Python Review and some Advanced Python

Announcement

- Please come to office hours, if you have questions for homework or bonus projects!
- Thursday, 4/21, guest lecture by Aaron Knoll on Scientific Computing and Volume rendering!

Python in Review

Lecture Notes modeled after: https://automatetheboringstuff.com/

Pratice Python at: http://www.pythontutor.com/index.html

Choose Python 3.3

Other Python Learning Tools on the web

- http://cscircles.cemc.uwaterloo.ca/
- https://wiki.python.org/moin/BeginnersGuide/NonProgrammers
- http://www.pythonchallenge.com/
- https://automatetheboringstuff.com/
- https://wiki.python.org/moin/ProblemSets
- https://projecteuler.net/

Maths in Python

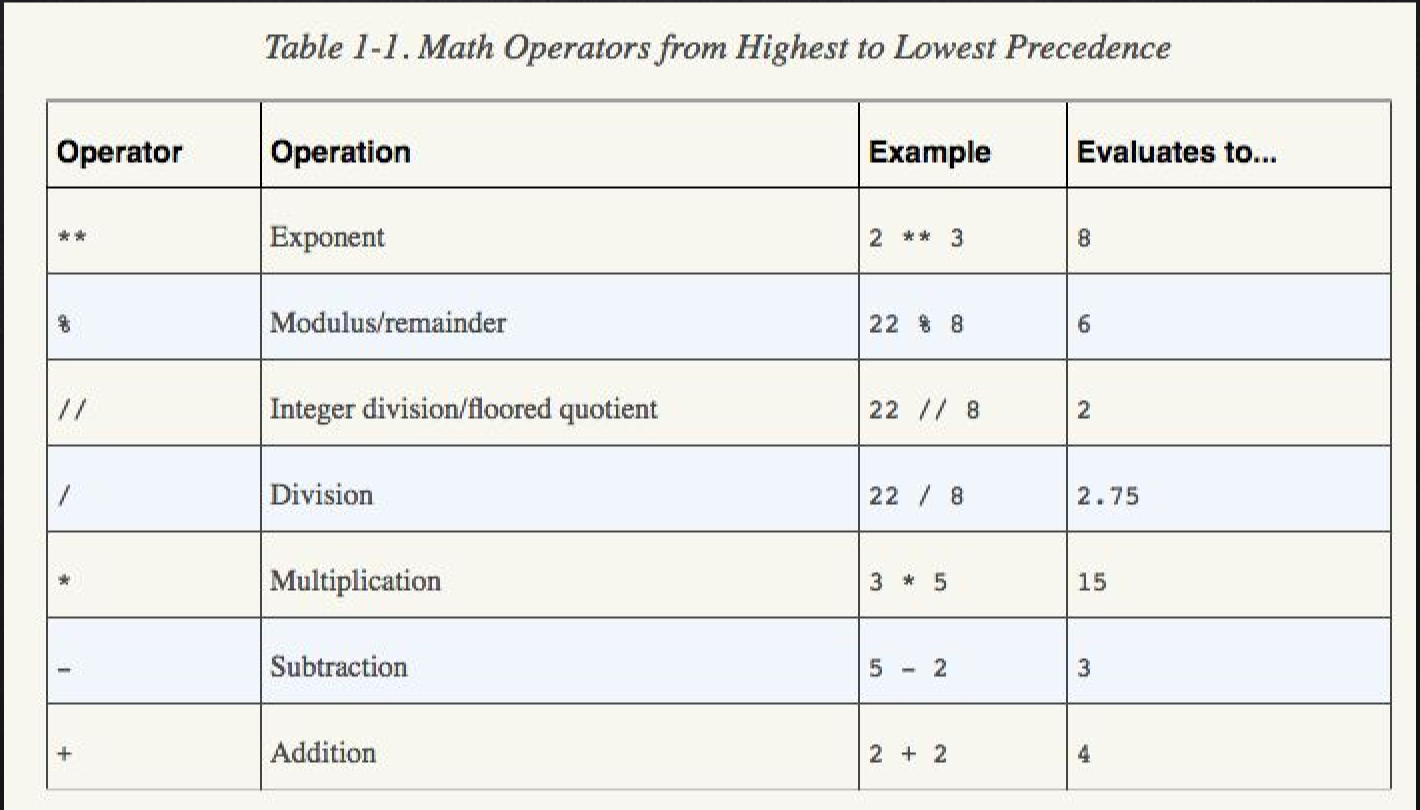


Table 1-1. Math Operators from Highest to Lowest Precedence

Operator	Operation	Example	Evaluates to...
`**`	Exponent	`2 ** 3`	`8`
`%`	Modulus/remainder	`22 % 8`	`6`
`//`	Integer division/floored quotient	`22 // 8`	`2`
`/`	Division	`22 / 8`	`2.75`
`*`	Multiplication	`3 * 5`	`15`
`-`	Subtraction	`5 - 2`	`3`
`+`	Addition	`2 + 2`	`4`

```
print (2+3) and (2–2)
print (2+1) or (2–2)
print (3*2)
print (3**2)
print (3/2)
print (3//2)
print (22/8)
```

```
5
3
6
9
1.5
1
2.75
```

Types in Python

int, float, bool, str

Think of examples of integers, floating-point numbers, strings

Store values in variables

The print() function

The len() function

```
# The bracket operator: selects a single character from a string

fruit = "banana is delicious"

print(fruit[1:4])
print(fruit[:])
print(fruit[1:])
print('a' in fruit)
```

```
ana
banana is delicious
anana is delicious
True
```

```
# Length of a string, indexing
fruit = "banana is delicious"

fruit_len = len(fruit)
print(fruit_len)
print(fruit[-2])

another_fruit = 'pineapple'
print(fruit + ' ' + another_fruit)

print((another_fruit+' ')*5)
```

```
19
u
banana is delicious pineapple
pineapple pineapple pineapple pineapple pineapple
```

The input() function

```
print("what's your name?")
name = input()
print("hello "+name+" !")
```

```
what's your name?
Bei
hello Bei !
```

str(), int(), float()

```
print(str(29))
print('Are you '+ str(3) + ' years old?')
print(int(42.34))
print(int(–99.2))
print(float('3.1415926'))
print(float('3'))
```

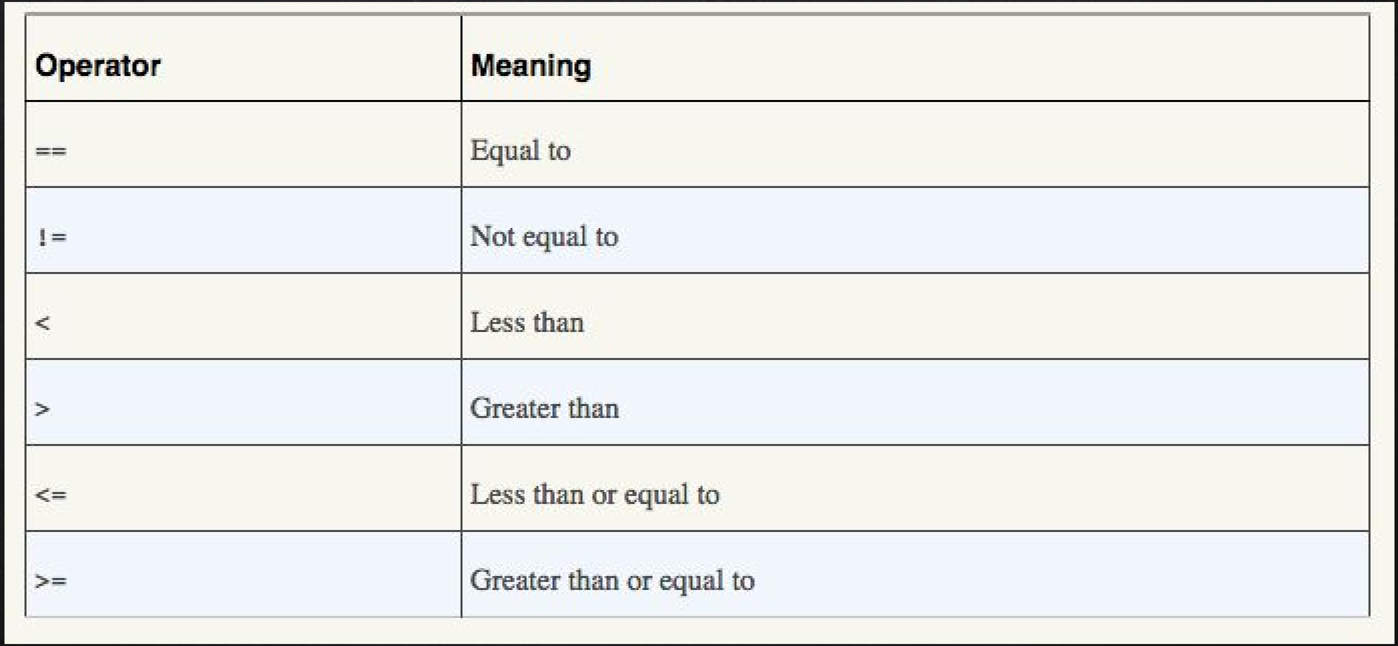
```
29
Are you 3 years old?
42
-99
3.1415926
3.0
```

Text and number equivalence

```
print(23 == '23')
print(12 == 12.0)
print(234.0 == 00234.00)
```

```
False
True
True
```

Flow control



Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Boolean

```
print(True and True)
print(True and False)
print(False or True)
print(False != True)
```

```
True
False
True
True
```

If statement

```python
name = 'Ali Baba'
password = 'Open Sesame'
if name == 'Ali Baba':
    print('Hello Ali Baba')
    if password == 'Open Sesame':
        print('Access granted.')
    else:
        print('Wrong password.')
```

Hello Ali Baba
Access granted.

While statement

```
repeat = 0
while repeat < 5:
    print('Click Here!')
    repeat = repeat + 1
```

Click Here!

Click Here!

Click Here!

Click Here!

Click Here!

For loop

```
for i in range(1, 10, 2):
    print(i, )
```

```
1
3
5
7
9
```

Functions

```
def hello(name):
   return('Hello ' + name)

print(hello('Alice'))
print(hello('Bob'))
```

```
Hello Alice
Hello Bob
```

Lists

```
print([1, 2, 3])
print(['cat', 'dog', 'rat', 'snake'])
print(['hello', 3.1415926, ['a','b','c'], None, 23])
```

```
[1, 2, 3]
['cat', 'dog', 'rat', 'snake']
['hello', 3.1415926, ['a', 'b', 'c'], None, 23]
```

```
a = [1, 2, 3]
b = ['cat', 'dog', 'rat', 'snake']
c = ['hello', 3.1415926, ['a','b','c'], None, 23]

print(a[0])
print(a[1:2])
print(b[3])
print(c[2])
print(c[2][2])
```

```
1
[2]
snake
['a', 'b', 'c']
c
```

```
animals = ['cat', 'dog', 'rat', 'tiger']
print (animals[-1]+ ' is afraid of '+animals[-3])
```

tiger is afraid of dog

```
animals = ['cat', 'dog', 'rat', 'tiger']
print (animals[2:4])
```

```
['rat', 'tiger']
```

```
print([1, 2, 3] + ['A', 'B', 'C'])
print(['X', 'Y', 'Z'] * 3)

mylist = [1,2,3,4,5,6]
del mylist[2]
print(mylist)

print(1 in mylist)
```

```
[1, 2, 3, 'A', 'B', 'C']
['X', 'Y', 'Z', 'X', 'Y', 'Z', 'X', 'Y', 'Z']
[1, 2, 4, 5, 6]
True
```

```
greetings = ['hello', 'hi', 'howdy', 'heyas']

g1, g2, g3, g4 = greetings

print(g1)
print(g2)
print(g3)
print(g4)
```

hello
hi
howdy
heyas

List-like types: strings and tuples: Not Mutable

```
breakfast = ('egg', 'bread', 'milk')
print(breakfast[1:2])
print(len(breakfast))
breakfast[0]='orange'
```

```
('bread',)
3
TypeError: 'tuple' object does not support item assignment
```

Conversion between lists and tuples

```
print(tuple(['cat', 'dog', 9]))

print(list(('cat', 'dog', 9)))

print(list('hello'))
```

```
('cat', 'dog', 9)
['cat', 'dog', 9]
['h', 'e', 'l', 'l', 'o']
```

Strings are immutable: can't change a character in the string

A bit of advanced Python

Dictionaries

K-value pairs

```
myCat = {'size': 'fat', 'color': 'white', 'noise': 'loud'}

print(myCat['size'])

print('My cat has ' + myCat['color'] + ' fur.')
```

fat
My cat has white fur.

Dictionaries v.s. lists:
Dictionaries are not ordered

```
meals = ['egg', 'milk', 'bacon']
lunch = ['milk', 'egg', 'bacon']
print(meals==lunch)

meals = {'protein': 'egg', 'drink': 'milk', 'meat': 'bacon'}
dinner = {'protein': 'egg', 'meat': 'bacon', 'drink': 'milk'}
print(meals == dinner)
```

```
False
True
```

keys(), values(), items()

```
meals = {'protein': 'egg', 'drink': 'milk', 'fruit': 'orange'}

for v in meals.values():
    print(v)

for k in meals.keys():
    print(k)

for i in meals.items():
    print(i)
```

```
orange
milk
egg
fruit
drink
protein
('fruit', 'orange')
('drink', 'milk')
('protein', 'egg')
```

Classes and Objects

Further reading: http://www.openbookproject.net/thinkcs/python/english2e/ch13.html

```
class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def distance_from_origin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5

p = Point(3, 4)
print(p.x)
print(p.y)
print(p.distance_from_origin())
```

```
3
4
5.0
```

Coming Up Next: Scientific Visualization

THANKS!

Any questions?

You can find me at
beiwang@sci.utah.edu

http://www.sci.utah.edu/~beiwang/teaching/cs1060.html

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash