# Python Brain Teasers

EXERCISE YOUR MIND

```
1 class Player:
2     # Number of players in the Game
3     count = 0
4
5     def __init__(self, name):
6         self.name = name
7         self.count += 1
8
9
10 p1 = Player('Parzival')
11 print(Player.count)
```

What will this code print?

30 mind bending teasers & solutions

Miki Tebeka

# Copyright

# A Task to Do

*tasks.py*

```
from heapq import heappush, heappop

# Tasks priority queue
# A task is a tuple of (priority, payload)
tasks = []
heappush(tasks, (30, 'work out'))
heappush(tasks, (10, 'wake up'))
heappush(tasks, (20, 0xCAFFE))
heappush(tasks, (20, 'feed cat'))
heappush(tasks, (40, 'write book'))

while tasks:
    _, payload = heappop(tasks)
    print(payload)
```

Try to guess what the output is before moving to the next page.

This code will raise a `TypeError` exception.

The built-in heapq module implements min-heap over lists.

It's common to use a heap for a priority queue. Pushing and deleting form the heap are `log(N)` operations and the first item in the heap (e.g. `tasks[0]`) is always the smallest.

To compare items in the heap, `heapq` uses the comparison defined it the object's type (using the `<` operator which maps to the specific type's `__lt__` special method). The objects in the `tasks` heap are tuples. Python orders tuples in a lexicographical order - very much like books are ordered in the library. [1] Lexicographical order compares the first two items, then the second two ... finally if all items equal, the longer tuple is considered bigger.

In line 13, you pop the first item from `tasks`, which is `(10, 'wake up')`. After this item is removed from the heap, `heapq` will move the smallest item to the top of the heap. There are two candidates `(20, 'feed cat')` and `(20, 0xCAFFE)`, [2] since the first items in these tuples are equal - Python will try to compare the second items. Comparing `'feed cat'` (a `str`) with `0xCAFFE` (an `int`) - will raise an exception.[3]

# Further Reading

- Heap data structure
- Lexicographical order
- Tuples and Sequences

[1] This is also how strings and lists are ordered.

[2] `0xCAFFE` is a hexadecimal (base 16) number. Writing "English" this way is called Leet.

[3] In Python 3, Python 2 will happily compare `str` and an `int`.