

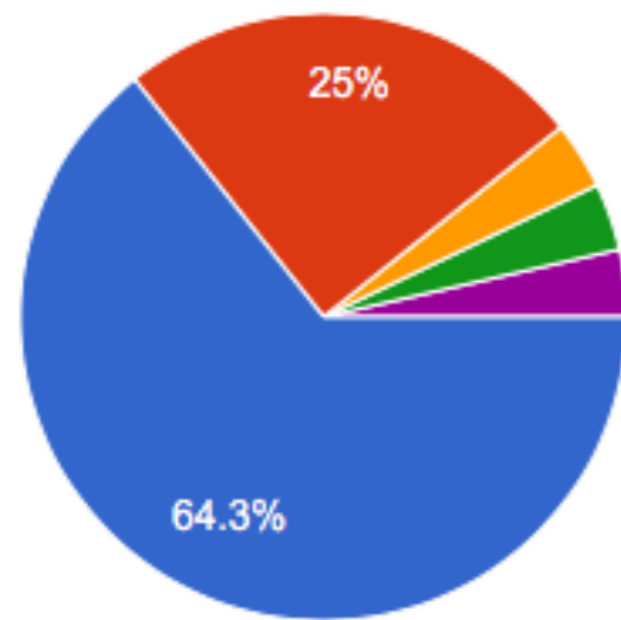
#5 Object Oriented Programming

TA: Jerry Chen (jerry.c@berkeley.edu)

Did you hear about the new object oriented get-rich-quick scheme? It's called "inheritance."

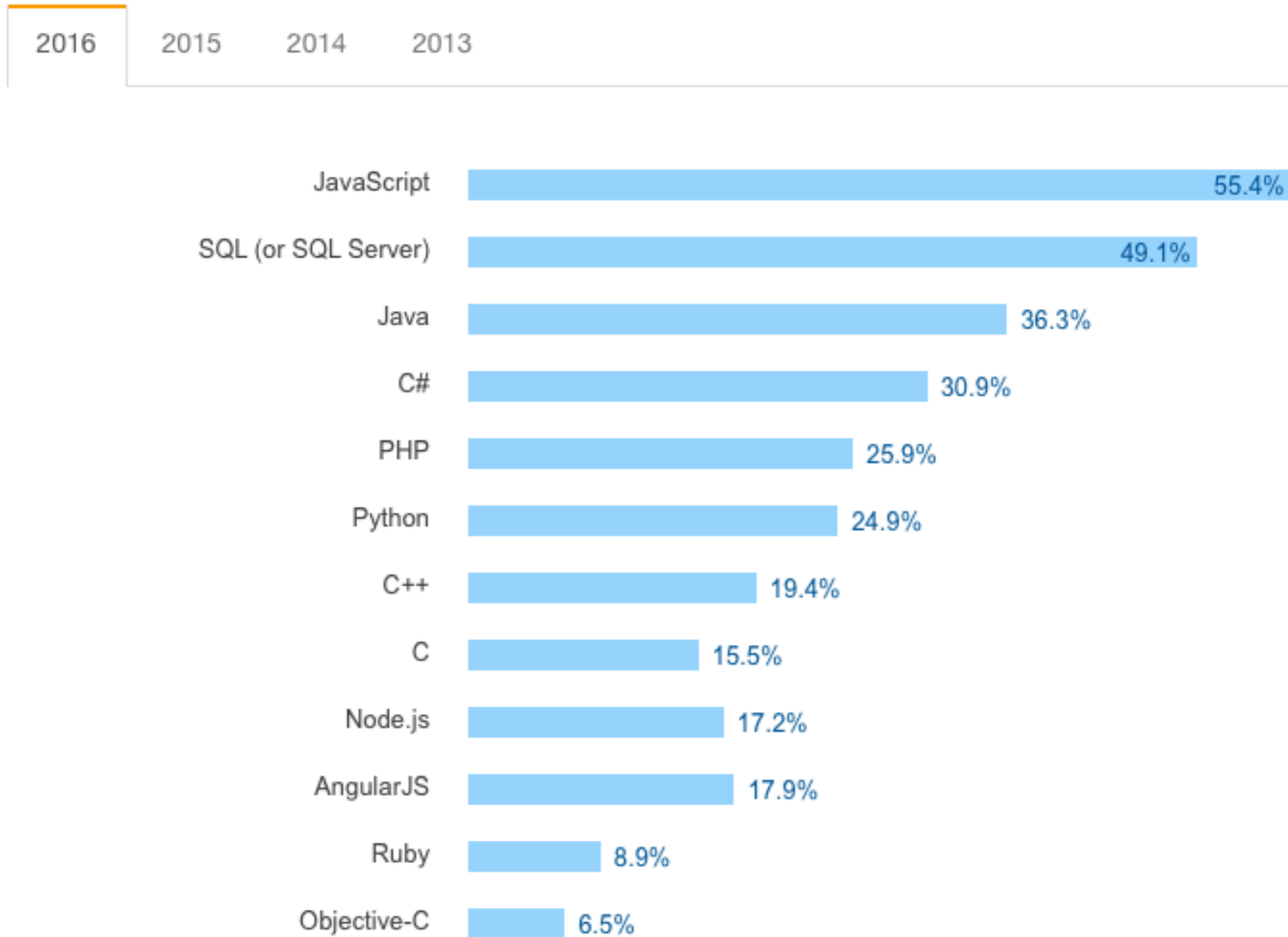
Who would win in a fight? Hog or Maps?

28 responses

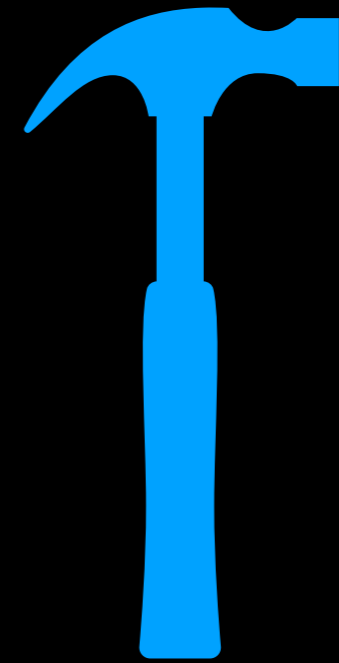


- Hog
- Maps
- Ants
- Tom
- Neither :(

I. Most Popular Technologies







OOP in Python

Some vocabulary

- Class - a "template" for a object
- Object - an instance of a class
- Attribute - data
 - **Instance attributes** are specific to an object
 - **Class attributes** are shared with objects of the class
- Method - a function that has been bound to an object


```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
```

```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
```

```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
```

```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
```

```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
9 dumbledore.lecture("Magic")
```

```
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
9 dumbledore.lecture("Magic")
```

```
1 class Car:
2     def drive(self):
3         print("I am definitely a car")
4
5 class Boat:
6     def __init__(self):
7         self.is_car = 'Nope'
8 b = Boat()
9
10 Car.drive(b)
11 b.drive()
12 Car.drive("car")
13 Car.drive()
```

```
1 class Car:
2     def drive(self):
3         print("I am definitely a car")
4
5 class Boat:
6     def __init__(self):
7         self.is_car = 'Nope'
8 b = Boat()
9
10 Car.drive(b) ✓
11 b.drive()
12 Car.drive("car")
13 Car.drive()
```



```
1 class Car:
2     def drive(self):
3         print("I am definitely a car")
4
5 class Boat:
6     def __init__(self):
7         self.is_car = 'Nope'
8 b = Boat()
9
10 Car.drive(b) ✓
11 b.drive() ✗
12 Car.drive("car")
13 Car.drive()
```

```
1 class Car:
2     def drive(self):
3         print("I am definitely a car")
4
5 class Boat:
6     def __init__(self):
7         self.is_car = 'Nope'
8 b = Boat()
9
10 Car.drive(b) ✓
11 b.drive() ✗
12 Car.drive("car") ✓
13 Car.drive()
```



```
1 class Car:
2     def drive(self):
3         print("I am definitely a car")
4
5 class Boat:
6     def __init__(self):
7         self.is_car = 'Nope'
8 b = Boat()
9
10 Car.drive(b) ✓
11 b.drive() ✗
12 Car.drive("car") ✓
13 Car.drive() ✗
```

```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```

```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```



```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```





```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```




✓

✗ ✓





```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```




```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10     def drive():
11         print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```



```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```



```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```



```
1 class Car:
2     def __init__(not_self):
3         not_self.tires = 10
4
5 class Funky:
6     def __init__():
7         print("No self?")
8
9 class BoatCar(Boat):
10    def drive():
11        print("I am definitely... a boatcar")
12
13 b = BoatCar()
14 b.drive()
15 BoatCar.drive()
```

