



# Lesson - 18

# Data structures

SkillUp, by Vitali Cernomschi

# План занятия



1. **Вопросы по домашнему заданию**
2. **Add custom script to NPM**
3. **Data structures**
4. **Primitives vs Objects: копирование по значению и ссылке**
5. **Arguments**
6. **Date and Time**

# NPM scripts



## pre- and post-

```
{  
  "scripts": {  
    "potato": "potato --mash ./index.js"  
    "prepotato": "echo SO HUNGRY",  
    "postpotato": "echo YUM YUM"  
  }  
}
```

npm run potato

```
> prepotato  
SO HUNGRY  
> potato  
mashing...  
done.  
> postpotato  
YUM YUM
```

# Properties and methods



JavaScript allows us to work with primitives (strings, numbers, etc.) as if they were objects.

Primitives:

`string, number, boolean, symbol, null and undefined`

An object

# String



- Properties
  - length, constructor, prototype
- Methods
  - charAt()
  - charCodeAt()
  - concat()
  - endsWith()
  - fromCharCode()
  - includes()
  - indexOf()
  - lastIndexOf()
  - localeCompare()
  - match()
  - repeat()
  - replace()
  - search()
  - slice()
  - split()
  - startsWith()
  - substr()
  - substring()
  - toLocaleLowerCase()
  - toLocaleUpperCase()
  - toLowerCase()
  - toString()
  - toUpperCase()
  - trim()
  - valueOf()

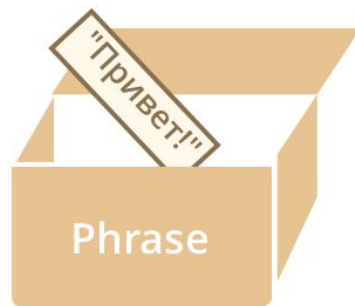
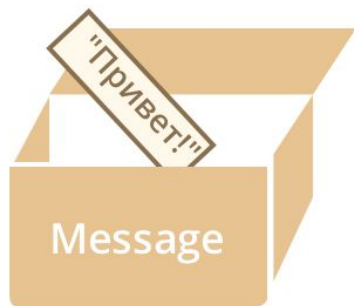
# Number



- Properties
  - constructor
  - MAX\_VALUE
  - MIN\_VALUE
  - NEGATIVE\_INFINITY
  - NaN
  - POSITIVE\_INFINITY
  - prototype
- Methods
  - isFinite()
  - isInteger()
  - isNaN()
  - isSafeInteger()
  - toExponential(x)
  - toFixed(x)
  - toPrecision(x)
  - toString()
  - valueOf()

# Примитивные типы: Копирование по значению

```
1 var message = "Привет!";  
2 var phrase = message;
```



# Объекты: Копирование по ссылке

```
1 var user = { name: "Вася" }; // в переменной – ссылка  
2  
3 var admin = user; // скопировали ссылку
```





# Псевдомассив аргументов "arguments"

## Code:

```
function checkAllArguments(b, c, e) {  
    var a = [ 'first item' ];  
  
    console.log(a);  
    console.log(arguments);  
  
    console.log(typeof a);  
    console.log(typeof arguments);  
  
    console.log(arguments instanceof Object);  
    console.log(arguments instanceof Array);  
}  
  
checkAllArguments('first argument', 'second argument', 'third argument');
```

## Result

```
▶ ["first item"]  
▶ Arguments(3) ["first argument", "second argument", "third argument",  
object  
object  
true  
false
```

# Создание даты



```
new Date()  
new Date(year, month, day, hours, minutes, seconds, milliseconds)  
new Date(milliseconds)  
new Date(date string)
```

# Получение компонентов даты



Method	Description
getFullYear()	Get the <b>year</b> as a four digit number (yyyy)
getMonth()	Get the <b>month</b> as a number (0-11)
getDate()	Get the <b>day</b> as a number (1-31)
getHours()	Get the <b>hour</b> (0-23)
getMinutes()	Get the <b>minute</b> (0-59)
getSeconds()	Get the <b>second</b> (0-59)
getMilliseconds()	Get the <b>millisecond</b> (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

# Установка компонентов даты



Method	Description
setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

# Автоисправление, форматирование и вывод дат

```
1 var date = new Date(2014, 11, 31, 12, 30, 0);
2
3 var options = {
4   era: 'long',
5   year: 'numeric',
6   month: 'long',
7   day: 'numeric',
8   weekday: 'long',
9   timezone: 'UTC',
10  hour: 'numeric',
11  minute: 'numeric',
12  second: 'numeric'
13 };
14
15 alert( date.toLocaleString("ru", options) ); // среда, 31 декабря 2014 г. н.э. 12:30:00
16 alert( date.toLocaleString("en-US", options) ); // Wednesday, December 31, 2014 Anno Domini
```

# Lexical environment

1. Создание `LexicalEnvironment`
2. Выполнение `LexicalEnvironment`
3. Очистка `LexicalEnvironment`

```
1 function sayHi(name) {  
2   // LexicalEnvironment = { name: 'Вася', phrase: undefined }  
3   var phrase = "Привет, " + name;  
4  
5   // LexicalEnvironment = { name: 'Вася', phrase: 'Привет, Вася'}  
6   alert( phrase );  
7 }  
8  
9 sayHi('Вася');
```

# References



1. NPM Scripts:
  - a. <https://docs.npmjs.com/misc/scripts>
2. String data type:
  - a. [https://www.w3schools.com/jsref/jsref\\_obj\\_string.asp](https://www.w3schools.com/jsref/jsref_obj_string.asp)
  - b. [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String)
3. 4. Number data type:
  - a. [https://www.w3schools.com/jsref/jsref\\_obj\\_number.asp](https://www.w3schools.com/jsref/jsref_obj_number.asp)
  - b. [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number)
4. Структуры данных:
  - a. RU: <https://learn.javascript.ru/data-structures>
  - b. EN: <https://javascript.info/data-types>
5. Date:
  - a. [https://www.w3schools.com/js/js\\_dates.asp](https://www.w3schools.com/js/js_dates.asp)
  - b. [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date)
6. Интернационализация:
  - a. <https://learn.javascript.ru/intl>

# Домашнее задание - теоретическая часть

1. Прочитать главу про структуры данных:

EN: <https://learn.javascript.ru/data-structures>

RU: <https://javascript.info/data-types> (except 7,8,9 as it is ES6 related)





# Домашнее задание - практическая часть



1. Напишите функцию, которая сложивает 2 значение и выводит их сумму. Если результат является числом, то его надо вывести с 2 знаками после запятой.
2. Написать unit tests для функции вычисления суммы.

# Домашнее задание - по желанию

1. Напишите функцию `sum(...)`, которая возвращает сумму всех своих аргументов:

```
1 sum() = 0
2 sum(1) = 1
3 sum(1, 2) = 3
4 sum(1, 2, 3) = 6
5 sum(1, 2, 3, 4) = 10
```

- a. **Advanced part:** написать unit tests для функции вычисления суммы.

2. Создайте функцию `getWeekDay(date)`, которая выводит текущий день недели в коротком

```
1 var date = new Date(2012,0,3); // 3 января 2012
2 alert( getWeekDay(date) );      // Должно вывести 'вт'
```

- a. **Advanced part:** написать unit tests для функции вычисления дня недели.