UNIVERSITEIT STELLENBOSCH UNIVERSITY

Day four: building our knowledge

- Detecting exceptions to protect users
- Getting frequencies in collections.Counters

UNIVERSITEIT STELLENBOSCH UNIVERSITY

Exceptions result from failed expectations

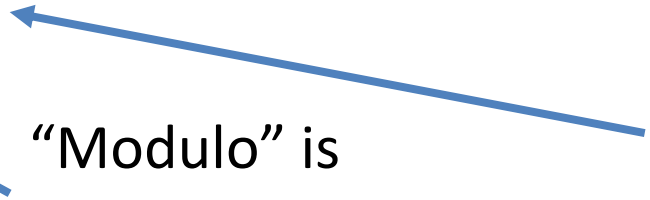
- You expected a number but got a letter.
- You tried to do something against rules.
- You try to read a file that doesn't exist.

Exceptions let your software survive these disconnects

```
try:
    print (1/0)
except ZeroDivisionError:
    print("Division by zero is not allowed!")
```

Users cause a lot of problems

```
try:
    YourNumber = int(input("What number shall I check? "))
    if (YourNumber%2 == 1):
        print("Odd!")
    else:
        print("Even!")
except ValueError:
    print("Sorry, I can only work with integers.")
    quit()
```



"Modulo" is remainder from division

Receive input

UNIVERSITEIT STELLENBOSCH UNIVERSITY

Opening a file? What could go wrong?

```
try:
    FASTAFile = open(ThisFASTA, "r")
except IOError:
    print("Error: We experienced an error opening the file!")
    quit()
```

Dict data structure

```
MyCD = {
    "Title": "Nevermind",
    "Artist": "Nirvana",
    "Year": 1991
}
ThisYear = MyCD["Year"]
```

Counters are advanced dicts

```
import collections
LetterFreqs = collections.Counter()
LetterFreqs.update("abracadabra")
LetterFreqs
```

Counter({'a': 5, 'b': 2, 'r': 2, 'c': 1, 'd': 1})

```
LetterFreqs["a"]
```

5