Hello, my name is Andrei. I will tell you about Lodash. The topics that I will discover it is:

* What is Lodash?
* Why do I need it?
* How can I use it?
* And few examples working with Lodash.

So, what is Lodash&? **Lodash** is a [JavaScript](https://en.wikipedia.org/wiki/JavaScript) library which provides utility functions for common programming tasks using the [functional programming](https://en.wikipedia.org/wiki/Functional_programming) paradigm. Original author was John-David Dalton. Initial release was in April, 2012. Lodash draws most of its ideas from [Underscore.js](https://en.wikipedia.org/wiki/Underscore.js).

Why do I need it? Lodash provides powerful adult functions which help to coding your program. As well the library help to developer don't think about compatibility between browsers. Thanks to that Lodash can be used in Front-end and Back-end to solve some daily tasks what happened every day. There are some benefits why society of programists does like in Lodash:

* Allow build semantic data flow;
* Help writes pure simple functions;
* Documentation is nice and concise;
* Ease to start with.

Lodash can divide on few families, which contain targeted functions, for examples:

* array;
* collection;
* date;
* function;
* lang;
* math;
* number;
* object;
* seq;
* string;
* utils;
* properties.

One family has own functions to work with something. If you work only with collection you can install just that family.

Ok, go to documentation. Documentation has pretty good explanation of all the functions. Here is have line for searching, the list of function families, link to source where you can see code of function, field arguments it is a data which relay into the function, field what this function will return and the most important some examples how to use.

How can I establish Lodash?

You can download full build or core build of the library. Or just include link to CDN resource to your site. Actually installation too ease, in browser you put one line script. Library can be installed via npm. As you can see you can import whole library or just some piece.

Ok, let’s go ahead. I want to show you how Lodash functions are useful. First function named range, what she do, she create an array of numbers (positive and/or negative) with a step. She accepts three arguments: start point, end point and a step. It is a very useful function for create an array of predefined numbers. I have thought how I would implement the same effect uses only ordinary JS. Only the Array.from method came to my mind. I have an example with him here, but as you see it is a little more difficult to describe than the range function.

Next function is drop, she creates sliced array without N elements from the beginning. She accepts two arguments: array and number of elements that we don’t needed. I have thought, what will be if I send not array but a string. Well, Lodash is still working. He transforms the string to elements of array and continue makes his job.

There are similar function dropRight. From name it is understand that she slice elements from the ending and return the rest. It also works with strings.

Initial, simple function that gets all but the last element of array. Maybe in some cases it needed. The analog this function I think is dropRight without second argument.

Tail, this function is some reverse initial function. Tail function gets all but the first element of array. The analog this function is drop.

Take and takeRight. Take creates a slice of array with n elements taken from the beginning. TakeRight from the ending accordingly. In some sense these functions is reverse of drop functions. Lodash has many such functions what have a reverse effect.

Next section is rounding functions for float numbers. Floor, ceil and round. In the usual JavaScript they also have. But in the Lodash they have a second argument is precision. This is a comfortable thing where you say to what level you want to round. You can also attribute negative precision, which will be very comfortable in the financial area.

Next section is cases function. These functions are intended to transform a string into a specific pattern. camelCase, kebabCase, lowerCase, snakeCase, startCase, uppercase they all are supported in Lodash.

Next will be trim functions. Trim removes leading and trailing whitespace or specified characters from string. Usual JavaScript doesn’t have such function, so you would have to use slice or match functions. As you can see trim removes not only whitespaces but other specific characters entered as second argument from a string. You can remove them only from the beginning or end of the line uses trimStart of trimEnd accordingly.

If we can remove the unnecessary from the line, we can add something to a line. This makes the pad function. You can still add to the beginning or only to the end of the line.

To functions. To functions uses when we want convert one value to another. Look at the first example. We want to translate an object into an array. In a common JavaScrip, we would at least use Object.keys and Object.values. But lodash easily translates one value into another. If you start writing such a function from the beginning, it would take a long time and there would be no assurance that it will work in other browsers.

In conclusion I want to say that Lodash is more oriented for human understanding. This library has an extensive arsenal for any type of data, it is very cool. These functions work with a variety of data types and do not break if they transfer the wrong data type to them. Special respect deserves the fact that it will work the same everywhere. In my speech, I did not show a quarter of the opportunities that Lodash has behind it. Maybe in the future they will be in the usual JavaScript, who knows, time will show. On this I finish, thank you for your attention. That’s all.