



Java Exceptions - Try...Catch

[< Previous](#)[Next >](#)

Java Exceptions

When executing Java code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things.

When an error occurs, Java will normally stop and generate an error message. The technical term for this is: Java will throw an **exception** (throw an error).

Java try and catch

The **try** statement allows you to define a block of code to be tested for errors while it is being executed.

The **catch** statement allows you to define a block of code to be executed, if an error occurs in the try block.

The **try** and **catch** keywords come in pairs:

Syntax

```
try {  
    // Block of code to try  
}  
catch(Exception e) {  
    // Block of code to handle errors  
}
```

Consider the following example:

This will generate an error, because **myNumbers[10]** does not exist.

```
public class Main {  
    public static void main(String[ ] args) {  
        int[] myNumbers = {1, 2, 3};  
        System.out.println(myNumbers[10]); // error!  
    }  
}
```

The output will be something like this:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10  
    at Main.main(Main.java:4)
```

Try it Yourself »

If an error occurs, we can use **try...catch** to catch the error and execute some code to handle it:

Example

```
public class Main {  
    public static void main(String[ ] args) {  
  
        int[] myNumbers = {1, 2, 3};  
        System.out.println(myNumbers[10]);  
    }  
}
```

```
        System.out.println("Something went wrong.");  
    }  
}  
}
```

The output will be:

```
Something went wrong.
```

[Try it Yourself »](#)

Finally

The **finally** statement lets you execute code, after **try...catch** , regardless of the result:

Example

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            int[] myNumbers = {1, 2, 3};  
            System.out.println(myNumbers[10]);  
        } catch (Exception e) {  
            System.out.println("Something went wrong.");  
  
            System.out.println("The 'try catch' is finished.");  
        }  
    }  
}
```

The output will be:

```
Something went wrong.  
The 'try catch' is finished.
```

[Try it Yourself »](#)

ADVERTISEMENT

The throw keyword

The **throw** statement allows you to create a custom error.

The **throw** statement is used together with an **exception type**. There are many exception types available in Java: **ArithmeticException** , **FileNotFoundException** , **ArrayIndexOutOfBoundsException** , **SecurityException** , etc:

Example

Throw an exception if **age** is below 18 (print "Access denied"). If age is 18 or older, print "Access granted":

```
public class Main {  
    static void checkAge(int age) {  
        if (age < 18) {  
            // ...  
        }  
        else {  
            System.out.println("Access granted - You are old enough!");  
        }  
    }  
}
```

```
    }  
}  
  
public static void main(String[] args) {  
    checkAge(15); // Set age to 15 (which is below 18...)  
}  
}
```

The output will be:

```
Exception in thread "main" java.lang.ArithmeticException: Access denied - You mu  
    at Main.checkAge(Main.java:4)  
    at Main.main(Main.java:12)
```

Try it Yourself »

If **age** was 20, you would **not** get an exception:

Example

```
checkAge(20);
```

The output will be:

```
Access granted - You are old enough!
```

Try it Yourself »

Test Yourself With Exercises

Exercise:

Insert the missing parts to handle the error in the code below.

```
{
int[] myNumbers = {1, 2, 3};
System.out.println(myNumbers[10]);
}      (Exception e) {
System.out.println("Something went wrong.");
}
```

[Submit Answer »](#)

[Start the Exercise](#)

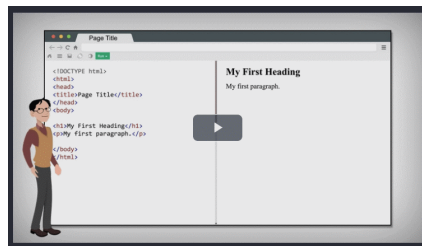
[< Previous](#)

[Next >](#)

ADVERTISEMENT

NEW

We just launched
W3Schools videos



Explore now

COLOR PICKER



Get certified
by completing
a Java
course today!



Get started

CODE GAME



Play Game

ADVERTISEMENT

ADVERTISEMENT

ADVERTISEMENT



Report Error

Spaces

Pro

Buy Certificate

Top Tutorials

[HTML Tutorial](#)

[CSS Tutorial](#)

[JavaScript Tutorial](#)

- How To Tutorial
- SQL Tutorial
- Python Tutorial
- W3.CSS Tutorial
- Bootstrap Tutorial
- PHP Tutorial
- Java Tutorial
- C++ Tutorial
- jQuery Tutorial

Top References

- HTML Reference
- CSS Reference
- JavaScript Reference
- SQL Reference
- Python Reference
- W3.CSS Reference
- Bootstrap Reference
- PHP Reference
- HTML Colors
- Java Reference
- Angular Reference
- jQuery Reference

Top Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- SQL Examples
- Python Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- Java Examples
- XML Examples
- jQuery Examples

Get Certified

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- Front End Certificate
- SQL Certificate
- Python Certificate
- PHP Certificate
- jQuery Certificate
- Java Certificate
- C++ Certificate
- C# Certificate
- XML Certificate

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.
W3Schools is Powered by W3.CSS.

