Values and Variables

Introduction to Values and Variables

In this section, we begin to learn the JavaScript language, starting with values and variables. We will use persons as examples, such as a person's name, age, or job.

Understanding Values

A value is basically a piece of data. It is the most fundamental unit of information in programming. For example, the text "Jonas" is a value. Similarly, the number 23 is a value. Mathematical operations combine values to form new values, such as $40 + 8 + 23 - 10 = 61$. A value is the smallest unit of information in JavaScript.

javascript Code Sample

```
console.log("Jonas");
```

javascript Code Sample

```
console.log(23);
```

javascript Code Sample

```
console.log(40 + 8 + 23 - 10);
```

Storing Values in Variables

One extremely useful thing we can do with values is to store them into variables. This allows us to reuse them over and over again. For example, we can declare a variable and assign a value to it.

javascript Code Sample

```
let firstName = "Jonas";
```

This is called declaring a variable. It creates a real variable in the computer's memory and stores the value inside that variable. For example, we can declare a variable called JS and assign the value "amazing" to it.

javascript Code Sample

```
let JS = "amazing";
```

Variables are like boxes in the real world that can hold objects, such as a book. We can write a label on the box to describe what is in it, and then find the object later by using that label. Variables work in the same way. Here, we have a box called **firstName**, and into that box, we put the value "Jonas". To use this value, we simply use the variable name.

javascript Code Sample

```
console.log(firstName);
```

If we change the value of **firstName** to "Bob", then the output will be Bob. We can use the variable name multiple times, and whenever JavaScript sees the variable name, it replaces it with the original value assigned to the variable. This is extremely useful because if we want to change the value, we only need to do it in one place.

javascript Code Sample

```
firstName = "Matilda";
```

Variable Naming Conventions and Rules

Now that we know what a variable is, let us discuss conventions and rules for naming variables. We should not give random names to variables. The way we named the variable here is called camelCase. CamelCase means that when there are multiple words in a variable name, the first word is lowercase and all subsequent words start with an uppercase letter.

javascript Code Sample

```
let firstNamePerson;
```

Other naming styles include using underscores, which is popular in other languages like Ruby.

javascript Code Sample

```
let first_name;
```

In JavaScript, camelCase is the standard. Usually, variable names are written using camelCase notation.

Rules for Variable Names

Variable names cannot start with a number. For example, the following is illegal:

javascript Code Sample

```
let 3years = 3;
```

If we try to load this in JavaScript, we get a syntax error. Variable names can only contain numbers, letters, underscores, or the dollar sign. Symbols like ampersand (&) are not allowed.

javascript Code Sample

```
let jonas_and_matilda = "JM";
```

Another error occurs when we try to name a variable using a reserved JavaScript keyword, such as **new** or **function**.

javascript Code Sample

```
let new = 27;

let function = 1;
```

To fix this, we can start the variable name with an underscore or a dollar sign. These are the only two symbols allowed besides letters and numbers.

javascript Code Sample

```
let _function = 1;

let $function = 1;
```

The word **name** is kind of reserved but still allowed to use. However, it is better to use more descriptive names like **firstName**.

javascript Code Sample

```
let name = "Jonas";
```

More Naming Conventions

We should not start a variable name with an uppercase letter. This is a convention, not a rule. Variable names with an uppercase letter are used for specific cases in JavaScript, such as object-oriented programming. For now, always start variable names with a lowercase letter.

javascript Code Sample

```
let Person = "Jonas";
```

Variables that are all uppercase are reserved for constants that will never change, such as the number PI.

javascript Code Sample

```
const PI = 3.1415;
```

If it is a real constant, write it in uppercase. This is a common convention in programming.

Descriptive Variable Names

It is important to make sure that variable names are descriptive. This helps write cleaner code. Variable names should make it easy to understand what value the variable is holding just by reading the name.

javascript Code Sample

```
let myFirstJob = "programmer";

let myCurrentJob = "teacher";
```

This is much better than using names like **job1** and **job2**.

javascript Code Sample

```
let job1 = "programmer";

let job2 = "teacher";
```

It is much easier to understand what **myFirstJob** and **myCurrentJob** mean just by looking at the variable names. This approach is better for code readability.

Recap: What is a Variable?

A variable is basically a box into which we can store a value. We give the box a name, such as **firstName** or **myFirstJob**, and then store a value in it. Later in the code, we can reference that variable over and over again.

javascript Code Sample

```
console.log(myFirstJob);
```

If we change the value of **myFirstJob** to "coder", it will change across the entire program.

javascript Code Sample

```
myFirstJob = "coder";
```

Now we get "coder" as the output. This is what variables are all about. They are one of the most important things in programming, so make sure to really understand them before moving on.

Key Takeaways

- Values are the most fundamental units of information in programming and can be stored in variables for reuse.

- Variables act as labeled boxes that store values, allowing for easy reference and modification throughout the code.
- JavaScript variable naming conventions include camelCase, and there are specific rules about allowed characters and reserved keywords.
- Descriptive variable names improve code readability and maintainability.