WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

Principles of Software Programming: Basic Data Structures

Svitlana Vakulenko, MSc.

WS 2017

Acknowledgement: slides adopted from Dr. Jürgen Umbrich GZP SS 2014 lecture slides

This Episode

- 13:00-15:45
- Data structures & operations:
  - list: Stack & Queue
  - set
  - tuple
  - dictionary

Data Structure: List

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list

Data Structure: List


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list

List Slicing


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']

shopping_list[1]
shopping_list[-1]
shopping_list[0:-1]
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list
http://pythoncentral.io/how-to-slice-listsarrays-and-tuples-in-python/

List Functions


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']

len(shopping_list)

'Milk' in shopping_list
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list

https://www.programiz.com/python-programming/list

List Methods


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']

shopping_list.append("Chocolate")
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list

https://www.programiz.com/python-programming/list

List Methods


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']

shopping_list.append("Chocolate")
shopping_list.insert(0, "Chocolate")
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list

https://www.programiz.com/python-programming/list

List Methods


```
shopping_list = ['Milk', 'Apples',
'Eggs', 'Toilet rolls', 'Bananas',
'Bread']

shopping_list.append("Chocolate")
shopping_list.insert(0, "Chocolate")

shopping_list.remove("Chocolate")

shopping_list.sort()
print(shopping_list)
```

http://www.bbc.co.uk/scotland/education/wwww/buildings/standard/shopping/?item=list
https://wiki.python.org/moin/HowTo/Sorting

Types of Lists

https://www.npmjs.org/package/stac

http://www.skyrac.co.uk/please-form-an-orderly-queue/

Stack (LIFO)

```
list.append("War and Peace")

list.pop()
```

http://www.womansday.com/life/saving-money/cheap-books-online#slide-3

Queue (FIFO)


```
list.append("Mazda3")

list.pop(0)
```

http://www.theepochtimes.com/n2/world/french-strikes-force-cut-in-airline-flights-44413.html

Loop

```
>>> authors = ['William Shakespeare', 'Jane Austen', 'J.K. Rowling']
```

```
>>> i = 0
>>> while i < len(authors):
...     print authors[i]
...     i += 1
William Shakespeare
Jane Austen
J.K. Rowling
```

```
for x in shopping_list:
    print ("I need " + x)
```

http://www.icrar.org/__data/assets/pdf_file/0008/1436615/challenge09b-notes3.pdf

Ex.1: Shopping Basket

http://www.kegkits.com/kegerator2.htm

http://koshersamurai.wordpress.com/2012/10/17/unhealthy-interests-part-7-grocery-shopping/

http://www.wheeledbaskets.co.uk/supermarket_basket.aspx?ref=41

String

```
s = 'FizzBuzz'

s[0]
s[4:]

len(s)

for x in s:
    print x
```

http://codingbat.com/doc/python-strings.html

https://wiki.python.org/moin/ForLoop

String methods

```
>>> s = 'Hello, world'
>>> s = s.replace('world', 'Sveta')

>>> s.split()
```

http://en.wikibooks.org/wiki/Python_Programming/Strings
http://www.pythoncentral.io/cutting-and-slicing-strings-in-python/

String methods

```
>>> s = 'Hello, world'
>>> s = s.replace('world', 'Sveta')

>>> s.split()
['Hello,', 'Sveta']
```

http://en.wikibooks.org/wiki/Python_Programming/Strings
http://www.pythoncentral.io/cutting-and-slicing-strings-in-python/

String methods

```
>>> s = 'Hello, world'
>>> s = s.replace('world', 'Sveta')

>>> s.split()
['Hello,', 'Sveta']
```

```
1  >>> tim = '16:30:10'
2  >>> hrs, mins, secs = tim.split(':')
3  >>> hrs
4  '16'
5  >>> mins
6  '30'
7  >>> secs
8  '10'
```

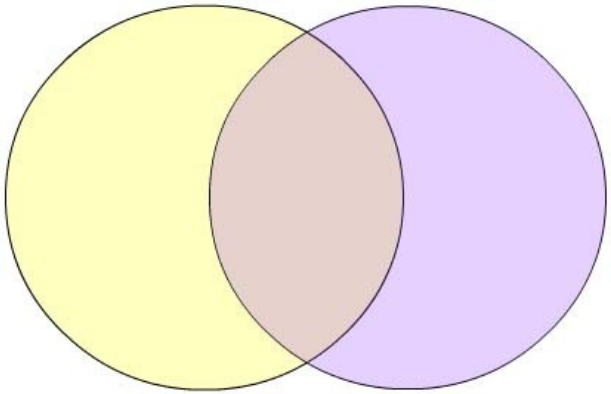
http://en.wikibooks.org/wiki/Python_Programming/Strings
http://www.pythoncentral.io/cutting-and-slicing-strings-in-python/

Set

```
numbers1 = set([1,1,2,3,4])
numbers2 = {1, 3}

if numbers2.issubset(numbers1):
    print("Is a subset")

if numbers1.issuperset(numbers2):
    print("Is a superset")
```



```
set1 = set(["cat", "dog"])     # Initialize set from a list          #set1 is now {'cat', 'dog'}
set2 = set(["dog", "mouse"])                                        #set2 is now {'mouse', 'dog'}
set3 = set1 & set2             # Intersection                       #set3 is now {'dog'}
set4 = set1 | set2             # Union                              #set4 is now {'dog', 'cat', 'mouse'}
set5 = set1 - set3             # Set difference                     #set5 is now {'cat'}
```

http://en.wikibooks.org/wiki/Python_Programming/Sets
http://www.dotnetperls.com/set-python
http://learningtogive.org/lessons/unit391/lesson2.html
https://github.com/thehackerwithin/PyTrieste/wiki/Python2-ListsDictsSetsTuples
https://docs.python.org/3/tutorial/datastructures.html#sets

Ex.2: Unique chars

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

FizzBuzz

Data Structure: Tuple

```
>>> b = ("Bob", 19, "CS")
>>> (name, age, studies) = b     # tuple unpacking
>>> name
'Bob'
>>> age
19
>>> studies
'CS'
```

julia_string = “Julia Roberts”

x, y = julia_string.split()

http://openbookproject.net/thinkcs/python/english3e/tuples.html
http://www.pythoncentral.io/cutting-and-slicing-strings-in-python/

Data Structure Composability

```
julia_more_info = ( ("Julia", "Roberts"), (8, "October", 1967),
                    "Actress", ("Atlanta", "Georgia"),
                    [ ("Duplicity", 2009),
                      ("Notting Hill", 1999),
                      ("Pretty Woman", 1990),
                      ("Erin Brockovich", 2000),
                      ("Eat Pray Love", 2010),
                      ("Mona Lisa Smile", 2003),
                      ("Oceans Twelve", 2004) ])
```

```
students = [
    ("John", ["CompSci", "Physics"]),
    ("Vusi", ["Maths", "CompSci", "Stats"]),
    ("Jess", ["CompSci", "Accounting", "Economics", "Management"]),
    ("Sarah", ["InfSys", "Accounting", "Economics", "CommLaw"]),
    ("Zuki", ["Sociology", "Economics", "Law", "Stats", "Music"])]
```

http://openbookproject.net/thinkcs/python/english3e/tuples.html

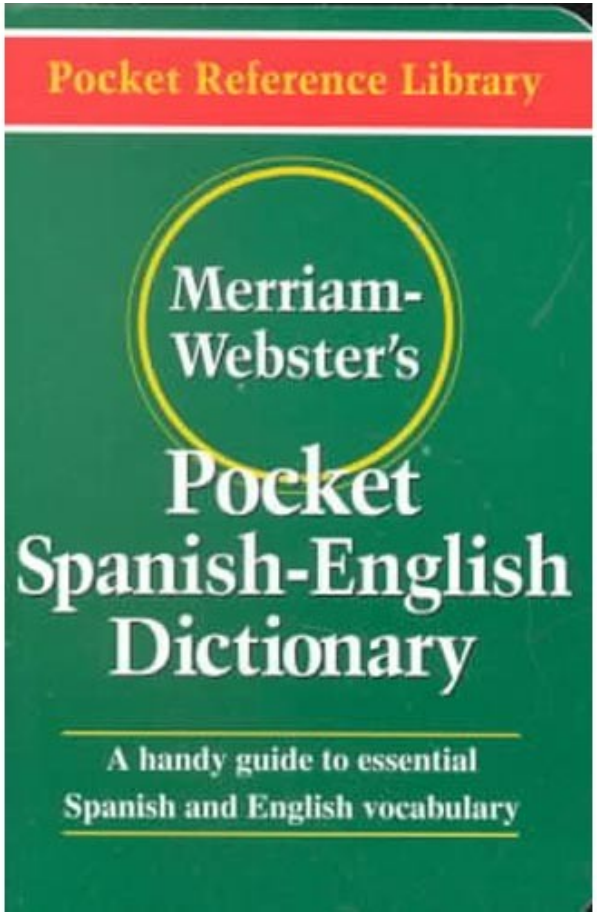
Loop

```
a = [3,4,5,6]
for i, val in enumerate(a):
  print i, val
```

```
0 3
1 4
2 5
3 6
```

https://stackoverflow.com/questions/126524/iterate-a-list-with-indexes-in-python

Data Structure: Dictionary




```
spanish = dict()
spanish['hello'] = 'hola'
spanish['yes'] = 'si'
spanish['one'] = 'uno'
spanish['two'] = 'dos'
spanish['three'] = 'tres'
spanish['red'] = 'rojo'
spanish['black'] = 'negro'
spanish['green'] = 'verde'
spanish['blue'] = 'azul'

print(spanish['two'])
print(spanish['red'])
```

http://www.walmart.com/ip/1784417

http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/dictionaries.html

Dictionary operations

- **Create**

  basket = { 'oranges': 12, 'pears': 5, 'apples': 4 }
- **Read**

  basket['apples']

  basket.keys()

  basket.values()
- **Update**

  basket['bananas'] = 5
- **Delete**

  basket.pop('pears')

  del basket['apples']

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Dictionaries

http://zetcode.com/lang/python/dictionaries/

Ex.3: Inventory

Part No.	Description	Quantity	Unit Price
00010-100	Monitor	1	$350.00
00010-200	Desk lamps	3	$55.00
00025-275	Phone	5	$85.00

$70

$40

$55

https://www.google.com/shopping

http://help.adobe.com/en_US/livecycle/9.0/designerHelp/index.htm?content=000179.html

http://www.worldstores.co.uk/c/Desk_Lamps.htm

Quiz

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

kahoot.it