



Binary & WebSockets

```
[0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,  
0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,  
1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1,  
1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,  
1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,  
0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1,  
1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0]
```

Agenda

- Binary data: <beep> <boop>!
- Python's relationship with bits, bytes, chars and strings
- A WebSocket server implementation, step-by-step

Binary Refresher

The basics, mainly for my benefit

Binary Refresher

Python, The Drink



- 50 mL White Rum
- 30 mL Mint Liqueur
- 30 mL Pineapple Juice
- Juice of Half a Lemon
- Sprite or 7 Up

1. Shake together with ice and strain.
2. Pour into a glass and add Sprite or 7 Up.
3. Garnish with a slice of lime.

This cocktail must be served fast - its froth disappears quickly. This drink has a fresh pineapple taste.

https://github.com/the-teacher/cocktails_for_programmers/blob/master/cocktails_for_programers.md

Binary Refresher

- We'll be dealing with basic definitions:
 - Bit:
 - 0 | 1
 - Bytes (or, Octets): 8 bits
 - 'a' == 97 == '01100001'
 - Worrying only about unsigned values
 - Endianness is a concern

Binary Refresher

Jupyter Notebook

Working With Bytes

in Python

Working With Bytes

- In Python 2 (a rapidly fading memory) bytes, and strings were one and the same
- The `str()` class was used for readable text and for binary storage
- Unicode was still a second-class citizen

Working With Bytes

- In Python 3, everything's changed
 - The `str()`, `bytes()`, `bytearray()` types, handle Unicode strings, immutable bytes, mutable bytes
 - Other features all exist support value transformation and base conversions

Working With Bytes

Jupyter Notebook

Nibbles and Bits

Python Techniques

Nibbles and Bits

- Python has no built-in data type for representing binary bits
- Thanks to Python's infinite integer representation, you could use an **int** value to pack and store the bits
- Still, it's not a convenient way to see, test or modify bits, as if they were in a list, for example

Nibbles and Bits

- We can choose from a few options for creating easy to use bit field lists
 - Manual process (see Jupiter Notebook)
 - The *bitarray* package
 - The *bitstring* package
 - The *BitVector* package

Nibbles and Bits

- Also, working with packed bits can be done with
 - *ctypes*
 - *struct*

Nibbles and Bits

- Today, we'll focus on the painful process of manually working with bits

Nibbles and Bits

Jupyter Notebook

Case Study

*tinymce: A WebSocket Server
Implementation*



Conclusion

Q & A

