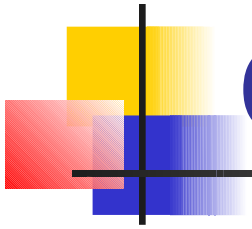
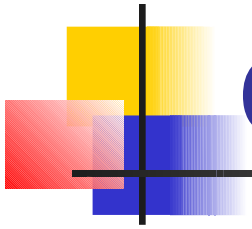


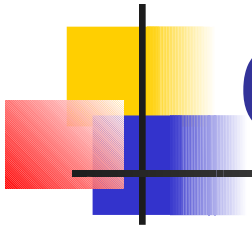
CS105

STRING
LIST
TUPLE
DICTIONARY



Characteristics of Sequence

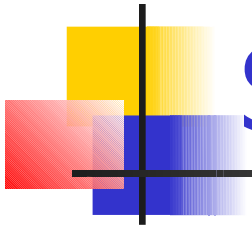
- What is sequence data type?
  - It stores several objects
  - Each object has an order
  - Each object can be referred by an index
- Data types in sequence
  - List
  - Tuple
  - String



Operations in sequence

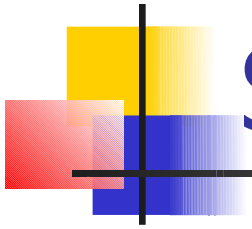
- Indexing e.g., s[i]
- Slicing e.g., s[1:5]
- Concatenation e.g., s + t
- Repetition e.g., s * 5
- Membership test e.g., ‘a’ in s
- Length e.g., len(s)

➔ IR ML CS



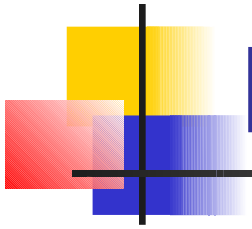
String

- Characteristics of sequence data types
- Definition of string
- Modification of string
- Formatting
- String methods
- String modules



String definition

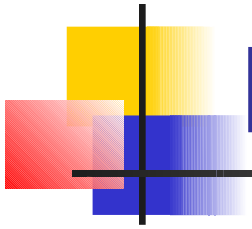
- One line string
  - Use ’ ’ or ” ”
- Multiple line string
  - Use ”’ ”’ or ”””” ””””
- Escape sequence
  - Use backspace



Modification of string

- Not possible to modify it directly
- Substitute with a new name
  - >>> s = ‘spam and egg’
  - >>> s = s[:5] + ‘cheese‘ + s[5:]
  - >>> s

    ‘spam cheese and egg’



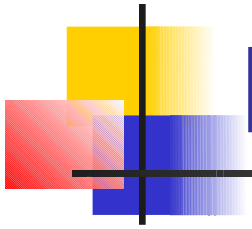
Formatting of string

- Separate format field and data
- E.g.,
  - >>> S = ‘spam and egg’
  - >>> names = [‘H. Cho’, ‘Bora Cho’]
  - >>> for name in names:

        print ‘Hi, %s’ % name

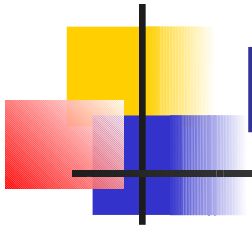
    Hi, H. Cho

    Hi, Bora Cho



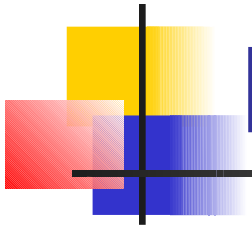
Methods in string

- upper()
- lower()
- capitalize()
- count(s)
- find(s)
- rfind(s)
- index(s)



Methods in string (cont.)

- strip(), lstrip(), rstrip()
- replace(a, b)
- expandtabs()
- split()
- join()
- center(), ljust(), rjust()



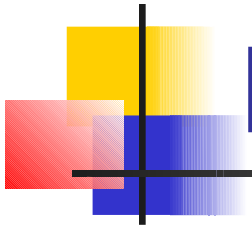
Formatting of string

- Separate format field and data
- E.g.,
  - >>> S = ‘spam and egg’
  - >>> name = [‘H. Cho’, ‘Bora Cho’]
  - >>> for name in names:

        print ‘Hi, %s’ % name

    Hi, H. Cho

    Hi, Bora Cho



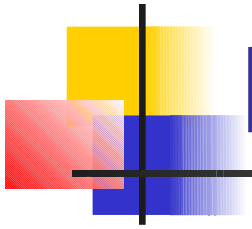
List

- Operations
- Nested lists
- Methods
- List comprehension
- Range
- Examples

Operations in List

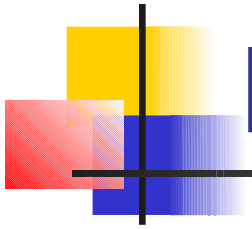
- Indexing e.g., L[i]
- Slicing e.g., L[1:5]
- Concatenation e.g., L + L
- Repetition e.g., L * 5
- Membership test e.g., 'a' in L
- Length e.g., len(L)

➔ IR ML CS



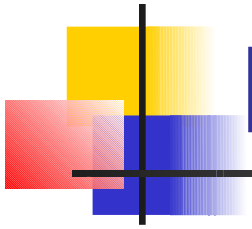
Nested List

- List in a list
- E.g.,
  - >>> s = [1,2,3]
  - >>> t = [‘begin’, s, ‘end’]
  - >>> t
  - [‘begin’, [1, 2, 3], ‘end’]
  - >>> t[1][1]
  - 2



Methods in List

- append
- insert
- index
- count
- sort
- reverse
- remove
- pop
- extend



List Comprehension

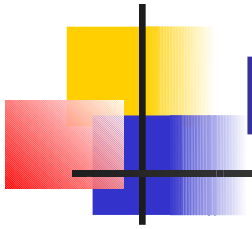
- Function to create list

- E.g.,
  - >>> L = [k * k for k in range(10)]
  - >>> L

    [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
  - >>> L = []
  - >>> for k in range(10):

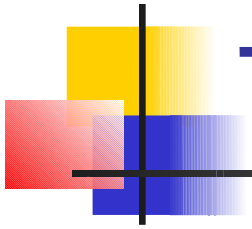
        L.append(k*k)
  - >>> L

    [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]



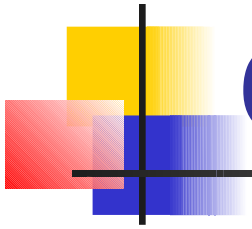
List Comprehension (cont.)

- E.g.,
  - >>> [(i, j, i*j) for i in range (2,100,2)
    for j in range(3,100,3)
    if (i + j) % 7 == 0]



Tuple

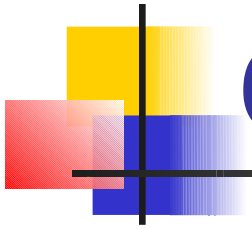
- Operations in Tuple
- List vs. Tuple



Operations in Tuple

- How to make a tuple?
  - E.g.,

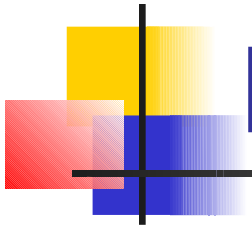
```
>>> t = ()
>>> t = (1, 2, 3)
>>> t = 1, 2, 3
>>> t = (1, )
>>> t = 1,
```



Operations in Tuple

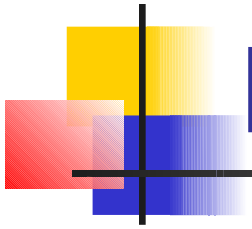
- Indexing e.g., T[i]
- Slicing e.g., T[1:5]
- Concatenation e.g., T + T
- Repetition e.g., T * 5
- Membership test e.g., ‘a’ in T
- Length e.g., len(T)

➔ IR ML CS



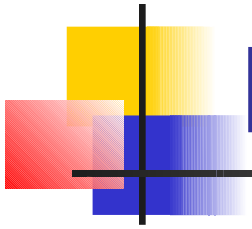
List vs. Tuple

- What are common characteristic?
  - Both store arbitrary data objects
  - Both are of sequence data type
- What are differences?
  - Tuple doesn’t allow modification
  - Tuple doesn’t have methods
  - Tuple supports format strings
  - Tuple supports variable length parameter in function call.



Dictionary

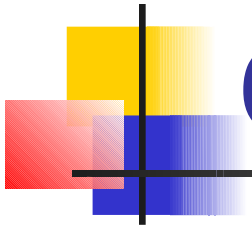
- Operations in Dictionary
- Methods in Dictionary
- Symbol table



Dictionary

- What is dictionary?
  - Refer value through key
- E.g.,

```
>>> member = {‘basketball’:5,
  ‘soccer’:11, ‘baseball’:9}
>>> member[‘baseball’]
9
```



Operations in Dictionary

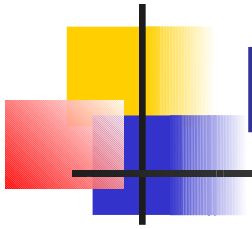
- >>> member[‘volleyball’] = 7
- >>> member[‘volleyball’] = 6
- >>> member

{‘soccer’:11, ‘volleyball’:6, ‘basketball’:5,
‘baseball’:9}

- >>> len(member)

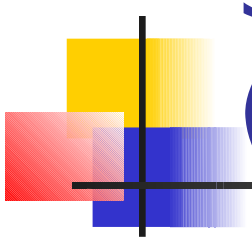
4

- >>> del member[‘basketball’]



Methods in Dictionary

- keys()
- values()
- items()
- has_key(key)
- clear()
- copy()
- get(key[,x])
- setdefault(key[,x])
- update(D)
- popitem()



Symbol table (represented in Dictionary)

- What is symbol table?
  - A table that stores a value corresponding to a symbol
  - Global / local symbol table
    - Globals()
    - Locals()
  - Symbol table of objects
    - __dict__
    - Example of dictionary
    - Example of function