Writing Functions

- Define a function:

```
def <function name>(<parameter list>):
  """documentation"""      # optional doc string
  # The code would go here…
```

The function body is indented one level:

```
def computeSquare(x):
    return x * x
#anything here is not the part of the function
```

Example Functions

```
def gcd(a, b):
    "greatest common divisor"
    while a != 0:
        a, b = b%a, a    # parallel assignment
    return b

>>> gcd.__doc__
'greatest common divisor'
>>> gcd(12, 20)
4
```

Error Handling-try/except

- Use try/except blocks, similar to try/catch:

```
fridge_contents = {"egg":8, "mushroom":20,
"pepper":3, "cheese":2,
"tomato":4, "milk":13}
try:
    if fridge_contents["orange juice"] > 3:
        print("Sure, let’s have some juice!")
except KeyError:
    print("Awww, there is no orange juice.")
```

Error Handling:raise

- raise IndexError
- raise IndexError("k out of range")
- raise IndexError, "k out of range"
- Last caught exception info:
  - sys.exc_info() == (exc_type, exc_value, exc_traceback)
- Last uncaught exception (traceback printed):
  - sys.last_type, sys.last_value, sys.last_traceback

```
import sys
try:
    1/0
except:     # catch everything
  print "Oops:", sys.exc_info()
  raise     # reraise
```