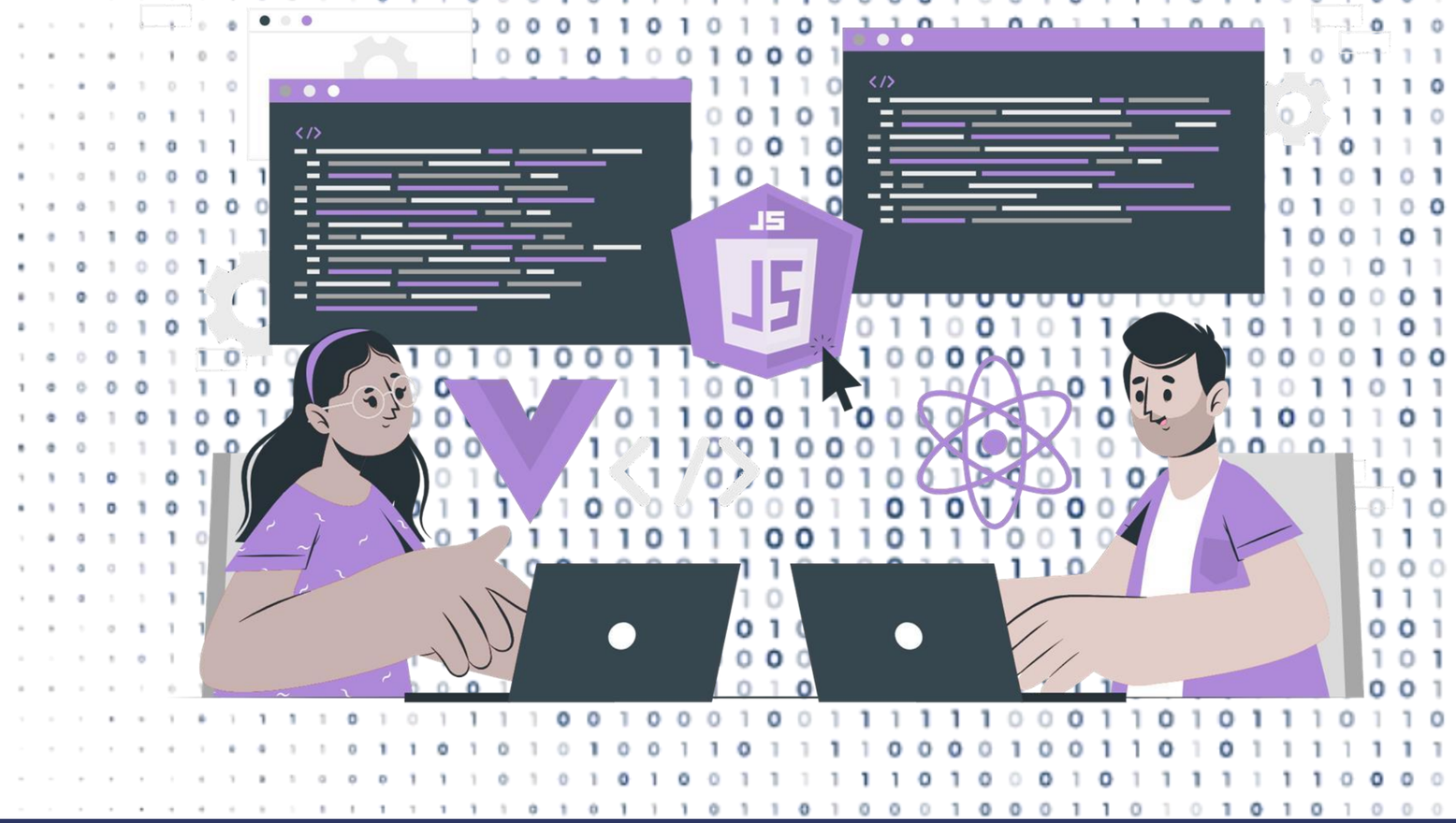
Variables and typeof



Lecture CheckList

1. Introduction to variables.
2. Creation of variables.
3. Naming variables in Javascript.
4. Assigning values to variables.
5. Introduction to typeof.
6. Benefits of typeof.

Introduction to variables.

Variables are like containers, they are used to hold the information we'll need when programming. Variables store data of any datatype that can be used throughout a program.

Variable means anything that can vary. Variables hold the data value and it can be changed anytime we want.

Creation of variables.

Creating a variable is also called declaring a variable. There are four ways to create a variable in JavaScript.

1. var keyword.
2. let keyword.
3. const keyword.
4. No definition.

Creation of variables.

Creating a variable is also called declaring a variable. There are four ways to create a variable in JavaScript.

1. var keyword.
2. let keyword.
3. const keyword.
4. No definition.

Naming variables in JavaScript

1. Variable names should begin with either a letter or an underscore or a dollar sign.
2. Variable names should not begin with numbers or special characters except the underscore and dollar signs.
3. Keywords are reserved words that have a specific meaning and cannot be used as variables. Keywords like if, else, for should not be used as variable names.
4. Variable names are case-sensitive. That means name and Name are different variable names.

Naming Conventions

To ensure consistency in naming your variables you can adopt one of the following naming conventions in naming your variables.

JS

```
var name1;
var name1 = value1;
var name1 = value1, name2 = value2;
var name1, name2 = value2;
var name1 = value1, name2, /* …, */ nameN = valueN;
```

Assigning values to a variable.

Storing data in a variable is also called assigning a value to a variable. To store data in a variable(assign value to a variable), use the = symbol. Place the variable name on the left side of the = symbol and place the value you want to store in the variable goes on the right side of the = symbol.

The = symbol is called the assignment operator.

Variables can be created before assigning values to them.

Whenever you create a variable without assigning a value to it, by default javascript stores undefined [absence of the value].

JS

```
var bla;
bla = 2;
```

Initializing a Variable.

Values can also be assigned to variables at the moment of creating them. Creating variables and assigning values to them at the same time is known as initializing a
variable.

JS

```
var a = 1;
let a = 2; // SyntaxError: Identifier 'a' has already been declared
```

JS

```
let a = 1;
{
  var a = 1; // SyntaxError: Identifier 'a' has already been declared
}
```

JS

```
var a = 1;
{
  let a = 2;
}
```

typeof

The "typeof" operator is a JavaScript operator that allows you to check the data type of a given variable. It can be used with any data type, including objects, arrays, and even null values.

typeof operator is very useful for determining the data type of a given variable. In addition, it can also be used to check for null values.

Benefits of using typeof

There are a few benefits to using the "typeof" operator in JavaScript.

- It is a convenient way to check if a variable is of a certain data type without having to check for conditions.
- It can be used as a debugging tool to help check why a particular piece of code is not working as expected if any datatype issues.
- It can help prevent errors in the code by giving a clear understanding of the data types that are being used.

How can the "typeof" operator be used to detect errors?

The "typeof" operator can be used to detect errors in the code. If we try to access a variable that has not been declared, we will get an error message. This is because "typeof" returns "undefined" for undeclared variables.

THANK YOU