

PASCAL - VARIANTS

http://www.tutorialspoint.com/pascal/pascal_variants.htm

Copyright © tutorialspoint.com

Pascal supports a unique type of storage named variants. You can assign any simple type of values in a variant variable. The type of a value stored in a variant is only determined at runtime. Almost any simple type can be assigned to variants: ordinal types, string types, int64 types.

Structured types such as sets, records, arrays, files, objects and classes are not assignment-compatible with a variant. You can also assign a pointer to a variant.

Free Pascal supports variants.

Declaring a Variant

You can declare variant type like any other types using the **var** keyword. The syntax for declaring a variant type is –

```
var  
  v : variant;
```

Now, this variant variable v can be assigned to almost all simple types including the enumerated types and vice versa.

```
type  
  color = (red, black, white);  
var  
  v : variant;  
  i : integer;  
  b : byte;  
  w : word;  
  q : int64;  
  e : extended;  
  d : double;  
  en : color;  
  as : ansistring;  
  ws : widestring;  
  
begin  
  v := i;  
  v := b;  
  v := w;  
  v := q;  
  v := e;  
  v := en;  
  v := d;  
  v := as;  
  v := ws;  
end;
```

Example

The following example would illustrate the concept –

```
Program exVariant;  
  
uses variants;  
type  
  color = (red, black, white);  
  
var  
  v : variant;  
  i : integer;  
  r : real;  
  c : color;
```

```

as : ansistring;

begin
  i := 100;
  v:= i;
  writeln('Variant as Integer: ', v);

  r:= 234.345;
  v:= r;
  writeln('Variant as real: ', v);

  c := red;
  v := c;
  writeln('Variant as Enumerated data: ', v);

  as:= ' I am an AnsiString';
  v:= as;
  writeln('Variant as AnsiString: ', v);
end.

```

When the above code is compiled and executed, it produces the following result –

```

Variant as Integer: 100
Variant as real: 234.345
Variant as Enumerated data: 0
Variant as AnsiString: I am an AnsiString

```