

FONKSİYONLAR

Hazırlayan : Muhammed Esad Goncaloğlu

FONKSİYONLAR

- Fonksiyonlar, birlikte bir görevi gerçekleştiren bir ifadeler gruplarıdır.
- Her C programının en az bir fonksiyon vardır, bu da main() fonksiyonudur.
- C kütüphaneleri, programınızın çağırabileceği çok sayıda gömülü fonksiyon barındırırlar.
- Örneğin string.h kütüphanesinde bulunan strcat() fonksiyonu iki diziyi birleştirmek için kullanılır.
- Yine aynı kütüphanenin fonksiyonlarından olan memcpy(), bir bellek konumunu başka bir konuma kopyalamak için kullanılır.

```
return_type function_name( parameter list ) {  
    body of the function  
}
```

```
int max(int num1, int num2) {  
    int result;  
  
    if (num1 > num2){  
        result = num1;  
    }else{  
        result = num2;  
    }  
    return result;  
}
```

FONKSİYON ÇAĞIRMA

- Bir C fonksiyonu oluştururken, fonksiyonun ne yapması gerektiğine dair bir tanım verirsiniz.
- Bir fonksiyonu kullanmak için, tanımlanan görevi gerçekleştirmek üzere o fonksiyonu çağırmanız gerekir.
- Bir işlevi çağırarak için, işlev adıyla birlikte gerekli parametreleri iletmeniz yeterlidir.
- Fonksiyonları önceden ya da sonradan tanımlayabilirsiniz.

```
#include <stdio.h>
void sayhello(){
    printf("Hello World!");
}
int main(){
    sayhello();
    return 0;
}
```

```
#include <stdio.h>
void sayhello();
int main(){
    sayhello();
    return 0;
}
void sayhello(){
    printf("Hello World!");
}
```

BAZI KÜTÜPHANE FONKSİYONLARI

- math.h kütüphanesinin bazı gömülü fonksiyonları şunlardır;
- $\text{sqrt}()$ = sayının karekökünü alır.
- $\text{pow}(a,b)$ = a sayısının b. kuvvetini alır.
- $\log_{10}(x)$ = x'in 10'lu tabanındaki logaritmasını verir.
- $\log(x)$ = x'in e sayısı (2,71) tabanındaki logaritmasını verir.

```
#include <stdio.h>
#include <math.h>
int main(){
    int d;
    printf("Please enter a number : ");
    scanf("%d",&d);
    int c=pow(d,d);
    printf("%d. power of %d = %d\n",d,d,c);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main(){
    float d;
    printf("Please enter a number : ");
    scanf("%f",&d);
    float c=sqrt(d);
    printf("Root of %f = %f\n",d,c);
    return 0;
}
```

RECURSION (ÖZYİNELEME)

- Özyineleme, bir fonksiyonun çağrısının kendisini yapma tekniğidir.
- Bu teknik, karmaşık problemleri çözmesi daha kolay olan basit problemlere ayırmanın bir yolunu sunar.
- Özyinelemeyi anlamak biraz zor olabilir.
- Bu tekniğin nasıl çalıştığını anlamanın en iyi yolu onunla deney yapmaktır.

```
int sum(int k);

int main() {
    int result = sum(10);
    printf("%d", result);
    return 0;
}

int sum(int k) {
    if (k > 0) {
        return k + sum(k - 1);
    } else {
        return 0;
    }
}
```

KAYNAKÇA

- https://www.tutorialspoint.com/cprogramming/c_functions.htm
- <https://www.tutorialspoint.com/header-files-stdio-h-and-stdlib-h-in-c>
- https://www.tutorialspoint.com/c_standard_library/string_h.htm
- https://www.w3schools.com/c/c_functions_recursion.php