

The mystery of typename

dkfz.

DEUTSCHES
KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT



50 Jahre – Forschen für
ein Leben ohne Krebs

The first use of typename

- No Difference between „class“ and „typename“

```
template<class T1, class T2> ...
```

```
template<typename T1, typename T2>...
```

Second use of typename

- Resolve ambiguities with nested dependent type name

- Dependent Type Name:

Name depends direct on template.

```
template <class C>  
    ...  
    C temp = value;
```

- Nested Dependent Type Name:

Name depends on a template which is
nested.

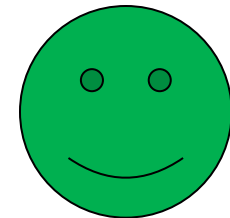
```
template <class C>  
    ...  
    C::iterator iter1;  
    std::vector<C>::iterator iter2;
```

Examples

```
template <typename T>
void foo(const T& t)
{
    T::bar * p;
    vector<T>::iterator iter;
}
```



```
template <typename T>
void foo(const T& t)
{
    typename T::bar * p;
    typename vector<T>::iterator iter;
}
```

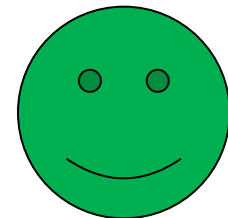


Examples

```
template <typename T>  
void foo(const T& t)  
{  
    typename vector<int>::iterator iter;  
}
```



```
template <typename T>  
void foo(const T& t)  
{  
    vector<int>::iterator iter;  
}
```

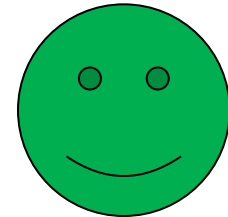


The loved exception

```
template <class T>  
class C1 : typename T::InnerType  
{  
};
```



```
template <class T>  
class C2 : A<typename T::InnerType>  
{  
};
```



Rules for typename usage

- In template declaration: interchangeable with „class“
- Only within templates
- Only if the template name is nested
- Not in class base list
- Not for fully defined templated classes
- **Each compiler handles it different**