

JavaScript

Основы программирования, часть 3

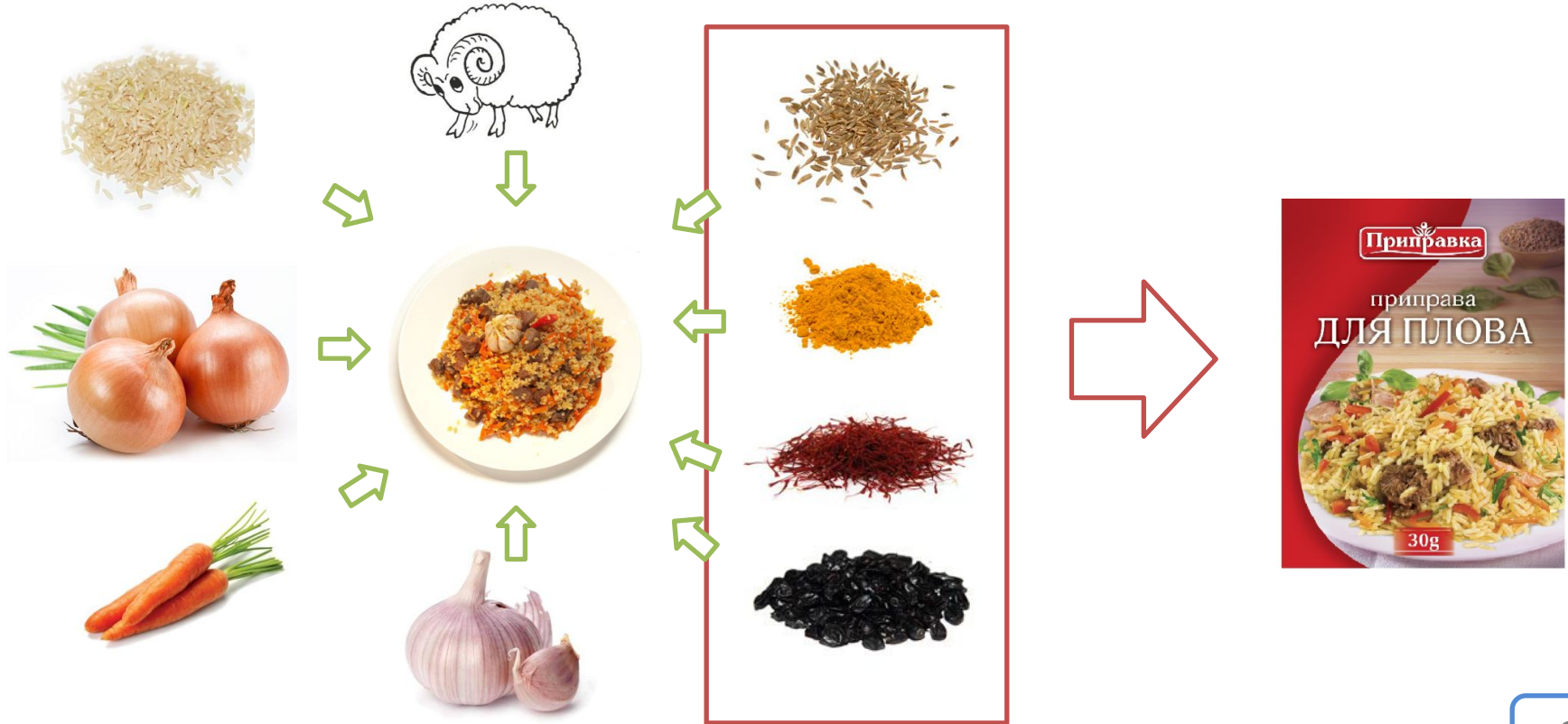


Web.Dev.
Courses

www.courses.dp.ua

Функции

Функция – фрагмент кода, у которого есть имя, который можно вызывать из любого места в программе. **Функции** уменьшают количество кода в программе, код функции пишется один раз, используется многократно.



Функции

Функция – фрагмент кода, у которого есть имя, который можно вызывать из любого места в программе. **Функции** уменьшают количество кода в программе, код функции пишется один раз, используется многократно.

С функциями мы уже сталкивались: **alert()**, **prompt()**, **parseInt()** и т.д. Для нас это просто названия, мы не знаем как они устроены, но мы знаем, что передав ей определенные параметры мы получим на выходе ожидаемый результат.

Функции

Функция – фрагмент кода, у которого есть имя, который можно вызывать из любого места в программе. **Функции** уменьшают количество кода в программе, код функции пишется один раз, используется многократно.

```
1 <script>
2   function imt(weight, height){
3       var index = weight / (height * height);
4       return index;
5   }
6
7   console.log("Пётр: " + imt(78, 1.91));
8   console.log("Иван: " + imt(67, 1.69));
9   console.log("Ирина: " + imt(62, 1.74));
10 </script>
```

Пётр: 21.380992845590857

[ex02.html:7](#)

Иван: 23.458562375266975

[ex02.html:8](#)

Ирина: 20.478266613819528

[ex02.html:9](#)

>

Функции

Основной отличительный знак **функций** – круглые скобки: **alert()** Круглые скобки делают сразу два дела: говорят браузеру что мы хотим **выполнить функцию** имя которой стоит перед скобками, и позволяет передать **функции параметры** (если они необходимы для работы функции):

Имя функции
которую вызываем
(выполняем).

Скобки, признак
того, что мы именно
вызываем
функцию.

Параметры функции
(данные которые
функция
обрабатывает).

**alert("Сообщение на
экран");**

Параметры функции и возвращаемое значение

Параметры функции (аргументы) – такие себе «переменные» которые передаются функции при каждом вызове, и могут влиять на результат её работы.

Возвращаемое значение – возможность функции вернуть результат своей работы, при необходимости. Возврат делается оператором **return**.

```
1 <script>
2 function imt(weight, height){
3     var index = weight / (height * height);
4     return index;
5 }
6
7 console.log("Пётр: " + imt(78, 1.91));
8 console.log("Иван: " + imt(67, 1.69));
9 console.log("Ирина: " + imt(62, 1.74));
10 </script>
```

Подробнее о параметрах функции

Параметры внутри **функции** ведут себя как временные переменные, которые живут только пока **функция выполняется**, при старте им присваиваются те значения которые **переданы функции** при вызове.

Не обязательно передавать **функции** все **параметры** которые предусмотрены её заголовком, НО в таком случае

```
1 <script>
2
3 //функция с тремя параметрами
4 function demo(a, b, c){
5     console.log("Получено: ", a, b, c);
6     var x = a + b + c;
7     return x;
8 }
9
10 var res = demo(1,2,3);
11 console.log("Результат: ", res);
12
13 res = demo(1,2);
14 console.log("Результат: ", res);
15
16 res = demo();
17 console.log("Результат: ", res);
18
19 </script>
```

Получено: 1 2 3	ex02.html:5
Результат: 6	ex02.html:11
Получено: 1 2 undefined	ex02.html:5
Результат: NaN	ex02.html:14
Получено: undefined undefined undefined	ex02.html:5
Результат: NaN	ex02.html:17

Практически любой код можно преобразовать в функцию

```
1 <script>
2
3   function day_profit(sum, percent){
4       return ((sum * (percent/100))/365)*0.805;
5   }
6
7   console.log("Для 1000 грн, под 23%: ", day_profit(1000, 23));
8   console.log("Для 5500 грн, под 19.5%: ", day_profit(5500, 19.5));
9
10 </script>
```

Для 1000 грн, под 23%:	0.5072602739726028	<u>ex05.html:7</u>
Для 5500 грн, под 19.5%:	2.3653767123287675	<u>ex05.html:8</u>

Формула расчёта дневного дохода депозита

Практическая ценность функций

1. Уменьшаем дублирование (повторение) кода;
2. Проще вносить изменения;
3. Абстрагирование от деталей;

```
1 <script>
2   var imt = function(weight, height){
3       if(height > 3){ //Если рост введён в см. то переводим в метры.
4           height = height / 100;
5       }
6       var index = weight / (height * height);
7
8       return index;
9   }
10
11   console.log(imt(70, 1.9));
12   console.log(imt(70, 190));
13 </script>
```

Функции используются для обработки событий

Событие – действие о котором браузер должен уведомить нашу программу;

Смысл событий в JS - сказать браузеру: «когда произойдёт клик по элементу, то выполни вот эту функцию».

```
1 <html>
2   <head>
3     <script>
4       function on_event() {
5         alert("функция вызвана!");
6       }
7     </script>
8   </head>
9   <body>
10    <p onclick="on_event();">Text Text Text</p>
11  </body>
12 </html>
```

События /

Events

Вариантов событий много, задача программиста выбрать нужное

HTML DOM Events

DOM: Indicates in which DOM Level the property was introduced.

Mouse Events

Event	Description	DOM
onclick	The event occurs when the user clicks on an element	2
oncontextmenu	The event occurs when the user right-clicks on an element to open a context menu	3
ondblclick	The event occurs when the user double-clicks on an element	2
onmousedown	The event occurs when the user presses a mouse button over an element	2
onmouseenter	The event occurs when the pointer is moved onto an element	2
onmouseleave	The event occurs when the pointer is moved out of an element	2
onmousemove	The event occurs when the pointer is moving while it is over an element	2
onmouseover	The event occurs when the pointer is moved onto an element, or onto one of its children	2
onmouseout	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children	2
onmouseup	The event occurs when a user releases a mouse button over an element	2

Keyboard Events

Event	Description	DOM
onkeydown	The event occurs when the user is pressing a key	2
onkeypress	The event occurs when the user presses a key	2
onkeyup	The event occurs when the user releases a key	2

http://www.w3schools.com/jsref/dom_obj_event.asp

Немного практики

Служебные функции браузера.

document.querySelectorAll("css_selector") – возвращает массив объектов (элементов HTML-документа, тегов) которые соответствуют *css-селектору* который передан ей в качестве параметра;

document.querySelector("css_selector") – возвращает первый объект (элемент HTML-документа, тег) из тех которые соответствуют *css-селектору* который передан ей в качестве параметра.

document.createElement("tag_name") – метод создающий пустой элемент (объект, тег), которые еще не входит в документ, но его свойства уже можно наполнять необходимыми данными;

Добавим на страницу код который сделаем скиду на товары цена которых выше средней

Смартфоны++ Есть скидки!

% Показать скидки!

iPhone 8	899 \$
iPhone 7	799 \$
Apple iPhone 6	199 \$
Samsung Galaxy S9	650 \$
Nokia 3310	99 \$
HTC Two M3	433 \$
Lenovo H6000	250 \$
Microsoft Lumia 950	195 \$
Sony Ericsson K750 Gold	999 \$



Смартфоны++ Есть скидки!

% Показать скидки!

iPhone 8	809 \$
iPhone 7	719 \$
Apple iPhone 6	199 \$
Samsung Galaxy S9	585 \$
Nokia 3310	99 \$
HTC Two M3	433 \$
Lenovo H6000	250 \$
Microsoft Lumia 950	195 \$
Sony Ericsson K750 Gold	899 \$

Скопируйте заготовку разметки в
notepad++

<http://files.courses.dp.ua/web/10/ex01.html>

```

31 <script>
32     function action(){
33         //Просим у браузера найти все теги которые подходят селектору.
34         //Браузер вернёт теги в виде массива.
35         var prices = document.querySelectorAll("li > .tag");
36
37         //Заготавливаем переменную под сумму цен.
38         var sum = 0;
39
40         //Перебираем массив с тегами которые хранят цены.
41         for(var i = 0; i < prices.length; i++){
42             //Преобразуем строку с ценой в число и накапливаем сумму.
43             var price = parseInt(prices[i].innerHTML);
44             sum = sum + price;
45         }
46
47         //Расчитываем среднюю цену
48         var avr = sum / prices.length;
49
50         //Снова перебираем массив с ценами
51         for(var i = 0; i < prices.length; i++){
52             var price = parseInt(prices[i].innerHTML);
53
54             //Нас будут интересовать только теги с ценой выше средней
55             if(price > avr){
56
57                 //Создаём еще один тег с ценником.
58                 var new_price = document.createElement("span");
59                 //Устанавливаем для него классы (в т.ч. класс делающий тег красным).
60                 new_price.className = "tag tag-danger tag-pill pull-xs-right";
61                 //Присваиваем теги новое значение, а именно цену со скидкой.
62                 new_price.innerHTML = Math.floor(price * 0.9) + " $";
63                 //Добавляем тег, к li в качестве потомка.
64                 prices[i].parentNode.appendChild(new_price);
65
66                 prices[i].remove();
67             }
68         }
69
70         //Находим кнопку и отключаем её.
71         var button = document.querySelector("h1 + button");
72         button.disabled = true;
73     }
74 </script>

```

Добавим на страницу код который сделаем скиду на товары цена которых выше средней



Домашнее задание

**Сделайте, по нажатию кнопки, перевод цен из
долларов в гривну (курс на ваше**

отре

% Показать скидки!

iPhone 8	899 \$
iPhone 7	799 \$
Apple iPhone 6	199 \$
Samsung Galaxy S9	650 \$
Nokia 3310	99 \$
HTC Two M3	433 \$
Lenovo H6000	250 \$
Microsoft Lumia 950	195 \$
Sony Ericsson K750 Gold	999 \$



% Показать скидки!

iPhone 8	23374 грн
iPhone 7	20774 грн
Apple iPhone 6	5174 грн
Samsung Galaxy S9	16900 грн
Nokia 3310	2574 грн
HTC Two M3	11258 грн
Lenovo H6000	6500 грн
Microsoft Lumia 950	5070 грн
Sony Ericsson K750 Gold	25974 грн

Заготовка по
адресу

<http://files.courses.dp.ua/web/10/ex01.html>