Writing a FOR Loop

Created by Math Works for
Structural Dynamics

Math Works

(Refer Slide Time: 00:05)

```
for k = 1:99
    value(k+1) = (1+r)*value(k);
end

ok
```
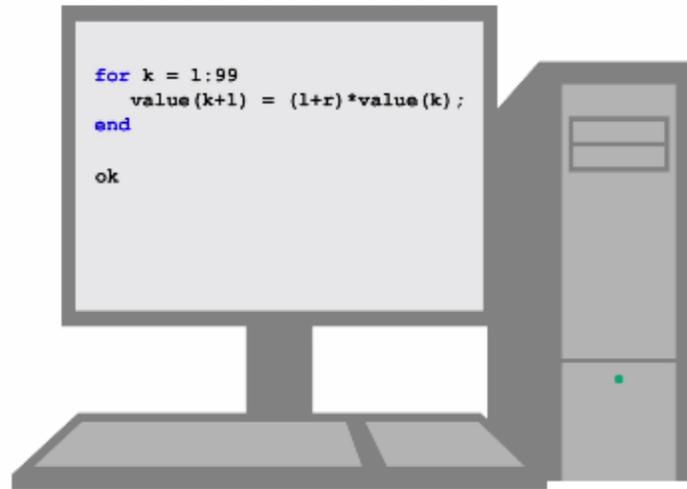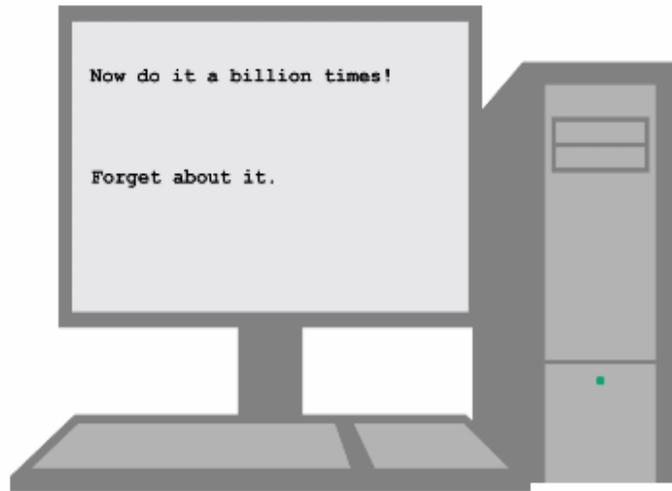
One of the big advantages computers have over humans is that.

They do not complain when asked to repeat a task hundreds thousands even millions of times.

(Refer Slide Time: 00:14)

Now do it a billion times!

Forget about it.

Well as long as you ask them the right way.

(Refer Slide Time: 00:17)

Savings Acount Balance

$$New\ Balance = (1 + r)(Current\ Balance)$$

Take the task of modeling the growth of a savings account balance over a hundred years sure we could compute each and every balance one after another but who has the time to do that, instead let us see how we can use a for loop and MATLAB to automate tasks like this.

```
                    MATLAB Commands
r = 0.02;
balance = zeros(1,100);
balance(1) = 20000;
balance(2) = (1+r)*balance(1);
balance(3) = (1+r)*balance(2);
balance(4) = (1+r)*balance(3);
balance(5) = (1+r)*balance(4);
```

First let us set up some initial variables we have the interest rate R then we are going to need somewhere to save the computed balances for the next 100 years, so we create a vector of zeroes balance, we will also assign the first element of balance to be the initial deposit amount, using the formula we can compute and store the second balance and the third and the fourth and the fifth but we need to do this for a hundred years.

```
MATLAB Commands
r = 0.02;
balance = zeros(1,100);
balance(1) = 20000;
for k = 1:99
    balance(k+1) = (1+r)*balance(k);
end
```

Let us see how we can automate this with a for loop.

FOR Loop

```
for k = 1:99
    balance(k+1) = (1+r)*balance(k);
end
```

A for loop allows a command or collection of commands to be repeated a specified number of times.

(Refer Slide Time: 01:14)

Loop Keywords

```
for k = 1:99
    balance(k+1) = (1+r)*balance(k);
end
```

A for loop is denoted by the loop keywords for and end which specify the start and end of the for loop.

Loop Variable

```
for k = 1:99
    balance(k+1) = (1+r)*balance(k);
end
```

The for key word is followed by the loop variable which changes value to cover each entry in a range in this case the loop variable K will assume integer values from 1 to 99.

Loop Body

```
for k = 1:99
    balance(k+1) = (1+r)*balance(k);
end
```

The remaining commands in between the four and end keywords are called the loop body.

$$k = 2$$

```
for k = 1:99
    balance(2) = (1+r)*balance(1);
end
```

⊞ **balance**   [20000,20400,0,0,0...0]

When the for loop is executed the command in the loop body will be executed for every value of K in the range, so first k starts off as one and the command executed is next k will assume the value too.

$$k = 2$$

```
for k = 1:99
    balance(3) = (1+r)*balance(2);
end
```

▦ **balance** [20000,20400,20808,0,0...0]

And the command becomes and so on.

```
        k = 99

    for k = 1:99
        balance(100) = (1+r)*balance(99);
    end

balance  [...,136536,139267,142052]
```
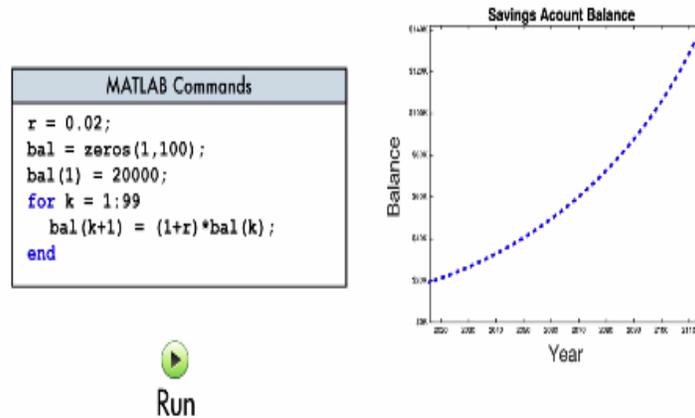
The loop body repeats 99 times with the balance being updated each time giving us the account balances for the first100 years.

(Refer Slide Time: 02:07)

Now that we have completed the loop we are ready to run our code and analyze the results, great looks like we will septuplet the original deposit, now I just have to live to be a hundred and forty.

MathWorks