

# 10 Списки

## 10.1 Введение в списки

Язык HTML предлагает несколько механизмов создания списков информации. В каждом списке должен быть один или несколько элементов списков. Списки могут содержать:

- Неупорядоченную информацию.
- Упорядоченную информацию.
- Определения.

Предыдущий список, например, не упорядочен, он создан с помощью элемента [UL](#):

```
<UL>
<LI>Неупорядоченную информацию.
<LI>Упорядоченную информацию.
<LI>Определения.
</UL>
```

Упорядоченный список, создаваемый с помощью элемента [OL](#), может содержать информацию, в которой важен порядок, например, рецепт:

1. Тщательно смешать сухие ингредиенты.
2. Влить жидкость.
3. Смешивать 10 минут.
4. Выпекать в течение часа при температуре 300 градусов.

Списки определений, создаваемые с помощью элемента [DL](#), могут содержать ряд пар термин/определение (хотя списки определений могут иметь и иные применения). Например, список определений можно использовать в рекламе изделия:

### **Низкая цена**

Новая модель этого изделия существенно дешевле предыдущей!

### **Проще работа**

Мы изменили изделие, так что с ним теперь легко работать!

### **Безопасно для детей**

Вы можете оставить детей в комнате, и изделие не причинит им вреда (не гарантируется).

На языке HTML он определяется следующим образом:

```
<DL>
<DT><STRONG>Низкая цена</STRONG>
<DD> Новая модель этого изделия существенно дешевле предыдущей!
<DT><STRONG>Проще работа</STRONG>
<DD>Мы изменили изделие, так что с ним теперь легко работать!
<DT><STRONG> Безопасно для детей </STRONG>
<DD> Вы можете оставить своих детей в комнате, и изделие не причинит им вреда (не га-
рантируется) .
</DL>
```

Списки могут быть вложенными, разные типы списков можно использовать вместе, как в следующем примере, где список определений содержит неупорядоченный список (ингредиенты) и упорядоченный список (процедуру):

### **Ингредиенты:**

- 100 г муки
- 10 г сахара
- 1 стакан воды
- 2 яйца
- соль, перец

### **Процедура:**

1. Тщательно смешайте сухие ингредиенты.
2. Влейте жидкие ингредиенты.
3. Смешивайте 10 минут.
4. Выпекайте в течение часа при температуре 300 градусов.

## Примечания:

Можно добавить изюм.

Точное представление трех типов списков зависит от агента пользователя. Не стоит использовать списки для создания отступов в тексте. Это делают с помощью таблиц стилей.

## 10.2 Неупорядоченные списки ([UL](#)), упорядоченные списки ([OL](#)) и элементы списков ([LI](#))

```
<!ELEMENT UL - - (LI)+ -- неупорядоченный список -->
<!ATTLIST UL
  %attrs; -- %coreattrs, %i18n, %events --
>
<!ELEMENT OL - - (LI)+ -- упорядоченный список -->
<!ATTLIST OL
  %attrs; -- %coreattrs, %i18n, %events --
>
```

Начальный тег: **обязателен**, Конечный тег: **обязателен**

```
<!ELEMENT LI - O (%flow)* -- элемент списка -->
<!ATTLIST LI
  %attrs; -- %coreattrs, %i18n, %events --
>
```

Начальный тег: **обязателен**, Конечный тег: **не обязателен**

Определения атрибутов

type = информация о стиле [\[CI\]](#)

**Нежелателен.** Этот атрибут устанавливает стиль элемента списка. Доступные в настоящее время значения предназначены для визуальных агентов пользователей. [Возможные значения](#) описаны ниже (включая информацию о регистре).

start = [число](#) [\[CN\]](#)

**Нежелателен.** Только для [OL](#). Этот атрибут задает начальный номер первого элемента в упорядоченном списке. По умолчанию начальный номер - "1". Помните, что, хотя значением этого атрибута является целое число, соответствующая метка может быть нецифровая. Если в качестве стиля выбраны латинские буквы верхнего регистра (A, B, C, ...), start=3 означает "C". Если в качестве стиля выбраны римские цифры нижнего регистра, start=3 означает "iii" и т.д.

value = [число](#) [\[CN\]](#)

**Нежелательно.** Только для [LI](#). Этот атрибут устанавливает номер текущего элемента списка. Помните, что, хотя значением атрибута является целое число, соответствующая метка может быть нечисловая (см. атрибут [start](#)).

compact [\[CI\]](#)

**Нежелателен.** Если этот логический атрибут установлен, он сообщает визуальным агентам пользователей о том, что генерировать список нужно более компактно. Интерпретация этого атрибута зависит от агента пользователя.

Атрибуты, определяемые в любом другом месте

- [id](#), [class](#) ([идентификаторы в пределах документа](#))
- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))
- [title](#) ([заголовок элемента](#))
- [style](#) ([встроенная информация о стиле](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([внутренние события](#))

Упорядоченные и неупорядоченные списки генерируются одинаково за исключением того, что визуальные агенты пользователей нумеруют упорядоченные списки. Агенты поль-

завателей могут представлять эти номера несколькими способами. Элементы неупорядоченного списка не нумеруются.

Оба эти типа списков состоят из последовательностей элементов списков, определяемых элементом LI (конечный тег которого можно опустить).

В этом примере показана общая структура списка.

```
<UL>
  <LI> ... первый элемент списка...
  <LI> ... второй элемент списка...
  ...
</UL>
```

Списки могут быть вложенными. ПРИМЕР НЕЖЕЛАТЕЛЬНОГО ИСПОЛЬЗОВАНИЯ:

```
<UL>
  <LI> ... Уровень один, номер один...
  <OL>
    <LI> ... Уровень два, номер один...
    <LI> ... Уровень два, номер два...
    <OL start="10">
      <LI> ... Уровень три, номер один...
    </OL>
  </OL>
  <LI> ... Уровень два, номер три...
</OL>
<LI> ... Уровень один, номер два...
</UL>
```

**Информация о порядке номеров.** В упорядоченных списках невозможно продолжать нумерацию автоматически из предыдущего списка или убрать нумерацию для некоторых элементов. Однако авторы могут пропустить несколько элементов списка, установив для них атрибут `value`. Нумерация для последующих элементов списка продолжается с нового значения. Например:

```
<ol>
<li value="30"> элемент списка номер 30.
<li value="40"> элемент списка номер 40.
<li> элемент списка номер 41.
</ol>
```

## 10.3 Списки определений: элементы DL, DT и DD

```
<!-- списки определений - DT - термин, DD - его определение -->

<!ELEMENT DL - - (DT|DD)+ -- список определений -->
<!ATTLIST DL
  %attrs; -- %coreattrs, %i18n, %events --
>
```

**Начальный тег: *обязателен*, Конечный тег: *обязателен***

```
<!ELEMENT DT - O (%inline;)* -- термин -->
<!ELEMENT DD - O (%flow;)* -- определение -->
<!ATTLIST (DT|DD)
  %attrs; -- %coreattrs, %i18n, %events --
>
```

**Начальные тег: *обязателен*, Конечный тег: *не обязателен***

*Атрибуты, определяемые в любом другом месте*

- [id](#), [class](#) (идентификаторы в пределах документа)
- [lang](#) (информация о языке), [dir](#) (направление текста)
- [title](#) (заголовок элемента)
- [style](#) (встроенная информация о стиле)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (внутренние собы-

[тия](#))

Списки определений незначительно отличаются от других типов списков - тем, что элементы состоят из двух частей: термина и определения. Термин обозначается с помощью элемента [DT](#) и может иметь только встроенное содержимое. Описание указывается с помощью элемента [DD](#), имеющим содержимое уровня блока. Пример:

```
<DL>
  <DT>Dweeb
  <DD>young excitable person who may mature
    into a <EM>Nerd</EM> or <EM>Geek</EM>

  <DT>Cracker
  <DD>hacker on the Internet

  <DT>Nerd
  <DD>male so into the Net that he forgets
    his wife's birthday
</DL>
```

Вот пример с несколькими терминами и определениями:

```
<DL>
  <DT>Center
  <DT>Centre
  <DD> A point equidistant from all points
    on the surface of a sphere.
  <DD> In some field sports, the player who
    holds the middle position on the field, court,
    or forward line.
</DL>
```

Другим применением элемента [DL](#), например, может быть разметка диалогов, где каждый элемент [DT](#) означает говорящего, а в каждом элементе [DD](#) содержатся его слова.

### 10.3.1 Визуальное отображение списков

*Примечание.* Ниже приводится описание поведения некоторых имеющихся на настоящий момент визуальных агентов пользователей при форматировании списков. Таблицы стилей предоставляют большие возможности управления форматированием (например, в отношении нумерации, соглашений, используемых в разных языках, отступов и т.д.).

Визуальные агенты пользователей обычно сдвигают вложенные списки соответственно уровню вложенности. Для элементов [OL](#) и [UL](#) атрибут [type](#) определяет параметры генерации для визуальных агентов пользователей. Для элемента [UL](#) возможными значениями атрибута [type](#) являются `disc`, `square` и `circle`. Значение, используемое по умолчанию, зависит от уровня вложенности текущего списка. Эти значения не учитывают регистр. Представление каждого значения зависит от агента пользователя. Агенты пользователей должны пытаться представлять "disc" в виде небольшого заполненного кружка, "circle" - в виде окружности, а "square" в виде небольшого квадрата.

Графические агенты пользователь могут генерировать их как:

- для значения "disc"
- для значения "circle"
- для значения "square"

Для элемента [OL](#) возможные значения атрибута [type](#) приведены в следующей таблице (они учитывают регистр):

Type	Стиль нумерации	
1	арабские цифры	1, 2, 3, ...
a	буквы нижнего регистра	a, b, c, ...
A	буквы верхнего регистра	A, B, C, ...

i	римские цифры в нижнем регистре	i, ii, iii, ...
I	римские цифры в верхнем регистре	I, II, III, ...

Помните, что использование атрибута `type` нежелательно, и стили списков должны определяться с помощью таблиц стилей. Например, с помощью CSS можно указать, что стиль нумерации для элементов списка в нумерованном списке - римские цифры нижнего регистра. В приведенном ниже примере каждый элемент `OL`, принадлежащий классу "withroman", обозначается римской цифрой.

```
<STYLE type="text/css">
OL.withroman { list-style-type: lower-roman }
</STYLE>
<BODY>
<OL class="withroman">
<LI> Шаг один ...
<LI> Шаг два ...
</OL>
</BODY>
```

Генерация списка определений также зависит от агента пользователя. Например, список:

```
<DL>
  <DT>Dweeb
  <DD>young excitable person who may mature
    into a <EM>Nerd</EM> or <EM>Geek</EM>

  <DT>Cracker
  <DD>hacker on the Internet

  <DT>Nerd
  <DD>male so into the Net that he forgets
    his wife's birthday
</DL>
```

может генерироваться следующим образом:

```
Dweeb
  young excitable person who may mature into a Nerd or Geek
Cracker
  hacker on the Internet
Nerd
  male so into the Net that he forgets his wife's birthday
```

## 10.4 Элементы DIR и MENU

Использование элементов `DIR` и `MENU` нежелательно.

Формальное определение см. в [Переходном DTD](#).

*Атрибуты, определяемые в любом другом месте*

- [id](#), [class](#) (идентификаторы в пределах документа)
- [lang](#) (информация о языке), [dir](#) (направление текста)
- [title](#) (заголовок элемента)
- [style](#) (встроенная информация о стиле)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (внутренние события)

Элемент `DIR` предназначался для создания многостраничных списков каталогов. Элемент `MENU` предназначался для использования в списках меню, состоящих из одного столбца.

Оба эти элемента имеют ту же структуру, что и элемент `UL`, различаясь только при генерации. На практике агенты пользователей генерируют списки `DIR` или `MENU` точно так же, как список `UL`.

Настоятельно рекомендуется использовать вместо этих элементов элемент `UL`.