If-Then-Else

```
if (condition) then
    statements if condition is true
else  ⇦ the "else" is optional!
    statements if condition is false
end if
```

<	less than	<=	less than or equal to
>	greater than	>=	greater than or equal to
=	equal to	not=	not equal to

Example – Voting Age
3b. Add new code – another option

```
var age : int          % declare a variable for age

% ask the user's age
put "How old are you? "..
get age

% if they are 18 or older, they can vote
if (age >= 18) then
    put "You can vote!"
else % if they are under 18, they cannot vote
    put "You are not old enough to vote."
end if
```

The For Loop

The counted loop is so common that a special type of loop was created to streamline the code.

```
for count : 1 .. 10
  put count
end for
```

Notes:

1. Do NOT declare the variable count. The for loop will take care of that automatically

Example – Blast Off!

```
put “Begin Count Down...”

for decreasing count : 10 .. 1
   put count
   delay(1000)
end for

put “Blast Off!”
```

Changing the Increment/Decrement

So far, we have incremented or decremented by 1. It is possible to take larger steps using the “by” command to specify the (integer) step size.

```
for count : 1 .. 10 by 2
   put count
end for

for decreasing count : 10 .. 1 by 3
   put count
end for
```

Rolling a Die (simulation)

```
% roll a single die 5 times

var roll : int

for count : 1 .. 5
  randint ( roll, 1, 6 )
  put “You rolled a “, roll
end for
```

Conditional Loops
Exit Condition Placement

For consistency and readability, the exit condition will only appear at the very beginning or very end of the loop.

```
loop
    exit when (condition)
    do something
end loop
```

```
loop
    do something
    exit when (condition)
end loop
```

Conversion Operations – Turing

```
var num : int

% declare a constant value for pi
const pi : real := 3.14

num := floor(pi)    % truncated to 3
num := round(pi)    % rounded to 3
num := ceil(pi)     % rounded up to 4

num := floor(2.71) % truncated to 2
num := round(2.71) % rounded to 3
num := ceil(2.71)  % rounded up to 3
```

Integer Division - Turing

In Turing, there are operators for both the remainder and the whole number (quotient).

```
var quotient : int
var remainder : int

quotient := 7 div 3   % is 2
remainder := 7 mod 3  % is 1

put "7/3 = ", quotient ..
put " remainder ", remainder
```