## Функции

1. Создайте функцию **p** и **a**, которые будут служить короткими именами для **prompt** и **alert** соответственно, то есть запрашивать окно с полем ввода и выводить любое пользовательское сообщение в стандартном модальном окне. Напишите функцию **d**, которая будет служить коротким именем для **debugger**, то есть запускать процесс отладки. Используя эти псевдонимы получите значение из prompt и отобразите в alert. А процесс работы скрипта проконтролируйте с помощью функции, которая служит для дебага.

|  |
| --- |
| d();  let name = p( 'Enter your name', '');  a(name);  a( 'Hello World' ); |

1. Используя стрелочный синтаксис ES6, напишите функцию **max** (аналог Math.max), которая сравнивает два числа и *возвращает* большее:

|  |
| --- |
| console.log( max(0, -1) ); // 0 |

1. Используя синтаксис ES5, напишите функцию-аналог **Math.min()**. Функция принимает любое количество чисел и *возвращает* меньшее из них:

|  |
| --- |
| console.log( min(0, -1, 100, 500, 100500) ); // -1 |

1. Напишите функцию, которая в переданной строке ищет самое длинное слово и возвращает его длину.

|  |
| --- |
| console.log( findLongestWordLength('Аргентина манит негра') ); // 9 |

1. Изучите *перебирающие методы массивов*: **forEach**, **filter**, **map**. Создайте массив объектов users (~10 объектов), каждый объект имеет поля firstname, lastname, age с разными значениями, у некоторых есть поле middlename. Используя встроенные функции массивов:
   1. Отфильтруйте пользователей младше 18 лет
   2. Добавьте каждому объекту поле fullName, которое является конкатенацией firstname, middlename и lastname, если есть все три, и только firstname и lastname, если middlename нет
   3. Сформируйте новый массив, который содержит только полное имя пользователей

1. Не используя методы массива, напишите функцию аналог метода **shift**. Функция удаляет из переданного в параметре массива первый элемент и возвращает новый массив.
2. Не используя методы массива, напишите функцию аналог метода массива **push**. Функция добавляет в конец переданного в параметре массив произвольное количество элементов.
3. Напишите функцию аналог метода ES6 **Object.assign()**. Первый параметр функции - целевой объект, поля которого будут изменены или расширены. Остальные параметры - объекты-источники, полями которых будет расширяться целевой объект.

|  |
| --- |
| var source = {firstname: 'Tom', age: 10}  var s = extend(source, {firstname: 'John'}, {lastname: 'Doe'});  console.log(source); // {firstname: 'John', age: 10, lastname: 'Doe'}  console.log(s); // {firstname: 'John', age: 10, lastname: 'Doe'} |

1. Напишите функцию **ask** с тремя параметрами: **question**, **defaultValue**, **callback**. Функция **ask** запрашивает с помощью **prompt** вопрос **question**, ответ по умолчанию - **defaultValue**. Что делать с полученным значением - определяет **callback**

|  |
| --- |
| ask('How old are you?', '10', function(response) {  console.log(response);  }); |

1. Найдите и исправьте ошибки в коде:

|  |
| --- |
| function foo() {  function bar(a) {  i = 3;  return i + a;  }    for (var i = 0; i < 10; i++) {  console.log( bar(i \* 2) );  }  }  foo(); // 3 5 7 9 11 13 15 17 19 21 |

## Level Up

1. “История в картинках”. Создайте массив тематически связанных [строковых картинок](https://unicode.org/emoji/charts/full-emoji-list.html). Перебирая массив в цикле с определенной задержкой (setTimeout) выводите на страницу один из элементов массива (одну из картинок). Тема - свободная, ограничена только набором доступных строковых картинок.

