Relational operators:- Assum variable A hold 10 and variable B holds 20, then

operator	description	Examp
=	check if the values of two operands are equal or not, if yes then condition becom true	(A=B) not tr
<>	checks if the values of two operators are equal or not, if values are not equal then condition becom true	(A<>) true.
>	checks if the value of left operand is greater than the value of the right operand, if yes, then condition becom true.	(A>B) not tr
<	check if the value of the left operand is less than the value of the right operand, if yes, then condition becom true.	(A<B) true.
>=	Check if the value of the left operand is greater than or equel to the value of the right operand if yes, then condition becom true	(A>=B is no true
<=	check if the value of the left operand is less than or equel to the value of the right operator if yes, then condition is true	(A<=B is tr

Boolean Operators

Following table shows all the Boolean operators supported by Pascal language. All these operators work on Boolean operands and produce Boolean results. Assume variable **A** holds true and variable **B** holds false, then:

Operator	Description	Example
and	Called Boolean AND operator. If both the operands are true, then condition becomes true.	A and B) is false.
and then	It is similar to the AND operator, however, it guarantees the order in which the compiler evaluates the logical expression. Left to right and the right operands are evaluated only when necessary.	(A and then B) is false.
or	Called Boolean OR Operator. If any of the two operands is true, then condition becomes true.	(A or B) is true.
or else	It is similar to Boolean OR, however, it guarantees the order in which the compiler evaluates the logical expression. Left to right and the right operands are evaluated only when necessary.	(A or else B) is true.
<=	Called Boolean NOT Operator. Used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	not (A and B) is true.

The following example illustrates the concept:

```
program beLogical;
var
a, b: boolean;
begin
   a := true;
   b := false;

   if (a and b) then
      writeln('Line 1 - Condition is true' )
   else
      writeln('Line 1 - Condition is not true');
   if  (a or b) then
      writeln('Line 2 - Condition is true' );

   (* lets change the value of a and b *)
   a := false;
   b := true;
   if  (a and b) then
      writeln('Line 3 - Condition is true' )
   else
      writeln('Line 3 - Condition is not true' );
   if not (a and b) then
   writeln('Line 4 - Condition is true' );
end.
```

Reserved word in Pascal:

the Statement in Pascal are designed with Some specific pascal words, which are called Reserved words. For Ex:- Program, input, output, var, begin, real, writeline, readline and end are all reserved words.

there are more keywords like.

(and, array, case, Const, div, do, down to, else, file, for, Function, goto, if, in, label, mod, nil, not, of, or, packed, Procedure, record, repeat, set, then, to, type, until, while, with

Different b/w Read & ReadLine

* ReadInStruti - discards all other values on the same line.
but read does not.

"Decision Making"

(1)

if - then statement :- is the simplest Fo
of control statement, Frequently use
in decision Making and changing the
control Flow of the Program executi

syntax :-

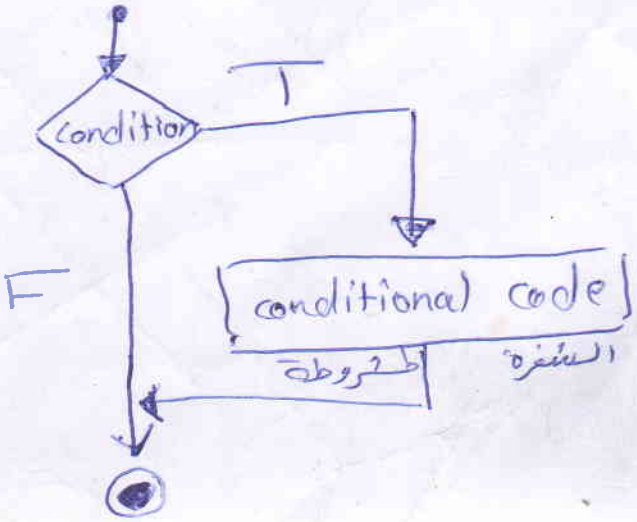
IF Condition then statemen

ex :- IF (a <= 20) then

c := c+1 ;

where condition :- is relational condition o
Boolean.

→ if the condition ~~is~~ evaluated to true,
the th block of code inside if statemen
will be executed, if the condition evaluate
to false the First set of the code af
~~if statement~~ the end of if statement w
be executed.



ex:- Program ifchecking;

```
var
  a: integer;
begin
  a:=10;          ← read لا داعي لكتابة
                    لانتحتجنا يا اطفال
                    لما البرنامج
  if( a<20) then        (بد رمز)
  writeln('a is less than 20');
  writeln('value of a is:', a);

end.
```

② if - then - else statement

~~An if then statement can be fo~~

else statement will executes when the Boolean expression is False (condition)

Syntax: if condition then S_1 else S_2

when the condition is true the S_1 is executed and S_2 is skipped. if the condition is False the S_2 is executed.

ex:-
```
if color = red then
   writeln('You have chosen a red car')
else
   writeln('please choose a color for your
```

Program ifelse checking;

```
var
 a: integer;
begin
 a:=100;
 if (a<20) then
 writeln('a is less than 20')
 else
 writeln('a is not less than 20');
 writeln('value of a is:', a);
end.
```

~~[crossed out]~~: (if-then-else-if-then-else statement

Syntax:

if (condition1) then

S1 {executes when the con1 is Tru

else if (condition2) then

S2 {executes when the con2 is True}.

else if (condition 3) then

S3 {executes when the con3 is T

else

S4 (executes when none of the above conditions is True) -

~~output~~

م.م. فاطمة حسن عبد

ex: Program if-else2.s

```
var
   a: integer;
begin
   a:= 100;
   if ( a=10) then
   writeln ('value of a is 10');
   else if ( a=20) then
   writeln ('value of a is 20')
   else if ( a=30) then
   writeln ('value of a is 30')
   else
   writeln ('None of the value
                is Matching');
   writeln ('Exact value of a is
   end.
```

output

```
None of the value is Matching
Exact value of a is: 100
```