Regular Expressions

April 4, 2013

Today

- Regular Expressions
- Using regular expressions in Python
- New data structure: *Iterator*

Today

- Regular Expressions – a way to **match strings**
- Using regular expressions in Python
- New data structure: *Iterator*

Play Time!

Play Time!

Go to regexpal.com and copy this text into the lower box:

```
Ahab was born in 1802.  Starbuck, in '98 --- 1998, that is.  And (as
everyone knows), Ishmael was born in 2036 but died in 1879 (making him
[1879-2036=]-57 years old at the time of his death, or 48, depending on
how you count).  And the white whale was immortal.

Am I telling this story right?  Anyhow, each of them had access to a time
machine, a harpoon, and twenty sheep for barter.  Including the whale.  As
was common in his era, Starbuck had his own name tattooed on the back of
his arm: "*$", it read, "Star-buck", a kind of pun.

Our story begins in interstellar space...
```

Try entering different things in the upper box. Any surprises?

Let's get serious now...

- In `ACT3-2.py`, Run `nameGame` with your name

```
def nameGame(name):
 name = name.lower()
 c = name[0]
 if (c == 'a') or (c == 'e') or (c == 'i') or (c == 'o') or (c == 'u'):
   suffix = name
 else:
   suffix = name[1:len(name)]

 myStr = name + ' ' + name + ' bo b' + suffix + '\n'
 myStr = myStr + 'banana fana fo f' + suffix + '\n'
 myStr = myStr + 'me mi mo m' + suffix + '\n'
 myStr = myStr + name
 return myStr
```

Let's get serious now...

```
jadrian jadrian bo-badrian
banana-fana fo-fadrian
me my mo madrian
jadrian!
```

Do Task 1

Regular Expressions

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets		

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter		

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace		

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit		

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	

Regular Expressions

The qu1ck brown fox jumped over the lazy d0g.

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.		

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.		

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character		

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character	`.`	

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character	`.`	`T h e ' ' q u ...`

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character	`\s.+\s`	

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character	`\s.+\s`	`qu1ck brown fox ...`

Today

- Regular Expressions – a way to **match strings**
- Using regular expressions in Python
- New data structure: *Iterator*

Regular Expressions

`The qu1ck brown fox jumped over the lazy d0g.`

Special Syntax	Meaning	R.E	Result
[ ]	Match anything between brackets	`[qjd]u`	`qu ju`
\w	Match any letter	`o\w`	`ow ox ov`
\s	Match any whitespace	`e\s\w`	`'e q' 'e l'`
\d	Match any digit	`\d\w`	`1c 0g`
+	Match **one or more** of the previous things.	`\s\w+\s`	`The brown fox ...`
*	Match **zero or more** of the previous things.	`\w\d*\w`	`The qu1ck brown ...`
.	Match any character	`\s.+\s`	`qu1ck brown fox ...`

Do Task 2

Regular Expressions

Just the beginning… see the 'PythonRE' link for lots more:

Pattern	Description
^	Matches beginning of line.
$	Matches end of line.
.	Matches any single character except newline. Using m option allows it to match newline as well.
[...]	Matches any single character in brackets.
[^...]	Matches any single character not in brackets
re*	Matches 0 or more occurrences of preceding expression.
re+	Matches 1 or more occurrence of preceding expression.
re?	Matches 0 or 1 occurrence of preceding expression.
re{ n}	Matches exactly n number of occurrences of preceding expression.
re{ n,}	Matches n or more occurrences of preceding expression.
re{ n, m}	Matches at least n and at most m occurrences of preceding expression.
a| b	Matches either a or b.
(re)	Groups regular expressions and remembers matched text.
…	…

Today

- Regular Expressions – a way to **match strings**
- Using regular expressions in Python
- New data structure: *Iterator*

Data Structures

Lists

‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’

Data Structures

Lists

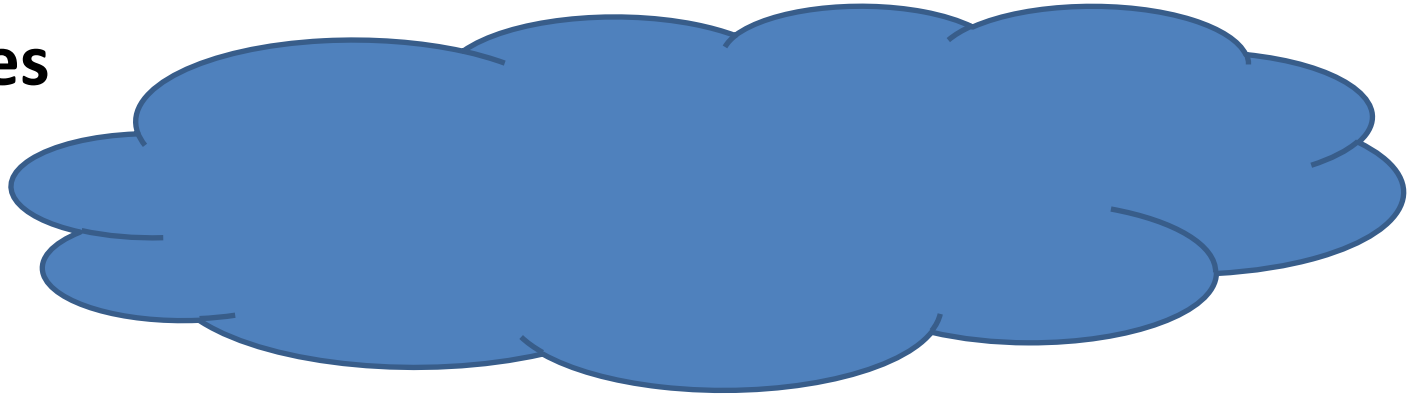
content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

Dictionaries



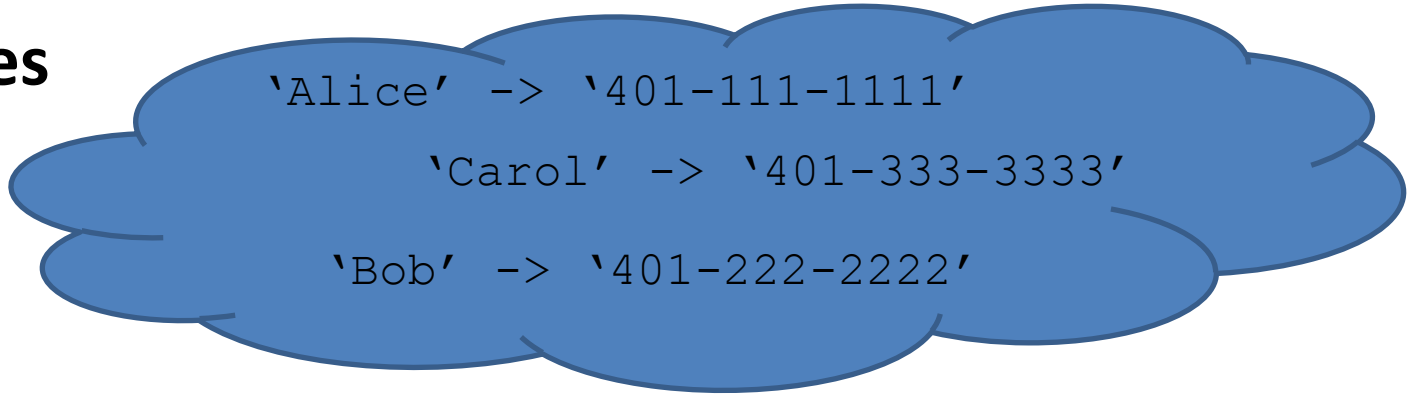
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

Dictionaries

keys & values



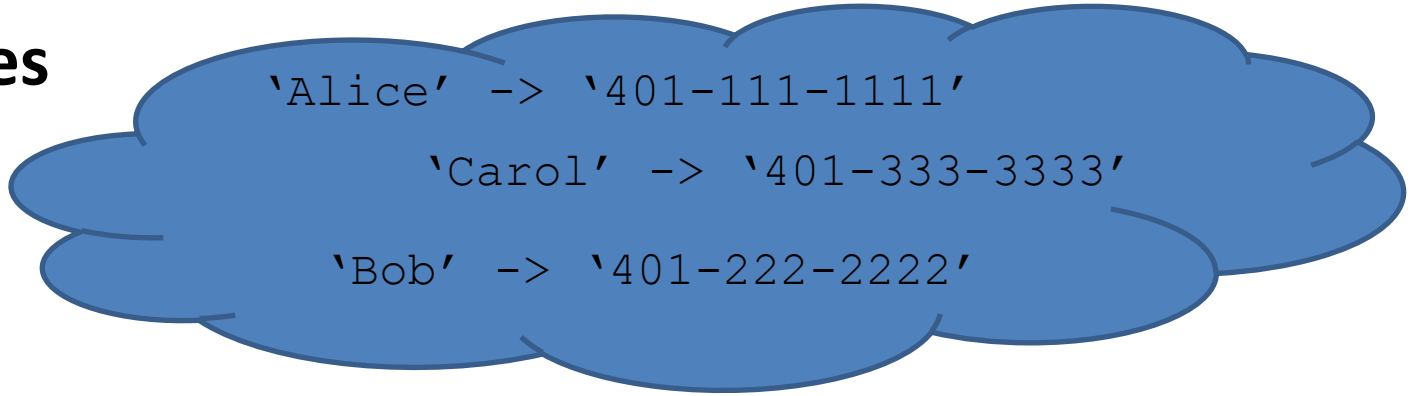
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

Dictionaries

keys & values



Iterators

Match Objects

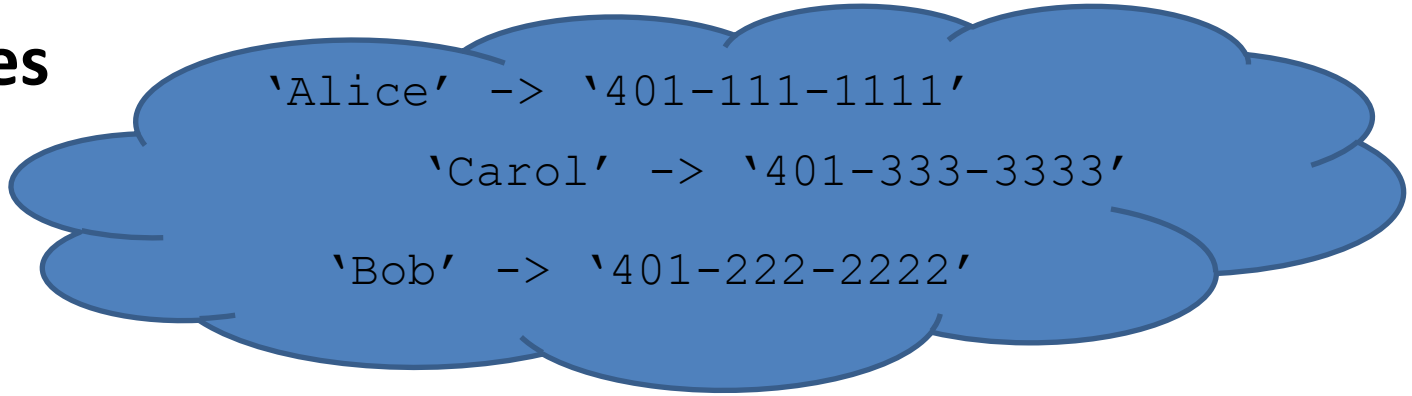
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

Dictionaries

keys & values



Iterators

Match Objects

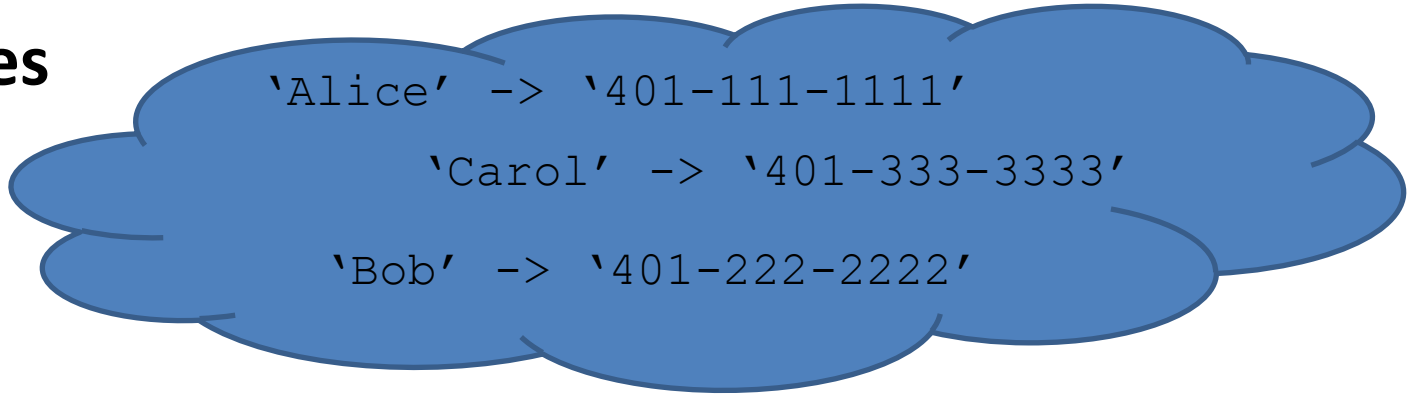
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

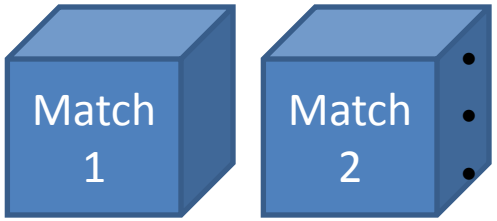
Dictionaries

keys & values



Iterators

Match Objects



- Matched String
- Matched String Start
- Matched String End

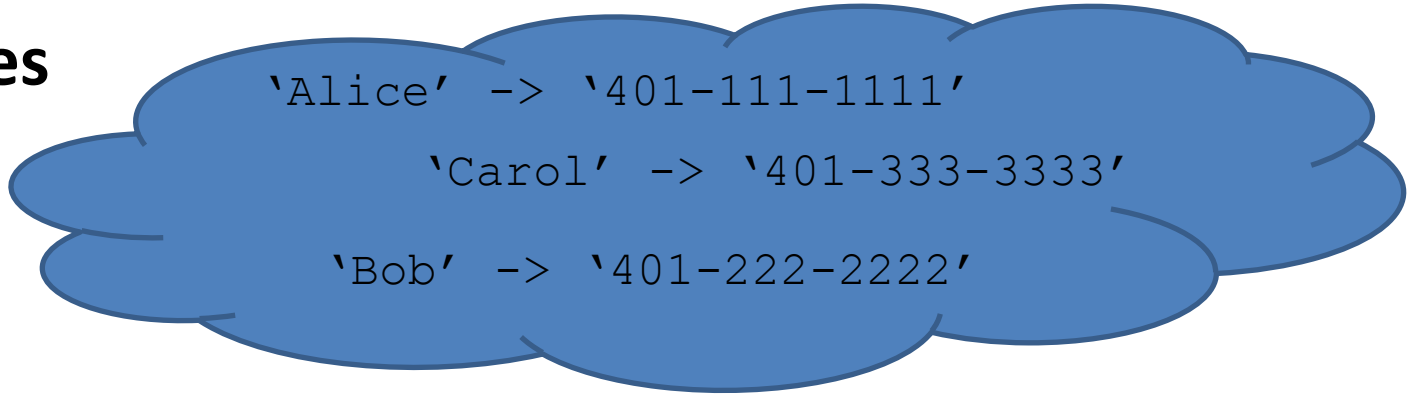
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

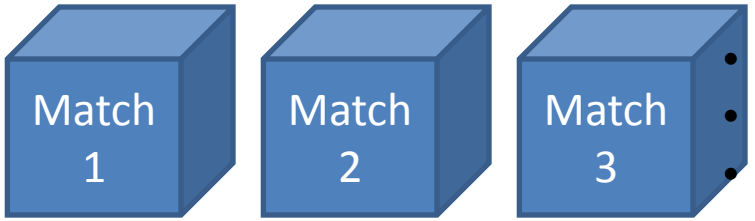
Dictionaries

keys & values



Iterators

Match Objects



- Matched String
- Matched String Start
- Matched String End

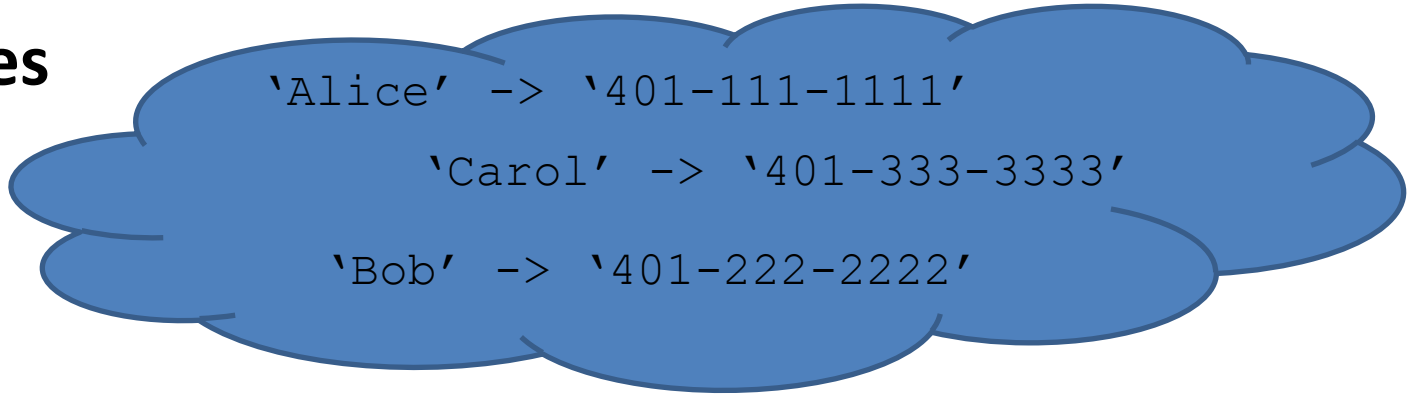
Data Structures

Lists

content	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’
indices	0	1	2	3	4	5	6	7	8	9

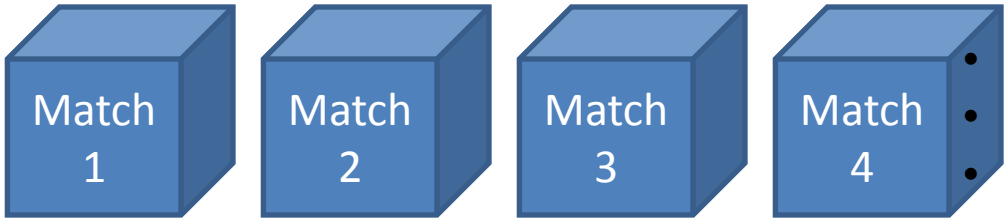
Dictionaries

keys & values



Iterators

Match Objects



- Matched String
- Matched String Start
- Matched String End

Iterators

The cat in the hat sat on a mat

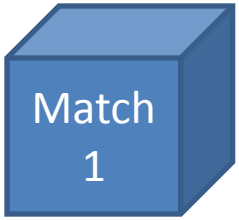
Regular Expression:'\wat'

Iterators

The cat in the hat sat on a mat

Regular Expression:'\wat'

Iterator

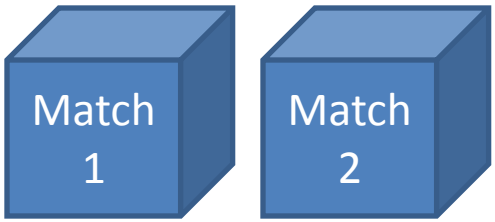


Iterators

The cat in the hat sat on a mat

`Regular Expression:'\wat'`

Iterator

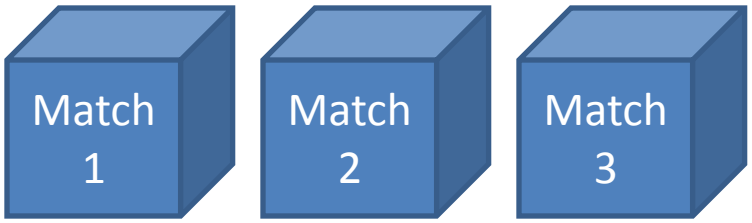


Iterators

The cat in the hat sat on a mat

Regular Expression:'\wat'

Iterator

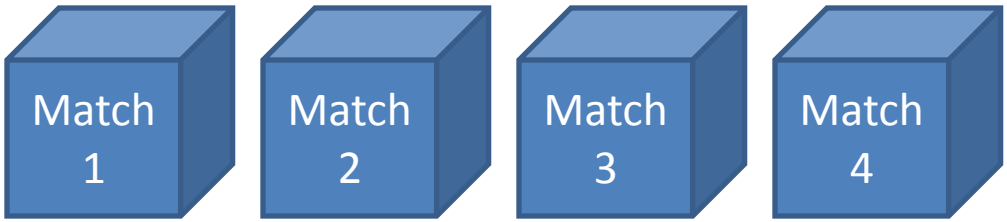


Iterators

The cat in the hat sat on a mat

Regular Expression:'\wat'

Iterator



Using Regular Expressions in Python

```
def printRegEx(regex,myStr):
    '''Prints all occurrences of the regular expression.'''

    myIter = re.finditer(regex,myStr) # Iterator!

    # This iterator contains MATCHES of the regex.
    # The following functions work on regex matches:
    #  group(0): returns the string that matches the regex
    #  start(0): the starting position of the string in myStr
    #  end(0): the ending position of the string in myStr

    # For loops work on iterators in addition to lists
    # Each match of the regex in myStr is stored in
    # a variable called 'occ'
    for occ in myIter:
        print 'matches',occ.group(0), \
            'at positions',occ.start(0),'-',occ.end(0)
    return
```