Функции в JavaScript

Математическая функция

Функция одной переменной

$$y = f(x)$$

Функция нескольких переменных

$$y = f(x,y)$$

SyntaxError: Illegal return statement

```
function cheer(score) {
  if (score === 147)
    return "Maximum!";
  }
  if (score > 100) {
    return "Century!";
  }
}
```

Объявление функции ES5 - Function Declaration:

```
function название(parm1, parm2, …) {
   console.log('что-то')
}
```

Объявление функции в ES5 - Function Expression:

```
const название = function (parm1, parm2, …)
  console.log('что-то')}
```

Объявление стрелочной функции в ES6 только Function Expression:

```
const переменная = (parm1, parm2, …) => { return }
```

Вызов

```
название(arg1, arg2, …)
```

Функция

```
// Реализация функции
function add(x, y) {
    var total = x + y;
    return total;
}

> add(2, 3, 4);
```

В чем несоответствие ?

Присвоение значения из функции

```
// Реализация функции
function add(x, y) {
    let total = x + y;
    return total;
}
let result = add(2, 3, 4);
// Какие типы можно возвращать из
 функции ?
```

Кол-во аргументов `arguments[]`

```
function add() {
    let sum = 0;
    let j = arguments.length;
    for (var i = 0, i < j; i++)
    {
        sum += arguments[i];
    }
    return sum;
}

> add(2, 3, 4, 5)
```

Область видимости переменной

```
let sum = 0;
function add() {
    let sum = 0;
    let j = arguments.length;
    for (let i = 0; i < j; i++)
    {
        sum += arguments[i];
    }
    return sum;
}
add(2,3);
>sum // Чему равна переменная ?
```

Область видимости переменной

```
var let = 0; ← глобальная
      переменная


function add() {
    var let = 0; ← локальная
                       переменная
    let j = arguments.length;
    for (var i = 0; i < j; i++){
        sum += arguments[i];
    }
    return sum;
}
```

Перегрузка фунции

```
function addSomeNumber(num){
    return num + 100;
}
function addSomeNumber(num) {
    return num + 200;
}
var result = addSomeNumber(100);
// 300
```

В результате перегрузки иногда возникают конфликты старой и новой библиотеки. Т.к js подключается последовательно.

Что будет если вызов функции возникнет раньше ее реализации?

```
alert(sum(10, 10));
function sum(num1, num2){
    return num1 + num2;
}
```

Передача функции в качестве аргумента.

```
function execf(parm,mass){
   alert(parm(2,3));
}

function sum(num1, num2){
   return num1 + num2;
}
execf(sum);
```

Функция как метод объекта.

```
var cat = {
    name: 'Barsik',
     age: 3,
     mew: function() {
          console.log('Мяу.. покорми меня !')
   }
}
```

Вызов метода:

```
cat.mew();
```

Передача аргументов.

```
var cat = {
    name: ' Barsik',
    age: 3,
    mew: function(str) {
          console.log(str);
   }
}
```

Вызов метода с аргументом:

```
cat.mew('Мяу.. покорми меня !');
```

Объявление функции ES5 - Function Declaration:

```
function название(parm1, parm2, …) {
   console.log('что-то')
}
```

Объявление функции в ES5 - Function Expression:

```
const название = function (parm1, parm2, …)
  console.log('что-то')}
```

Объявление стрелочной функции в ES6 только Function Expression:

```
const переменная = (parm1, parm2, …) => { return }
```

Вызов

```
название(arg1, arg2, …)
```

SyntaxError: Function statements require a function name (V8-based)

callback

https://learn.javascript.ru/callbacks

Задачи.

1.Написать функцию которой передается объект а она выводит его содержимое.

2.Реализовать функцию из п.2 в виде метода объекта. Добавить еще пару произвольных методов объекту.