

Функции и события в JavaScript

JS

Задача

- Какие значения выведет следующий фрагмент кода?
- `let k = 0;`
- `while (++k < 5) document.write(k + "
");`
- А если заменить `++k` на `k++` ?

Ответ 1

- 1
- 2
- 3
- 4

Ответ 2

- 1
- 2
- 3
- 4
- 5

Задача

- **Задача:** вывести на пяти строчках 😊 – на первой один смайлик, на второй – два и т.д.

<https://js.do/>

Решение

```
<script>
var linesNum=5;
var line="";
var subpower=1;
while (subpower<=linesNum)
{line+=":) ";
document.write(line+"<br \\/>");
subpower++;}
```

```
:)
:):)
:):):)
:):):):)
:):):):):)
```

Задача

- Какие значения выведет следующий фрагмент кода?
- `let j = 3;`
- `while (j) {`
- `document.write(j--);`
- `}`

Ответ

- 321

Задача

- Какие значения выведет следующий фрагмент кода?
- `for (let i = 2; i < 10; i++) {`
- `if (i % 2 == 0) {`
- `alert(i);`
- `}`
- `}`

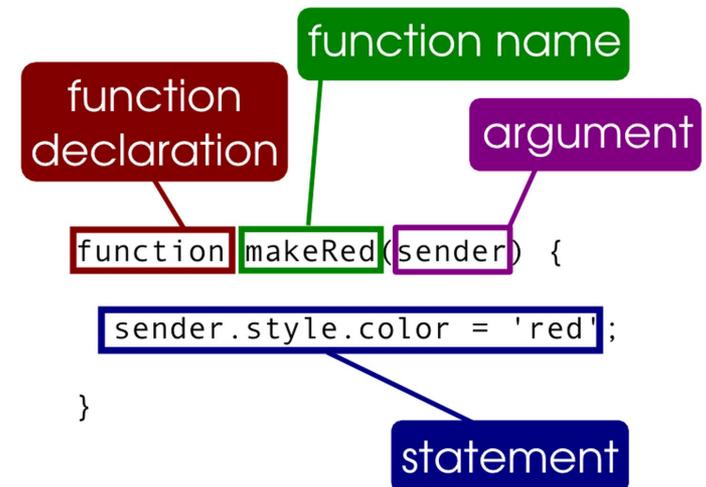
Ответ

- 2
- 4
- 6
- 8

Функции JS

- Зачастую нам надо повторять одно и то же действие во многих частях программы.
- Чтобы не повторять один и тот же код во многих местах, придуманы функции. Функции являются основными «строительными блоками» программы.

Примеры встроенных функций вы уже видели – это `alert(message)`, `prompt(message, default)` и `confirm(question)`. Но можно создавать и свои.



Объявление функции

- Для создания функций мы можем использовать *объявление функции*.
- Вначале идёт ключевое слово `function`, после него *имя функции*, затем список *параметров* в круглых скобках через запятую (в вышеприведённом примере он пустой) и, наконец, код функции, также называемый «телом функции», внутри фигурных скобок.

```
function имя(параметры) {  
...тело...  
}
```

```
1 function showMessage() {  
2     alert( 'Всем привет!' );  
3 }
```

Выбор имени функции

- Функция – это действие. Поэтому имя функции обычно является глаголом. Оно должно быть простым, точным и описывать действие функции, чтобы программист, который будет читать код, получил верное представление о том, что делает функция.
- Как правило, используются глагольные префиксы, обозначающие общий характер действия, после которых следует уточнение.
- Например, функции, начинающиеся с "show" обычно что-то показывают.
- Функции, начинающиеся с...
- "get..." – возвращают значение,
- "calc..." – что-то вычисляют,
- "create..." – что-то создают,
- "check ..." – что-то проверяют и возвращают логическое значение,

```
1 showMessage(..) // показывает сообщение
2 getAge(..)      // возвращает возраст (в каком либо значении)
3 calcSum(..)    // вычисляет сумму и возвращает результат
4 createForm(..) // создаёт форму (и обычно возвращает её)
5 checkPermission(..) // проверяет доступ, возвращая true/false
```

Параметры функций в JavaScript

- Мы можем передать внутрь функции любую информацию, используя параметры (также называемые *аргументами функции*).
- В нижеприведённом примере функции передаются два параметра: `from` и `text`

```
function showMessage(from, text) { // аргументы: from, text
  alert(from + ': ' + text);
}
showMessage('Аня', 'Привет!'); // Аня: Привет! (*)
showMessage('Аня', "Как дела?"); // Аня: Как дела? (**)
```

Возврат значения

- Функция может вернуть результат, который будет передан в вызвавший её код.
- Простейшим примером может служить функция сложения двух чисел:

```
1 function sum(a, b) {  
2   return a + b;  
3 }  
4  
5 let result = sum(1, 2);  
6 alert( result ); // 3
```

Директива `return` может находиться в любом месте тела функции. Как только выполнение доходит до этого места, функция останавливается, и значение возвращается в вызвавший её код

Какие значения выведет
следующий фрагмент кода?

```
function func(num) {  
    return num;  
    let result = num * num;  
    return result;  
}  
alert( func(3) );
```

Ответ

- 3

Какие значения выведет следующий фрагмент кода?

```
function func(num) {  
  if (num <= 0) {  
    return Math.abs(num);  
  } else {  
    return num * num;  
  }  
}  
alert( func(10) );  
alert( func(-5) );
```

Ответ

- 100
- 5

События

- — это действия, которые происходят в программируемой вами системе, о которых система сообщает вам, чтобы вы могли, если нужно, как-то реагировать на них. Например, если пользователь нажимает кнопку на веб-странице, вы можете ответить на это действие, отобразив информационное окно.
- Справочник - <https://meliorem.ru/frontend/javascript/spravochnik-javascript-sobytij-dom-elementov/>

События мыши:

- `click` – происходит, когда кликнули на элемент левой кнопкой мыши (на устройствах с сенсорными экранами оно происходит при касании).
- `contextmenu` – происходит, когда кликнули на элемент правой кнопкой мыши.
- `mouseover` / `mouseout` – когда мышь наводится на / покидает элемент.
- `mousedown` / `mouseup` – когда нажали / отжали кнопку мыши на элементе.
- `mousemove` – при движении мыши.

События на элементах управления:

- `submit` – пользователь отправил форму `<form>`.
- `focus` – пользователь фокусируется на элементе, например нажимает на `<input>`.

Клавиатурные события:

- `keydown` и `keyup` – когда пользователь нажимает / отпускает клавишу.

События документа

- `DOMContentLoaded` – когда HTML загружен и обработан.

Обработчики событий

- Событию можно назначить *обработчик*, то есть функцию, которая сработает, как только событие произошло.
- Именно благодаря обработчикам JavaScript-код может реагировать на действия пользователя.

Использование атрибута HTML

- Обработчик может быть назначен прямо в разметке, в атрибуте, который называется `on<событие>`.
- Например, чтобы назначить обработчик события `click` на элементе `input`, можно использовать атрибут `onclick`, вот так:

```
<input value="Нажми меня"  
onclick="alert('Клик!)" type="button">
```

- При клике мышкой на кнопке выполнится код, указанный в атрибуте `onclick`.

Следующий пример по клику запускает функцию

```
1 <script>
2   function countRabbits() {
3     for(let i=1; i<=3; i++) {
4       alert("Кролик номер " + i);
5     }
6   }
7 </script>
8
9 <input type="button" onclick="countRabbits()" value="Считать кроликов!">
```

Считать кроликов!

Использование свойства DOM-объекта

```
1 <input type="button" id="button" value="Кнопка">
2 <script>
3   button.onclick = function() {
4     alert('Клик!');
5   };
6 </script>
```

Кнопка

Спасибо за внимание!

