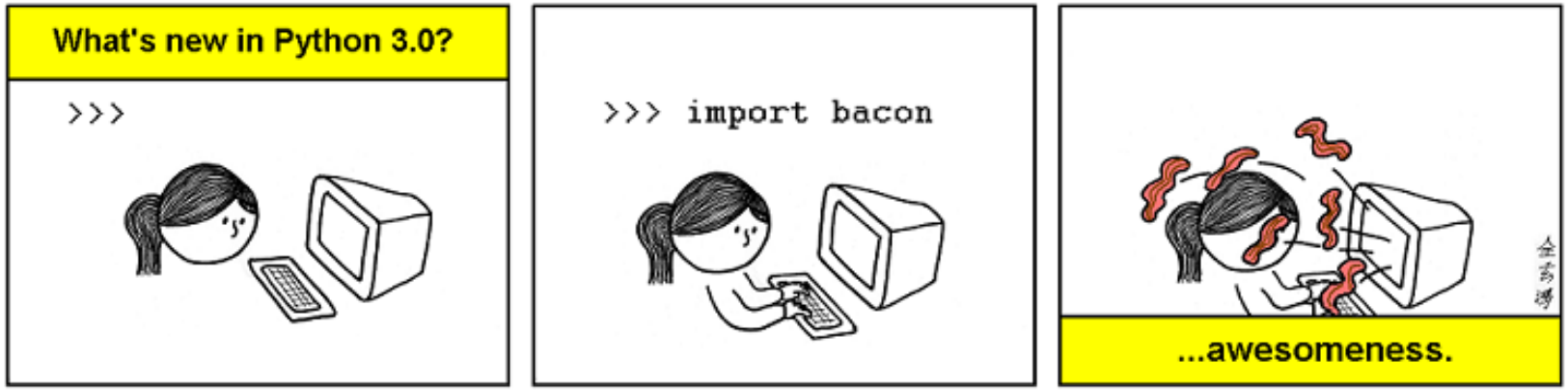
# My file is **TOO BIG**!

A (brief) introduction to parsing in python.




http://abstrusegoose.com/

# Remembering classes

What are the two things that every class needs to be defined in python?

If some object “carl” has an attribute “hairLoss”, how would I print out that value?

If “carl” has a function “applyWig()” how would I use it?

How are special functions used by the python processer named? (we used one of these a lot.)

# Remembering classes

```
class Dinosaur:
    def __init__(self, color, height, call = "Rawr!"):
        self.height = height
        self.color = color
        self.call = call

    def __str__(self):
        return "\"I am a mighty "+self.color+" dinosaur! I stand "+str(self.height)+" meters tall. "+self.call+"\""

bob = Dinosaur("green", 200)
print(bob)
print("Bob is "+str(bob.height/3.28)+" feet tall.\n")

betty = Dinosaur("blue", 232, "Beep boop bop!")
print(betty)
print("Betty is "+str(betty.height/3.28)+" feet tall.\n")
```

# Remembering classes

```
1  class Dinosaur:
2      def __init__(self, color, height, call = "Rawr!"):
3          self.height = height
4          self.color = color
5          self.call = call
6
7      def __str__(self):
8          return "\"I am a mighty "+self.color+" dinosaur! I stand "+str(self.height)+" meters tall. "+self.call+"\""
9
10 bob = Dinosaur("green", 200)
11 print(bob)
12 print("Bob is "+str(bob.height/3.28)+" feet tall.\n")
13
14 betty = Dinosaur("blue", 232, "Beep boop bop!")
15 print(betty)
16 print("Betty is "+str(betty.height/3.28)+" feet tall.\n")
```

```
"I am a mighty green dinosaur! I stand 200 meters tall. Rawr!"
Bob is 60.9756097561 feet tall.

"I am a mighty blue dinosaur! I stand 232 meters tall. Beep boop bop!"
Betty is 70.7317073171 feet tall.
```

An **Iterator** is used to step through an object's data.

```
def myIterator(myList):
    for item in mylist:
        yield (item)
```

Iterators use the yield keyword

An **Iterator** is used to step through an object's data.

```
def myIterator(myList):
    for item in mylist:
        yield (item)
```

```
aList = [1,5,6,2]
for thing in myIterator(aList):
    print(thing)
```

To use an iterator you just use a for loop

An **Iterator** is used to step through an object's data. But you can specialize it depending on your needs.

```
def mySquareIterator(myList):
    for item in mylist:
        yield (item**2)
```

An **Iterator** is used to step through an object's data. But you can specialize it depending on your needs.

```
def mySquareIterator(myList):
    for item in mylist:
        yield (item**2)
```

```
aList = [1,5,6,2]
for thing in mySquareIterator(aList):
    print(thing)
```

Objects can have **iterator** functions…

```
class Dinosaur:
    def __init__(self, color, height, call = "Rawr!"):
        self.height = height
        self.color = color
        self.call = call

    def __iter__(self):
        yield self.height
        yield self.color
        yield self.call

    def __str__(self):
        return "\"I am a mighty "+self.color+" dinosaur! I stand "+str(self.height)+" meters tall. "+self.call+"\""

bob = Dinosaur("green", 200)

for thing in bob:
    print(thing)
```

Objects can have **iterator** functions…

```
class Dinosaur:
    def __init__(self, color, height, call = "Rawr!"):
        self.height = height
        self.color = color
        self.call = call

    def __iter__(self):
        yield self.height
        yield self.color
        yield self.call

    def __str__(self):
        return "\"I am a mighty "+self.color+" dinosaur! I stand "+str(self.height)+" meters tall. "+self.call+"\""

bob = Dinosaur("green", 200)

for thing in bob:
    print(thing)
```

```
200
green
Rawr!
```