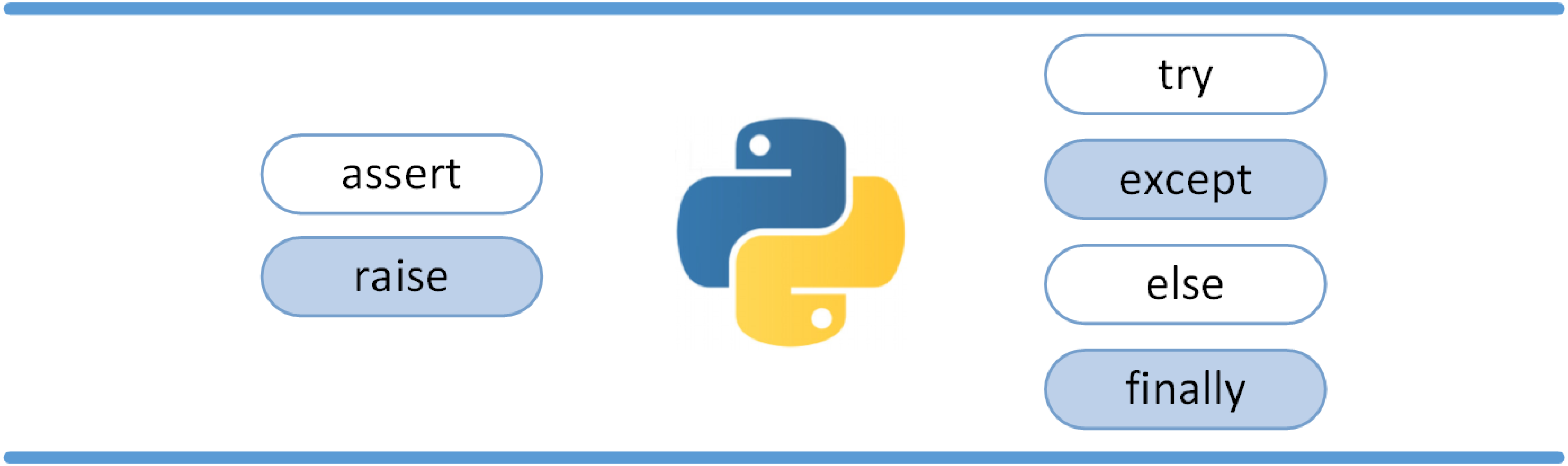
Intro to Exceptions

Intro to Exceptions

1. Differentiating Exceptions From Syntax Errors
2. Raising an Exception
3. Using the `assert` Keyword
4. Catching Exceptions With the `try...except` Block
5. Handling the Success Case With `else`
6. Catching a Built-in Exception
7. Cleaning Up After Using `finally`
8. Summary

Intro to Exceptions



assert

raise

try

except

else

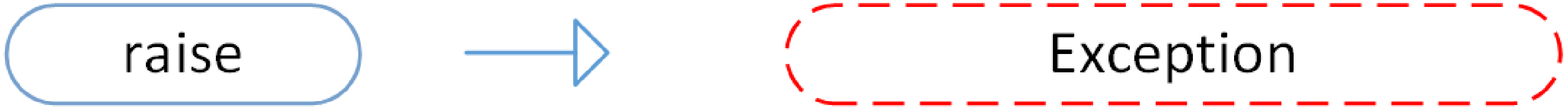
finally

Differentiating Exceptions From Syntax Errors

- Parsing Error: `SyntaxError`
- Execution Error: `Exception`

Raising an Exception

Use raise to force an exception:

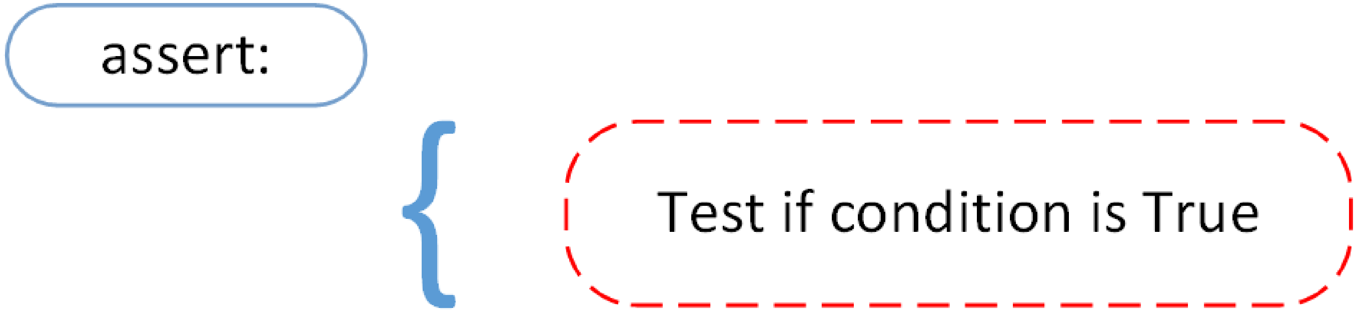


raise

Exception

Using the `assert` Keyword

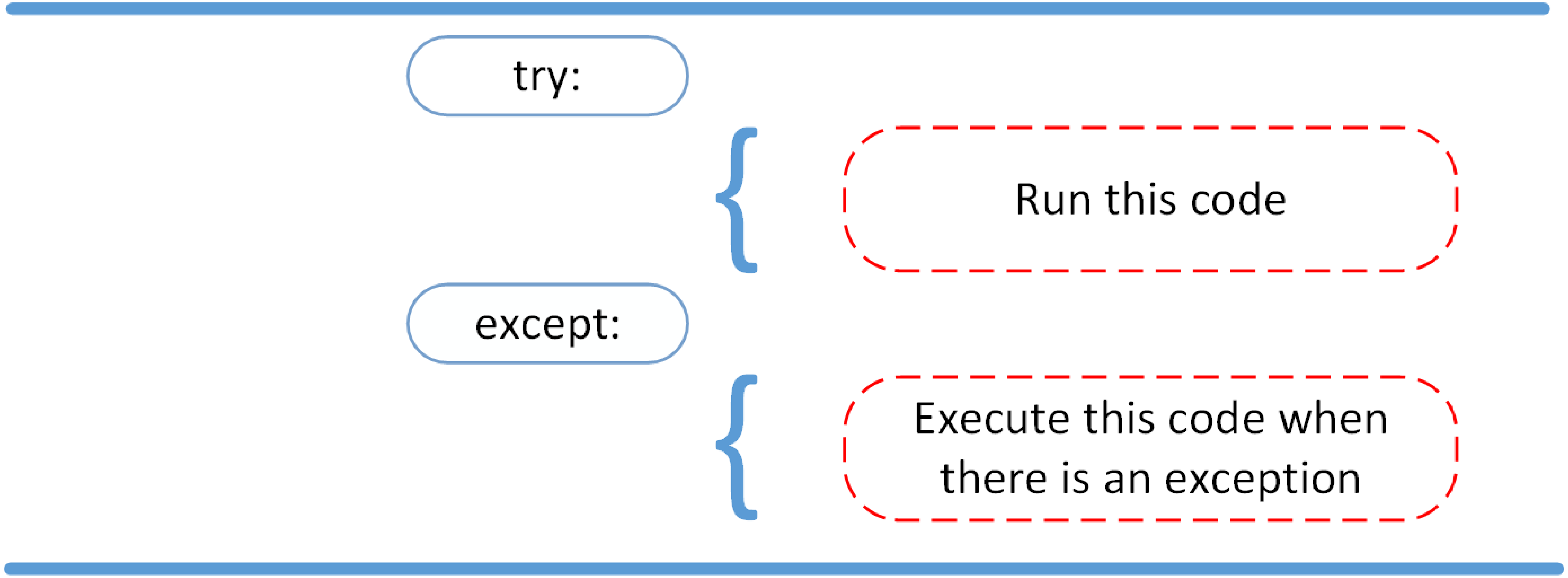
Assert that a condition is met:



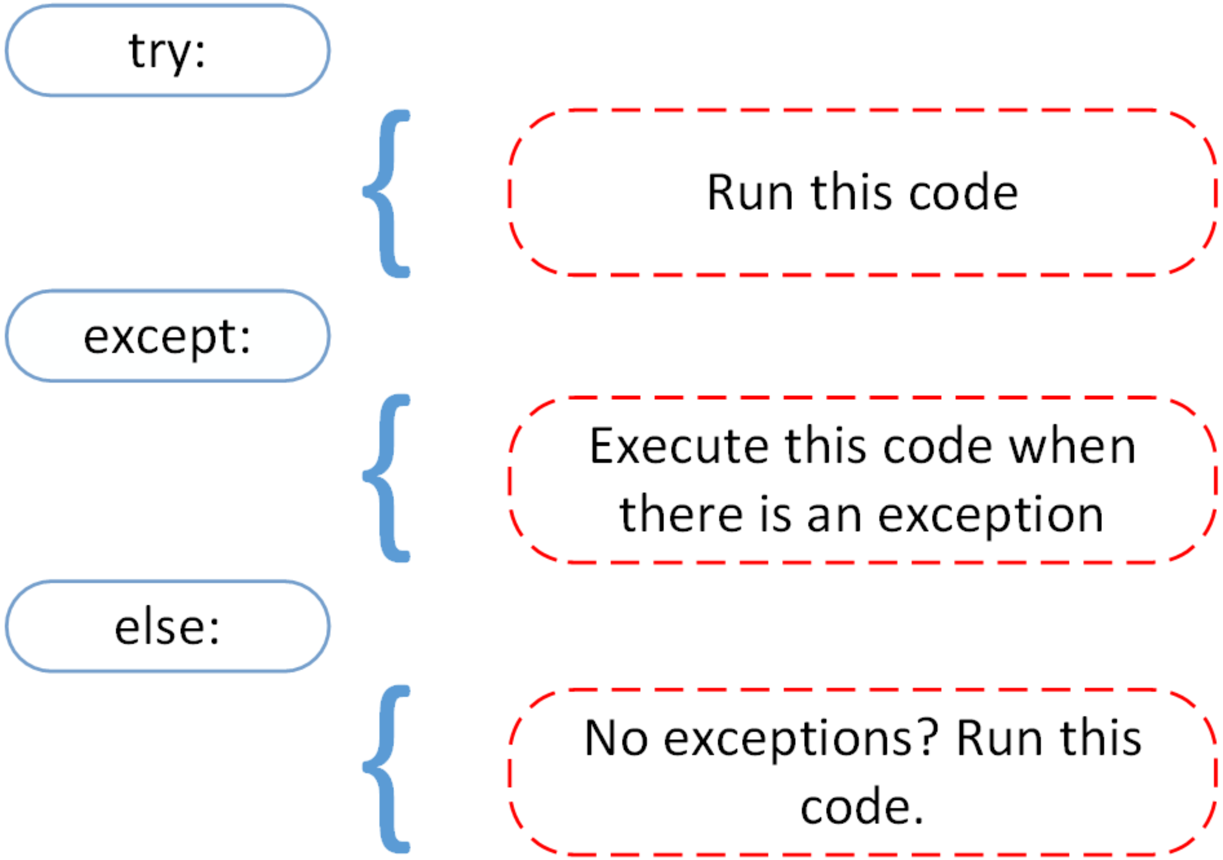
assert:

Test if condition is True

Catching Exceptions With the `try...except` Block



Handling the Success Case With `else`



try:

Run this code

except:

Execute this code when there is an exception

else:

No exceptions? Run this code.

Catching a Built-in Exception

- `FileNotFoundError`

Built-in Exceptions

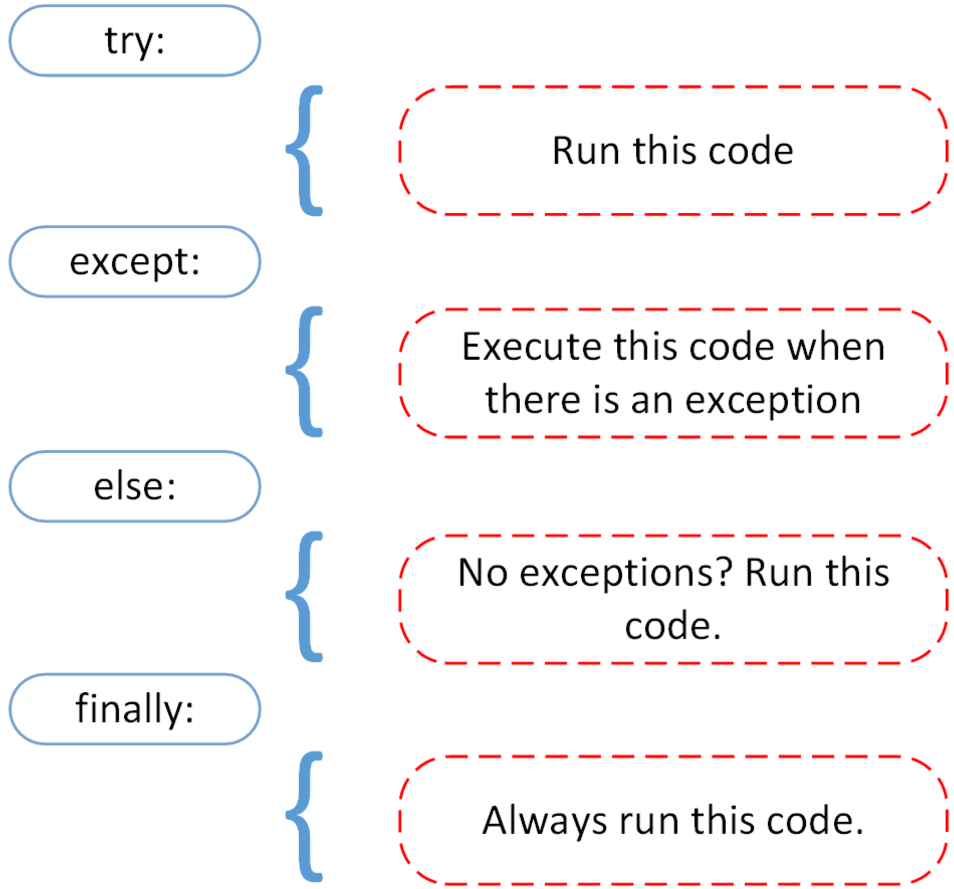
```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- StopAsyncIteration
      +-- ArithmeticError
      |    +-- FloatingPointError
      |    +-- OverflowError
      |    +-- ZeroDivisionError
      +-- AssertionError
      +-- AttributeError
      +-- BufferError
      +-- EOFError
      +-- ImportError
      |    +-- ModuleNotFoundError
      +-- 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
      |    +-- RecursionError
      +-- SyntaxError
      |    +-- IndentationError
      |         +-- TabError
      +-- SystemError
      +-- TypeError
      +-- ValueError
      |    +-- UnicodeError
      |         +-- UnicodeDecodeError
      |         +-- UnicodeEncodeError
      |         +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
           +-- ImportWarning
           +-- UnicodeWarning
           +-- BytesWarning
           +-- EncodingWarning
           +-- ResourceWarning
```

Catching a Built-in Exception

- `FileNotFoundError`

Cleaning Up After Using finally



try:

Run this code

except:

Execute this code when there is an exception

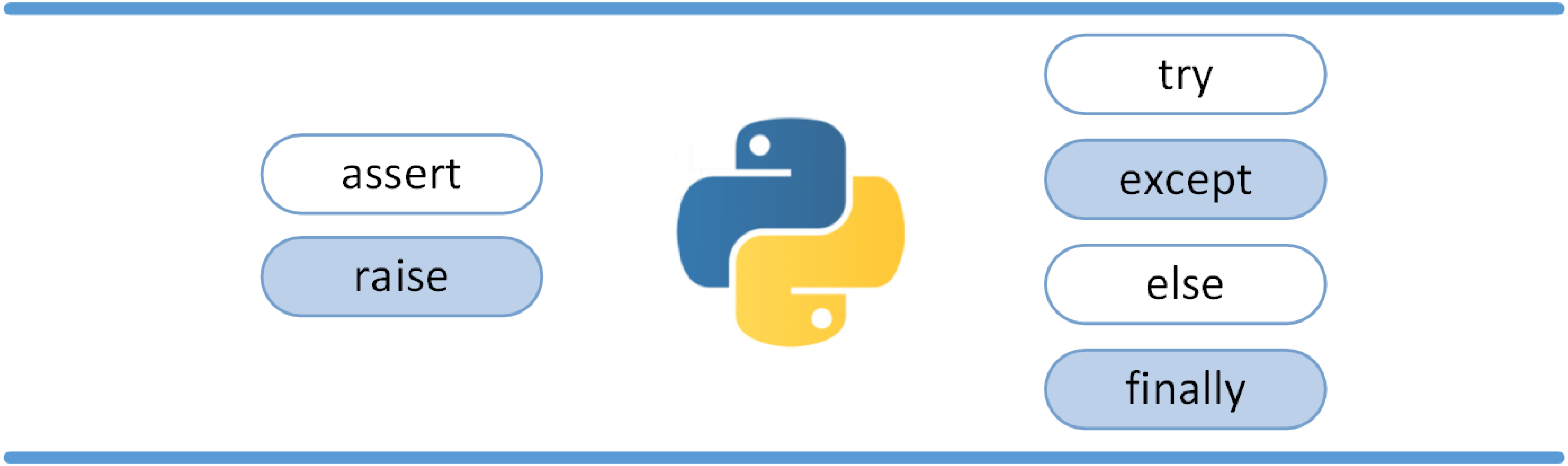
else:

No exceptions? Run this code.

finally:

Always run this code.

Summary



assert

raise

try

except

else

finally

Summary

1. Differentiating Exceptions From Syntax Errors
2. Raising an Exception
3. Using the `assert` Keyword
4. Catching Exceptions With the `try...except` Block
5. Handling the Success Case With `else`
6. Catching a Built-in Exception
7. Cleaning Up After Using `finally`

Intro to Exceptions