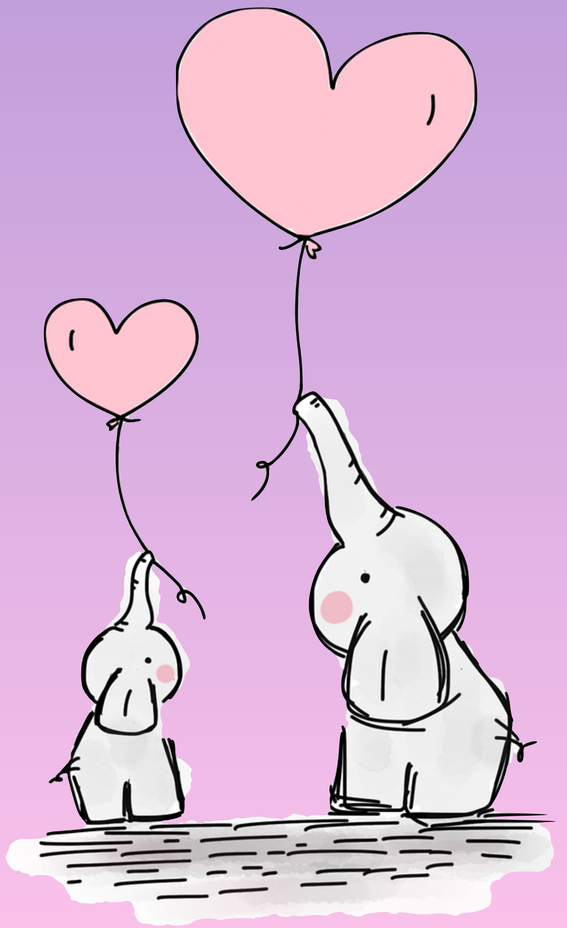
Exceptions Review



try and except

```
try:
  # Something dangerous
except ValueError:
  # Handle safely
```

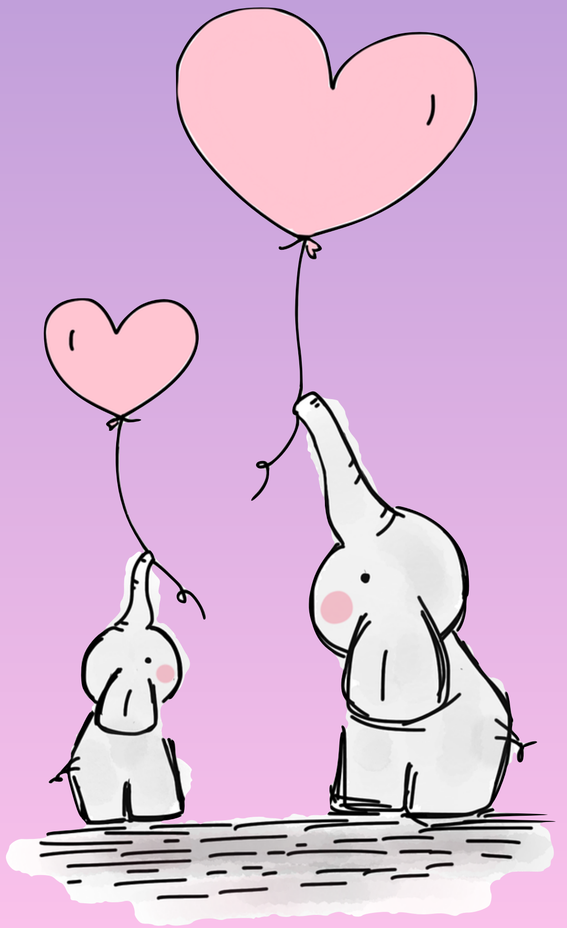
`try` and `except` in Practice

```
while True:
    try:
        course = int(input("What class are you in? "))
        if course == 41:
            print("You're in the right place!")
        else:
            print("Go find the right Zoom link!")
        break
    except ValueError:
        print("Please enter a course number.")
```

Exceptions: Summary

- `try/except/finally` is one of the paradigms of control flow in Python.
- In Python, it is typically easier to ask for forgiveness than for permission; therefore, using `try/except/finally` is often stylistically superior to error checking in advance with an `if/elif/else` block.

Exceptions



Handling Multiple Errors

```
try:
  # Something dangerous
except ValueError:
  # Handle ValueErrors
except KeyError:
  # Handle KeyErrors
```

Catching Everything

```
try:
  # Something dangerous
except:
  # Handles any exception
```

Why might this be dangerous?

Advanced Error Catching

```
try:
    # Something dangerous
except ValueError:
    # Handle ValueErrors.
else:
    # Execute if no errors have been raised.
finally:
    # Execute regardless of whether an error
    # has been raised.
```