://learn.javascript.ru/document>

# Свойства `.children` и `.nodeChild` МАССИВЫ С ПОТОМКАМИ ТЕГА

```
<div>  
  <p>Text #1</p>  
  <p>Text #2</p>  
  <p>Text #3</p>  
  <p>Text #4</p>  
  <p>Text #5</p>  
</div>
```

```
▼ NodeList[11] ⓘ  
  ▶ 0: text  
  ▶ 1: p  
  ▶ 2: text  
  ▶ 3: p  
  ▶ 4: text  
  ▶ 5: p  
  ▶ 6: text  
  ▶ 7: p  
  ▶ 8: text  
  ▶ 9: p  
  ▶ 10: text  
  length: 11  
  ▶ __proto__: Object
```

***.childNodes***

```
▼ [p, p, p, p, p] ⓘ  
  ▶ 0: p  
  ▶ 1: p  
  ▶ 2: p  
  ▶ 3: p  
  ▶ 4: p  
  length: 5  
  ▶ __proto__: Object
```

***.children***

Немного практики

# Немного практики

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>DOM.task1</title>
5 <script>
6   window.onload = function(){
7     //Начало вашего кода.....
8
9     //.....Конец вашего кода
10  }
11 </script>
12 </head>
13 <body>
14 <h1>Каталог телефонов <button>Отсортировать</button></h1>
15 <div id="phone_list">
16   <div>iPhone 3Gs <span>582$</span></div>
17   <div>Lenovo <span>997$</span></div>
18   <div>HTC <span>698$</span></div>
19   <div>Samsung <span>242$</span></div>
20   <div>Fly <span>793$</span></div>
21   <div>Asus <span>771$</span></div>
22   <div>Nokia <span>108$</span></div>
23   <div>LG <span>214$</span></div>
24   <div>IPhone 5s <span>216$</span></div>
25   <div>IPhone 6+ <span>326$</span></div>
26   <div>IPhone 6s+ <span>799$</span></div>
27 </div>
28 </body>
29 </html>
```

## Каталог телефонов

iPhone 3Gs 582\$  
Lenovo 997\$  
HTC 698\$  
Samsung 242\$  
Fly 793\$  
Asus 771\$  
Nokia 108\$  
LG 214\$  
IPhone 5s 216\$  
IPhone 6+ 326\$  
IPhone 6s+ 799\$



*По нажатию на кнопку необходимо отсортировать список по возрастанию цены.*

Скопируйте код:

<http://courses.dp.ua/files/is/tasks/task1.html>

# Немного практики

```
5 <script>
6   window.onload = function() {
7     //Начало вашего кода.....
8     var btn = document.querySelector("button");
9     btn.onclick = function() {
10      var list = document.querySelector("h1 + div");
11      var elements = list.children;
12      for(var i = 0; i < elements.length - 1; i++){
13        for(var j = 0; j < elements.length - 1 - i; j++){
14          var price1 = parseInt(elements[j].querySelector("span").innerHTML);
15          var price2 = parseInt(elements[j+1].querySelector("span").innerHTML);
16          if(price1 > price2) {
17            list.insertBefore(elements[j+1], elements[j]);
18          }
19        }
20      }
21    }
22    //.....Конец вашего кода
23  }
24 </script>
```

*По нажатию на кнопку необходимо  
отсортировать список по возрастанию  
цены.*

Данные необходимо не  
ТОЛЬКО выдавать но и  
получать



# Формы & элементы ВВОДА

## Bootstrap Form Example

User name:

Fill required!

Email address

Fill required correct email!



Скачайте заготовку в

Notepad++

<http://files.courses.dp.ua/web/15/ex01.html>

# Элементы ввода, формы

*Получение данных от  
пользователя*

Регистрация пользователя

Имя:

Email:

**<form> ... </form>**

**Элементы ввода,**

**формы**

*Получение данных от  
пользователя.*

**<form> ... </form>**

**<form action="..." >**

*Атрибут **action** позволяет указать какой странице будут переданы данные для обработки (какой странице на сервере).*

# <form method="..."

GET



POST

```
<form action="file.php" method="GET">
```

```
<form action="file.php" method="POST">
```

Параметры передаются в открытом виде, в строке URL.

Длина строки параметров ограничена (255 байт, ограничение связано с совместимостью).

Является методом по умолчанию.

Параметры передаются неявно, в теле HTTP запроса.

Объём передаваемых параметров неограничен (зависит от настроек сервера).

# Элементы ввода

`<input ... >`

`type="text"`

*Для размещения элементов ввода на форме применяется тег `input` с различными значениями атрибута `type`.*

# Элементы ввода

`<input ... >` `<label>`

*Тег `label` применяется для оформления подписей к элементам ввода, связывается с элементом ввода по `id`.*

# <input 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="Выбрати..."/> Файл не выбрано.
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="Скинути"/>
submit	Кнопка для отправки данных формы на сервер.	<input type="submit" value="Надіслати"/>
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.	<input type="text"/>

*Изначальный набор элементов ввода*

# <input placeholder="...">

```
<label for="user_name_id">Имя: </label>  
<input type="text" name="user_name" id="user_name_id" placeholder="Enter name">
```

## Form Example

Имя:  Email:



*Атрибут **placeholder** позволяет задать замещающий текст, который даст подсказку о назначении поля ввода.*



# <input type="..."> HTML5

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.

*HTML5 принёс новые типы элементов ввода*

# Однако

....

```
12 <label for="user_name_id">Дата рождения: </label>  
13 <input type="date" name="user_bday" id="user_bday_id">  
14
```



*Поддержка «новых» элементов ввода в браузерах оставляет желать лучшего.*

# Примеры HTML5...

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP JQUERY TUTORIALS REFERENCES EXAMPLES

## HTML Tutorial

- HTML HOME
- HTML Introduction
- HTML Editors
- HTML Basic
- HTML Elements
- HTML Attributes
- HTML Headings
- HTML Paragraphs
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Computercode
- HTML Comments
- HTML Colors
- HTML CSS
- HTML Links
- HTML Images
- HTML Tables
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Layout
- HTML Responsive
- HTML Iframes
- HTML JavaScript
- HTML Head
- HTML Entities
- HTML Symbols
- HTML Charset
- HTML URL Encode

### Input Type: date

The **<input type="date">** is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

#### Example

```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

[Try it Yourself >](#)

You can add restrictions to the input:

#### Example

```
<form>
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
</form>
```

[Try it Yourself >](#)

[http://www.w3schools.com/html/html\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html_form_input_types.asp)

# Параметры в URL

# Параметры в URL-

```
/file.html?userName=Ivan&userEmail=ivan%40mail.com
```

*Передача параметров файлу, через  
URL*

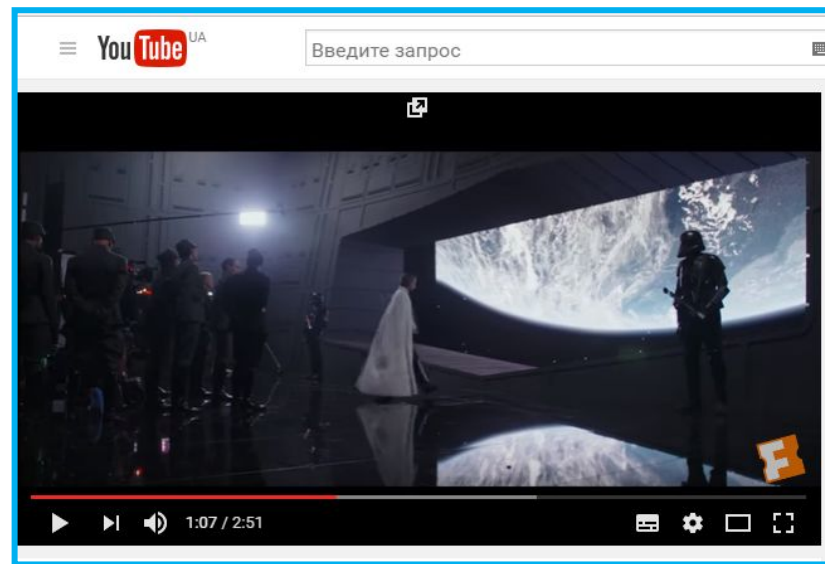
- ?** – спецсимвол, говорящий, что в этом месте заканчивается адрес файла, и начинается перечень параметров.
- =** – спецсимвол разделяющий имя параметра и его значение.
- &** – спецсимвол разделяющий пары (имя параметра и его значение) друг от друга.

*Параметры передаваемые из формы добавляются в URL адрес, эти параметры есть возможность обработать на стороне сервера*

# Параметры в URL-адресе

<https://www.youtube.com/watch?v=IMHJgaqPepo>

[https://www.youtube.com/watch?v=4B6\\_y1s-Hco](https://www.youtube.com/watch?v=4B6_y1s-Hco)




*Страница одна – параметры разные, как результат и содержимое разное.*

# Оформление элементов ввода (Bootstrap)

# Оформление элементов ВВОДА

```
6
7
8
9
10
11
12
13
14 <head>
```

```
<style>
  input {
    border: 2px solid green;
    border-radius: 15px 40px;
    background: lightblue;
    font-size: 20pt;
  }
</style>
```



*Элементы форм можно оформлять при помощи CSS*



# Оформление элементов ввода

**Form Example**

Имя:

Email:

*Элементы форм можно оформлять при помощи  
CSS*

# Оформление элементов ввода /

```
14 <div class="container">
15
16 <h1> Bootstrap From Example</h1>
17
18 <form>
19 <div class="form-group">
20 <label for="user_name_id" class="form-control-label">User name: </label>
21 <input type="text" name="user_name" class="form-control" id="user_name_id" placeholder="Enter user name, ex: Ivan">
22 <small class="form-text text-muted">Fill required!</small>
23 </div>
24 <div class="form-group">
25 <label for="user_email_id" class="form-control-label">Email address</label>
26 <input type="email" name="user_email" class="form-control" id="user_email_id" placeholder="Enter email, ex: ivan@mail.com">
27 <small id="emailHelp" class="form-text text-muted">Fill required correct email!</small>
28 </div>
29 <button type="submit" class="btn btn-primary">Send data</button>
30 </form>
31
32 </div>
```

## Bootstrap From Example

User name:

Fill required!

Email address

Fill required correct email!



*Bootstrap содержит в себе стили для оформления элементов ввода (пример на основе стартового шаблона*

*bootstrap)*

# Оформление элементов ввода /

```
14 <div class="container">
15
16 <h1> Bootstrap From Example</h1>
17
18 <form>
19 <div class="form-group has-warning">
20 <label for="user_name_id" class="form-control-label form-control-warning">User name: </label>
21 <input type="text" name="user_name" class="form-control form-control-warning"
22     id="user_name_id" placeholder="Enter user name, ex: Ivan">
23 <small class="form-text text-muted">Fill required!</small>
24 </div>
25 <div class="form-group has-danger">
26 <label for="user_email_id" class="form-control-label">Email address</label>
27 <input type="email" name="user_email" class="form-control form-control-danger"
28     id="user_email_id" placeholder="Enter email, ex: ivan@mail.com">
29 <small id="emailHelp" class="form-text text-muted">Fill required correct email!</small>
30 </div>
31 <button type="submit" class="btn btn-primary">Send data</button>
32 </form>
33
34 </div>
```

## Bootstrap From Example

User name: ⚠

Enter user name, ex: Ivan ⚠

Fill required!

Email address

Enter email, ex: ivan@mail.com ✖

Fill required correct email!

Send data

*Bootstrap содержит в себе стили классы для обозначения неверно заполненных полей.*



# Валидация данных

# Валидация данных

**Bootstrap Form Example**

User name:

Fill required!

Email address

Fill required correct email!

[Send data](#)



Скачайте заготовку в

Notepad++

<http://files.courses.dp.ua/web/15/ex01.html>

# Валидация данных

(проверка на

*Валидация данных вводится к проверке соответствия ли введённые данные шаблону или нет. Что считать корректным, а что нет – полностью определяет разработчик.*

# Валидация данных средствами HTML

# Валидация (проверка на корректность) данных

```
</div>
<div class="form-group">
  <label for="user_email_id" class="form-control-label">
  <input type="email" name="user_email" class="form-control">
  <small id="emailHelp" class="form-text text-muted">Fill in this field
</div>
<button type="submit" class="btn btn-primary">Send data
</form>
```

Email address

Fill required correct email!

**Send data**

! Адрес электронной почты должен содержать символ "@". В адресе "lalala" отсутствует символ "@".

*Встроенные средства валидации данных появились вместе с новыми элементами в HTML5,*

*НО...*



# Валидация данных средствами

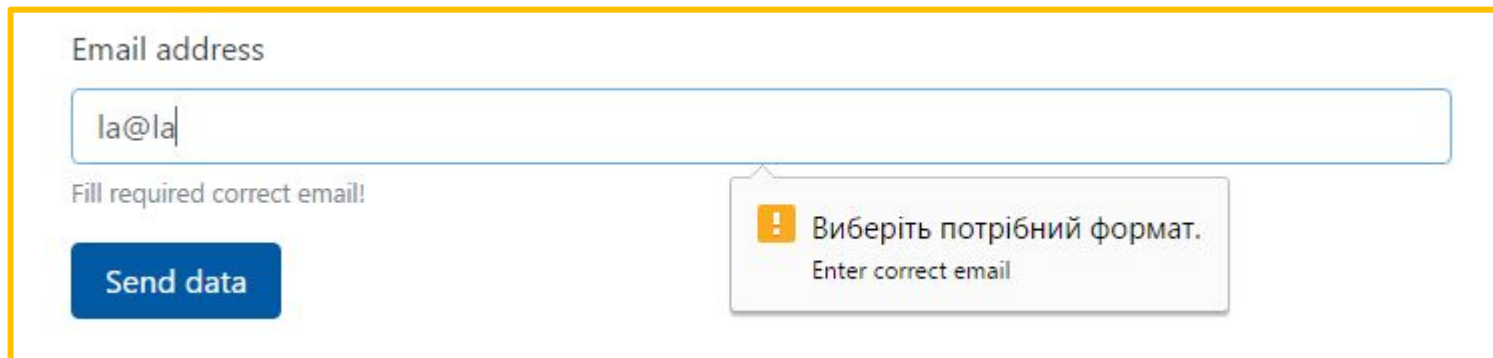
## HTML

```
<input type="text" required pattern="\S+@\S+\.\S+" title="Enter correct email">
```

Атрибут **required** говорит о том, что поле должно быть заполнено, хоть чем-то.

Атрибут **pattern** задаёт регулярное выражение которым будет проверены введённые данные.

Атрибут **title**, помимо основного предназначения, будет выступать подсказкой при ошибке валидации поля.



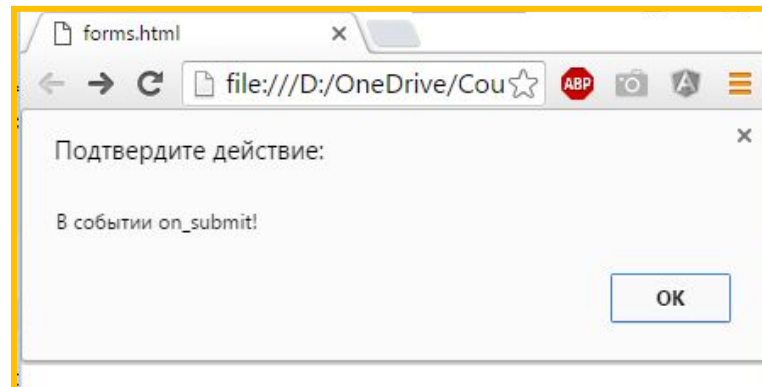
The screenshot shows a web form with the following elements:

- A label "Email address" above the input field.
- An input field containing the text "la@la".
- A message "Fill required correct email!" below the input field.
- A blue button labeled "Send data" below the input field.
- A validation error message box with a yellow warning icon, containing the text "Виберіть потрібний формат. Enter correct email".

# У форм есть событие onsubmit

*Которое случается после нажатия кнопки отправки данных, но до самой отправки, обработав его, можно проверить данные и при необходимости отменить отправку.*

```
3 <script>
4   window.onload = function() {
5       var form = document.querySelector("form");
6       form.onsubmit = function(event_info) {
7           alert("В событии on_submit!");
8       }
9   }
10 </script>
```



# Регулярные выражения

Короткие записи популярных символьных классов

Короткая запись	Полная запись	Описание
<code>\s</code>	<code>[\f\t\n\r]</code>	Пробельный символ
<code>\S</code>	<code>[^\f\t\n\r]</code>	Любой символ, кроме пробельного
<code>\d</code>	<code>[0-9]</code>	Цифра
<code>\D</code>	<code>[^0-9]</code>	Любой символ, кроме цифры
<code>\w</code>	<code>[a-zA-Z0-9]</code>	Латиница или цифра
<code>\W</code>	<code>[^a-zA-Z0-9]</code>	Любой символ, кроме латиницы или цифры
<code>.</code>	<code>[^\n\r]</code>	Любой символ, кроме перевода строки
<code>\b</code>		Граница слова
<code>\B</code>		Не граница слова
<code>\A</code>		Начало строки
<code>\Z</code>		Конец строки

*Регулярное выражение – шаблон которым проверяется строка, строка может соответствовать шаблону, а может не соответствовать.*

# Валидация данных средствами JavaScript

# Алгоритм

## валидации:

1. Подписываемся на событие отправки формы (*onsubmit*);
2. В обработчике событие проверяем **данные** на соответствие;
3. Если данные не корректны сообщаем об это пользователю и **отменяет отправку данных**;
4. Если данные корректны, не мешаем браузеру отправлять данные.

## У каждого элемента `input` есть свойство `value`

С помощью которого можно получить (или установить), содержимое элемента ввода.

## Отправку формы можно отменить

Для этого достаточно вызвать метод `.preventDefault()` которая есть у объекта содержащего информацию о событии (который передаётся первым параметром в функции обработчике события) или **просто вернуть `false`** из функции обработчика события.

```
10 window.onload = function(){
11
12     var form = document.querySelector("form");
13
14     form.onsubmit = function(e){
15         e.preventDefault();
16         alert("Отправка данных отменена.");
17     }
18
19 }
```

# Валидация данных

```
8 <script>
9
10 window.onload = function(){
11
12     var form = document.querySelector("form");
13
14     form.onsubmit = function(e){
15
16         var name     = user_name_id.value;
17         var email    = user_email_id.value;
18
19         var problem = false;
20
21         if(name.length < 1){
22             problem = true;
23             user_name_id.classList.add("form-control-danger");
24             user_name_id.parentNode.classList.add("has-danger");
25         }else{
26             user_name_id.classList.remove("form-control-danger");
27             user_name_id.parentNode.classList.add("has-danger");
28         }
29
30         if(!email.match(/\S+@\S+\.\S+/)){
31             problem = true;
32             user_email_id.classList.add("form-control-danger");
33             user_email_id.parentNode.classList.add("has-danger");
34         }else{
35             user_email_id.classList.remove("form-control-danger");
36             user_email_id.parentNode.classList.remove("has-danger");
37         }
38
39         if(problem){
40             e.preventDefault();
41         }
42     }
43 }
44
45 </script>
```



# Элементы ввода и интерактивность



# Элементы ввода и ИНТЕРАКТИВНОСТЬ

## Bootstrap From Example

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse volutpat dapibus arcu, sed tristique ipsum euismod blandit. Vestibulum commodo tortor laoreet tortor blandit volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur molestie auctor egestas. Quisque erat erat, dignissim ac massa et, semper faucibus erat. Proin sagittis nibh non porta aliquet. Ut egestas tempus dolor vitae imperdiet. Aenean hendrerit vulputate ipsum, eu ultrices elit sodales sit amet. Pellentesque imperdiet justo a enim pulvinar, ut pulvinar risus ultrices. Aliquam erat volutpat. Cras posuere, nisl ac maximus lacinia, eros lectus aliquam lorem, sit amet congue massa velit nec justo.



Скачайте заготовку в

*Notepad++*

<http://files.courses.dp.ua/web/15/ex02.html>


# Элементы ввода и интерактивность

```
8 <script>
9
10 window.onload = function(){
11
12     sizer.oninput = function(e){
13         var size = parseInt(sizer.value);
14         var text = document.querySelector("p");
15         text.style.fontSize = size + "pt";
16     }
17 }
18
19 </script>
```

*У элементов ввода есть события отвечающие за изменения данных в них (например когда пользователь перемещает ползунок, или вводит посимвольно фразу). Одно из них **oninput**, подписавшись на которое мы можем в динамике отслеживать*

# Элементы ввода и интерактивность

Bootstrap From Example

—  —  
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Suspendisse volutpat dapibus  
arcu, sed tristique ipsum  
euismod blandit. Vestibulum  
commodo tortor laoreet tortor

*Скрипт отслеживает изменение «ползунка» и пропорционально изменяет размер шрифта для текста.*

# Домашнее задание

# Немного практики

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>DOM.task1</title>
5 <script>
6   window.onload = function(){
7     //Начало вашего кода.....
8
9     //.....Конец вашего кода
10  }
11 </script>
12 </head>
13 <body>
14 <h1>Каталог телефонов <button>Отсортировать</button></h1>
15   <div id="phone_list">
16     <div>iPhone 3Gs <span>582$</span></div>
17     <div>Lenovo <span>997$</span></div>
18     <div>HTC <span>698$</span></div>
19     <div>Samsung <span>242$</span></div>
20     <div>Fly <span>793$</span></div>
21     <div>Asus <span>771$</span></div>
22     <div>Nokia <span>108$</span></div>
23     <div>LG <span>214$</span></div>
24     <div>IPhone 5s <span>216$</span></div>
25     <div>IPhone 6+ <span>326$</span></div>
26     <div>IPhone 6s+ <span>799$</span></div>
27   </div>
28 </body>
29 </html>
```

## Каталог телефонов

iPhone 3Gs 582\$  
Lenovo 997\$  
HTC 698\$  
Samsung 242\$  
Fly 793\$  
Asus 771\$  
Nokia 108\$  
LG 214\$  
IPhone 5s 216\$  
IPhone 6+ 326\$  
IPhone 6s+ 799\$



*По нажатию на кнопку необходимо отсортировать список по возрастанию цены. При повторном нажатии сортировка должна выполняться по убыванию цены. И так по очереди переключать направление сортировки.*

**Скопируйте код:**

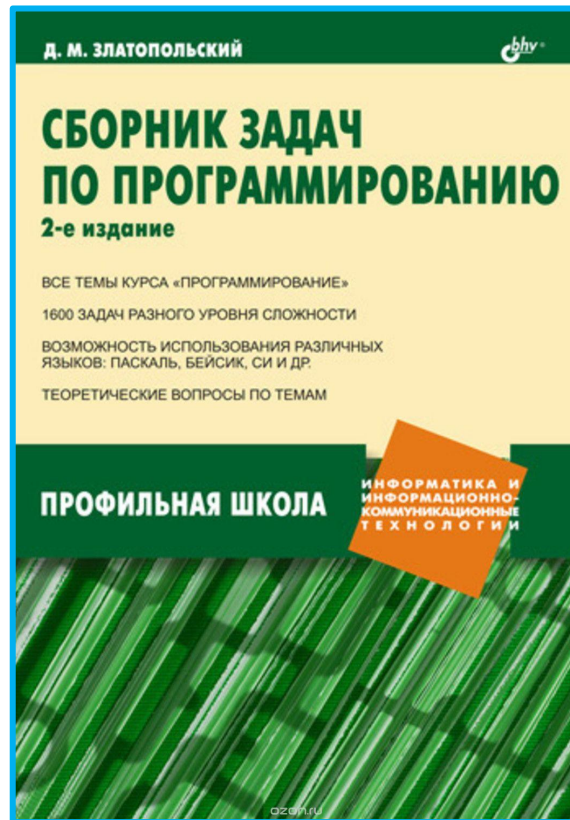
<http://courses.dp.ua/files/is/tasks/task1.html>

# Где брать задачи для тренировки?

*Чтобы научиться программировать – нужно тренироваться...*

*Чтобы тренироваться нужны задачи...*

*Чтобы были задачи нужно уметь программировать...*



**Д. М. Златопольский**  
**Сборник задач по программированию**