Передача параметров из HTML в JavaScript

Часто при инициализации JS-компонента нужно найти определенный DOM-элемент с которым компонент будет работать, а также достать из HTML параметры (настройки). О способах хранения и передачи в JS этих параметров я и хотел бы поговорить.

Раньше я решал эту задачу так: простые параметры передавал через классы, сложные — через дополнительные (не всегда валидные) атрибуты или дочерние элементы.

Предположим, у нас есть DOM-элемент:

<span id="test">тест</span>

При клике на этот элемент мы хотим показывать некий попап. Для показа попапа нужно задать следующие параметры: ширина, высота, таскаемость попапа и текст. Раньше я написал бы примерно следующее:

<span id="test" class="width\_250 height\_150 draggable"><span class="text">Трам-парам!</span>тест</span>

Потом я бы долго и муторно получал значения этих параметров путем выпарсивания их из классов и доставания из дочерних элементов.

Так бы и мучился дальше, если бы мой коллега [Дима Филатов](http://dfilatov.blogspot.com/) не подсказал гораздо более элегантное решение — прописать атрибут onclick в котором возвращать объект с настройками.

<span id="test" onclick="return {width: 250, height: 150, draggable: true, text: 'Трам-парам!'};">тест</span>

Как достать этот объект:

var element = document.getElementById('test');

// Забираем параметры

var options = element.onclick instanceof Function ? element.onclick() : {};

// Заметаем следы - обнуляем onclick

element.onclick = null;

element.removeAttribute('onclick');

// Навешиваем обработчик onclick для показа попапа

element.onclick = function () {

showPopup(options);

};

По сути, для передачи настроек используется формат JSON в котором есть шесть типов значений: объект, массив, строка, число, булевское значение и null. Используя эти типы, можно передавать довольно сложную структуру данных, немыслимую для передачи через классы.

По-моему, очень круто. Блин, почему я сам не додумался до этого? :)

**Передача параметров JS в HTML**

Добрый день. Такой вопрос. У меня в файле JS есть массив массивов со значениями стилей. Как в HTML-документе div-ам присваивать значения из этого массива.(То есть ссылаться из JS файла в HTML). На переменную upDateTime не обращаем внимание. Интересно, как присвоить цвет диву, и высоту заданную в процентах.

|  |
| --- |
| <div align="center" id="red"></div> |

[показать чистый исходник в новом окне](http://javascript.ru/forum/#viewSource)[Скрыть/показать номера строк](http://javascript.ru/forum/#noList)[печать кода с сохранением подсветки](http://javascript.ru/forum/#printSource)

|  |  |  |
| --- | --- | --- |
| 01 | var params = { | |
| 02 | lines: [ |

|  |  |
| --- | --- |
| 03 | { |
| 04 | background: '#00F', | |

|  |  |  |
| --- | --- | --- |
| 05 | updateTime: 1000, | |
| 06 | elements: [{ |

|  |  |  |
| --- | --- | --- |
| 07 | background: 'yellow', | |
| 08 | width: 25 |

|  |  |
| --- | --- |
| 09 | }, |
| 10 | { | |

|  |  |  |
| --- | --- | --- |
| 11 | background: 'green', | |
| 12 | width: 50 |

|  |  |  |
| --- | --- | --- |
| 13 | }, | |
| 14 | { |

|  |  |  |
| --- | --- | --- |
| 15 | background: 'red', | |
| 16 | width: 25 |

|  |  |  |
| --- | --- | --- |
| 17 | } | |
| 18 | ] |

|  |  |
| --- | --- |
| 19 | }, |
| 20 | { | |

|  |  |  |
| --- | --- | --- |
| 21 | background: 'yellow', | |
| 22 | updateTime: 1000, |

|  |  |
| --- | --- |
| 23 | elements: [{ |
| 24 | background: '#00F', | |

|  |  |  |
| --- | --- | --- |
| 25 | width: 25 | |
| 26 | }, |

|  |  |
| --- | --- |
| 27 | { |
| 28 | background: '#00F', | |

|  |  |  |
| --- | --- | --- |
| 29 | width: 50 | |
| 30 | }, |

|  |  |
| --- | --- |
| 31 | { |
| 32 | background: '#00F', | |

|  |  |  |
| --- | --- | --- |
| 33 | width: 25 | |
| 34 | }] |

|  |  |
| --- | --- |
| 35 | } |
| 36 | , |

|  |  |
| --- | --- |
| 37 | { |
| 38 | background: '#00F', | |

|  |  |  |
| --- | --- | --- |
| 39 | updateTime: 1000, | |
| 40 | elements: [{ |

|  |  |  |
| --- | --- | --- |
| 41 | background: '#00F', | |
| 42 | width: 25 |

|  |  |
| --- | --- |
| 43 | }, |
| 44 | { | |

|  |  |  |
| --- | --- | --- |
| 45 | background: '#00F', | |
| 46 | width: 50 |

|  |  |  |
| --- | --- | --- |
| 47 | }, | |
| 48 | { |

|  |  |  |
| --- | --- | --- |
| 49 | background: '#00F', | |
| 50 | width: 25 |

|  |  |  |
| --- | --- | --- |
| 51 | }] | |
| 52 | } |

|  |  |
| --- | --- |
| 53 | ] |
| 54 | } |

|  |  |
| --- | --- |
| 55 | var a = params.lines.length; |
| 56 | lines: [{background: 'red'}] | |

|  |  |
| --- | --- |
| 57 | function createDiv1(){ |
| 58 | for (var i = 0; i < params.lines.length; i++){ | |

|  |  |
| --- | --- |
| 59 |  |
| 60 | } | |

|  |  |
| --- | --- |
| 61 | } |
| 62 | document.getElementById('red').style.height = 100/a; | |

|  |  |
| --- | --- |
| 63 | document.getElementById('blue').style.height = 100/apx; |
| 64 | document.getElementById('green').style.height = 100/apx; | |

|  |  |
| --- | --- |
| 65 |  |
| 66 | document.getElementById('red').style.backgroundColor = params.lines[0].background; | |

# Передача переменной из html в javascript

В процессе разработке сайтов часто появляется потребность передать какое-либо значение (переменную, массив) в javascript,в таких случаях обычно советуют что то вроде

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1  2  3  4 | <script>  var variable = <?php $variable ;?>  alert(variable);  <script> |

Javascript записанный инлайном вызывает смешанные чувства.

Во первых это почти всегда не по кодстайлу.

Во вторых трудно отлаживать.

В третьих Такой код не кушируется браузером.

Внимание дорогой читатель  пример выше **неправильный более чем полностью**, о чем предупреждаю сразу и это как раз то как делать не нужно, а вот о том как правильно вдумчиво и внимательно можно **прочитать ниже**.

Но если вы к примеру не человек высоких моральными принципов, можете смело копировать и делать с этим  вообще  что угодно, я вам этого не советовал и  более того даже отговаривал.

Я убежден что javascript должен храниться исключительно в файлах,  
html дабы обеспечить нормальное кеширование браузерами, и облегчить поиск этого самого скрипта в своем проекте.

Но как же передать значение из php и при этом сохранить красивый код?

Ответ использовать eval() в Javascript.

Рассмотрим пример с подключенимм google maps.

Для начала подключим google maps (библиотеку)

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1 | <script src='<https://maps.googleapis.com/maps/api/js?key=AIzaSyCdrLjcwRWoPkHw8bTvGD2gtNCn_6brs0w&amp;sensor=true&amp;language=ru>'></script> |

инициализируем начало отрисовки карт сразу после загрузки DOM

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | var $ = jQuery.noConflict();  $(document).ready(function () {      if ($('#map-canvas').length) {          function initialize() {              var place = eval($("#map-canvas").attr('data-place'));              var pos = new google.maps.LatLng(place[0], place[1]);              var mapOptions = {                  center: pos,                  zoom: 17,                  mapTypeId: google.maps.MapTypeId.ROADMAP,                  mapTypeControlOptions: {                      style: google.maps.MapTypeControlStyle.DROPDOWN\_MENU                  },                  scrollwheel: false,                  rotateControl: true              };                var map = new google.maps.Map(document.getElementById("map-canvas"), mapOptions);              map.setTilt(45);                var marker = new google.maps.Marker({                  position: pos,                  map: map              });          }            google.maps.event.addDomListener(window, 'load', initialize);      }  }); |

Создадим необходимые элементы html

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1 | <div id="map-canvas" data-place="[61.784625, 34.350755]"></div> |

Объяснение этой уличной магии

в js нас интересует строка :

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1 | var place = eval($("#map-canvas").attr('data-place')); |

здесь

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1 | $("#map-canvas").attr('data-place') |

Получает атрибут data-place  элемента с id map-canvas после чего eval() преобразует строку [61.784625, 34.350755] в js код , конкретней в массив т.к. присутствуют квадратные скобки.

По спецификации HTML5 ,браузеры пропускают атрибуты тегов начинающихся с data- влиять на отрисовку этот атрибут не будет, но его значение можно использовать в javascript.

Код html можно модифицировать:

[?](https://alkoweb.ru/peredacha-peremennoj-iz-html-v-javascript/)

|  |  |
| --- | --- |
| 1 | <div id="map-canvas" data-place="<?php echo '[61.784625, 34.350755]' ;?>"></div> |

с использованием php , так простыми средствами можно передавать целые массивы переменных из php в js не используя инлайнового написания : )

# События

## Введение в обработку событий

**Последнее обновление: 1.11.2015**

Для взаимодействия с пользователем в JavaScript определен механизм событий. Например, когда пользователь нажимает кнопку, то возникает событие нажатия кнопки. В коде JavaScript мы можем определить возникновение события и как-то его обработать.

В JavaScript есть следующие типы событий:

* События мыши (перемещение курсора, нажатие мыши и т.д.)
* События клавиатуры (нажатие или отпускание клавиши клавиатуры)
* События жизненного цикла элементов (например, событие загрузки веб-станицы)
* События элементов форм (нажатие кнопки на форме, выбор элемента в выпадающем списке и т.д.)
* События, возникающие при изменении элементов DOM
* События, возникающие при касании на сенсорных экранах
* События, возникающие при возникновении ошибок

Рассмотрим простейшую обработку событий. Например, на веб-странице у нас есть следующий элемент div:

|  |  |
| --- | --- |
| 1 | <div id="rect" onclick="alert('Нажато')" style="width:50px;height:50px;background-color:blue;"></div> |

Здесь определен обычный блок div, который имеет атрибут **onclick**, который задает **обработчик события** нажатия на блок div. То есть, чтобы обработать какое-либо событие, нам надо определить для него обработчик. Обработчик представляет собой код на языке JavaScript. В данном случае обработчик выглядит довольно просто:

|  |  |
| --- | --- |
| 1 | alert('Нажато') |

И при нажатии на кнопку будет выскакивать сообщение:

Также можно было бы вынести все действия по обработке события в отдельную функцию:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <!DOCTYPE html>  <html>  <head>      <meta charset="utf-8" />  </head>  <body>  <div id="rect" onclick="displayMessage()" style="width:50px;height:50px;background-color:blue;"></div>  <script>  function displayMessage(){        alert('Нажато');  }  </script>  </body>  </html> |

Теперь обработчиком события будет выступать функция displayMessage.

### Передача параметров в обработчик события

В обработчик можно передавать параметры. Например, мы можем передать текущий объект, на котором возникает событие:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <a href="page1.html" onclick="return handler(this)">Станица 1</a>  <script>  function handler(obj){        alert(obj.href);      return false;  }  </script> |

Ключевое слово this указывает на текущий объект ссылки, на которую производится нажатие. И в коде обработчика мы можем получить этот объект и обратиться к его свойствам, например, к свойству href.

Кроме того, надо отметить, что здесь обработчик возвращает результат. Хотя в первом примере с блоком div от обработчика не требовалось возвращения результата. Дело в том, что для некоторых обработчиков можно подтвердить или остановить обработку события. Например, нажатие на ссылку должно привести к переадресации. Но возвращая из обработчика false, мы можем остановить стандартный путь обработки события, и переадресации не будет. Если же возвращать значение true, то событие обрабатывается в стандартном порядке.

Если же мы вовсе уберем возвращении результата, то событие будет обрабатываться, как будто возвращается значение true:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <a href="page1.html" onclick="handler(this)">Станица 1</a>  <script>  function handler(obj){        alert(obj.href);  }  </script> |

Кроме непосредственно элемента-источника события в обработчик мы можем передавать объект event. Этот объект не определяется разработчиком, это просто аргумент функции обработчика, который хранит всю информацию о событии. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <!DOCTYPE html>  <html>  <head>      <meta charset="utf-8" />      <style>      #rect{          width:50px;          height:50px;          background-color:blue;      }      </style>  </head>  <body>  <div id="rect" onclick="handler(event)"></div>  <script>  function handler(e){        alert(e.type); // получаем тип события  }  </script>  </body>  </html> |

В данном случае с помощью свойства type объекта event получаем тип события (в данном случае тип click).

# Советы по JavaScript

# Передача и обработка данных в html-файле

**Передача данных в html-файл:**

Передачу данных в html-файл можно произвести через URL-адрес. После адреса ставиться знак вопроса и после него параметр, который Вы хотите передать. Если этих параметров несколько, то между ними ставиться амперсанд (&).

**Пример 1:**

"youfile.html?parametr"

**Пример 2:**

"youfile.html?parametr1&parametr2&parametr3"

Также данные можно передавать при помощи форм. Для этого надо воспользоваться методом GET.

Вот как буде выглядеть URL-адрес в строке броузера:

**Пример:**

"youfile.html?name1=value1&name2=value2&name3=value3"

**Получение переданных данных:**

Получение переданных данных выполняется при помощи **JavaScript**.

Сначало мы присваиваем переменной все то, что находится после знака вопроса:

var p\_url=location.search.substring(1);

Если Вы передавали один параметр, то переменную *p\_url* можно обрабатывать.

Если параметров было много, то их надо разделить:

var parametr=p\_url.split("&");

У нас получился массив *parametr*, который содержит полученные данные.

Если Вы пересылали данные при помощи форм, то надо избавиться от знака равенства и присвоить каждой переменной свое значение:

var values= new Array();for(i=0; i var j=parametr.split("="); values[j[0]]=unescape(j[1]);}

**Пример:**

Если строка была:

"youfile.html?name=Serg&city=Рязань&age=21"

то получим массив:

values[name]="Serg";values[city]="Рязань";values[age]="21";

или можно так:

values[0]="Serg";values[1]="Рязань";values[2]="21";

# Чем заменить SSI, или необычное использование JavaScript

Структура подавляющего количества интернет-страничек примерно одинакова: это заголовок странички, навигационное меню и, непосредственно, само содержимое. Нет проблем, когда на Вашем сайте несколько страниц. Если Вы захотели изменить дизайн, то на это уйдет немного времени. А если страниц не 8-10 а 100-200? Чтобы поменять дизайн нужен долгий однообразный кропотливый труд. А если в меню появился новый пункт? Опять же везде на каждой странице надо вводить изменения.

Помочь в этом может **SSI** - возможность собрать страничку из отдельных блоков. Т.е у Вас есть меню, которое одинаково на всех страницах. Вы его выносите в отдельный файл, а затем включаете его в нужное место на странице. Теперь, чтобы изменить меню на всех страницах, надо изменить его только в одном файле.

Это конечно удобно, но данная возможность имеется только при платном хостинге, или Вам придется вставлять баннеры хостера, что может быть нежелательно для Вас.

Справиться с этой проблемой может язык **JavaScript**, а именно атрибут *SRC* в теле тега <SCRIPT>.

Атрибут *SRC* задает URL файла, в котором находится код программы **JavaScript**, а в самом файле с расширением \*.js мы запишем то, что должно выводиться на экран при помощи метода *document.write*

Например у нас есть меню состоящее из двух пунктов и страница, в которую надо вставить это меню:

**Файл menu.js:**

document.write("<B>Меню:</B>

<BR>"); document.write("<A href="page1.html">Ссылка1</A>

<BR>"); document.write("<A href="page2.html">Ссылка2</A>");

**Файл page.html:**

<html> <head> <title>Проверка</title> </head> <body> <h1>Моя страница</h1> <script src="menu.js"></script> . . . </body> </html>

Т.е. вместо

<script src="menu.js"></script>

браузер вставил:

<B>Меню:</B>

<BR> <A href="page1.html">Ссылка1</A>

<BR> <A href="page2.html">Ссылка2</A>

И если вместо меню на всех страницах вставить *<script src="menu.js"></script>*, то для изменения меню на всех страницах нужно изменить только файл *menu.js*

Но у этого метода есть небольшие недостатки:

# Как сделать страницу стартовой

Сделав страницу стартовой при запуске Internet Explorer браузер пользователя будет автоматически загружать указанный адрес.

Вот код ссылки, нажав которую пользователь сделает страницу стартовой:

<a href="#" onClick="this.style.behavior="url(#default#homepage)"; this.setHomePage("http://www.spravkaweb.ru/"); return false;"> Сделать страницу стартовой</a>

# Как добавить страницу в избранное

Добавляет вашу страницу в "Избранное".

Эта возможность позволяет посетителю вашей страницы сделать закладку, если он не успевает просмотреть или просто запомнить адрес вашего ресурса.

<a href="#" onClick="window.external.addFavorite ("http://www.spravkaweb.ru/", "Краткий справочник WEB-языков"); return false;"> Добавить сайт в избранное</a>

# Как вывести дату последнего изменения страницы

Эта возможность позволяет автоматически указывать дату последней модификации данной страницы.

Но данная возможность таит и свой минус: требуется почаще обновлять страницу, а иначе пользователь будет думать, что срок годности вашего ресурса уже давно истек.

<script language="JavaScript"> <!-- document.write ("Дата последнего изменения: " + document.lastModified); //--> </script>

# Как убрать в теге SELECT рамку и стрелку

При помощи каскадных таблиц стилей (**CSS**) можно изменить стандартный вид объекта <SELECT>:

<SELECT size=1 style="position:absolute;width:100px; clip:rect(2px 82px 20px 2px);background:pink;top:10px;">

<OPTION VALUE=1>One

<OPTION VALUE=2>Two

<OPTION VALUE=3>Three </SELECT>

# Интересный JavaScript - побуквенное появление текста

Данный эффект позволяет выводить текст побуквенно. Есть возможность задавать количество выводимых за один раз символов и скорость вывода.

Вот код:

<script>

var source,dest,len,now=0,delay=100,letters=1;

function show\_text()

{

source = document.getElementById("pageTextSource");

dest = document.getElementById("pageText");

len = source.innerHTML.length;

show();

}

function show()

{

dest.innerHTML += source.innerHTML.substr(now,letters);

now+=letters;

if(now<len)

setTimeout("show()",delay);

}

</script>

<body onload="show\_text();">

<p style="display:none" id="pageTextSource">Текст для вывода...

<p id="pageText"></p>

</body>