

Оглавление

Data – атрибуты app.js	2
data-action-click	2
data-action-focusout	2
data-js-init	2
«Режим отладки» (логирование)	3
Функции	3
app.init()	3
app.developer(action, func, text, log_status = true)	3
app.data()	3
• data-action-click	3
• bootstrap tooltip	4
• data-action-focusout	4
• data-js-init	4
app.click(obj, func)	4
app.focus(obj, func)	5
app.focusOut(obj, func)	5
setLocation(curLoc)	5
print_result(wrapper, text, method = 'html_')	6
addPostPage	6
addPostPage.init()	6
addPostPage.selected(selectors = [], selector_themes = [])	6

app.js

app.js - основной скрипт, через него запускаются события ([click](#), [focusout](#), [focus](#)), а также вызовы функций через **data** - атрибуты и т.д. с целью контроля логики и происходящих событий скриптов (для того, что бы отследить события, необходимо установить **log_status = true** в функции [app.developer\(\)](#)). При загрузке документа происходит запуск [app.init\(\)](#), в нем записаны все вызовы, которые необходимо выполнить после загрузки страницы.

Data – атрибуты app.js

*Поиск и инициализация всех указанных ниже **data**-атрибутов происходит в [app.data\(\)](#).*

[data-action-click](#) – событие клика, параметром передается ключ объекта для вызова функции из **actions.js - click**.

Пример:

```
<a data-action-click="name_func"> CLICK ME </a> ...
```

Вызовет **click.name_func(obj)** из **actions.js**

[data-action-focusout](#) – событие при потере фокуса с объекта, параметром передается ключ объекта для вызова функции из **actions.js - focusOut**.

Пример:

```
<input data-action-focusout="name_func" /> ...
```

Вызовет **focusOut.name_func(obj)** из **actions.js**

[data-js-init](#) – вызов функции на странице, параметром передается имя объекта.

Пример:

app.js

```
<div data-js-init = "name_func" /> ...
```

Вызовет **name_func.init(obj)**.

«Режим отладки» (логирование)

*В данном режиме логируются все события скрипта. Для активации режима необходимо в функции **app.developer()** установить значение **true** для параметра **log_status**. Для подключения события к логированию, смотрите описание [app.developer\(\)](#).*

Функции

app.init() – вызовы функций после загрузки страницы.

app.developer(action, func, text, log_status = true) – «режим отладки». При условии **log_status = true** в консоли логируются все события и инициализации с целью отладки документа.

Параметры:

- **action** – событие, которое выполняется.
- **func** – имя функции из которой производится инициализация.
- **text** – дополнительное описание.
- **log_status** – значение *false* исключает из логирования, *true* – логирует.

app.data() – инициализация скриптов и событий согласно *data* – атрибутов.

- **data-action-click** – записывает в DOM событие клика по объекту. Параметром передает объект, на котором произошло событие. В качестве значения передается имя функции (см. [data - атрибуты](#)), которую необходимо выполнить. Данный метод реализован с целью создания общего шаблона события, что

app.js

позволяет уменьшить количество кода, а также для того, чтоб не упустить ни одно событие клика из логирования при активном [режиме разработчика](#).

Пример:

```
<a data-action-click='showTopMenu'> Link </a> ...
```

при клике инициализирует click.showTopMenu() из actions.js.

- [bootstrap tooltip](#) – инициализация всплывающих подсказок (Bootstrap).
- [data-action-focusout](#) – записывает в DOM событие потери фокуса с объекта. Параметром передает объект, на котором произошло событие.

Пример:

```
<input data-action-focusout='name_func' /> ...
```

при клике инициализирует focusOut.name_func() из actions.js.

- [data-js-init](#) – инициализирует функцию ***.init()** по [data](#) - атрибуту в документе.

Пример:

```
<div data-js-init="plugin"> ...
```

Инициализирует **plugin.init()** после загрузки страницы.

[app.click\(obj, func\)](#) – записывает в DOM событие клика по объекту. Параметром передает объект, на котором произошло событие.

Параметры:

- **obj** – объект, по которому выполняется событие.
- **func** – имя функции, которая запустится по событию клика.

Пример:

```
app.click("#block", "myFunc")...
```

при клике по \$('#block') вызовет myFunc.init(obj).

app.js

`app.focus(obj, func)` – записывает в DOM событие при фокусе на объект. Параметром передает объект, на котором произошло событие.

Параметры:

- **obj** – объект, по которому выполняется событие.
- **func** – имя функции, которая запустится по событию фокуса.

Пример:

```
app.focus("input#name", "myFunc")...
```

при фокусе на `$('input#name')` вызовет `myFunc.init(obj)`.

`app.focusOut(obj, func)` – записывает в DOM событие при потере фокуса с объекта. Параметром передает объект, на котором произошло событие.

Параметры:

- **obj** – объект, по которому выполняется событие.
- **func** – имя функции, которая запустится по событию потери фокуса.

Пример:

```
app.focusOut("input#name", "myFunc")...
```

при потере фокуса с `$('input#name')` вызовет `myFunc.init(obj)`.

`setLocation(curLoc)` – устанавливает url при ajax, вырезая из строки «ajax/», используется для фильтров на странице.

Параметры:

- **curLoc**– строка, которую необходимо вставить в адресную строку.

Пример:

```
Var url = '/ajax/page/id4'
```

```
setLocation(url')...
```

в адресной строке установит `https://site.com/page/id4`.

app.js

`print_result(wrapper, text, method = 'html_')` – функция вывода результата (разметки, текста) в указанный блок. Используется в ајах запросах. Данная функция используется, как шаблон, с целью контроля при активном режиме отладки. Функция выполняет: `append`, `html`, `text`.

Параметры:

- **wrapper** – блок для вывода результата.
- **text** – текст или разметка для вставки.
- **method** – `html_` (метод `.html()`), `text_` (метод `.text()`), `append` (метод `.append()`).

Пример:

```
var text = '<a href="#"> Link </a>';  
print_result('#wrapper', text, 'append');  
добавит '<a href="#"> Link </a>' в блок $('#wrapper') методом .append().
```

addPostPage

Выполняется на странице добавления\редактирования публикации. Загружается по атрибуту [data-js-init](#) в документе.

`addPostPage.init()` – выполняется при загрузки страницы.

`addPostPage.selected(selectors = [], selector_themes = [])` – событие **change** внутри `select` (наука); для пары `select` (наука и тема науки). При выборе науки вызывает **`select_science.init()`** из **`actions.js`** – ајах запрос для получения перечня тем выбранной науки, результат выводится во второй `select` пары (тема науки).

Параметры:

- **selectors[]** – Массив классов или объектов `select` – науки (`['.science-1', 'input[name="science-2"]']`).

app.js

- **Selector_themes[]** – Массив классов или объектов select – темы науки (['.scienceThemes-1', 'input[name="scienceThemes-2"]']).