

JAVASCRIPT

ФУНКЦИИ



ФУНКЦИИ

- избавляют от повторений кода;
- улучшают структуру программы;
- уменьшают вероятность появления ошибок и повышают устойчивость к ошибкам.

КОГДА ДЕЛАТЬ

- переписываем одни и те же последовательности;
- много мелких подробностей;
- слишком длинная основная программа;
- имеются слишком сложные подзадачи;
- необходимость повторного использования кода в других программах.

ВСТРОЕННЫЕ ФУНКЦИИ

alert();

console.log();

И так далее...

Это точно такие же функции, как и те, которые мы будем создавать, только они уже зарезервированы языком JavaScript.

ОБЪЯВЛЕНИЕ ФУНКЦИИ

```
function functionName (argument) {  
    // body of function  
}
```

functionName – имя функции. Именно его будем использовать в дальнейшем коде

АРГУМЕНТ

```
function functionName (argument) {  
    // body of function  
}
```

argument - Аргумент. Это то что мы передаем в функцию на обработку (какие-то данные)

ТЕЛО ФУНКЦИИ

```
function functionName (argument) {  
    // body of function  
}
```

body of function - Непосредственно
тело функции.

ПРИМЕР ФУНКЦИИ

```
function helloWorld () {  
    alert("Hello, World!");  
}
```

helloWorld(); // Выведет на экран Hello, World!

helloWorld(); // Еще раз выведет на экран Hello, World!

ОБЛАСТЬ ВИДИМОСТИ

```
function helloWorld () {  
    var hello = "hello";  
    console.log(hello);  
}
```

helloWorld(); // hello

console.log(hello); // *ReferenceError: hello is not defined*

ОБЛАСТЬ ВИДИМОСТИ

```
function helloWorld () {  
    hello = "hello"  
    console.log(hello);  
    hello2 = hello + "!";  
}  
  
helloWorld();           // hello  
  
console.log(hello2);    //hello!
```

ОБЛАСТЬ ВИДИМОСТИ

```
var world = "world!";
```

```
function helloWorld () {
```

```
    var world = "hello!";
```

```
    console.log(world);
```

```
}
```

```
helloWorld();      //hello!
```

```
console.log(world); //world!
```

ОБЛАСТЬ ВИДИМОСТИ

```
function count() {  
  for (var i = 0; i < 3; i++) {  
    var j = i * 2;  
  }  
  alert( i );    // i=3  
  alert( j );    // j=4  
}  
count()
```

ОБЛАСТЬ ВИДИМОСТИ

```
function count() {  
    var i, j; // передвинули объявления var в начало  
    for (i = 0; i < 3; i++) {  
        j = i * 2;  
    }  
    alert( i ); // i=3  
    alert( j ); // j=4  
}  
count();
```

ОБЛАСТЬ ВИДИМОСТИ

```
function count() {  
  for (let i = 0; i < 3; i++) {  
    var j = i * 2;  
  }  
  alert( i );    // i=3  
  alert( j );    // j=4  
}  
count();
```

ВНЕШНИЕ ПЕРЕМЕННЫЕ

```
var world = " world!";
```

```
function helloWorld () {
```

```
    hello = "hello";
```

```
    console.log(hello+world);
```

```
}
```

```
helloWorld(); // hello world!
```

АРГУМЕНТЫ

```
function functionName (argument) { //функция принимает аргумент  
    // body of function  
}
```

argument - Аргумент. Это то, что мы передаем в функцию на обработку (какие-то данные)

functionName(5); - Передаем 5 в качестве аргумента
в функцию

АРГУМЕНТЫ

Пример:

```
function myFunc(argument) {  
    console.log(argument+95);  
}  
  
myFunc(5); // 100
```

НЕСКОЛЬКО АРГУМЕНТОВ

```
function myFunc(arg, arg2, arg3, someText) {  
    console.log("arg = " + arg);  
    console.log("arg2 = " + arg2);  
    console.log("arg3 = " + arg3);  
    console.log("someText = " + someText);  
}  
  
myFunc(2, 3, 5, "Какой-то текст");
```

НЕСКОЛЬКО АРГУМЕНТОВ

```
function myFunc(arg, arg2, arg3, someText) {  
    console.log("arg = " + arg);  
    console.log("arg2 = " + arg2);  
    console.log("arg3 = " + arg3);  
    console.log("someText = " + someText); //undefined  
}  
  
myFunc(2, 3, 5);
```

ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ

```
function myFunc(arg, someText = "Hello, world!") {  
  console.log("arg = " + arg);  
  console.log("someText = " + someText);  
}
```

```
myFunc(2); // arg = 2, someText = Hello, world!
```

НЕСКОЛЬКО АРГУМЕНТОВ

```
function myFunc(arg, someText = "Hello, world!") {  
    console.log("arg = " + arg);  
    console.log("someText = " + someText);  
}
```

```
myFunc(2,3);    // arg = 2, someText = 3
```

```
myFunc (2);
```

НЕСКОЛЬКО АРГУМЕНТОВ

```
function myFunc(arg, someText) {  
    if (someText === undefined) {someText = "Текст по  
    умолчанию"}  
  
    console.log("arg = " + arg);  
  
    console.log("someText = " + someText);  
  
}  
  
myFunc(2);
```

НЕСКОЛЬКО АРГУМЕНТОВ

```
function myFunc(arg, someText) {  
    someText = someText || "Какой-то текст";  
    console.log("arg = " + arg);  
    console.log("someText = " + someText);  
}  
  
myFunc(2);
```

ВОЗВРАТ ЗНАЧЕНИЯ - RETURN

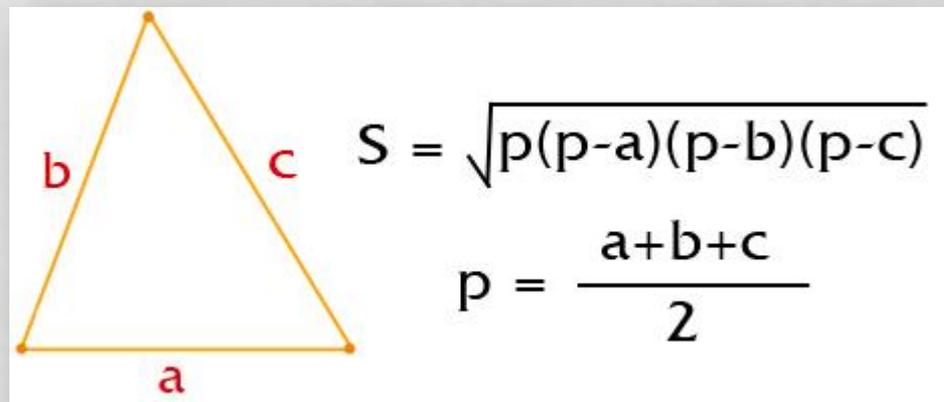
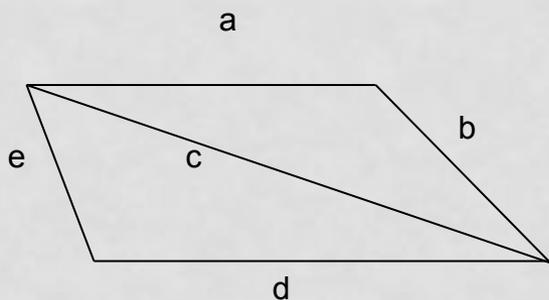
```
function myFunc (argument) {  
    return argument*argument;  
}  
  
console.log (myFunc(4));    //4*4=16
```

ВАРИАНТ ВЫЗОВА ФУНКЦИИ

```
function myFunc (argument) {  
    return argument*argument;  
}  
  
var func = myFunc(4);  
  
console.log(func);
```

ПРИМЕР

Даны числа a , b , c , d , e . Найти площадь фигуры. Для подсчета площади треугольника использовать функцию.



$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

$$p = \frac{a+b+c}{2}$$

РЕШЕНИЕ

```
function squareTriangle (x, y, z) {  
    var p = (x + y + z) / 2;  
    return Math.sqrt(p * (p - x) * (p - y) * (p - z));  
}  
  
var a=4, b=4, c = 6, e = 3, d = 5, square;  
square = squareTriangle (a, b, c) + squareTriangle (c, d, e);  
console.log (square);
```

ПРИМЕР

Даны два целых числа. Поменять местами их значения.

```
function remove (x,y) {  
  let z = x;  
  x = y;  
  y = z;  
  return [x, y];  
}
```

ПРИНЯТИЕ РЕЗУЛЬТАТОВ

```
let a = 2, b = 5;
```

```
var c = remove (a,b);
```

```
a = c[0];
```

```
b = c[1];
```

```
console.log (`a=${a}, b=${b}`)
```

```
[a,b] = remove (a,b);
```

```
console.log (`a=${a}, b=${b}`);
```

ИМЯ ФУНКЦИИ

showMessage(..) // "показать" сообщение

getAge(..) // "получает" возраст

calcD(..) // "вычисляет" дискриминант

createForm(..) // "создает" форму

checkPermission(..) // "проверяет" разрешение

jQuery - \$

Prototype - \$\$

LoDash - _

ФУНКЦИОНАЛЬНЫЕ ВЫРАЖЕНИЯ

```
function sayHello() {  
    alert( "Hello" );  
}
```

```
alert( sayHello );    // выведет код функции
```

ФУНКЦИОНАЛЬНЫЕ ВЫРАЖЕНИЯ

```
function sayHello() {           // (1)
    alert( "Hello" );
}
```

```
var func = sayHello;           // (2)
func();    // Hello           // (3)
```

```
sayHello = null;
sayHello();                     // ошибка (4)
```

ОБЪЯВЛЕНИЕ FUNCTION EXPRESSION

```
var f = function(параметры) {  
    // тело функции  
};
```

```
var sayHello = function(name) {  
    alert( "Привет, " + name );  
};
```

```
sayHello('Вася');
```

СРАВНЕНИЕ С FUNCTION DECLARATION

// Function Declaration

```
function sum(a, b) {  
    return a + b;  
}
```

// Function Expression

```
var sum = function(a, b) {  
    return a + b;  
}
```

СРАВНЕНИЕ С FUNCTION DECLARATION

```
sayHello("Вася");    // Привет, Вася
```

```
function sayHello(name) {  
    alert( "Привет, " + name );  
}
```

```
sayHello("Вася");    // ошибка!
```

```
var sayHello = function(name) {  
    alert( "Привет, " + name );  
}
```

СРАВНЕНИЕ С FUNCTION DECLARATION

```
var sayHello = function(name) {  
  alert( "Привет, " + name );  
}
```

```
sayHello("Вася");      // Привет, Вася
```

АНОНИМНЫЕ ФУНКЦИИ

```
function ask(question, yes, no) {  
    if (confirm(question)) yes()  
    else no();  
}
```

```
function showOk() {  
    alert( "Вы согласились." );  
}
```

```
function showCancel() {  
    alert( "Вы отменили выполнение." );  
}
```

```
// использование  
ask("Вы согласны?", showOk, showCancel);
```

АНОНИМНЫЕ ФУНКЦИИ

```
function ask(question, yes, no) {  
  if (confirm(question)) yes()  
  else no();  
}
```

```
ask(  
  "Вы согласны?",  
  function() { alert("Вы согласились."); },  
  function() { alert("Вы отменили выполнение."); }  
)
```

РЕКУРСИЯ

$$\text{pow}(x, n) = x * \text{pow}(x, n - 1);$$

$$x^n = x * x^{n-1}$$

$$\text{pow}(2, 4) = 2 * \text{pow}(2, 3)$$

$$\text{pow}(2, 3) = 2 * \text{pow}(2, 2)$$

$$\text{pow}(2, 2) = 2 * \text{pow}(2, 1)$$

$$\text{pow}(2, 1) = 2$$

РЕКУРСИЯ

```
function pow(x, n) {  
  if (n !== 1) { // пока n !== 1, сводить вычисление pow(x,n) к  
    pow(x,n-1)  
    return x * pow(x, n - 1);  
  } else {  
    return x;  
  }  
}  
  
var result = pow(2, 3);  
console.log(result);
```

КОНТЕКСТ ВЫПОЛНЕНИЯ

Контекст: { x: 2, n: 3, строка 1 }

```
function pow(x, n) {  
  if (n != 1) {  
    return x * pow(x, n - 1);  
  } else {  
    return x;  
  }  
}
```

row(2, 3)

x=2, n=3

Контекст: { x: 2, n: 3, строка 1 }

row(2, 2)

x=2, n=2

Контекст: { x: 2, n: 3, строка 3 }

Контекст: { x: 2, n: 2, строка 1 }

row(2, 1)

x=2, n=1

Контекст: { x: 2, n: 3, строка 3 }

Контекст: { x: 2, n: 2, строка 3 }

Контекст: { x: 2, n: 1, строка 1 }

Выход из pow(2, 1)

```
function pow(x, n) {  
  if (n != 1) {  
    return x * pow(x, n - 1);  
  } else {  
    return x; // первая степень числа равна самому числу  
  }  
}
```

Контекст: { x: 2, n: 3, строка 3 }

Контекст: { x: 2, n: 2, строка 3 }

Выход из pow(2, 2)

Контекст: { x: 2, n: 3, строка 3 }

Выход из pow(2, 3)

```
function pow(x, n) {  
  var result = x;  
  for (var i = 1; i < n; i++) {  
    result *= x;  
  }  
  return result;  
}
```

ПРИМЕР

Напишите функцию `sumTo(n)`, которая для данного `n` вычисляет сумму чисел от 1 до `n`, например:

$$\text{sumTo}(1) = 1$$

$$\text{sumTo}(2) = 2 + 1 = 3$$

$$\text{sumTo}(3) = 3 + 2 + 1 = 6$$

$$\text{sumTo}(4) = 4 + 3 + 2 + 1 = 10$$

...

$$\text{sumTo}(100) = 100 + 99 + \dots + 2 + 1 = 5050$$

РЕШЕНИЕ ЦИКЛОМ

```
function sumTo(n) {  
    var sum = 0;  
    for (var i = 1; i <= n; i++) {  
        sum += i;  
    }  
    return sum;  
}  
  
console.log ( sumTo(100) );
```

РЕШЕНИЕ РЕКУРСИЕЙ

```
function sumTo(n) {  
    if (n == 1) return 1;  
    return n + sumTo(n - 1);  
}  
  
console.log ( sumTo(100) );
```

РЕШЕНИЕ ПО ФОРМУЛЕ

```
function sumTo(n) {  
    return n * (n + 1) / 2;  
}  
  
console.log ( sumTo(100) );
```

ЗАДАЧИ

1. Описать функцию, вычисляющую вторую, третью и четвертую степени числа A и возвращающую эти степени соответственно в переменных B , C и D .
2. Описать функцию, добавляющую к целому положительному числу K справа цифру D . С помощью этой функции последовательно добавить к данному числу K справа данные цифры $D1$ и $D2$, выводя результат каждого добавления.

3. Описать функцию $\text{MinMax}(X, Y)$, записывающую в переменную X минимальное из значений X и Y , а в переменную Y – максимальное из этих значений. Используя четыре вызова этой процедуры, найти минимальное и максимальное из данных чисел A, B, C, D .
4. Описать функцию $\text{SortInc3}(A, B, C)$, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по возрастанию.

5. Описать процедуру $\text{ShiftLeft3}(A, B, C)$, выполняющую *левый циклический сдвиг*: значение A переходит в C , значение C – в B , значение B – в A .
6. Описать функцию $\text{Sign}(X)$, возвращающую для числа X следующие значения: -1 , если $X < 0$; 0 , если $X = 0$; 1 , если $X > 0$. С помощью этой функции найти значение выражения $\text{Sign}(A) + \text{Sign}(B)$ для данных чисел A и B .
7. Описать функцию $\text{RingS}(R1, R2)$, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами $R1$ и $R2$.

8. Функция принимает параметр - возраст пользователя.

Если число больше 16 – то выводим «добро пожаловать», если меньше – “вы еще молоды”.

9. Создайте функцию, которая считает длину массива и возвращает ее в виде числа. Массив в функцию передается как аргумент. Если аргумент не задан – выводится сообщение об ошибке.

10. Пользователь вводит числа. Если число больше 10, то функция возвращает квадрат числа, если меньше 7 – пишет, что число меньше 7. Если 8, 9 – то возвращает соответственно 7 или 8. Реализуйте решение с несколькими return.

11. Напишите игру «Угадай число». При загрузке страницы генерируется случайное число от 0 до 10. Пользователю дается три попытки угадать. При каждой проверке выдается подсказка: больше или меньше. При угадывании, завершении числа попыток выдается оповещение.

Количество попыток выводится на экран.

12. Магазин предлагает 2 размера гамбургеров:

- маленький (17грн, 250 ккал)
- большой (25грн, 340 ккал)

Бургер дополняется одной или несколькими начинками (обязательно одна должна быть):

- сыром (+ 4грн, + 25 ккал)
- салатом (+ 5грн, + 5 ккал)
- ветчиной (+ 10грн, + 50 ккал)

Дополнительно (не обязательно), гамбургер можно дополнить соусами (+5грн, 0 ккал) и полить майонезом (+ 4грн, + 10 ккал).

Напишите программу, рассчитывающую стоимость и калорийность бургера.