

Variables

Definition: A named piece of memory that stores information

Dim someVariable as Integer



01001000101010001101010100101001010100101010100101
01010101010100101010101011001010101000101010001001
01001001010100010110001010101100101001010100101010
00011010101001010010101001010101010010101010101010
01010101010110010101010001010100010010100100101010
00101100010101011000001101010100101001010100101010
10010101010101010100101010101011001010101000101010
00100101001001010100010110001010101100000110101010
01010010101001010101001010101010101010010101010101
10010101010001010100010010100100101010001011000101
01011001000101101010010100101010001010101010010101

Think of a variable as container for information

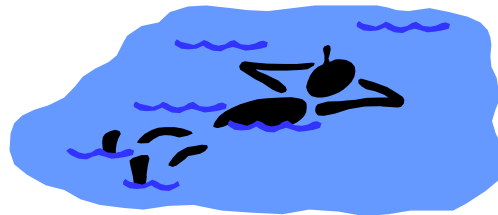
Imagine 3 types of containers: A cup, a paper bag and a swimming pool.

Each of these containers has different characteristics.

Size is one of the more obvious characteristic that makes a cup different from a swimming pool.

All can hold objects but common sense tells us that some objects are better stored in one particular type of container than another.

Do you really want to store your morning coffee in a swimming pool or imagine a cool refreshing swim in a paper bag full of water.



The choice of a variable(container) depends on the quantity and type of information you wish to store in that variable.

To create a variable(container) you might write something like this:

```
Dim myAge As Integer
```

What this means is that some variable has been created that will hold values of type integer.

It is very important to understand the difference between the variable **type** and the variable **name**.

There can be many identical variable types used in a program but each variable name must be unique.

Types versus Names

Lets go back to our container analogy for a moment.

Imagine a glass as a variable type. When you think of a glass you think of a container with certain characteristics.



What if you had 3 glasses though? How would you know what particular glass one was talking about?

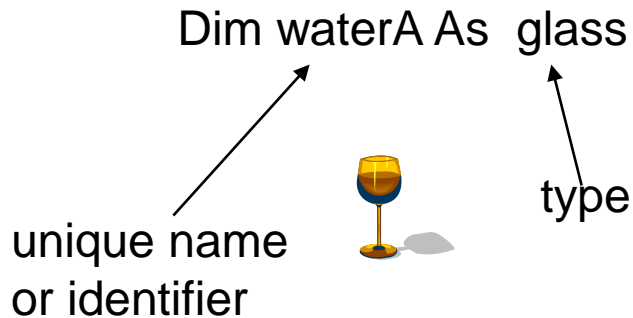


This is where the naming of specific variables(glasses) each with a unique name becomes very important.

In order to create a variable we must know at least 2 pieces of information about the variable.

We must know its type, in this case 'glass' and we must have a unique name for each specific glass that we have.

We give this information by writing the name of the type followed by a unique name (this is often called the identifier).



Now if we want another glass we simply use the same type name followed by different identifier.

Dim waterB As glass

We now have 2 specific variables, one called *waterA* and the other called *waterB* both of the type glass.

We call this process , **declaring a variable**

Declare the Following

- Declare a variable of type glass and call it waterC
- Declare a variable of type car and call it myWheels
- Declare a variable of type book and make up a suitable identifier for it.

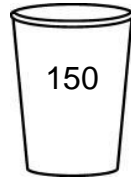
Returning to our water glasses, there is nothing in these glasses. They have been declared which means that space has been reserved for information to be put in but this space is still empty.

Once we have declared our variable...

Dim waterA As glass

...we can add water to our glass by simply writing the following.

waterA = 150



Assigning a value (in this case pouring 150 ml water into our glass) to a variable is also known as **initializing a variable**.

Your Turn.....

Create a container of type glass called waterB and fill it with 250 ml of water

Or

Using the terminology that we have learned over the past few slides we could restate the previous statement like this:

Declare a variable of type glass, call it waterB and initialize it with a value of 250

Dim waterB As glass

waterB = 250

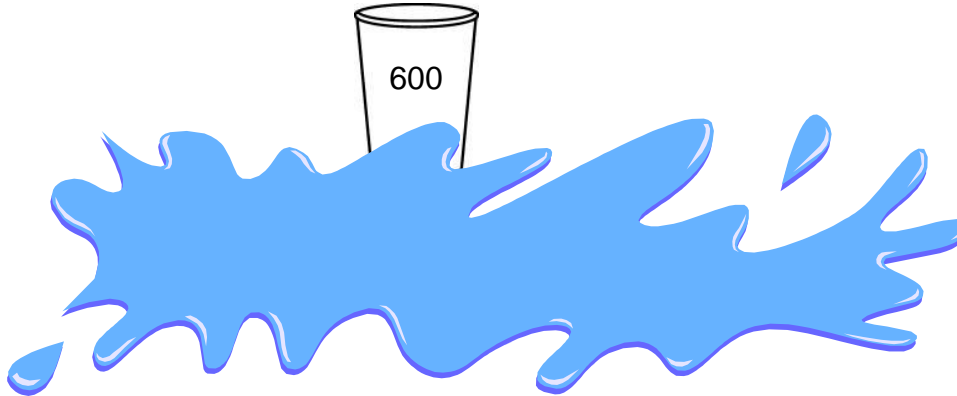


Variable Characteristics

When choosing a variable to hold information, it is important to know the limits or constraints of the variable.

Perhaps variables of type glass have a capacity of 500 ml.

Any attempt to put more than 500 ml in the glass and.....



...you'd better have a cloth handy!

Lets create a 3rd variable of type glass called waterC

We will not initialize it at this point but rather will leave it empty.

Dim waterC As glass

One very interesting characteristic about variables is that we can perform operations on them. Different variable types allow different operations.

The variable names become synonymous with the information they contain. Think carefully about this statement as we recall the values that waterA and waterB were initialized with.

The variable names become synonymous with the information they contain....

We assigned a value of 150 to waterA and a value of 250 to waterB.

Now we could write

$$\text{waterC} = \text{waterA} + \text{waterB}$$

Now waterC is a variable that contains a value of 400.

It is important to note that this operation is possible because waterA, waterB and waterC are all variables of the same type; glass.

A integer, being a variable, is a type of container. A byte rather than containing water as in the glass container analogy, holds integers values between -1.2 billion and +1.2 billion

Since an integer is the type of variable, declare the variable name(the identifier) followed by the type.

```
Dim myAge as Integer
```

We then have the option of assigning a value to the variable.

```
myAge = 16;
```

Camel or Hungarian Notation

With variable names `myAge` and `yourAge` you may have noticed the capital letter in the middle of the name. Variable names should provide a hint as to what kind of information will be contained in it.

Variables must not have any spaces in them.

By using capital letters at the beginning of each new word within a variable name, it is possible to use long descriptive names that are easily read without any spaces in the name. This sort of naming is referred to as camel or sometimes Hungarian notation. Long names with capitals in them have a kind of bump effect like a camel's hump.-I have no idea why it is sometimes called Hungarian notation.

`hereIsAnExampleOfALongNameUsingHungarianOrCamelNotation`

Given the following;

Dim myAge as Integer

Dim yourAge as Integer

myAge = 16

yourAge = 19

Dim differenceInAge as Integer

differenceInAge = yourAge – myAge;

What is the value of differenceInAge?

Your Turn....

- Using the following variable declarations.
- Write a line of code that will tell us how much savings you might have after you have spent some of your allowance.

Dim allowance as Integer

allowance = 25

Dim costOfMovie as Integer

costOfMovie = 8

Dim moneyLeftOver as Integer

In an earlier slide we stated that a variable became synonymous with the value it contains. To prove it, write the following code and observe the results.

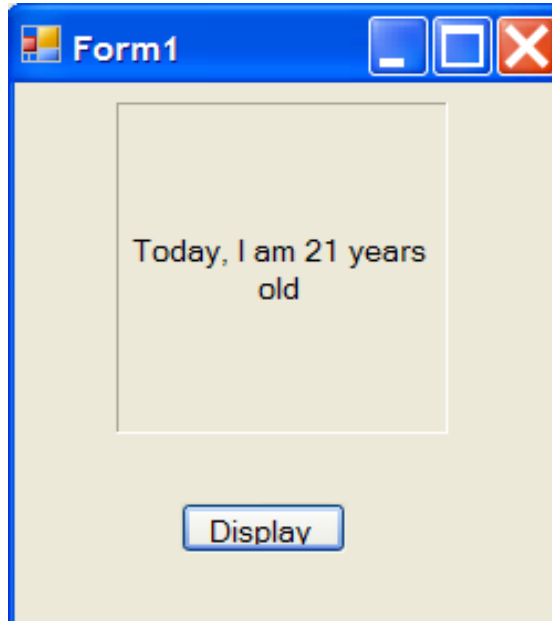
```
Dim myAge as Integer
```

```
myAge = 21
```

```
lblDisplay.Text = "Today, I am " + myAge + " years old"
```

What is the output of the above program?

Combining Text and Variables



```
Private Sub btnDisplay_Click(ByVal sender As System.Object,  
    Dim myAge As Integer  
    myAge = 21  
    lblDisplay.Text = "Today, I am " & myAge & " years old"  
  
End Sub
```

The variable 'myAge' became synonymous with the value it was initialized with, in this case **21**. We write in the variable 'myAge' in the code and we get **21** in the output.

Notice how we combine text and variables within the same program.

The characters that we write in quotes get output to the screen. By using the ‘&’ operator, we can combine these with the values of various variables.

Here is another example of combining text and variables.

```
Dim allowance As Integer
```

```
Integer = 25
```

```
“The number of dollars that my parents gave me today  
was “ & allowance
```

Notice the space between was and the final quotation. This space is necessary so that was and 25 don’t get squished together in the output.

Write this code in a short program called VariablePlusText.

HomeWork

Write a program called 'Money' which takes a variable called 'earnings', give it a value of 87.

Declare another variable called 'expenses' and give it a value of 25.

Output a variable called 'savings' to the screen by subtracting expenses from earnings.