Mini-Lecture 5

Strings

String: Text as a Value

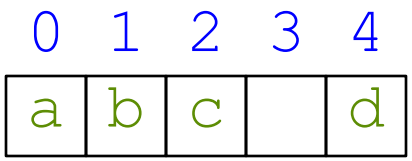
- String are quoted characters
  - `'abc d'` (Python prefers)
  - `"abc d"` (most languages)
- How to write quotes in quotes?
  - Delineate with "other quote"
  - **Example**: `" ' "` or `' " '`
  - What if need both `"` and `'` ?
- **Solution**: escape characters
  - Format: \ + letter
  - Special or invisible chars

Type: `str`

Char	Meaning
`\'`	single quote
`\"`	double quote
`\n`	new line
`\t`	tab
`\\`	backslash

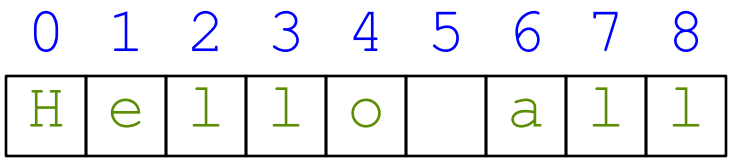
String are Indexed

- s = 'abc d'



- Access characters with []
  - s[0] is 'a'
  - s[4] is 'd'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'
- Called "string slicing"

- s = 'Hello all'

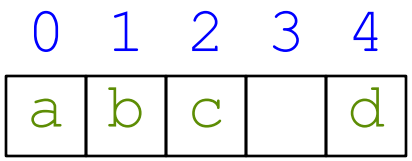


- What is s[3:6]?

A: 'lo a'
B: 'lo'
C: 'lo '
D: 'o '
E: I do not know

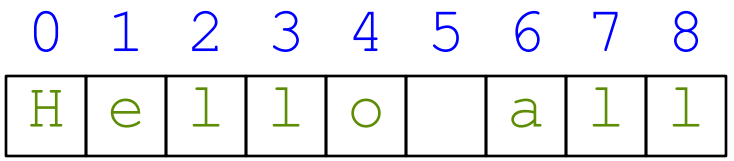
String are Indexed

- s = 'abc d'



- Access characters with []
  - s[0] is 'a'
  - s[4] is 'd'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'
- Called "string slicing"

- s = 'Hello all'

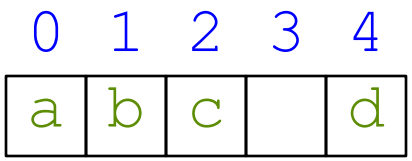


- What is s[3:6]?

A: 'lo a'
B: 'lo'
C: 'lo ' **CORRECT**
D: 'o '
E: I do not know

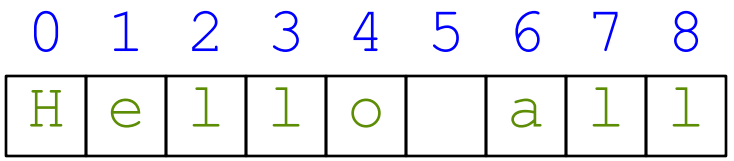
String are Indexed

- s = 'abc d'



- Access characters with []
  - s[0] is 'a'
  - s[4] is 'd'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'
- Called "string slicing"

- s = 'Hello all'

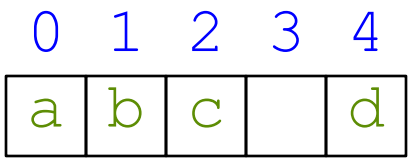


- What is s[:4]?

A: 'o all'
B: 'Hello'
C: 'Hell'
D: Error!
E: I do not know

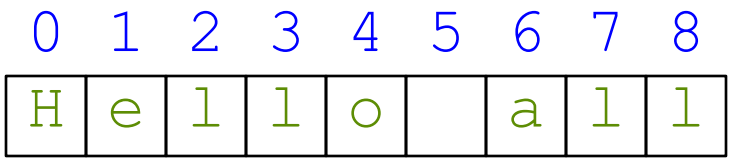
String are Indexed

- s = 'abc d'



- Access characters with []
  - s[0] is 'a'
  - s[4] is 'd'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'
- Called "string slicing"

- s = 'Hello all'



- What is s[:4]?

A: 'o all'
B: 'Hello'
C: 'Hell' **CORRECT**
D: Error!
E: I do not know

Other Things We Can Do With Strings

- **Operation** in: s_1 in s_2
  - Tests if s_1 "a part of" s_2
  - Say s_1 a *substring* of s_2
  - Evaluates to a bool
- **Examples**:
  - s = 'abracadabra'
  - 'a' in s == True
  - 'cad' in s == True
  - 'foo' in s == False
- **Function** len: len(s)
  - Value is # of chars in s
  - Evaluates to an int
- **Examples**:
  - s = 'abracadabra'
  - len(s) == 11
  - len(s[1:5]) == 4
  - s[1:len(s)-1] == 'bracadabr'

String Functions

- The introcs module has several string functions
  - Installed as part of Cornell Extensions
  - http://cs1110.cs.cornell.edu/docs/strings.html
- Use these instead of **methods** for now
  - Methods are an advanced programming feature
  - You will see them on Stack Overflow
  - Will come back to methods later
- Will need these functions for Assignment 1

Examples of String Functions

- introcs.index_str(s_1,s_2)
  - Position of the first instance of s_2 in s_1
- introcs.count_str(s_1,s_2)
  - Number of times s_2 appears inside of s_1
- introcs.strip(s)
  - A copy of s with white-space removed at ends

- s = 'abracadabra'
- from introcs import *
- index_str(s,'a') == 0
- index_str(s, 'rac') == 2
- count_str(s, 'a') == 5
- count_str(s, 'x') == 0
- strip(' a b ') == 'a b'

See IntroCS Docs for more