

# Рекурсия и Стек Числа и Строки

Занятие 4

# План занятия

- Рекурсия
- Стек
- Именованные функциональные выражения
- Структуры данных
- Методы и свойства
- Числа
- Строки

[illegible]

# Рекурсия

# RECURSION

Here we go again

# Рекурсия

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

1.  $\text{pow}(2, 4) = 2 * \text{pow}(2, 3)$
2.  $\text{pow}(2, 3) = 2 * \text{pow}(2, 2)$
3.  $\text{pow}(2, 2) = 2 * \text{pow}(2, 1)$
4.  $\text{pow}(2, 1) = 2$

# Рекурсия

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

```
console.log(pow(2, 3)); // 8
```

# Контекст выполнения, стек

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

```
console.log(pow(2, 3));
```

Контекст: { x: 2, n: 3, line 3 }

Контекст: { x: 2, n: 2, line 3 }

Контекст: { x: 2, n: 1, line 1 }

# Задача 1

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

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

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

- Рекурсия
- Цикл
- Формула арифметической прогрессии

## Задача 2

Написать функцию вычисляющую факториал числа  $n$

$$n! = n * (n - 1) * (n - 2) * \dots * 1$$



A scenic autumn landscape featuring a calm lake reflecting the surrounding trees. On the left, a large tree with vibrant yellow and orange foliage stands prominently. The background shows a dense forest of trees with varying autumn colors under a clear blue sky. The water in the lake is still, creating a perfect reflection of the trees and sky.

# Именованные функциональные выражения

# Named Function Expression

*// обычное функциональное выражение*

```
var f1 = function(a) { /* тело функции */ };
```

*// именованное функциональное выражение*

```
var f2 = function sayHi(a) { /* тело функции */ };
```

# Named Function Expression

```
var f = function sayHi(name) {  
    // изнутри функции – видно (выведет код функции)  
    console.log(sayHi);  
};  
  
// снаружи – не видно (ошибка: undefined variable 'sayHi')  
console.log(sayHi);
```

```
var test = function sayHi(name) {  
    sayHi = 'test'; // попытка перезаписи  
    console.log(sayHi); // function...  
};  
  
test();
```

# Named Function Expression

```
function f(n) {  
    return n ? n * f(n - 1) : 1;  
}
```

```
var g = f;  
f = null;
```

```
console.log( g(5) ); // error
```



# Named Function Expression

```
var f = function factorial(n) {  
    return n ? n*factorial(n-1) : 1;  
};
```

```
var g = f;  
f = null;
```

```
console.log( g(5) ); // 120
```

# Задача

Что выведет этот код?

```
function g() { return 1; }  
console.log(g);
```

А этот?

```
(function g() { return 1; });  
console.log(g);
```

# Структуры данных

## Методы и свойства

# СВОЙСТВО `str.length`

```
console.log('Hello world'.length); //11
```

```
var text = 'Hello GoIt';  
console.log(text.length); //10
```



# Методы toUpperCase и toLowerCase

```
var text = 'Hello GoIt';
```

```
console.log(text.toUpperCase()); //HELLO GOIT
```

```
console.log(text.toLowerCase()); //hello goit
```

# Метод num.toFixed(n)

```
var n = 65.432;
```

```
console.log( n.toFixed(2) ); // "65.43"  
console.log( n.toFixed(0) ); // "65"  
console.log( n.toFixed(5) ); // "65.43200"
```

# Внимание!

```
console.log( 12.34.toFixed(1) ); // 12.3  
console.log(12.toFixed(1)); // ошибка!  
console.log( 12..toFixed(1) ); // 12.0
```

# Числа

# Способы записи

```
console.log(0xFF); // 255 в шестнадцатеричной системе  
console.log(010); // 8 в восьмеричной системе
```

```
// еще пример научной формы: 3 с 5 нулями  
console.log(3e5); // 300000
```

```
// здесь 3 сдвинуто 5 раз вправо, за десятичную точку.  
console.log(3e-5); // 0.00003 <-- 5 нулей, включая начальный ноль
```

# Infinity

```
console.log( 1 / 0 ); // Infinity
console.log( 12345 / 0 ); // Infinity

console.log( Infinity > 1234567890 ); // true
console.log( Infinity + 5 == Infinity ); // true
console.log( -1 / 0 ); // -Infinity
console.log( 1e500 ); // Infinity
var x = Infinity;
```

# NaN

```
console.log( 0 / 0 ); // NaN
```

```
var n = 0 / 0;
```

```
console.log( isNaN(n) ); // true
```

```
console.log( isNaN('12') ); // false
```

```
console.log( NaN + 1 ); // NaN
```

# isFinite(n)

```
var value = prompt("Введите Infinity", 'Infinity');
```

```
var number = +value;
```

```
console.log( number ); // Infinity
```

```
console.log( isFinite(1) ); // true
```

```
console.log( isFinite(Infinity) ); // false
```

```
console.log( isFinite(NaN) ); // false
```



# Преобразование к числу

```
var s = '12.34';  
console.log( +s ); // 12.34  
  
console.log( +'12test' ); // NaN  
  
console.log( +' -12' ); // -12  
  
// 34, перевод строки \n является пробельным символом  
console.log( +' \n34 \n' );  
console.log( +' ' ); // 0, пустая строка становится нулем  
  
// NaN, пробел посередине числа – ошибка  
console.log( +'1 2' );  
  
console.log( '12.34' / '-2' ); // -6.17
```

# parseInt и parseFloat

```
console.log( parseInt('12px') ); // 12
```

```
console.log(parseInt('12px')) // 12, ошибка на символе 'p'
```

```
console.log(parseFloat('12.3.4')) // 12.3, ошибка на второй точке
```

```
console.log( parseInt('a123') ); // NaN
```

# Проверка на число

```
if (isNaN(x)) {  
    console.log( "Строка преобразовалась в NaN. Не число" );  
} else {  
    console.log( "Число" );  
}
```

```
console.log( isNaN(null) ); // false – не NaN, т.е. "число"  
console.log( isNaN('\n \n') ); // false – не NaN, т.е. "число"
```

```
function isNumeric(n) {  
    return !isNaN(parseFloat(n)) && isFinite(n);  
}
```

# toString

```
var n = 255;  
console.log( n.toString(16) ); // ff
```

```
var n = 4;  
console.log( n.toString(2) ); // 100
```

```
var n = 1234567890;  
console.log( n.toString(36) ); // kf12oi
```

# Округление

```
console.log( Math.floor(3.1) ); // 3  
console.log( Math.ceil(3.1) );  // 4  
console.log( Math.round(3.1) ); // 3
```

# Округление до заданной точности

```
var n1 = 12.34;  
console.log( n1.toFixed(1) ); // "12.3"
```

```
var n2 = 12.36;  
console.log( n2.toFixed(1) ); // "12.4"
```

```
var n3 = 12.34;  
console.log( n3.toFixed(5) ); // "12.34000"
```

```
var n4 = 12.34;  
console.log( +n4.toFixed(5) ); // 12.34
```

# Неточные вычисления

```
console.log( 0.1 + 0.2 == 0.3 );
```

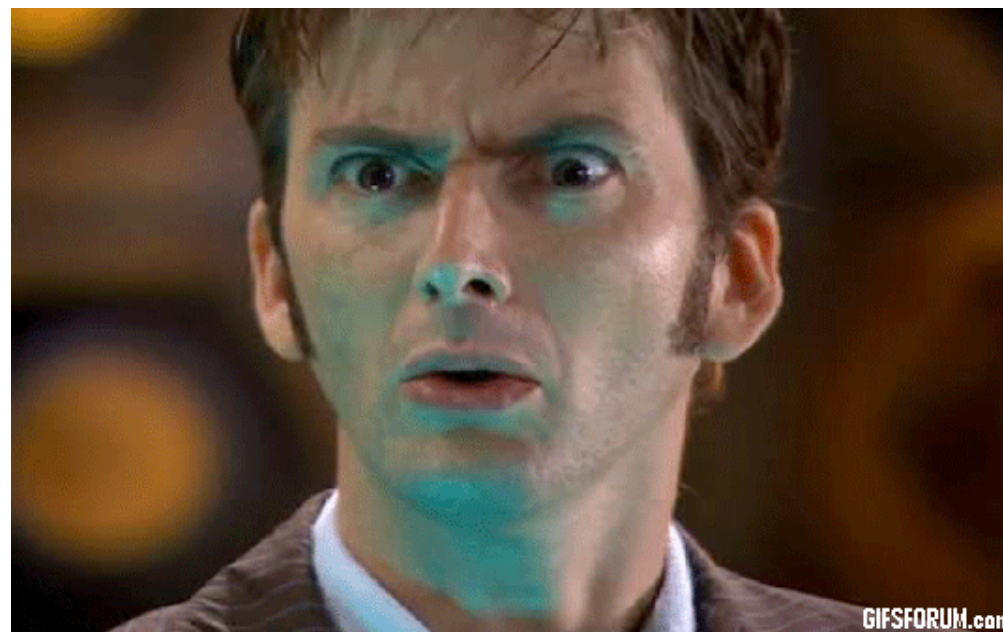
# Неточные вычисления

```
console.log( 0.1 + 0.2 == 0.3 ); // false
```



# Неточные вычисления

```
console.log( 0.1 + 0.2 ); // 0.30000000000000004
```



# Неточные вычисления

```
console.log( (0.1 * 10 + 0.2 * 10) / 10 ); // 0.3
```

```
var result = 0.1 + 0.2;  
console.log( +result.toFixed(10) ); // 0.3
```

# Функции общего назначения

**Math**.sqrt(x)

**Math**.log(x)

**Math**.pow(x, exp)

**Math**.abs(x)

**Math**.exp(x)

**Math**.max(a, b, c...)

**Math**.min(a, b, c...)

**Math**.random()

## Задача 1

Написать функцию которая возвращает случайное число из интервала (0, max)

## Задача 2

Написать функцию которая возвращает случайное число из интервала (min, max)

Uptat. Ut numsan ulla teu

Mod te magnisisi bla faci l

nullaor se core doloborper

nis alit in henisis nim nulp

ero odo

eugiam vulputet, quis niat

Строки

# Создание строк

```
var text = "моя строка";
```

```
var anotherText = 'еще строка';
```

```
var str = "012345";
```

# Специальные символы

```
alert( 'Привет\nМир' );
```

\b Backspace

\f Form feed

\n New line

\r Carriage return

\t Tab

\uNNNN

\u00A9 ©

# Экранирование спец символов

```
var str1 = 'I\'m a JavaScript programmer';
```

```
var str2 = "I'm a JavaScript \"programmer\" ";  
console.log( str2 ); // I'm a JavaScript "programmer"
```



# Длина length

```
var str = "My\n";
```

*// 3 символа. Третий – перевод строки*

```
console.log( str.length ); // 3
```

# Доступ к символам

```
var str = "jQuery";  
console.log( str.charAt(0) ); // "j"
```

```
var str = "Я – современный браузер!";  
console.log( str[0] ); // "Я"
```

# Изменения строк

```
var str = "строка";
```

```
str = str[3] + str[4] + str[5];
```

```
console.log( str ); // ока
```

# Методы toUpperCase и toLowerCase

```
var text = 'Hello GoIt';
```

```
console.log(text.toUpperCase()); //HELLO GOIT
```

```
console.log(text.toLowerCase()); //hello goit
```

# Поиск подстроки

```
var str = "Widget with id";
```

```
// 0, т.к. "Widget" найден прямо в начале str
```

```
console.log( str.indexOf("Widget") );
```

```
// 1, т.к. "id" найден, начиная с позиции 1
```

```
console.log( str.indexOf("id") );
```

```
// -1, не найдено, так как поиск учитывает регистр
```

```
console.log( str.indexOf("widget") );
```

```
// 12, поиск начат с позиции 2
```

```
console.log(str.indexOf("id", 2));
```

# Взятие подстроки

```
var str = "stringify";
```

```
// "s", СИМВОЛЫ С ПОЗИЦИИ 0 по 1 не включая 1.  
console.log(str.substring(0,1));
```

```
// ringify, СИМВОЛЫ С ПОЗИЦИИ 2 до конца  
console.log(str.substring(2));
```

```
// ring, со 2й позиции 4 символа);  
console.log(str.substr(2,4));
```

```
// "s", СИМВОЛЫ С ПОЗИЦИИ 0 по 1 не включая 1.  
console.log(str.slice(0,1));
```

# Взятие подстроки

```
// "testme", -2 становится 0
console.log( "testme".substring(-2) );

// -1 становится 0 -> получили substring(4, 0)
// 4 > 0, так что аргументы меняются
// местами -> substring(0, 4) = "test"
console.log( "testme".substring(4, -1) ); // "test"

// "me", от 2 позиции с конца
console.log( "testme".slice(-2) );

// "estm", от 1 позиции до первой с конца.
console.log( "testme".slice(1, -1) );
```

# Unicode

```
console.log( String.fromCharCode(1072) ); // 'а'
```

```
console.log( "абрикос".charCodeAt(0) ); // 1072, код 'а'
```



# Задача

Написать функцию которая возвращает строку с заглавным первым символом

```
upperFirst('hello') == 'Hello';  
upperFirst('') == ''; // нет ошибок при пустой строке
```

# План занятия

- Рекурсия
- Стек
- Именованные функциональные выражения
- Структуры данных
- Методы и свойства
- Числа строки



**2015**