

```

: String      2
S      Length('s')
:
:
: Edit      label
var a:string;
begin
a:=edit1.text;
label1.Caption := inttostr(length(A);
label2.Caption :=inttostr(length('asdfgh');
: Pchar      3
Var P:pchar;      : Pchar      P
'Hello      ;Form1.Caption := p      :      Begin
p:= All.....'
:
s: string;
begin
s:= 'HELLO ALL.....'#0      ;
p:= @s[1];
form1.Caption := p;
. Null      s      Null      #0
. s      @
: Variant (      )      4
:
var v1,v2,v3 :variant;
begin
v1:= 5;
v2:= 20;
v3 := v1+v2      ;
form1.Caption := v3;
: EvariantError
v1:= 5;
v2:= 'asd';
v3 := v1+v2      ;
:
v1:= 5;
v2:= '123';
v3 := v1+v2      ;

```

```

: Variant
compiler
interpreted
.
:
Interface
private
.
. Public
:
: Private
private
I :integer;
: Public
public
x:integer;
: OnCreate
I := 4 ; X := 24;
OnCreate
procedure TForm1.FormCreate ( X,I
.(
:
form2.Show ;
.( Uses
)
:
Form2.Caption := inttostr ( Form1.X) ;
X
Implementation Var

```

```

Implementation Var
Y: string = 'abcd'; F : double = 55;

```

: Const

Implementation Type

Implementation

Const

```

pi=3.14;
pi2=22/7;
programmer = 'Khaled';

```

Const

```

Pi :Real = 3.14;
pi2:single =22/7;
programmer :string= 'Khaled';

```

function

procedure

:

routine

result

parameters

()

procedure Hello;

begin

ShowMessage (' ');

end;

function Double (Value: Integer) : Integer;

begin

Double := Value * 2;

end;

Implementation

Hello

Form1.caption := inttostr(Double(6));

```

Procedure ProcedureName ( var1,var2,var3: anytype ;
var4,var5:any type) ;
Begin
...
...
end;

```

```

Procedure Add( x,y : integer) ;
Begin
Y:= x+y ;
End;

```

```

var i,j:integer;
begin
i:=1;j:=2;
add(i,j);
label1.Caption := inttostr(i);
label2.Caption := inttostr(j);

```

```

. y x Add
: ( ) 2
Procedure ProcedureName ( var1,var2,var3: anytype ;var
var4,var5:any type) ;
Begin
...
...
end;

```

```

Add
procedure add(x:integer;var y:integer);
begin
y:= x+y;
end;

```

```

. j y y x Add

```

:

```

Add (          Add( 1,j)          Add
                Add( 1,2)          I,2)

```

. ()

: 3

```

Procedure ProcedureName ( const var1,var2,var3: anytype ;
var4,var5:any type) ;
Begin
...
...
end;

```

Const

: Add

```

procedure add(const x:integer;var y:integer);
begin
y:= x+y;
end;

```

```

procedure add(const x:integer;const y:integer);
begin
y:= x+y;
end;

```

Left side cannot be assigned to :

Y

: 5

```

Procedure ProcedureName(var1,var2,var3: anytype ; var4:
anytype = Value ; var5:anytype= value) ;
Begin
...
...
end;

```

Add :

```
procedure add(var x:integer;var y:integer=5);
begin
x:= x+y;
end;
```

: Add

```
Add ( I,j) or Add( I )
```

: overload

4

Label :

```
procedure change( i :integer; var s :string) ; overload;
begin
s:=inttostr(i);
end;
procedure change( i:tdatetime; var s :string) ; overload;
begin
s:= datetostr(i)
end;
procedure change( i:tdatetime; var s1,s2 :string) ;
overload;
begin
S1:= datetostr(i);
S2 := datetostr( i+1);
end;
```

:

```
Var a:string ;
begin
change ( 5,a) ;
Label1.Caption := a ; end;
```

:

```
Var B:String ; begin
Change ( date,b);
Label2.Caption := b; End;
```

:

```
Var A,B:string; begin
```

<http://www.alshater.net>

Change (date,a,b);

Label1.Caption := A; Label2.Caption := B; end;

: Result 6

: Result

: Add

Function add(x,y:integer) : integer; Begin Add := x+y; End;	Function add(x,y:integer) : integer; Begin result := x+y; End;
--	---

:

. Interface

: Var Interface 1

Procedure Add(x,y : integer) ;

: Implementation (Begin... end)

Procedure Add(x,y : integer) ;

Begin

Y:= x+y ;

End;

. Type 2

Private Type :

Procedure Add(x,y : integer) ;

: Implementation Add

Procedure TForm1.Add(x,y : integer) ;

Begin

Y:= x+y ;

End;

: TForm1 Add

Form1.add (x,y) ;