



# Functions

# Functions

```
fun max(a: Int, b: Int): Int {  
    return if (a > b) a else b  
}
```

# Functions

```
fun max(a: Int, b: Int): Int {  
    return if (a > b) a else b  
}
```

# Functions

```
fun max(a: Int, b: Int): Int {  
  return if (a > b) a else b  
}
```

Convert to expression body



```
fun max(a: Int, b: Int): Int = if (a > b) a else b
```

# Functions

```
fun max(a: Int, b: Int): Int {  
  return if (a > b) a else b  
}
```

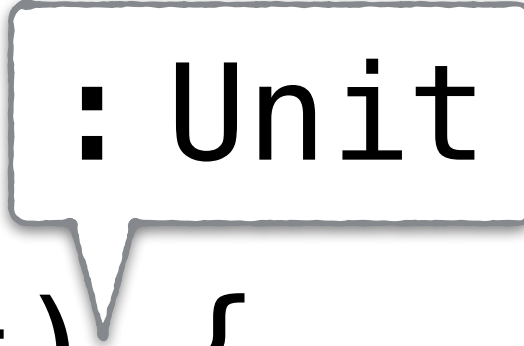
Convert to expression body



```
fun max(a: Int, b: Int) = if (a > b) a else b
```

# Function returning Unit

```
fun displayMax(a: Int, b: Int) {  
    println(max(a, b))  
}
```



An equivalent syntactic form:

```
fun displayMax(a: Int, b: Int): Unit {  
    println(max(a, b))  
}
```

# Functions everywhere

Top-level function:

```
fun topLevel() = 1
```

Member function:

```
class A {  
    fun member() = 2  
}
```

Local function:

```
fun other() {  
    fun local() = 3  
}
```



Is it possible to call a top-level function from Java? If yes, how?

1. You can't call it from Java
2. As a `static` function of the class, which name corresponds to a file name
3. As a member function of the class, which name corresponds to a file name







Is it possible to call a top-level function from Java? If yes, how?


1. You can't call it from Java
2. As a `static` function of the class, which name corresponds to a file name
3. As a member function of the class, which name corresponds to a file name

# Calling a top-level function from Java

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



# Calling a top-level function from Java

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



UsingFoo.java

```
package other;
```

```
import intro.MyFileKt;
```

```
public class UsingFoo {  
    public static void main(String[] args) {  
        MyFileKt.foo();  
    }  
}
```



# Calling a top-level function from Java

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



UsingFoo.java

```
package other;
```

```
import intro.MyFileKt;
```

```
public class UsingFoo {  
    public static void main(String[] args) {  
        MyFileKt.foo();  
    }  
}
```



# Calling a top-level function from Java

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



UsingFoo.java

```
package other;
```

```
import intro.MyFileKt;
```

```
public class UsingFoo {  
    public static void main(String[] args) {  
        MyFileKt.foo();  
    }  
}
```



# Calling a top-level function from Java

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



UsingFoo.java

```
package other;
```

```
import static intro.MyFileKt.*;
```

```
public class UsingFoo {  
    public static void main(String[] args) {  
        foo();  
    }  
}
```



# @JvmName

changes the JVM name of the class  
containing top-level functions



# Using @JvmName annotation

MyFile.kt

```
package intro
```

```
fun foo() = 0
```



JavaUsage.java

```
package other;
```

```
import intro.MyFileKt;
```

```
public class JavaUsage {  
    public static void main(String[] args) {  
        int i = MyFileKt.foo();  
    }  
}
```



# Using @JvmName annotation

Extensions.kt

```
@file:JvmName("Util")
```

```
package intro
```

```
fun foo() = 0
```



JavaUsage.java

```
package other;
```

```
import intro.Util;
```

```
public class JavaUsage {  
    public static void main(String[] args) {  
        int i = Util.foo();  
    }  
}
```

