Exceptions and Errors

Robert Smallshire
@robsmallshire
rob@sixty-north.com

Presenter

Austin Bingham
@austin_bingham
austin@sixty-north.com

pluralsight
hardcore dev and IT training

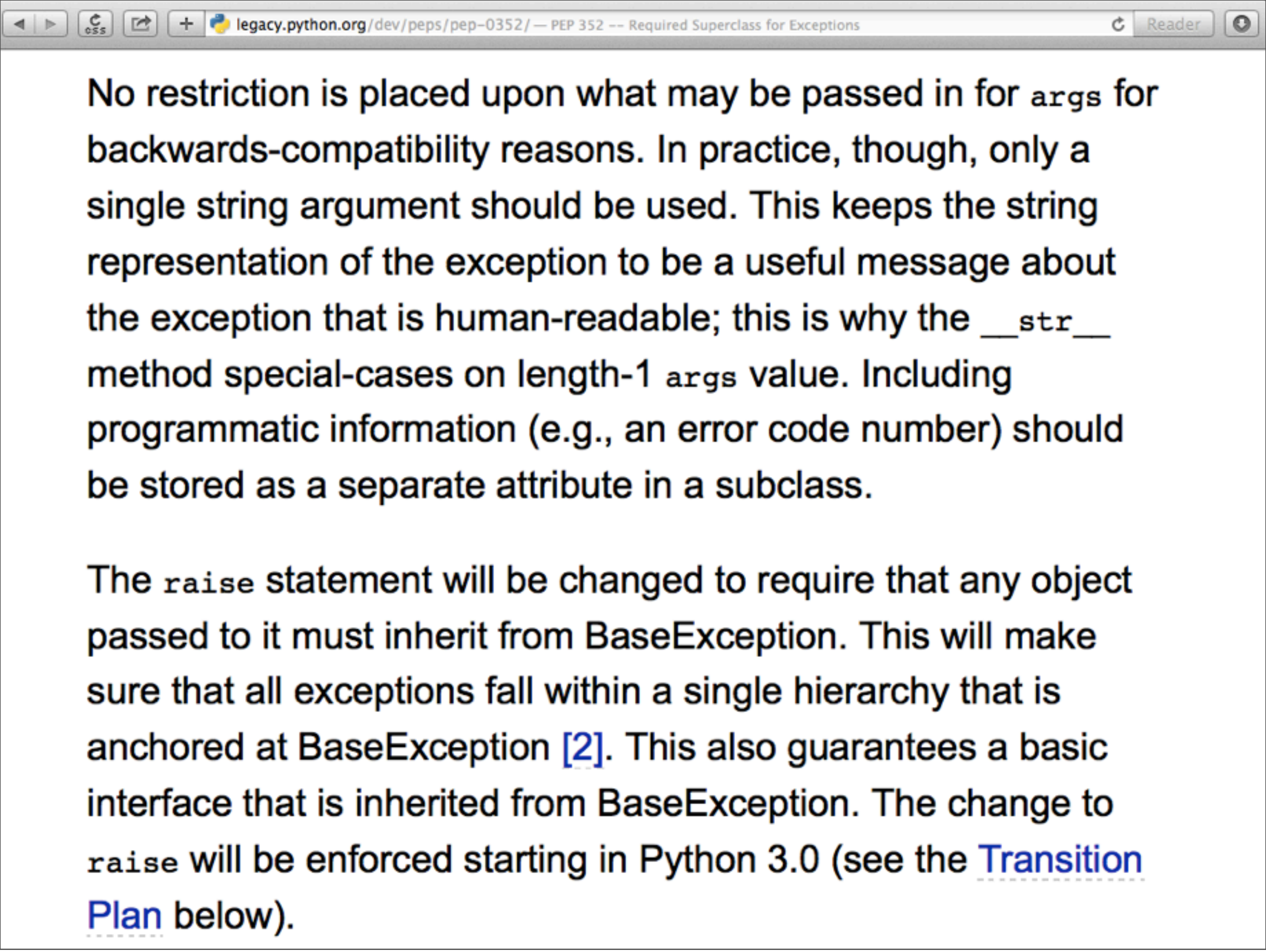
Avoid bad practices in Python exception handling

Exceptions are arranged in an **inheritance** hierarchy

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- ArithmeticError
      |    +-- FloatingPointError
      |    +-- OverflowError
      |    +-- ZeroDivisionError
      +-- AssertionError
      +-- AttributeError
      +-- BufferError
      +-- EOFError
      +-- ImportError
      +-- LookupError
      |    +-- IndexError
      |    +-- KeyError
      +-- MemoryError
      +-- NameError
      |    +-- UnboundLocalError
      +-- OSError
      |    +-- BlockingIOError
      |    +-- ChildProcessError
      |    +-- ConnectionError
      |    |    +-- BrokenPipeError
      |    |    +-- ConnectionAbortedError
      |    |    +-- ConnectionRefusedError
      |    |    +-- ConnectionResetError
      |    +-- FileExistsError
      |    +-- FileNotFoundError
      |    +-- InterruptedError
      |    +-- IsADirectoryError
      |    +-- NotADirectoryError
      |    +-- PermissionError
      |    +-- ProcessLookupError
      |    +-- TimeoutError
      +-- ReferenceError
      +-- RuntimeError
           +-- NotImplementedError
```

```
+-- Exception
     +-- SyntaxError
     |    +-- IndentationError
     |         +-- TabError
     +-- SystemError
     +-- TypeError
     +-- ValueError
     |    +-- UnicodeError
     |         +-- UnicodeDecodeError
     |         +-- UnicodeEncodeError
     |         +-- UnicodeTranslateError
     +-- Warning
          +-- DeprecationWarning
          +-- PendingDeprecationWarning
          +-- RuntimeWarning
          +-- SyntaxWarning
          +-- UserWarning
          +-- FutureWarning
          +-- ImportWarning
          +-- UnicodeWarning
          +-- BytesWarning
          +-- ResourceWarning
```

Exception payloads


legacy.python.org/dev/peps/pep-0352/ — PEP 352 -- Required Superclass for Exceptions
Reader

No restriction is placed upon what may be passed in for `args` for backwards-compatibility reasons. In practice, though, only a single string argument should be used. This keeps the string representation of the exception to be a useful message about the exception that is human-readable; this is why the `__str__` method special-cases on length-1 `args` value. Including programmatic information (e.g., an error code number) should be stored as a separate attribute in a subclass.

The `raise` statement will be changed to require that any object passed to it must inherit from BaseException. This will make sure that all exceptions fall within a single hierarchy that is anchored at BaseException [2]. This also guarantees a basic interface that is inherited from BaseException. The change to `raise` will be enforced starting in Python 3.0 (see the Transition Plan below).

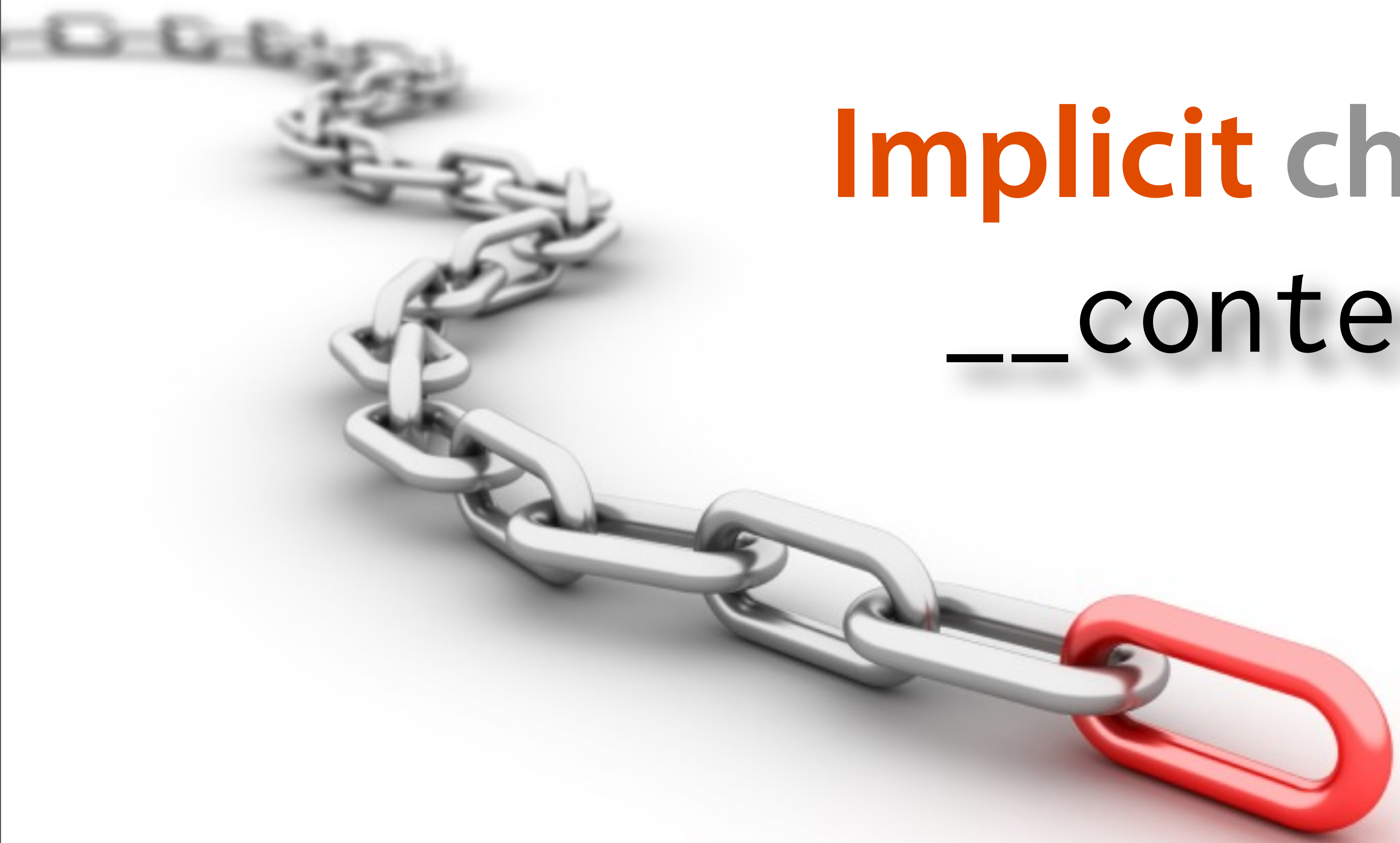
python™

Defining new exceptions

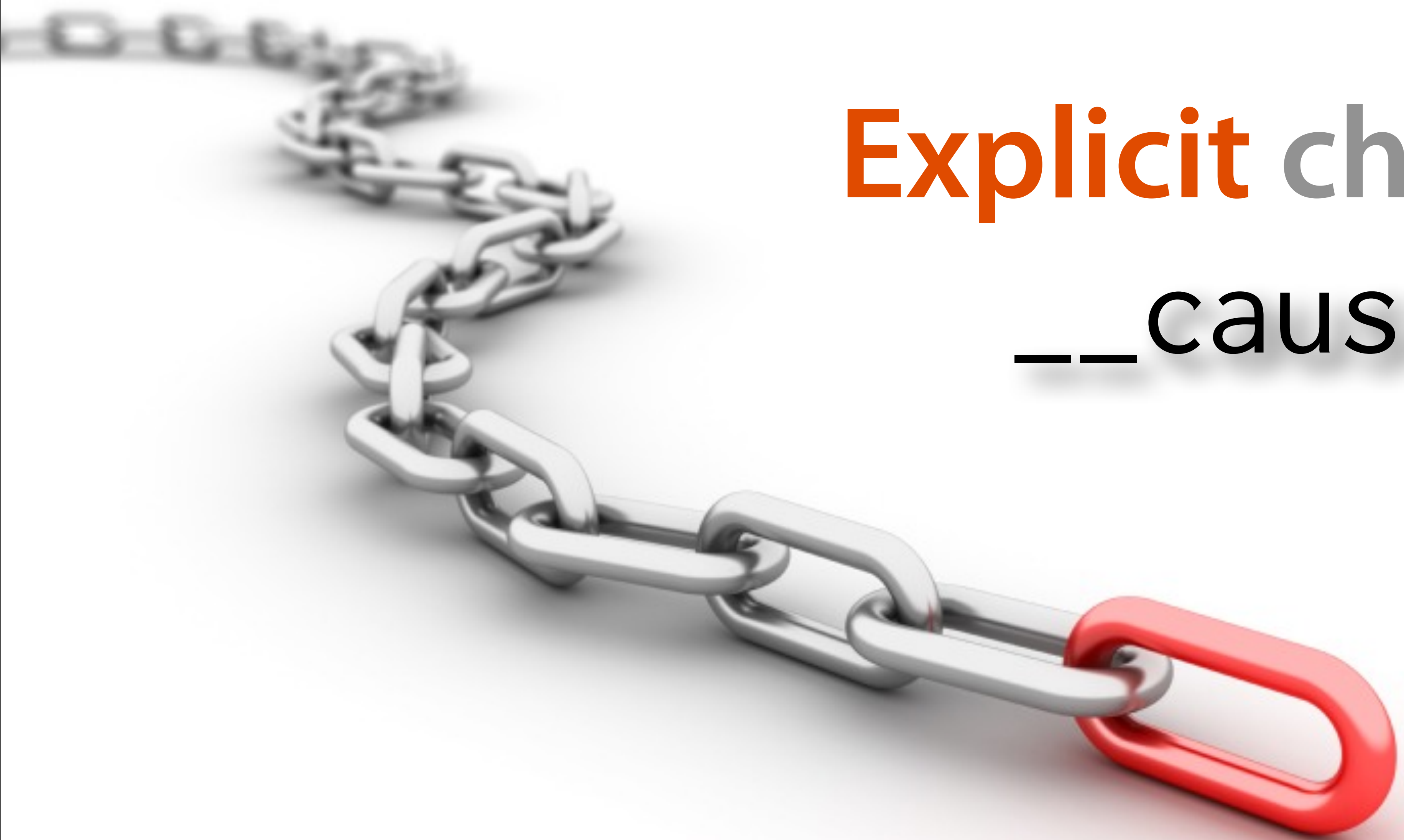
Chaining exceptions

Implicit chaining

`__context__`



Chaining exceptions



Explicit chaining

__cause__

python™

Tracebacks

`__traceback__`

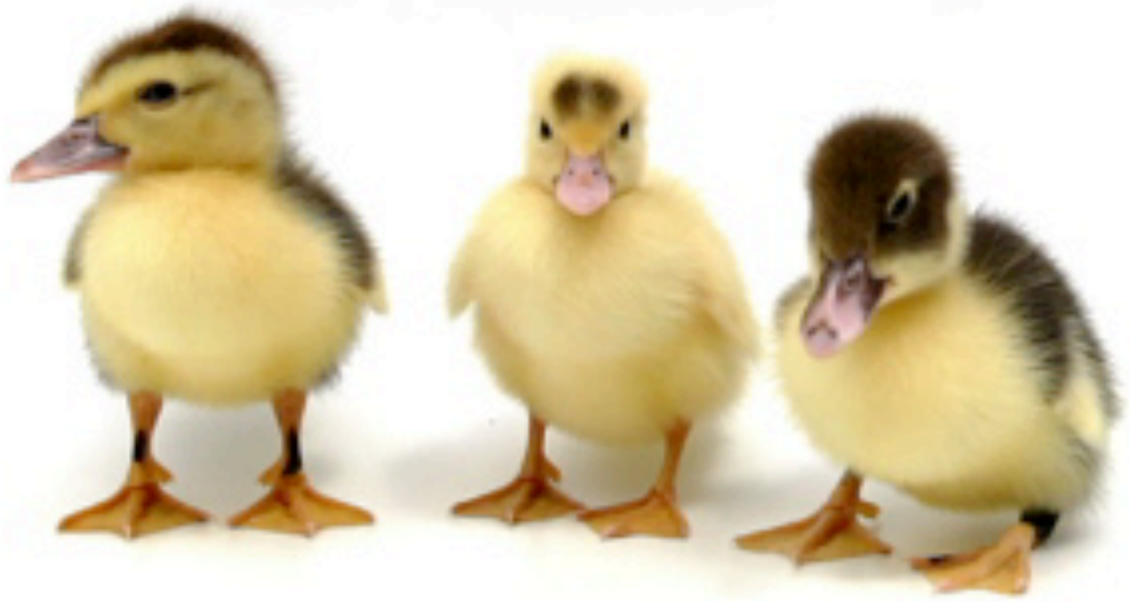
Checking invariants with Assertions

```
assert condition [,message]
```

False ⇒ AssertionError

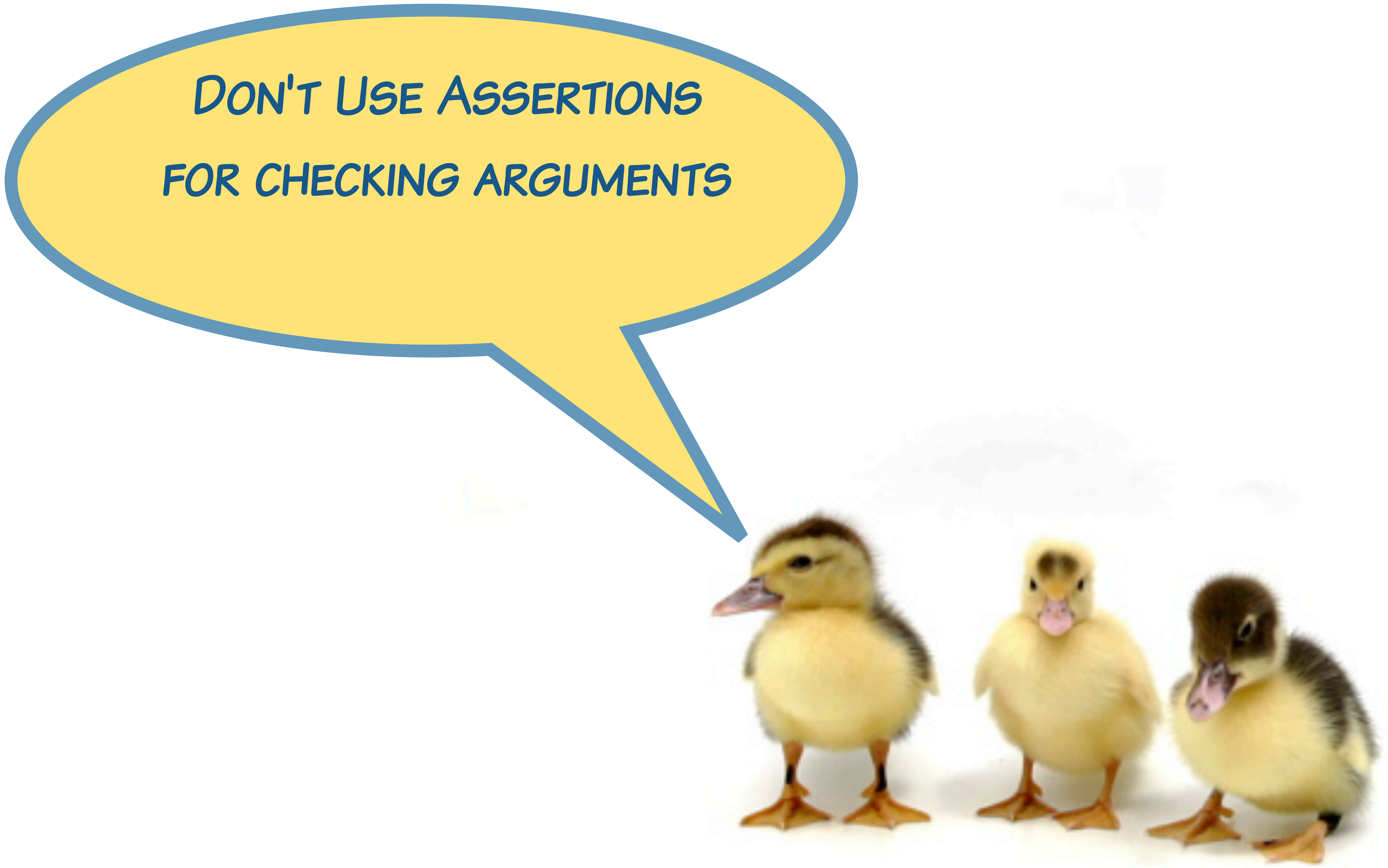
Duck Tails

Preconditions, post-conditions and assertions



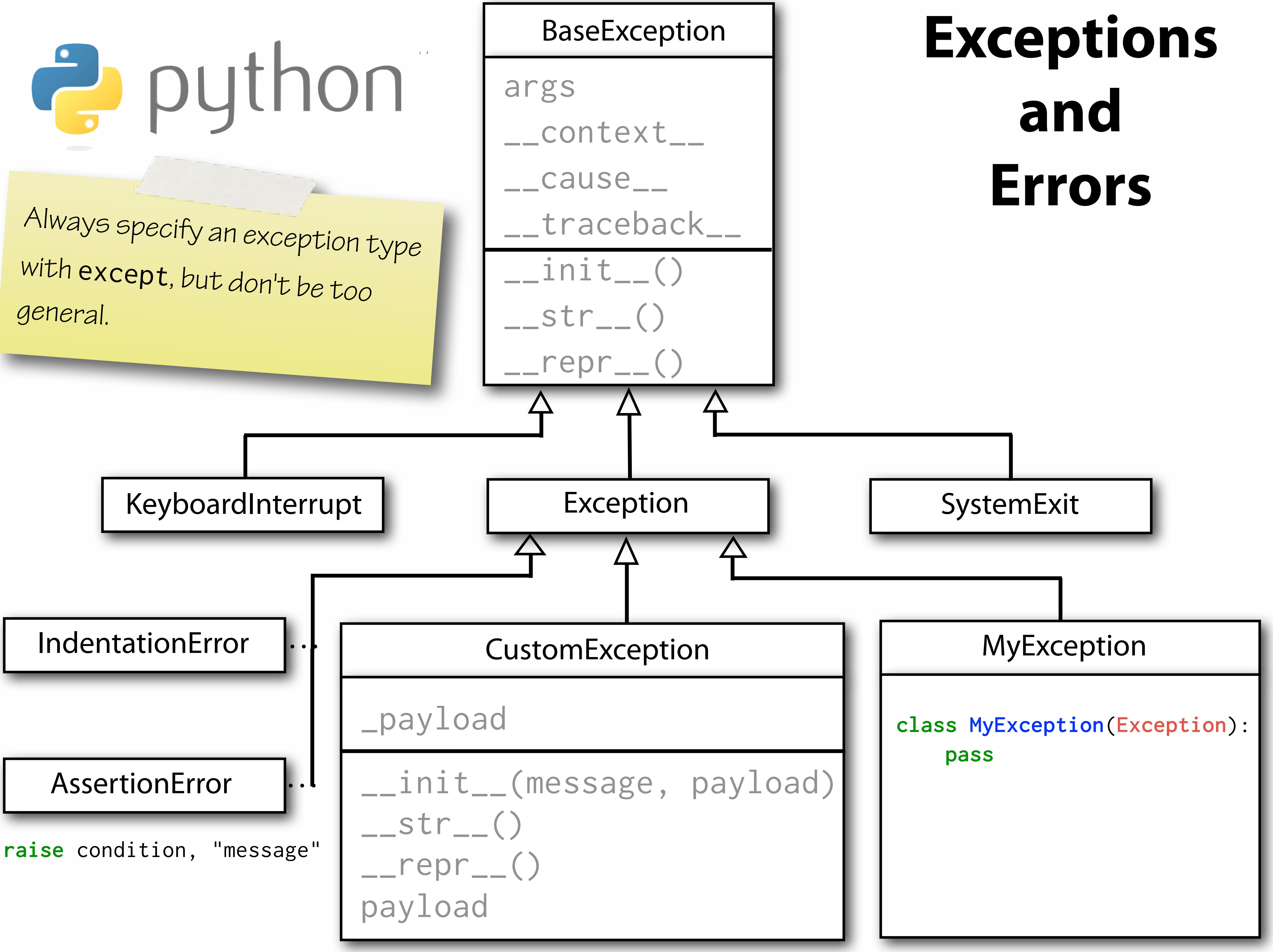
Duck Tails

Duck Tails



Exceptions and Errors

Always specify an exception type with `except`, but don't be too general.



```
class MyException(Exception):
    pass
```

```
raise condition, "message"
```