Objectives

- Computer's representations of data types

Review

- How can we find out the length of a string?
- How can we "get" the last character of a string? (2 ways)
- How can we "get" the first 3 characters of a string?
- What are the two ways to iterate through a string?
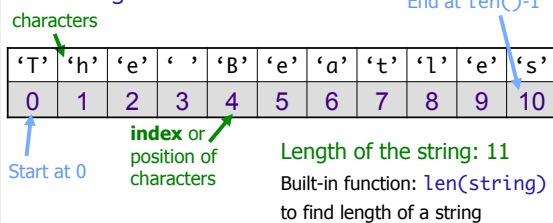- How can we tell if some string is part of some other string?

Review: Strings

- Actually a *sequence* of characters
  - Example:
    ```
    string = "The Beatles"
    ```

End at `len()`-1

characters

'T'	'h'	'e'	' '	'B'	'e'	'a'	't'	'l'	'e'	's'
0	1	2	3	4	5	6	7	8	9	10

Start at 0

index or position of characters

Length of the string: 11

Built-in function: `len(string)` to find length of a string

Review: Iterating Through a String

- For each character in the string

string of length 1
```
for char in string:
    print char
```

'T'	'h'	'e'	' '	'B'	'e'	'a'	't'	'l'	'e'	's'
0	1	2	3	4	5	6	7	8	9	10

An integer
```
for pos in xrange(len(string)):
    print string[pos]
```

- For each position in the string   Index into the string

Review: Substrings Operator []

- Look at a particular character in the string
  - Syntax: `mystr[<integer expression>]`
  - [Positive value]: index of character
  - [Negative value]: count backwards from end
- Look at a sequence of characters in the string
  - Syntax: `mystr[<start>:<end>]`

Review: Testing for Substrings

- Using the **in** operator
  - Used **in** before in **for** loops
- Syntax:
    ```
    substring in string:
    ```
  - Evaluates to `True` or `False`
- Example:
    ```
    if "Jr." in name:
        print name, "contains 'Jr.'"
    ```

HOW THE COMPUTER REPRESENTS DATA

Representations of Data

- Computer needs ways to represent different types of data
  - Eventually, all boils down to 1s and 0s
- Computer needs to translate between what humans know to what computer knows and back again

decimal, strings   binary   decimal, strings

Seems like a divergence on strings but just wait...

Decimal Representations

- Decimal is base 10
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each *position* in a decimal number represents a power of 10

Decimal Representations

- Decimal is base 10
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each *position* in a decimal number represents a power of 10
- Example: 54,087

5	4	0	8	7
10^4	10^3	10^2	10^1	10^0

- $= 5*10^4 + 4*10^3 + 0*10^2 + 8*10^1 + 7*10^0$
- $= 5*10,000 + 4*1000 + 0*100 + 8*10 + 7*1$

Number Representations

Characteristic	Decimal	Binary
Base	10	2
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1
Position represents	Power of 10	Power of 2

- Binary: two values (0, 1)
  - Like a light switch (either **off** or **on**) or booleans (either True or False)
- 0 and 1 are *binary digits* or **bits**
  - 64-bit machine: represents numbers (and other data) with 64 bits

---

Binary Representation

- Binary number: 1101

1	1	0	1
2^3	2^2	2^1	2^0

- $= 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$
- $= 1*8 + 1*4 + 0*2 + 1*1$
  - Decimal value: 13

- **Practice**: what is the decimal value of the binary number 10110?

Binary Representation

- Binary Number: 10110

1	0	1	1	0
2^4	2^3	2^2	2^1	2^0

- = $1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0$
- = 1*16 + 0*8 + 1*4 + 1*2 + 0*1
  - 22

Generalize this process into an algorithm…

Algorithm: Converting Binary to Decimal

Accumulator design pattern

1. Read in the binary number as a string
   - The starting exponent will be the length of the string-1
2. Initialize the result to zero
3. For each bit in the binary number
   - Multiply the bit by the appropriate power of 2
   - Add this to the result
   - Reduce the exponent by 1
4. Display the result

Implement algorithm binaryToDecimal.py

Algorithm: Converting Decimal to Binary

1. Read in the decimal as an integer
2. Initialize the result to the empty string
3. Repeat until the decimal is 0:
   - result = str(decimal % 2) + result
   - decimal = decimal / 2
4. Display the result

Try out algorithm with 22

Implement algorithm decimalToBinary.py

String Representations

- A **string** is a sequence of characters
- Each character is stored as a binary number
- **ASCII** (American Standard Code for Information Interchange) is one standard encoding for characters
  - Limitation: ASCII is based on the English language
  - Cannot represent other types of characters
- Unicode is a new standard

ASCII Questions

- Lowercase letters are represented by what range of numbers?

- Uppercase letters are represented by what range of numbers?

- What is the difference between the decimal encoding of 'M' and 'N'?
  - Between 'm' and 'n'?

ASCII Questions

- Lowercase letters are represented by what range of numbers?
  - 97--122
- Uppercase letters are represented by what range of numbers?
  - 65--90
- What is the difference between the decimal encoding of 'M' and 'N'?
  - Between 'm' and 'n'?
  - 1

Translating to/from ASCII

- Translate a character into its ASCII numeric code using **built-in function ord**
  - ord('a') ==> 97
- Translate an ASCII numeric code into its character using **built-in function chr**
  - chr(97) ==> 'a'

ascii_table.py

Encryption

- Process of encoding information to keep it secure
- One technique: Substitution Cipher
  - Each character in message is replaced by a new character

Caesar Cipher

- Replace with a character X places away
  - X is your *key*
- Julius Caesar used it to communicate with his generals
- "Wrap around"
- Write program(s) to do this in next lab

Caesar Cipher

- Using the ASCII handout, what would be the encoded messages?

Message	Key	Encoded Message
apple	5	
zebra	5	
the eagle flies at midnight	-5	

Next Lab

- Write an encoding/decoding program
  - Encode a message
  - Give to a friend to decode

This Week

- Broader Issue: Excel Bug
- Lab 5