**ABAP VERİ TİPLERİ**

Aşağıda ABAP programlarında kullanılan ön tanımlı tipler ve ABAP Dictionary’ de standart olarak tanımlı tipler anlatılmıştır.

**Ön Tanımlı Tipler (Predefined Data Types)**

Tüm ABAP programlarında kullanılabilirler.

**Sabit Uzunluklu Veri Tipleri (Fixed Length Data Types)**

Uzunluğu daima sabit olan veri tipleridir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Veri Tipi** | **Tanım** | **Başlangıç uzunluğu**  **(Byte)** | **Geçerli alan uzunluğu**  **(Byte)** | **Başlangıç değeri** | **Açıklama** |
| **Karakter** |  |  |  |  |  |
| C | Karakter | 1 | 1 – 65535 | yok |  |
| N | Nümerik metin | 1 | 1 – 65535 | 0 |  |
| D | Tarih | 8 | 8 | ‘00000000’ |  |
| T | Zaman | 6 | 6 | 0 |  |
| **Nümerik** |  |  |  |  |  |
| I | Tam sayı | 4 | 4 | 0 |  |
| P | Ondalıklı sayı | 8 | 1-16 | 0 |  |
| F | Ondalıklı sayı | 8 | 8 | 0 |  |
| decfloat16 | Ondalıklı sayı | 8 | 8 | 0 | 7.0 EHP2’ de eklendi |
| decfloat34 | Ondalıklı sayı | 16 | 16 | 0 | 7.0 EHP2’ de eklendi |
| **Heksadesimal** |  |  |  |  |  |
| X | On altılı sayı | 1 | 1 – 65535 | X’0′ |  |

C, N, P ve X veri tipleri genel tiplerdir. Kullanılmadan önce alan uzunluklarının belirtilmesi gerekir. P tipi için ondalık alanının da tanımlanması gerekir.

**Değişken Uzunluklu Veri Tipleri**

Çalışma anına kadar uzunlukları belirlenemeyen veri tipleridir.

|  |  |
| --- | --- |
| **Veri Tipi** | **Tanım** |
| STRING | Karakter Dizisi |
| XSTRING | Heksa decimal tiinde bir byte karakterter |

**ABAP Dictionary Ön Tanımlı Veri Tipleri**

ABAP dictionary’ de standart olarak tanımlanmış veri tipleri aşağıdaki gibidir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veri tipi** | **Tanım** | **Maksimum uzunluk n** | **ABAP veri tipi** |
| DEC | Hesap/miktar alanı | 1-31, 1-17 tabloda | P((n+1)/2) |
| INT1 | Bir byte tam sayı | 3 | Dahili kullanım |
| INT2 | İki byte tam sayı | 5 | Dahili kullanım |
| INT4 | Dört byte tam sayı | 10 | I |
| CURR | Para birimi alanı [BCD](https://d.docs.live.net/73bf8b331c50e228/ABAP%20Kitap/) formatında | 1-17 | P((n+1)/2) |
| CUKY | Para birimi alanı için para birimi anahtarı | 5 | C(5) |
| QUAN | [BCD](https://d.docs.live.net/73bf8b331c50e228/ABAP%20Kitap/) formatında miktar alanı | 1-17 | P((n+1)/2) |
| UNIT | Miktar alanları için birim anahtarı | 2-3 | C(n) |
| PREC | Artık kullanılmıyor | 16 | Dahili kullanım |
| FLTP | Kayan noktalı sayı | 16 | F(8) |
| NUMC | Nümerik metin | 1-255 | N(n) |
| CHAR | Karakter dizisi | 1-255 | C(n) |
| LCHR | Uzun byte karakter dizisi | 256-max | C(n) |
| STRING | Değişken uzunlukta String | 1-maksimum | STRING |
| RAWSTRING | Değişken uzunlukta Byte dizisi | 1-maksimum | XSTRING |
| DATS | Tarih | 8 | D |
| ACCP | Muhasebe dönemi YYYYMM | 6 | N(6) |
| TIMS | Zaman HHMMSS | 6 | T |
| RAW | Byte dizisi | 1-255 | X(n) |
| LRAW | Uzun byte dizisi | 256-max | X(n) |
| CLNT | Üst birim | 3 | C(3) |
| LANG | Dil | internal 1, external 2 | C(1) |

**Kullanıcı Tanımlı Veri Tipleri (User Defined Data Types)**

TYPES anahtar kelimesi kullanılarak program içerisinde kullanıcı tanımlı tipler tanımlanabilir.

Örnek: Kullanıcı tanımlı veri tipi kullanımını gösteren örnek.

TYPES: ty\_isim  TYPE c LENGTH 20,  
ty\_id    TYPE n,  
ty\_sonuc TYPE p DECIMALS 2,  
ty\_sira  TYPE i.  
DATA: gv\_isim  TYPE ty\_isim  VALUE ‘güney’,  
gv\_id    TYPE ty\_id    VALUE ‘1’,  
gv\_sonuc TYPE ty\_sonuc VALUE ‘89.20’,  
gv\_sira  TYPE ty\_sira  VALUE 193.  
WRITE : gv\_isim, gv\_id, gv\_sonuc, gv\_sira.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527548277d607.png

**Yapısal Veri Tipleri (Structured Data Types)**

Veri tiplerini bir isim altında gruplandırılarak oluşturulan veri tipleridir. Veri tipindeki bileşenlere tire (-) işareti ile erişilir.

**BEGIN OF** ve **END OF** anahtar kelimeleri arasında bileşenler tanımlanır.

Örnek: Yapısal veri tipi kullanımını gösteren örnek.

TYPES: ty\_isim(20) TYPE c,  
ty\_id       TYPE n,  
ty\_sonuc    TYPE p DECIMALS 2,  
ty\_sira     TYPE i.  
TYPES: BEGIN OF ty\_aday,  
isim      TYPE ty\_isim,  
id        TYPE ty\_id,  
sonuc     TYPE ty\_sonuc,  
sira      TYPE ty\_sira,  
adres(30) TYPE c,  
tarih     TYPE sy-datum,  
END OF ty\_aday.  
DATA: gs\_aday TYPE ty\_aday.  
gs\_aday-isim  = ‘güney’.  
gs\_aday-id    = ‘1’.  
gs\_aday-sonuc = ‘89.20’.  
gs\_aday-sira  = 193.  
gs\_aday-adres = ‘Abap sokak’.  
gs\_aday-tarih = sy-datum.  
WRITE: / gs\_aday-isim, gs\_aday-id, gs\_aday-sonuc,  
gs\_aday-sira, gs\_aday-adres, gs\_aday-tarih.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52754831baedc.png

**SABİTLER (CONSTANTS)**

Hafızada değiştirilemeyecek (sabit) verilerin tutulmasını sağlayan nesnelerdir. Değer, sabit tanımı yapılırken atanır ve daha sonra değiştirilemez.

Sabit tanımlamak için **CONSTANTS** anahtar kelimesi kullanılır. Değer atamak için VALUE ifadesi kullanılır.

XSTRING tipinde, referanslarda, internal tablolar ve internal tablo içeren yapılarda kullanılamaz.

Örnek 1: Sabit tanımlama örneği.

CONSTANTS : ay   TYPE i VALUE 12,  
gun  TYPE i VALUE 365,  
saat TYPE i VALUE 6.  
WRITE: ay, gun, saat.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275483b1930b.png

Örnek 2: Yapısal sabit tanımlama örneği.

CONSTANTS: BEGIN OF aday,  
isim(20) TYPE c            VALUE ‘güney’,  
id       TYPE n            VALUE ‘1’,  
sonuc    TYPE p DECIMALS 2 VALUE ‘89.20’,  
sira     TYPE i            VALUE 193,  
END OF aday.  
WRITE: /  aday-isim, aday-id, aday-sonuc, aday-sira.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275484495b72.png