

## - Variable Template:

- Her bir instantiation, bir değişken

## - Perfect Forwarding:

- Eğer yazacağımız fonksiyon, bir parametresine gelen değeri, *const'ünü ve value kategorisini koruyarak*, başka fonksiyona göndermek istiyorsa, *Parametre universal referans / forwarding ref. olmalı.*

→ Notasyon şu şekilde:

```
template <typename T>
void call_func(T&& x)
{
    func(std::forward<T>(x));
}
```

*T, T&, T&& gibi belirtilir.*

*func'a std::forward ile paketlenir.*

```
template<typename T, typename U, typename V, typename W>
void func(T &&t, U&& u, V&& v, W&& w)
{
    f(std::forward<T>(x), std::forward<U>(u), std::forward<V>(v), std::forward<W>(w));
}
```

→ Perfect forwarding *en sık*, *containerlere* var *eldeneme kullanılıyor*.

```
template<typename T>
void func(const T&&)
{
    ...
}
```

*Bu perfect forwarding değil! X*  
*Forwarding referansla, const / volatile qualifier olmaz! X X*

→ Forwarding referans *isn't* template argümanı !!

## - Variadic Template:

→ Bu template'leri oluştururken *her* template argümanı belirtilebilir. → *Tipi buna göre bir şey*

```
template<typename ...Args>
//
```

*...Args* *elips*

*parametre placeholder ismi.*

```
template<typename ...Ts>
void func(Ts& ...args)
{
    ...
}
```

*...args* *değişkenler artıktır & value parametreler yığılabilir olarak.*

```
template <typename ...Ts>
void func(Ts ...args)
{
    constexpr auto val = sizeof...(args);
}

int main()
{
    func(2, 3.4, 6.7, 5);
}
```

*val=4*  
*compile time da boye*  
*bir size of kullanabilir*

```
template <typename ...Ts>
void func(Ts ...args)
{
    int a[sizeof...(args)] = { 0 };
}
```

*=4 elementli array*

```
f(&args...); // expands to f(&E1, &E2, &E3)
f(n, ++args...); // expands to f(n, ++E1, ++E2, ++E3);
f(++args..., n); // expands to f(++E1, ++E2, ++E3, n);

f(const_cast<const Args*>(&args)...);
// f(const_cast<const E1*>(&X1), const_cast<const E2*>(&X2), const_cast<const E3*>(&X3))

f(h(args...) + args...); // expands to
// f(h(E1, E2, E3) + E1, h(E1, E2, E3) + E2, h(E1, E2, E3) + E3)
```

Devemine bir daha bak

STC'ye grrr yapti