

# Catching Exceptions

Specific Exception	Multiple Exceptions I	Multiple Exceptions II
<pre>try:     foo() except TypeError as e:     bar()</pre>	<pre>try:     foo() except (TypeError, NameError) as e:     bar()</pre>	<pre>try:     foo() except TypeError as e:     bar() except NameError as e:     another_bar()</pre>
All exceptions	Regardless of exceptions	If no exceptions were raised
<pre>try:     foo() except:     bar()</pre>	<pre>try:     foo() except:     # something finally:     bar()</pre>	<pre>try:     foo() except:     # something else:     bar()</pre>

# Defining Exceptions

```
class MyError(Exception):
    # do something
```

# Raising Exceptions

```
raise MyError("This is a message")
```

# Common built-in exceptions

Name	Description	Example
AttributeError	Invalid attribute name	<code>a [1, 2].foobar()</code>
ImportError	Failed <code>import</code> statement	<code>import unknown_module</code>
IndexError	Out-of-range sequence index	<code>a = b[len(b)]</code>
KeyError	Invalid dictionary key	<code>a = {"b": 1, "c": 2}["d"]</code>
Name Error	Undefined local or global name	<code>a = undefined_foobar</code>
TypeError	Type inappropriate for operation	<code>a = len(123)</code>
ValueError	Invalid value	<code>a = [1, 2].index(3)</code>
ZeroDivisionError	Division by 0	<code>a = 5 / 0</code>