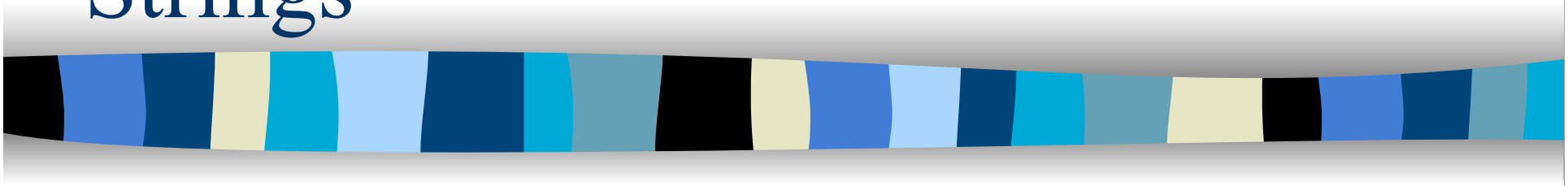
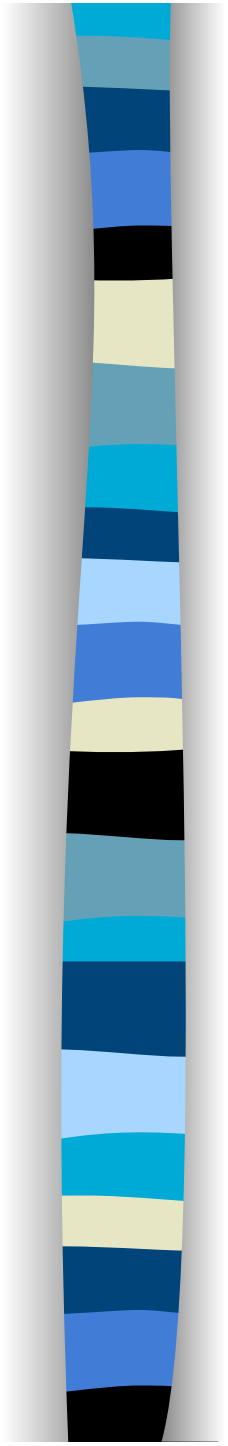
IS12 - Introduction to Programming Lecture 7: Processing Lists and Strings



Peter Brusilovsky

http://www2.sis.pitt.edu/~peterb/0012-161/

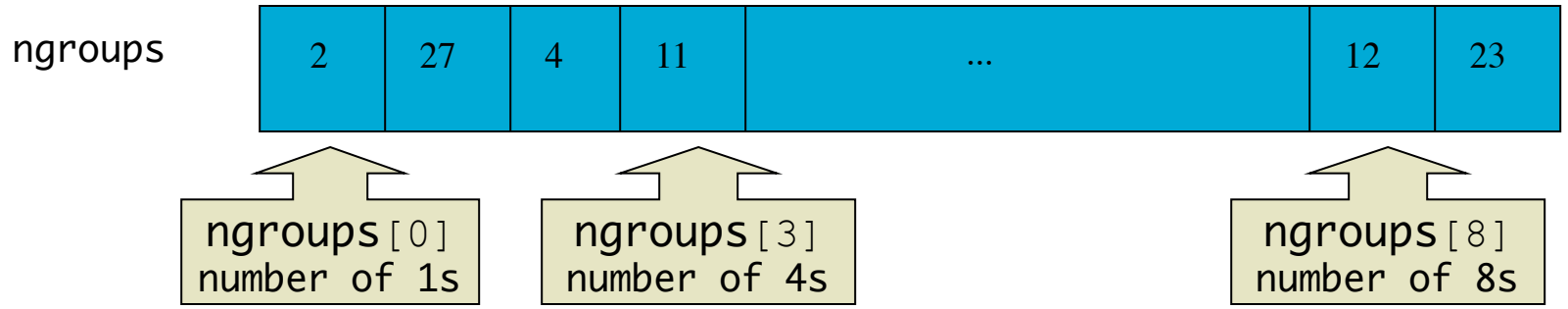


Outline

- More advances use of lists
  - Example: List as a set of counters
- Linear search in a list
  - Example: Team Search
- List slices
- Summary of list processing tools
  - Example: List of search Results
- Strings in Python
- Working with strings
  - Example: Palindrome

Using Lists for Counting

- Idea: Use a list as a set of counting variables
- Each element will count separate event
  - A group of people coming to a party

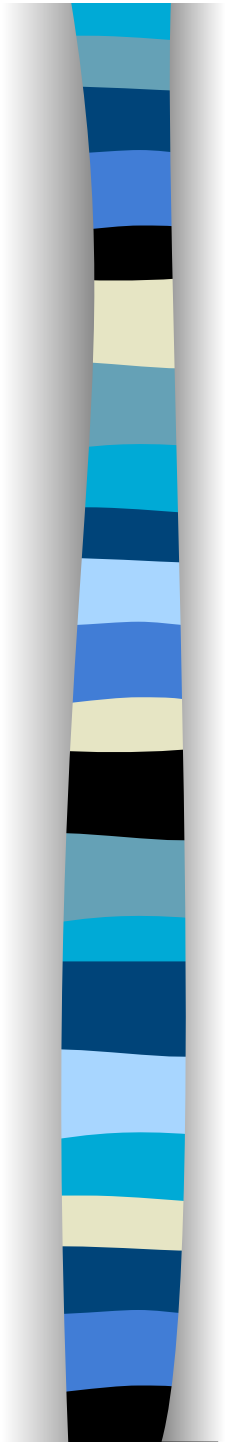


List groupsis used for counting digits (numeric characters)

Example: Group Counting

```
def main():
    print("Enter number of people in each group, 0 to stop")
    print("Enter 10 if a party has 10 or more people")
    nparties = [0] * 10
#   pre-reading first number
    npeople = int(input("First group: "))

# filling the list
    while npeople > 0:
          if(npeople < 10):
                  nparties[npeople - 1] += 1
          else:
                  nparties[9] += 1
          npeople = int(input("Next group: "))
#reporting
    for i in range(9):
          print("Number of {0:0}s is {1:0}".format(i + 1, nparties[i]))
    print("Number of large groups is {0:0}".format(nparties[9]))
```



Searching in a List

- We learned several list processing patterns
  - sum, count, max
- Search is one of the most important patterns of working with a list
- Is there a specific element in the list?
- Where is a specific element in the list?

Example: Search in a List (1)

```
# Is there value in the list?
def is_team (tlist, value):
   for i in range (len (tlist)):
       if tlist [i] == value:
              return True
   return False

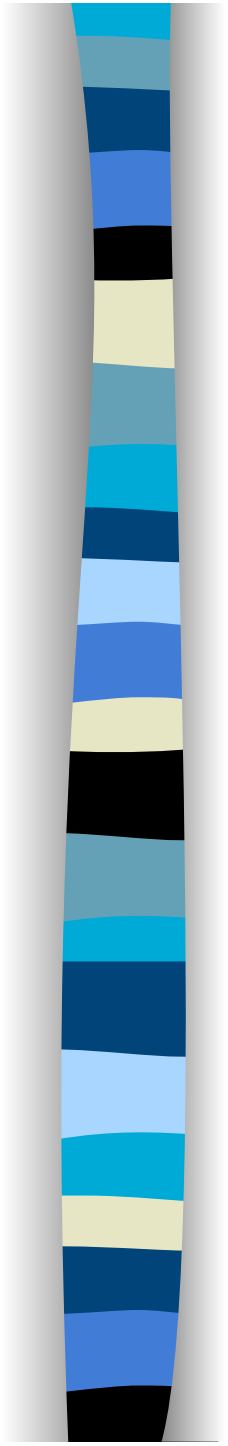
# In which position the value is in the list?

def which_team (tlist, value):
   for i in range (len (tlist)):
       if tlist [i] == value:
              return i
   return -1
```

Example: Search in a List (2)

```
# This function will create a list of teams
def make_list ():
    team_number = int (input ("How many teams in a hackathlon?"))
    teamlist = [0] * team_number
    for i in range(team_number):
          print("How many members in team", i)
          teamlist[i] = int (input ("? "))
    return teamlist

# Finding a value in a list - is there a team with specific n members?
def main ():
    teams = make_list ()
    number = int (input ("Which size of team to search for?"))
    if is_team (teams, number):
          tnum = which_team(teams, number)
          print ("Team number {0:0} has {1:0} members".format(tnum,
    number))
    else:
          print ("None of the teams has {0:0} members".format(number))
```



Exercise: Team search

- Modify team search program adding two new features
- No need to use is_team predicate
- User creates the list once and can search for teams many times

Standard tools for search

- Standard predicate <value> in <list>
  - True if there is a value in the list

    "a" in ["a", "b", "c"]
  - False if there is no value like it in the list

    "x" in ["a", "b", "c"]
- Method index()
  - Returns position if element found

    mylist.index("a")
  - Exception if there is no value like that!

Example: Python native search

```
# This function will create a list of teams
def make_list ():
    team_number = int (input ("How many teams in a hackathlon?"))
    teamlist = [0] * team_number
    for i in range(team_number):
          print("How many members in team", i)
          teamlist[i] = int (input ("? "))
    return teamlist

# Finding a value in a list - is there a team with specific number of
    members?
def main ():
    teams = make_list ()
    number = int (input ("Which size of team to search for?"))
    if number in teams:
          tnum = teams.index(number)
          print ("Team number {0:0} has {1:0} members".format(tnum, number))
    else:
          print ("None of the teams has {0:0} members".format(number))
```

List Slices

- Python allows to obtain list slices – continuous list segments
- Slices are lists!

```
>>> lista = [2, 4, 6, 8, 10, 12, 14, 16]
>>> listb = lista[2:5]
>>> print(alilista)
[6, 8, 10]
```

- Slice can start from the beginning of the list or run up to its end

```
>>> lista[:5]
[2, 4, 6, 8, 10]
>>> lista[5:]
[12, 14, 16]
```

- Negative values indicate positions from the end of the lisr

```
>>> lista[:-1]
[2, 4, 6, 8, 10, 12, 14]
```

List Slices

- Python allows to obtain list slices – continuous list segments
- Slices are lists!

```
>>> lista = [2, 4, 6, 8, 10, 12, 14, 16]
>>> listb = lista[2:5]
>>> print(alilista)
[6, 8, 10]
```

- Slice can start from the beginning of the list or run up to its end

```
>>> lista[:5]
[2, 4, 6, 8, 10]
>>> lista[5:]
[12, 14, 16]
```

- Negative values indicate positions from the end of the lisr

```
>>> lista[:-1]
[2, 4, 6, 8, 10, 12, 14]
```

Working with Lists in Python

- Accessing and modifying elements
- Adding, inserting, removing elements
- Concatenating and multiplying lists
- Finding elements and extracting slices
- Some combined methods: remove

```
>>> print(lista)
[2, 4, 6, 22, -5, 10, 12, 14, 16, -3]
>>> lista.remove(22)
>>> print(lista)
[2, 4, 6, -5, 10, 12, 14, 16, -3]
```

Example: List of Search Results

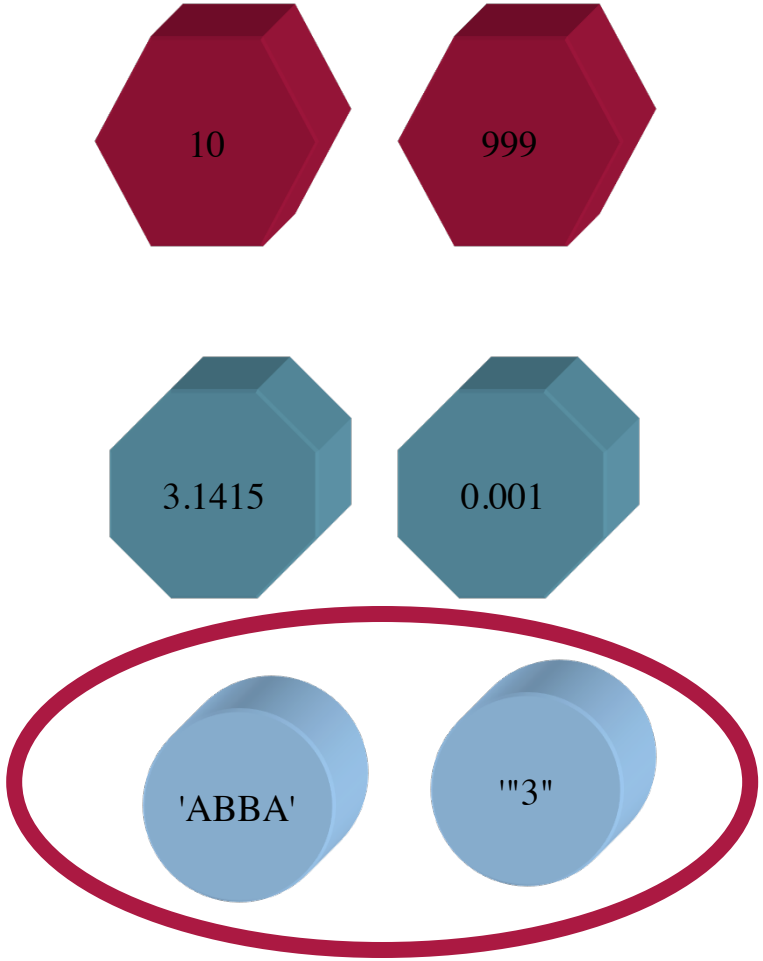
```
# Returns a list of positions of the value in the list?

def which_teams (tlist, value):
    resultlist = []
    for i in range (len (tlist)):
         if tlist [i] == value:
                  result.append(i)
    return resultlist

# Finding a value in a list - is there a team with specific number of
    members?
def main ():
    teams = make_list ()
    number = int (input ("Which size of team to search for?"))
    res_list = which_teams (teams, number)
    if len res_list > 0:
         print ("The following teams have", number, "members", res_list)
    else:
         print ("None of the teams has {0:0} members".format(number))
```

Strings as a standard data type

- Integer data
  - 1, 10, 999, 1000
- Floating point data
  - 3.1415, 0.001, 2.0
- Strings
  - "ABBA", 'x', "3"



Strings in Python

- Sequence of characters
- Standard Python type str
- String literals are delimited with " or '

```
st = 'apple'
st = "apple"
```

- Long strings could use tripple quotes

```
longstring= """This is a really
really
really long string"""
```

Strings in Python

- Sequence of characters
- Standard Python type str
- String literals are delimited with " or '

```
st = 'apple'
st = "apple"
```

- Long strings could use tripple quotes

```
longstring= """This is a really
really
really long string"""
```

Working with Strings in Python

- Strings are sequences, like lists
- Most Python tools for sequences work with strings
- String element access (read only!)

```
>>> mystring = "apple"
>>> print(mystring [3])
l
```

- String length

```
>>> print(len(mystring))
5
```

- String element access (read only!)

```
>>> mystring = "apple"
>>> print(mystring [3])
l
```

- String elements can't be changed! Try…

```
>>> mystring [0] = "A"
```

Strings: slicing and search

- String slicing:

```
>>> mystring = "Pineapple"
>>> print(mystring [0:2])
Pi
>>> print(mystring [4:])
apple
>>> print(mystring [:4])
Pine
>>> print(mystring [:-2])
Pineapp
```

- Strings search

```
>>> mystring = "apple"
>>> "a" in mystring
True
>>> mystring.index("l")
3
```

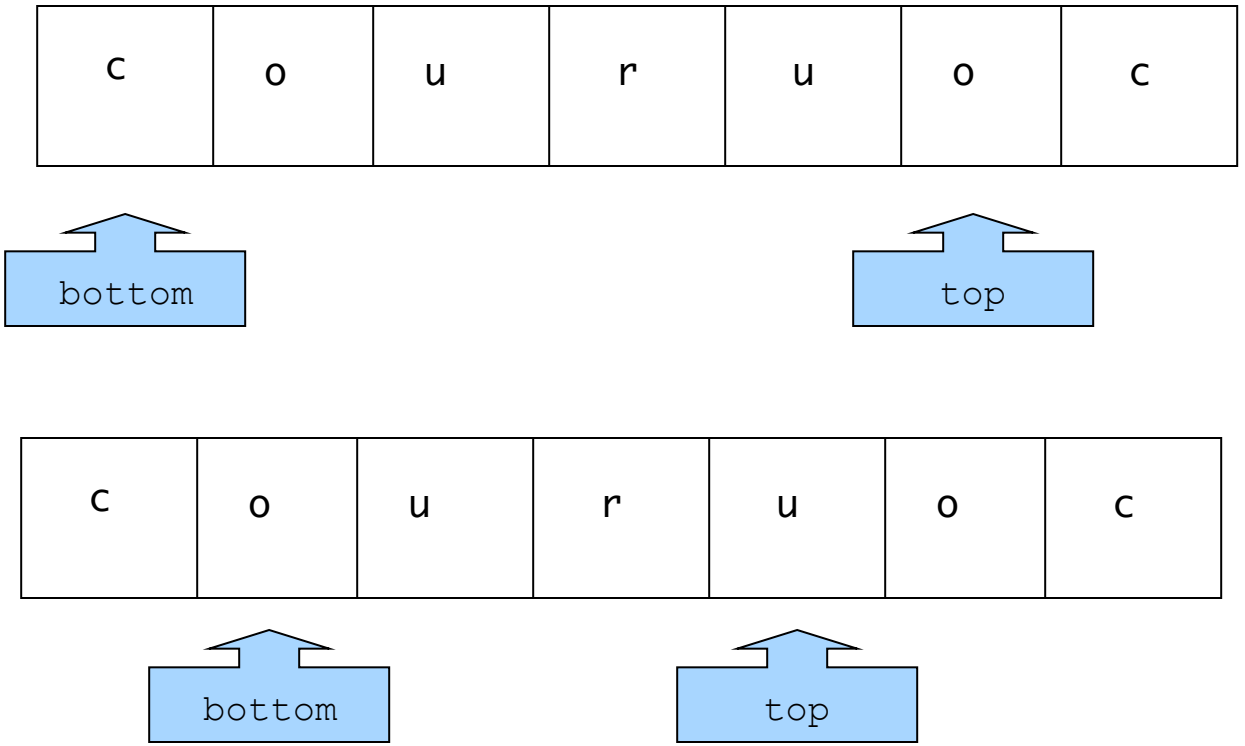
Example: Palindrome (main)

```
# Is the string a palindrome?
#  ABBA is a palindrome, BEATLES is not a palindrome

def main ():
   test_string = input ("Enter a string to check! ")

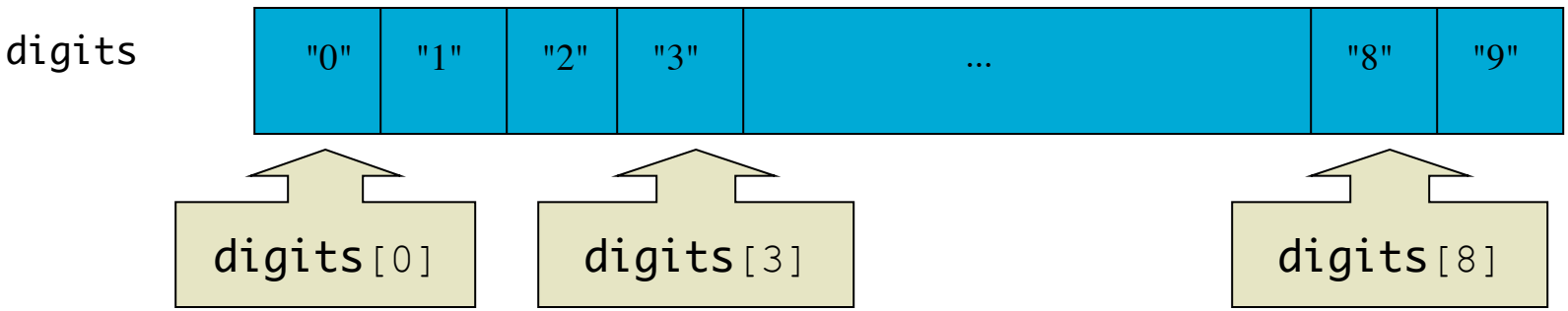
   if is_palindrome(test_string):
       print (test_string, "is a palindrome")
   else:
       print (test_string, "is not a palindrome")
```

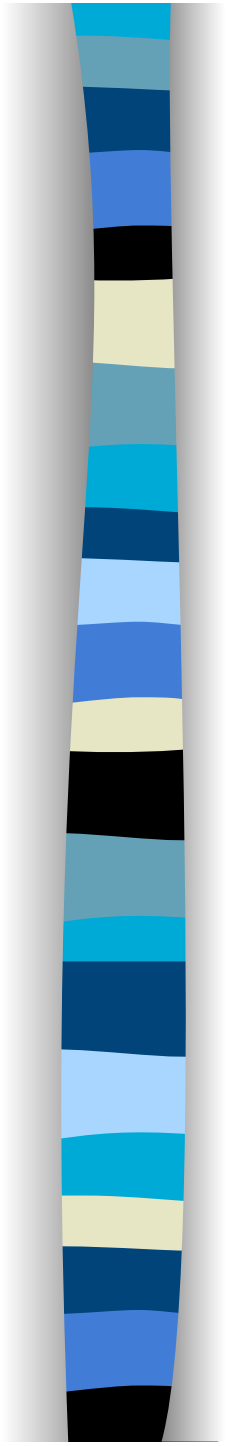
Example : Palindrome



Tip: Symbol to Digit Conversion

- Idea: Create a decoding list
- Each element with index K stores the corresponding symbol (0 – 9)
- If a symbol is in the list, it is a digit
- Then its value is its index in the list!





Before Next Lecture:

- Do reading assignment
  - Severance, Chapter 6: Strings
- Run and explore Classroom Examples
- Exercises:
  - Simplify party counting example
  - Finish classroom exercises
- Check yourself by working with MG system
- Prepare for exam
- Homework programming assignment due in 2 weeks