

# Data Types - 2 Mar ÖS 3:46

## Veri Tipleri

- INT1 : 0 ile 255 arası tam sayı değerleri alabilir.
- INT2 : -32,768 ile +32,767 arası tam sayı değerleri alabilir.
- I : -2,147,483,648 ile +2,147,483,647 arası tam sayı değerleri alabilir.
- INT8 : -9,223,372,036,854,775,808 ile +9,223,372,036,854,775,807 arası tam sayı değerleri alabilir.
- N : Nümerik karakterlerin saklanması sağlar. Uzunluğu yazılımcı tarafından belirlenir. Varsayılan uzunluğu 1'dir.
- DECFLOAT16 : Ondalıklı sayı tipidir. Maksimum uzunluğu 16 karakterdir.
- DECFLOAT34 : Ondalıklı sayı tipidir. Maksimum uzunluğu 34 karakterdir.
- P : Ondalıklı sayı tipidir. Float'tan farkı virgülden sonra kaç karakter olacağının yazılımcı tarafından belirlenebilmesidir. Virgülden sonra maksimum 14 karakter alabilir.
- C : Verilerin karakter olarak tutulmasını sağlar. Karakter uzunluğu belirlenebilir. Varsayılan uzunluğu 1'dir.
- STRING : Verilerin karakter olarak tutulmasını sağlar. Char'dan farkı karakter uzunluk sınırlaması yoktur.
- D : Tarih tipinde verilerin tutulmasını sağlar. Yazım formatı "yyyymmdd" şeklindedir.
- T : Saat tipinde verilerin tutulmasını sağlar. Yazım "hhmmss" şeklindedir.

Başlıca temel veri tipleri bunlardır. Şimdi ise bu veri tiplerini kullanarak nasıl değişken oluşturabiliriz bundan bahsedeceğim.

## Değişken Oluşturma

Değişken oluştururken DATA, TYPE, LENGTH ve VALUE anahtar kelimelerini kullanırız. Temel olarak değişken oluşturma syntax yapısı şu şekildedir.

DATA: degisken\_adi TYPE veri\_tipi LENGTH degisken\_uzunlugu VALUE ilk\_deger\_ataması.

- *Not: LENGTH ve VALUE komutu kullanılmak zorunda değildir. Ayrıca değişken isimlerinin 30 karakteri geçmemesi gerekmektedir.*

Bunu temel olarak ele aldığımız zaman diğer dillerden farklılığı sadece syntax yapısıdır. Örneğin c#'ı ele aldığım zaman `int i=234;` yaparken burada ise `i`'i data'a eşitliyoruz type ise `int` oluyor 234 ise value değerinin karşısına yazılıyor, basit örneklendirmeler aşağıdadır;

## INT1

| *DATA: lv\_byte TYPE int1 VALUE 200.*

## INT2

| *DATA: lv\_smallint TYPE int2 VALUE 5400.*

## I

| *DATA: lv\_integer TYPE i.*

## INT8

| *DATA: lv\_int8 TYPE int8 VALUE 759865.*

## N

| *DATA: lv\_number TYPE n LENGTH 25.*

## DECFLOAT16

| *DATA: lv\_float16 TYPE decfloat16 VALUE '1.26598965695'.*

## DECFLOAT34

| *DATA : lv\_float34 TYPE decfloat34 VALUE '25.69568658'.*

**P**

*DATA: lv\_decimal TYPE p DECIMALS 8. — DECIMALS 8 ile virgülden sonra kaç karakter olacağı belirtilir.*

**C**

*DATA: lv\_char(15) TYPE c. — Değişken isminden sonra yazılan (15) ile değişken uzunluğu belirtilir.*

**STRING**

*DATA: lv\_string TYPE string VALUE 'Kerem Türker'.*

**D**

*DATA: lv\_date TYPE d.*

**T**

*DATA: lv\_time TYPE t.*

Ve bunlardan sonra ise sabitler bulunuyor, her dilde ihtiyaç duyulduğu gibi abapta da sabitler vardır ve bunun için CONSTANTS anahtar kelimesi kullanılır.

**Örn;**

**CONSTANTS: c\_pi\_sayisi TYPE p DECIMALS 2 VALUE '3.14'.**

## **Structure Değişkenler**

Bundan sonra gelen konumuz ise structure değişkenler olmaktadır. Birden fazla değişkeni bir araya getirerek tek bir değişken içinde saklanmasını sağlar. Ana değişkenden al değişkenlere ulaşabilmek için tire (-) işareti kullanılır. Örneği aşağıdadır;

### **Oluşturma Örneği;**

```
DATA: BEGIN OF gs_bilgiler,  
ad(20) TYPE c,  
soyad(10) TYPE c,  
dogum_tarihi TYPE d,  
END OF gs_bilgiler.
```

### **Kullanım Örneği.**

```
gs_bilgiler-ad = 'Kerem'.  
gs_bilgiler-soyad = 'Türker'.  
gs_bilgiler-dogum_tarihi = '19901225'.
```