

MAKROLAR – MAKRO VS FONKSİYON

- **Etiketleme ve Fonksiyonel** amaç olarak iki kısma ayrılmaktadır.
- **MAKROLAR**
 - **Makro** kullanılarak programın **modüler tasarımı**na yardımcı oluruz
 - **Yalnızca 1 kez tanımlanır** ve sürekli çağrılabilir
 - Kod **derlemelerinde** teknik açıdan sorun yaşatmaması adına, **önişlem** sırasında satır içi olarak genişletilir.
 - **Yazılma sebebi**, kaynak kodun **yazılması** ve **yönetilmesi** kolaylaştırır
 - **Genel işlevli** durumlar için kullanılması **performans, çalışma zamanlaması, ek yük** olarak nitelendirildiği için **gömülü sistem** programlamada ciddi sıkıntılara yol açmaktadır.

Kullanım Alanları

- **Sabit değerli veya programa özgü parametrik değerlerin** kullanılmasında
- **Günlük mesajları yazdırma veya iddiaları işleme**
- **Basit hesaplamalar ve durum kontrolleri yapma**

FONKSİYON VS MAKRO

- **Fonksiyon**, derlenir ve **ilgili çağırma yapıldığında** bellekten yüklenir ve çalışır. **Makro**, derlenir ve **daima** bellekte saklanır
- **Fonksiyon**, işlev kodu programdan bağımsız kimliğini korur. **Makro**, koyulduğu yerin işini görür.
- **Fonksiyon**, Çalışan programda işlev çağırısı ile karşılaşıldığında, kontrol çalışma zamanı alt sistemi tarafından ona aktarılır ve çalışan programın içeriği (iade talimatı adresi) korunur.
- **Fonksiyon**, bir işlevi çağırırken performans düşüklüğüne sebep olur. (bağlam değiştirme, ana program talimatlarının dönüş adresini koruma, parametreleri geçirme ve dönüş değerlerini işleme vb.). Bu nedenle, fonksiyonun kullanımı sadece karmaşık kod blokları için (daha basit durumları işleyen makrolara karşı) istenir.
-

```

// Fonksiyon oluşturma türleri

// 1. Yöntem
#define Func(X) f(x) ; g(x)
if (Döngü)
Func(param);
else
exit();

// 2. Yöntem
#define Func(X) f(x) ; g(x)
if (Döngü)
Func(param) ;
else
exit() ;

// 3. Yöntem
#define Func(X) do{f(x) ; g(x) ; } while(0)
if (Döngü)
{
    Func(param) ;
}
else
{
    exit() ;
}

```

// Örnek Makro Tanımlaması

```

// 1. Yöntem
#define PI 3.1415
Area = PI * sqr(radius);

// 2. Yöntem
#define PI 3.1415
Area = PI*((radius)*(radius))

// 3. Yöntem
#define PI 3.1415
#define sqr(x) PI*x*x

```

// Başka bir örnek

/* 18 yaş sınırını geçmesi durumunda onay verme

- * Max(s1,s2) = Değişken yazımı
- * (s1 < s2) = Sıralama - Karşılaştırma - Atama durumu
- * ? = Durum sorgusu
- * s1 = Sıralama doğru
- * : s2 = Sıralama yanlış - s2 yazılsın
- * ## = İki fişi birleştirme
- * #error = Hata durumunda bir hata mesajı görüntüleyip derlemeyi durdurur
- * #pragma = Uygulama tanımlı işlerde kullanılır. İşlemci tanımazsa görmezden gelir
- */

#include <stdio.h>

#include <stdint.h>

#define Limit(s1,s2) (s1<s2) ? s1 : s2

int main ()

{

 const uint8_t s1 = 18;

 printf("%d\n", Limit(s1,20));

 return 0;

}