# Front-end development course

# JavaScript

Lecture #5

## Objects

Objects in JavaScript is just collections on key-value pairs. In other programming languages such data structures can be called Associative arrays or Maps.

In other languages, like Java, objects can be constructed only through constructor method on some class, but in JS you can create objects on the fly using “object literal”.

## this

To refer current object there is an identifier in in JavaScript called “***this***”, it may often seem that “***this***” is related to "object-oriented patterns", in JS this is a different mechanism. If a function has a “***this***” reference inside it, that “***this***” reference usually points to an object. But which object it points to depends on how the function was called. It's important to realize that this does not refer to the function itself.

Value “***this***” is called *invocation context* and will be determined at the time the function was called.

Usage of “***this***” guarantees that function is executing in the same object in which context it has been invoked.

There are four rules for how “***this***” gets set, and they're shown in a snippet in *this.js* examples.

## Indirect method invocation

There are three method for indirect method invocation:

- call (context, arg1, arg2, arg3, argN, ...) - invokes a function for a specific context with specific arguments.
- apply (context, [arg1, arg2, arg3, argN, …]) - same as call but arguments passed as array
- bind (context, arg1, arg2, arg3, argN, ...) - creates new function with predefined context for later use

Some tricks on bind(), call() and apply() https://habrahabr.ru/post/199456/

## Prototype

On creation every object get their prototype object. If prototype doesn’t explicitly setted, object get base prototype by default.

Any object can be a prototype of another object, and even that prototype can have it’s own prototype. If prototype has not-null link to the prototype - that band called **“prototype chain”**.

## new and Inheritance

Objects of same class in JavaScript === objects constructed from same prototype.

When constructing object using new JS engine does the following:

- New empty object is created.
- **"this"** get reference to that object.
- Function performs some work, set some properties and methods to that object.
- **"this"** is returned.

Any function in JS can works as a constructor.

## Homework

Homework assignment for this lecture is to implement data structure called Linked List.

All methods with usage examples are listed in **homework/homework.js**

Also in **homework/homework.spec.js** you can find autotests for LinkedList implementation, you can past your implementation to **homework/hw-node.js** and run autotests by executing  following in console:

```
npm install
npm test
```

And check the results in console.

More on Linked List can be found here

## Q&A

# Thank You

Netcracker