Loops and conditional statements

Table of Contents

Ex. 1: The *for* loop: Using for command

```
b = 3;
for k = 1:5 % start at 1, stop at 5
     b
end
```

```
b = 3
b = 3
b = 3
b = 3
b = 3
```

Ex. 2: The *for* loop: Utility of the dummy index

```
b = 3;
for k = 1:5 % start at 1, stop at 5
     b^k
end
```

```
ans = 3
ans = 9
ans = 27
ans = 81
ans = 243
```

```
sum1 = 0;
for k = 1:9 % start at 1, stop at 9
     sum1 = sum1 + k
end
```

```
sum1 = 1
sum1 = 3
sum1 = 6
sum1 = 10
sum1 = 15
sum1 = 21
sum1 = 28
sum1 = 36
sum1 = 45
```

```
sum1 = 0;
for k = 1:2:9 % start at 1, check every 2nd element, stop at 9
     sum1 = sum1 + k
end
```

```
sum1 = 1
sum1 = 4
sum1 = 9
sum1 = 16
sum1 = 25
```

Ex. 3: The *for* loop: Treatment of array within a loop

```
b = [3 8 9 4 7 5];
sum1 = 0;
for k = 1:4 % start at 1, stop at 4
 sum1 = sum1 + b(k)
end
```

```
sum1 = 3
sum1 = 11
sum1 = 20
sum1 = 24
```

```
b = [3 8 9 4 7 5];
sum1 = 0;
for k = 1:2:5 % start at 1, check every 2nd element, stop at 5
    sum1 = sum1 + b(k)
end
```

```
sum1 = 3
sum1 = 12
sum1 = 19
```

Ex. 4: The *for* loop: Double loop

```
sum1 = 0;
for n = 1:2 % start at 1, stop at 2
     for m = 1:3 % start at 1, stop at 3
        sum1 = sum1 + (n * m)
     end
end
```

```
sum1 = 1
sum1 = 3
sum1 = 6
sum1 = 8
sum1 = 12
sum1 = 18
```

```
for n = 1:2 % start at 1, stop at 2
    for m = 1:3 % start at 1, stop at 3
       fprintf('n = %3u, m = %3u \r', n, m)
    end
end
```

```
n =   1, m =   1
n =   1, m =   2
n =   1, m =   3
n =   2, m =   1
n =   2, m =   2
n =   2, m =   3
```

Ex. 5: The *for* loop: Advanced example

```
b = [2 5 7 4 9 8 3];
c = [2 3 5 7];
sum1 = 0;
for k = 1:4 % start at 1, stop at 4
 sum1 = sum1 + b(c(k))
end
```

```
sum1 = 5
sum1 = 12
sum1 = 21
sum1 = 24
```

Ex. 6: The *if* statement: Branching

```
num1 = 7;
if (num1 > 5)
 fprintf('%4u is greater than 5 \r', num1);
else
 fprintf('%4u is less than or equal to 5 \r', num1)
end
```

```
   7 is greater than 5
```

```
num1 = 3;
if (num1 > 5)
 fprintf('%4u is greater than 5 \r', num1);
else
 fprintf('%4u is less than or equal to 5 \r', num1)
end
```

```
   3 is less than or equal to 5
```

Ex. 7: The *if - elseif - else* statement

```
num1 = 4;
if (num1 >= 5)
 fprintf('%4i is greater than or equal to 5 \r', num1)
elseif (num1 > 1)
 fprintf('%4i is less than 5 but greater than 1 \r', num1)
elseif (num1 == 1)
 fprintf('%4i equals 1 \r', num1)
elseif (num1 > -3)
 fprintf('%4i is less than 1 but greater than -3 \r', num1)
else
 fprintf('%4i is less than or equal to -3 \r', num1)
end
```

```
4 is less than 5 but greater than 1
```

Ex. 8: An application - determine whether a given year is a leap year

```
nyear = 1975;
if (mod(nyear, 400) == 0)
 fprintf('%6u is a leap year', nyear)
elseif (mod(nyear,4) == 0) && (mod(nyear,100) ~= 0)
 fprintf('%6u is a leap year', nyear)
else
 fprintf('%6u is not a leap year', nyear)
end
```

```
1975 is not a leap year
```

Ex. 9: Combine looping and branching

```
sum1 = 0;
sum2 = 0;
N = 9
```

```
N = 9
```

```
for k = 1:N
    sum1 = sum1 + k;
    if (mod(k,3) == 0)
       sum2 = sum2 + k;
    end
end
sum1;
sum2;
```

Ex. 10: The *while* loop

```
x = 3;
while (x < 100)
    x = x * 3;
end
x;
```