Variables - Storing information

Storing information

In code, variables are used to store specific information in C#. We need variables to keep track of all kinds of information that our applications need.

Imagine a game of chess for example, we would need variable to store:

- All the pieces that's part of the game
- And their positions
- Whose turn it currently is

We can then read the information so that we can display the chess board to the player and also change the stored information in those variables when a player makes a move.

Variable definition, Assignment and Access

```
1 {}  string user;                // Definition
2     user = "Fredrik";           // Assignment
3     // Access the variable
4     Console.WriteLine(user);    // Prints Fredrik
5     Console.WriteLine("user");  // Prints user
```

A definition of a variable:

- Used to store data
- Consists of a type and a name (we will cover types in the next exercise)
- This is how you **declare** a variable
    - **string user;**

This declared variable now needs to be **assigned** a value before being **accessed**:

- **user = “Fredrik”;**

Then you can **access** its value:

- **Console.WriteLine(user);**

Notice that this is **different** for when you’d pass the variable name **WITH** quotation marks.

- **Console.WriteLine(“user”);**

Variable Initialization

```
string trinket = "The One Ring";
Console.WriteLine(trinket); // This prints The One Ring

string book = "Harry Potter";
Console.WriteLine(book);    // This prints Harry Potter

book = "Kungsgatan";        // Reusing the previous variable
Console.WriteLine(book);    // This prints Kungsgatan
```

Whenever a variable is assigned (=) a value for the first time, we call it **initialization**.

Before a variable is initialized we cannot access it:

- **string pet;**
- **Console.WriteLine(pet);**

As it produces an error, and you have to assign a value prior to using it.

You can directly **initialize** the variable when defining it:

- **string trinket = “The One Ring”;**

And you can also **assign** a new value at anytime:

- **trinket = “common earring”;**

Goal

Try to achieve this output by following these steps

```
Hello
World
World
Forsbergs
World
```

- Create a new console project named E5Variables
- Assign the value **“Hello”** to a new variable named **a**
- Output it to the console
- Assign the value **“World”** to the same variable
- Output the variable to the console again
- Assign the value of **a** to a new variable named **b**
- Output **b** to the console
- Assign the value **“Forsbergs”** to **a**
- Output **a** to the console
- Output **b** to the console

Feel free to put a breakpoint on the first line of code and then execute the application step by step.

Keep an eye on the local variables Window and see how the value of the variable changes at each step.