

School: №2		
Date: 00.00.0000	Teacher's name: Zhakipbayev Abai Kazbekovich	
Grade:	Number present:	absent:
Topic of the lesson: Variables and Data Types		
Learning objective(s) that this lesson is contributing to	Children will learn how to run JavaScript and understand the basics of working with JavaScript	
Lesson objectives	<p>All learners will be able to:</p> <ul style="list-style-type: none"> • Run JavaScript and build the first program <p>Most learners will be able to:</p> <ul style="list-style-type: none"> • Understand how variables work <p>Some learners will be able to:</p> <ul style="list-style-type: none"> • Will be able to work with different operations 	
Assessment Criteria	<p>Literacy - 1</p> <p>Building task - 1</p> <p>Work with variables - 1</p> <p>Pair Work - 1</p> <p>Team work - 1</p>	
Value links	Respect partner and group, understand the importance of time	
Previous learning	<p>How to run JavaScript</p> <p>Lines and numbers</p> <p>Variables</p> <p>Assignment operation</p>	
Cross curricular links	Mathematics, computer science, visual arts	
Time	Planned activities	Resources
	<p>1. Repeat JavaScript</p> <p>2. Card of five questions</p> <p>3. New topic</p> <p>4. Grading</p> <p>5. Homework</p>	
Beginning 5 min	<p>Before starting the lesson, I want to ask if you watched the video given as a homework, if so, please answer me the following questions (Five-card cards are given to children, and each correct answer is rated at 1 point)</p> <p>1. What is JavaScript?</p> <p>2. What is an interpreter?</p> <p>3. What is compilation?</p> <p>4. What can JavaScript do?</p> <p>5. What does JavaScript not know?</p>	<p>A computer</p> <p>A board</p> <p>A projector</p>
Middle 30 min	<p>JavaScript is intended to be run in a browser along with HTML and CSS. But, if these languages are intended for layout of the site structure, then JavaScript allows you to 'revive' web pages - make them responsive to user actions or demonstrate some dynamism (for example, changing pictures in a block or beautiful smoothly drop-down menus).</p>	

How to run JavaScript

There are two ways to write and run JavaScript: the first is that we write the code directly on the HTML page inside the <script> tag:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title> It's the title </title>

    <script>

      var name = 'Dimash';

      alert('Hello, '+name);

    </script>

  </head>

  <body>

    This is the main content of the page.

  </body>

</html>
```

The <script> tag can be placed anywhere on the page - both in the <head> and in the <body>.

The second option is that the JavaScript code is stored in a separate file (like CSS) and is also connected using the <script> tag with the src attribute, which indicates the path to the script file:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title> It's the title </title>

    <script src="path to a file with a script "></script>

  </head>

  <body>

    This is the main content of the page.

  </body>

</html>
```

JavaScript basics

Lines and numbers

The simplest JavaScript **data types** are **strings and numbers**.

Numbers denote themselves: 1, 12, 145, but **strings** must be quoted (single or double - no difference):

'string', "string"; // these are examples of strings

Variables

One of the most important and common objects in programming is a variable. A variable is an object that can store various data inside itself, for example, strings or numbers.

The variable name must consist of English letters: large or small, as well as numbers and the underscore.

In JavaScript, when declaring a variable, the **var** keyword must be written:

```
var a; // here we have announced a variable
```

```
var a, a1, isVar, is_var; // here we've announced a group of variables
```

Assignment operation

A very important element of programming is the assignment operation. Assignment Example:

```
var a = 4; // we set variable a to 4
```

Comments

In JavaScript code, as well as in HTML and CSS, you can leave comments. They can be multi-line and single-line:

```
var a = 4; // this is an example of a single line comment.
```

```
/*
```

This is an example of multi-line comment.

```
*/
```

```
var a = 4;
```

Comments are ignored by the browser when the code is executed, you can leave any notes in them or temporarily close the code from execution, so that you can return it later (if necessary).

Alert function

In JavaScript, there is a special alert function that allows you to display any text in the browser window in the form of a dialog box.

The following code displays a given text:

```
alert('Hello, World!'); // displays the phrase 'Hello, World!'
```

[Click on this link](#) to see such a window.

And in the following text code, a phrase is assigned, and then the contents of that variable are displayed on the screen:

```
var text = 'Hello, World!';
```

```
alert(text); // displays the phrase 'Hello, World!'
```

Math operations

In JavaScript, you can perform various mathematical operations between numbers:

```
alert(2 + 3); //displays 5
```

```
alert(5 - 1); //displays 4
```

```
alert(2 * 3); //displays 6
```

```
alert(6 / 2); //displays 3
```

Getting a specific string character

In JavaScript, you can access a specific character in a string by its number in this way: a [n] is the nth character of the string (note that numbering comes from scratch):

```
var a, b; // let's announce our variables
```

```
a = 'abcde'; // in the variable a will be stored the value 'abcde'
```

```
b = a[0]; // Variable b will be 'a'
```

```
b = a[1]; // Variable b will be 'b'
```

```
b = a[4]; // Variable b will be 'e'
```

Assignment Difficulties

Very often, newcomers do not understand that assignment is different from ordinary equality. See the following example:

```
var a = 1;
```

```
a = a + 2;
```

From a math point of view, the record is absurd, but not in terms of programming. In this case, variable **a** had a value of **1**, and then we assigned a variable **a** to a new value - the old variable value **a** plus **2**.

Increment and decrement operations

Operation **a++** or **++a** - increments the variable **a** by one. This operation is called **increment**.

The operation **a--** or **--a** - reduces the variable **a** by one. This operation is called **decrement**.

Examples:

```
var a = 1;
```

```
a++; // Increase a by 1, which corresponds to the code a = a + 1;
```

```
alert(a); //shows 2
```

```
var a = 1;
```

```
a--; // reduce a by 1, which corresponds to the code a = a - 1;
```

```
alert(a); //shows 0
```

Let's see in what cases the difference between **++a** and **a++** is manifested.

Suppose we have an alert code (**++a**) and an alert code (**a++**).

In the first case, the variable will first increase by one, and then it will be output, and in the second case, it will first be output and then increase.

Operations + =, - =, * =, / =

We have already looked at code that demonstrates difficulties with the assignment operation:

```
var a = 2;
```

```
a = a + 3;
```

In this case, we assign to the variable a its current value, increased by 2. However, JavaScript allows us to write this code even shorter using the += operator:

```
var a = 1;
```

```
a += 3; // this code is completely equivalent to the code a = a + 3;
```

In addition, there are operators -, *, /, which are equivalent to the following code:

```
var a = 2;
```

```
a -= 3; // this code is completely equivalent to the code a = a - 3;
```

```
var a = 2;
```

```
a *= 3; //this code is completely equivalent to the code a = a * 3;
```

```
var a = 2;
```

```
a /= 3; //this code is completely equivalent to the code a = a / 3;
```

Special values

In JavaScript, as in other programming languages, keywords exist for some special meanings. Here they are: undefined, null, true, false, NaN, Infinity, -Infinity.

Undefined and null values

An undefined value indicates uncertainty. For example, if we try to access a variable to which we have not yet set a value, then its value will be undefined.

```
var a;
```

```
alert(a); //shows undefined
```

A null value means 'nothing.' For example, we can set the variable to null to indicate that nothing is there.

This value is very similar to undefined, the difference is that undefined is not a specific value, and null is a specific value - nothing.

True and false

The true and false values indicate true and false, respectively. They are used for things that suggest two possible answers - yes or no.

For example, to the question 'are you already 18 years old?' you can answer yes, that is true, or not, that is false.

NaN value

The value NaN (Not-A-Number) means not a number. It can happen, for example, in this case - when you multiply a string with letters by a number:

```
alert('abc'*3); //shows NaN
```

Infinity and -Infinity Values

Infinity and **-Infinity** mean infinity and minus infinity, respectively. They are obtained if some number is divided by zero - in this case, JavaScript does not produce an error, as in other programming languages, but returns these values.

If we divide a positive number by zero, we get **Infinity**, and if negative - then **-Infinity**.

Prompt function

In addition to the alert function, which displays a dialog box, there is a **prompt** function that not only issues a window with text, but also allows you to receive any text from the user.

This text can be written into a variable and then perform any operations on it.

In the following example, we ask for the username, write it to the variable **name**, and use the **alert** function to display:

```
var name = prompt('Your name?');
```

```
alert('Your name'+name);
```

Confirm function

If you just need to ask the user 'Yes' or 'No', without giving him the opportunity to enter a different text - use the **confirm** function.

This function brings up a window with a question that needs to be answered by the user, and two buttons for answering: with the 'OK' button and with the 'Cancel' button.

If the user clicks 'OK', then the function will return **true**, and if 'Cancel' will return **false**.

In the following example, the confirm function will display a dialog box asking “Are you 18 years old?”

If you click 'Ok', then true will be written into the ok variable and the alert function will be displayed, and if you click 'Cancel', then **false**:

```
var ok = confirm('Are you 18 years old?');
```

```
alert(ok);
```

Typing Variables

What happens if you try to multiply, for example, a number and a string, like this: $3 * '3'$? As a result, you get the number 9. This means that JavaScript automatically performs type conversion when necessary, you do not need to worry about it.

However, there is a nuance: if we try to **add** a string and a number, then JavaScript will add them as strings, and not as numbers, like this: $'3' + 3$ will get the string '33', not the number 6.

In the case of, for example, with multiplication, JavaScript understood that it was impossible to multiply strings, so it translated strings into numbers and multiplied them. And the case of addition can be interpreted in two ways: add as strings or as numbers (plus it is used both for adding strings and numbers).

You can deal with this in the following way: you need to do an operation that is invalid for strings, for example, like this: $+ '3' + 3$ - put a plus in front of the string and it will be converted to a number.



The second option is this: you can tell javascript that we want to explicitly convert the string to a number. This is done using the **Number** function, like this: **Number('3') + 3**. The result is 6, not '33'.

Not only strings can be converted to numbers, but also any other data types, for example true can also be converted to a number in this way: **Number(true)**.

You can convert to other types using the functions **Boolean**, **String** and other similar ones.

**End
5 min**

Hometask: <http://old.code.mu/books/javascript/base/osnovy-yazyka-javascript-dlya-novichkov.html>

	<p style="text-align: center;">Ваша оценка урока 😊</p>  <ol style="list-style-type: none"> 1. Что я буду применять на практике 2. Самое интересное 3. Самым трудным было... 4. Меня удивило... 5. Мое пожелание себе 		
<p>Differentiation – how do you plan to give more support? How do you plan to challenge the more able learners?</p>	<p>Assessment – how are you planning to check learners' learning?</p>	<p>Health and Safety</p>	
<p>http://old.code.mu/books/javascript/base/osnovy-yazyka-javascript-dlya-novichkov.html</p>	<p style="text-align: center;">Ваша оценка урока 😊</p>  <ol style="list-style-type: none"> 1. Что я буду применять на практике 2. Самое интересное 3. Самым трудным было... 4. Меня удивило... 5. Мое пожелание себе 	<p>While working with a computer with children, we will conduct an exercise for the eyes.</p>	

