



Exceptions

No difference between checked and
unchecked exceptions

throw is an expression

```
val percentage =  
    if (number in 0..100)  
        number  
    else  
        throw IllegalArgumentException(  
            "A percentage value must be" +  
            "between 0 and 100: $number")
```

try is an expression

```
val number = try {  
    Integer.parseInt(string)  
} catch (e: NumberFormatException) {  
    return  
}
```

try is an expression

```
val number = try {  
    Integer.parseInt(string)  
} catch (e: NumberFormatException) {  
    null  
}
```

@Throws annotation

```
@Throws(IOException::class)  
fun foo() {  
    throw IOException()  
}
```



Is there any difference between calling
foo and bar from Java?

```
fun foo() { throw IOException() }
```

// Java

```
try {  
    DemoKt.foo();  
}  
catch (IOException e) {  
    // ...  
}
```



```
@Throws(IOException::class)  
fun bar() { throw IOException() }
```

// Java

```
try {  
    DemoKt.bar();  
}  
catch (IOException e) {  
    // ...  
}
```





Is there any difference between calling
foo and bar from Java?

```
fun foo() { throw IOException() }
```

```
@Throws(IOException::class)  
fun bar() { throw IOException() }
```

1. There's no difference
2. The code calling foo compiles, but
the code calling bar doesn't compile
3. The code calling bar compiles, but
the code calling foo doesn't compile





Is there any difference between calling
foo and bar from Java?

```
fun foo() { throw IOException() }
```

```
@Throws(IOException::class)  
fun bar() { throw IOException() }
```

1. There's no difference
2. The code calling foo compiles, but
the code calling bar doesn't compile
3. The code calling bar compiles, but
the code calling foo doesn't compile

Checked exceptions

```
fun foo() { throw IOException() }
```



```
// Java
```

```
try {  
    DemoKt.foo();  
}  
catch (IOException e) {  
    // ...  
}
```

Error: Exception java.io.IOException
is never thrown in
the corresponding try block



Checked exceptions

```
@Throws(IOException::class)  
fun bar() { throw IOException() }
```



```
// Java  
try {  
    DemoKt.bar();  
}  
catch (IOException e) {  
    // ...  
}
```

