



[Главная](#) » [HTML и HTML5](#) » [1.11. Семантические элементы HTML5](#) »

[1.11.11. Интерактивные элементы](#)

1.11.11. Интерактивные элементы

Опубликовано: 29 ноября 2021

Обновлено: 27 января 2022

4



Интерактивное содержимое представлено элементами, предназначенными для взаимодействия с пользователем.

Некоторые из этих элементов позволяют создавать виджеты — небольшие графические модули, решающие отдельные задачи.

HTML-элементы для создания виджетов

СОДЕРЖАНИЕ:

1. Элемент `<details>`
2. Элемент `<summary>`
3. Элемент `<dialog>`

1. Элемент `<details>`

Категории контента: потоковое содержимое, корневое секционное содержимое, интерактивное содержимое, видимое содержимое.



Объявление скрыто

Мы используем ваши ответы, чтобы подбирать для вас подходящую рекламу

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны [глобальные атрибуты](#), а также атрибут `open`, который делает дополнительную информацию видимой при загрузке страницы.

Элемент `<details>` представляет виджет раскрытия информации, который используется, чтобы показать или скрыть дополнительную информацию в конкретной области интерфейса. Не подходит для примечаний.

Если элемент `<details>` имеет дочерний элемент `<summary>`, то первый дочерний элемент `<summary>` представляет собой надпись на элементе. Если дочерний элемент `<summary>` отсутствует, браузер предоставляет свою собственную надпись (например, на английском «Details» или русском «Детали»).

HTML

```
<details>
  <summary><label for="fn">Name & Extension:</label></summary>
  <p><input type="text" id="fn" name="fn" value="Pillar Magazine.pdf">
  <p><label><input type="checkbox" name="ext" checked> Hide extension</label>
</details>
```

Примеры стилизации

▶ Name & Extension:

Согласно новой спецификации, Firefox отображает элемент `<summary>` как `display: list-item` и маркер можно стилизовать так же, как и элементы списка.

Согласно старой спецификации, в Chrome and Safari существует пользовательский псевдо-элемент `::-webkit-details-marker`, с помощью которого можно стилизовать маркер.



Для кроссбраузерной стилизации маркера, чтобы скрыть дефолтный и добавить пользовательский, необходимо для Firefox установить элементу `summary {display: block;}`, а для Chrome и Safari's установить `::-webkit-details-marker {display: none;}`. После этого дефолтный маркер будет скрыт, и можно установить свой маркер любым доступным способом стилизации, например:

CSS

```
details {
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #D0D4DC;
  margin-bottom: 20px;
}
summary {
  display: block;
  position: relative;
  padding-left: 25px;
  line-height: 18px;
}
summary::-webkit-details-marker {
  display: none;
}
summary::before {
  content: "";
  position: absolute;
  left: 0;
  width: 0;
  height: 0;
  border-top: 9px solid transparent;
  border-left: 14px solid;
  border-bottom: 9px solid transparent;
}
details[open] > summary::before {
  content: "";
  width: 0;
  height: 0;
  top: 2px;
  border-left: 9px solid transparent;
  border-right: 9px solid transparent;
  border-top: 14px solid;
}
details p {
  margin: 20px 0 0 0;
}
```

Поскольку атрибут `open` добавляется и удаляется автоматически, когда пользователь взаимодействует с элементом управления, его можно стилизовать по-разному в зависимости от состояния, например, можно плавно менять цвет надписи при открытии или закрытии элемента:



```
details > summary {  
  transition: color 1s;  
  color: black;  
}  
  
details[open] > summary {  
  color: red;  
}
```

2. Элемент <summary>

Категории контента: отсутствует.

Контекст, в котором этот элемент может быть использован: как первый дочерний элемент элемента <details> .

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны [глобальные атрибуты](#).

Элемент <summary> представляет заголовок или небольшое пояснение для остального содержимого родительского элемента <details> . Щелчок по элементу <summary> переключает состояние родительского элемента в открытое и закрытое состояния.

3. Элемент <dialog>

Поддержка браузерами

“

*IE: —
Edge: 79
Firefox: —
Chrome: 37
Safari: —
Opera: 24
iOS Safari: —
Opera Mini: —
Android Browser: 96
Chrome for Android: 96*

Категории контента: потоковое содержимое, корневое секционное содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны [глобальные атрибуты](#), а также атрибут `open` , который указывает, что элемент виден и пользователь может с ним взаимодействовать. Атрибут `tabindex` не должен использоваться с элементом <dialog> .



Элемент `<dialog>` представляет примитив графического интерфейса — диалоговое окно, с которым пользователь взаимодействует для выполнения конкретных задач.

Элемент `<dialog>` с помощью атрибута `method="dialog"` может интегрироваться с элементом `<form>`. При отправке формы диалоговое окно будет закрыто с `returnValue` равным `value` нажатой кнопки `submit`.

Для стилизации фона подложки элемента может быть использован псевдо-элемент `::backdrop`:

CSS

```
/* Подложка показывается только тогда, когда диалоговое окно открыто с помощью метода dialog.showModal()
dialog::backdrop {
  background: rgba(255,0,0,.25);
}
```

HTML

```
<dialog id="favDialog">
  <form method="dialog">
    <p>
      <label for="favAnimal">Любимые животные:</label>
      <select id="favAnimal">
        <option></option>
        <option>Вельш-корги</option>
        <option>Красная панда</option>
        <option>Рэгдолл</option>
      </select>
    </p>
    <p>
      <button id="cancel" type="reset">Отменить</button>
      <button type="submit">Отправить</button>
    </p>
  </form>
</dialog>
```

По материалам [Interactive elements](#)

< Формы

Скрипты >

Поделиться:

