02.19-exceptions

February 21, 2020

1

1.1 try & except

```
import math

while True:
    text = raw_input('> ')
    if text[0] == 'q':
        break
    x = float(text)
    y = math.log10(x)
    print "log10({0}) = {1}".format(x, y)
```

```
   q
  0
```

```
In [1]: import math

        while True:
            text = input('> ')
            if text[0] == 'q':
                break
            x = float(text)
            y = math.log10(x)
            print ("log10({0}) = {1}".format(x, y))
```

```
> 5
log10(5.0) = 0.6989700043360189
> 4
log10(4.0) = 0.6020599913279624
> 0
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

<ipython-input-1-568c099d4f47> in <module>
      6         break
      7     x = float(text)
----> 8     y = math.log10(x)
      9     print ("log10({0}) = {1}".format(x, y))


ValueError: math domain error
```

log10
try & except

```
import math

while True:
    try:
        text = raw_input('> ')
        if text[0] == 'q':
            break
        x = float(text)
        y = math.log10(x)
        print "log10({0}) = {1}".format(x, y)
    except ValueError:
        print "the value must be greater than 0"
```

try try **Python** except
try ValueErrorexcept except

```
In [2]: import math

        while True:
            try:
                text = input('> ')
                if text[0] == 'q':
                    break
                x = float(text)
                y = math.log10(x)
                print ("log10({0}) = {1}".format(x, y))
            except ValueError:
                print ("the value must be greater than 0")

> 2
log10(2.0) = 0.3010299956639812
```

```
> 5
log10(5.0) = 0.6989700043360189
> 3
log10(3.0) = 0.47712125471966244
>
```

```
---------------------------------------------------------------------------

IndexError                                Traceback (most recent call last)

<ipython-input-2-8ad4afdfe155> in <module>
  4     try:
  5         text = input('> ')
----> 6         if text[0] == 'q':
  7             break
  8         x = float(text)


IndexError: string index out of range
```

1.2

```
import math

while True:
    try:
        text = raw_input('> ')
        if text[0] == 'q':
            break
        x = float(text)
        y = 1 / math.log10(x)
        print "log10({0}) = {1}".format(x, y)
    except ValueError:
        print "the value must be greater than 0"
```

```
   y 1 / math.log10(x) 1
```

```
In [3]: import math

        while True:
            try:
                text = input('> ')
                if text[0] == 'q':
                    break
                x = float(text)
                y = 1 / math.log10(x)
```

```
            print ("log10({0}) = {1}".format(x, y))
        except ValueError:
            print ("the value must be greater than 0")
```

```
> 4
log10(4.0) = 1.660964047443681
> 8
log10(8.0) = 1.1073093649624541
> 1
```

```
---------------------------------------------------------------------------

ZeroDivisionError                         Traceback (most recent call last)

<ipython-input-3-592994153790> in <module>
  7             break
  8         x = float(text)
----> 9         y = 1 / math.log10(x)
 10         print ("log10({0}) = {1}".format(x, y))
 11     except ValueError:


ZeroDivisionError: float division by zero
```

```
except ZeroDivisionError
```

1.3

```
except Exception
```

```
In [6]: import math

        while T:
            try:
                text = input('> ')
                if text[0] == 'q':
                    break
                x = float(text)
                y = 1 / math.log10(x)
                print ("1 / log10({0}) = {1}".format(x, y))
            except Exception:
                print ("invalid value")
```

```
> 4
1 / log10(4.0) = 1.660964047443681
> 5
```

```
1 / log10(5.0) = 1.430676558073393
> 1
invalid value
> 0
invalid value
> -9
invalid value
> 1
invalid value
> q
```

1.4

ZeroDivisionError except

```
In [7]: import math

        while True:
            try:
                text = input('> ')
                if text[0] == 'q':
                    break
                x = float(text)
                y = 1 / math.log10(x)
                print ("1 / log10({0}) = {1}".format(x, y))
            except (ValueError, ZeroDivisionError):
                print ("invalid value")
```

```
> 4
1 / log10(4.0) = 1.660964047443681
> 5
1 / log10(5.0) = 1.430676558073393
> 1
invalid value
> 0
invalid value
> 1
invalid value
> q
```

```
In [8]: import math

        while True:
            try:
                text = input('> ')
```

```
                if text[0] == 'q':
                    break
                x = float(text)
                y = 1 / math.log10(x)
                print ("1 / log10({0}) = {1}".format(x, y))
            except ValueError:
                print ("the value must be greater than 0")
            except ZeroDivisionError:
                print ("the value must not be 1")
```

```
> 1
the value must not be 1
> 0
the value must be greater than 0
> 5
1 / log10(5.0) = 1.430676558073393
> -5
the value must be greater than 0
> q
```

„Exception

```
In [9]: import math

        while True:
            try:
                text = input('> ')
                if text[0] == 'q':
                    break
                x = float(text)
                y = 1 / math.log10(x)
                print ("1 / log10({0}) = {1}".format(x, y))
            except ValueError:
                print ("the value must be greater than 0")
            except ZeroDivisionError:
                print ("the value must not be 1")
            except Exception:
                print ("unexpected error")
```

```
> 1
the value must not be 1
> 2
1 / log10(2.0) = 3.321928094887362
> 3
1 / log10(3.0) = 2.095903274289385
> 0
the value must be greater than 0
> 4
```

```
1 / log10(4.0) = 1.660964047443681
> -5
the value must be greater than 0
> q
```

1.5

```
the value must be greater than 0
```

```
In [10]: float('a')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)

<ipython-input-10-688063d46f27> in <module>
----> 1 float('a')


ValueError: could not convert string to float: 'a'


ValueError
```

```
In [11]: import math

         while True:
             try:
                 text = input('> ')
                 if text[0] == 'q':
                     break
                 x = float(text)
                 y = 1 / math.log10(x)
                 print ("1 / log10({0}) = {1}".format(x, y))
             except ValueError as exc:
                 if exc == "math domain error":
                     print (exc)
                     print ("the value must be greater than 0")
                 else:
                     print (exc)
                     print ("could not convert '%s' to float" % text)
             except ZeroDivisionError:
                 print ("the value must not be 1")
             except Exception as exc:
                 print ("unexpected error:", exc.message)
```

```
> 4
1 / log10(4.0) = 1.660964047443681
> 5
1 / log10(5.0) = 1.430676558073393
> 1
the value must not be 1
> 0
math domain error
could not convert '0' to float
> -1
math domain error
could not convert '-1' to float
> q
```

```
   exc.message
```

```
ValueError: could not convert string to float: a
```

```
   message
```

```
could not convert string to float: a
```

```
   except Exception Exception Exception
```

```
try:
    pass
except:
    pass
```

1.6

```
In [12]: class CommandError(ValueError):
             pass
```

```
   ValueError
```

```
In [17]: valid_commands = {'start', 'stop', 'pause'}
```

```
         while True:
             command = input('> ')
             if command.lower() not in valid_commands:
                 raise CommandError('Invalid commmand: %s' % command)
```

```
> asasx
```

```
---------------------------------------------------------------------------

CommandError                                 Traceback (most recent call last)

<ipython-input-17-33d47ab515c9> in <module>
  4     command = input('> ')
  5     if command.lower() not in valid_commands:
----> 6         raise CommandError('Invalid commmand: %s' % command)


CommandError: Invalid commmand: asasx
```

```
raise
try/except
```

```
valid_commands = {'start', 'stop', 'pause'}

while True:
    command = raw_input('> ')
    try:
        if command.lower() not in valid_commands:
            raise CommandError('Invalid commmand: %s' % command)
    except CommandError:
        print 'Bad command string: "%s"' % command
```

```
CommandError ValueError except ValueError
```

1.7 finally

try/catch finally

try finally

```
In [19]: try:
             print(1)
         finally:
             print('finally was called.')

1
finally was called.
```

```
In [22]: try:
             print(1/0)
         finally:
             print('finally was called.')
```

```
finally was called.
```

```
        ---------------------------------------------------------------------------

        ZeroDivisionError                         Traceback (most recent call last)

        <ipython-input-22-fc8b5a4a0314> in <module>
          1 try:
    ----> 2     print(1/0)
          3 finally:
          4     print('finally was called.')


        ZeroDivisionError: division by zero
```

```
In [23]: try:
             print(1 / 0)
         except ZeroDivisionError:
             print ('divide by 0.')
         finally:
             print( 'finally was called.')
```

```
divide by 0.
finally was called.
```