Summary for midterm 1

CSE101 2018 Spring

Commands and Expressions

String	a sequence of characters	‘Hello, world!’
Integer	whole numbers	5
Floating-point	real numbers	12.36
Boolean		True OR False

Arithmetic in Python

- Follows the same PEMDAS rules. Don’t forget to use( ) when it is necessary.
- Symbols:

+	add
-	subtract
*	multiplication
/	division for real number(will return a floating-point num)
//	division for integers(when we don’t need the remainder)
%	gives you the remainder of an integer division
**	exponentiation

- Think about it:

I have an integer 125.

I want to get rid of the last digit and change it to 12, how to do that?

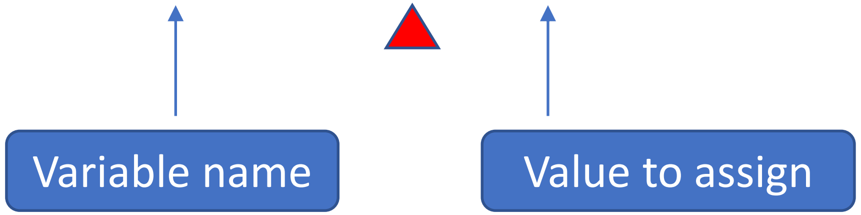
(use // 10)

I want to return the last digit 5. How to do that?

(use % 10)

Assignment Statements

- Number = 3



Variable name

Value to assign

You set a variable name, and you use = to assign the value to the variable.

If you want to change the value or update the variable's value, you can:

Number += 2 (5)

Number = Number + 2 (5)

Basically, they are the same thing.

Boolean & if-statement

- Boolean expressions:

==	is equal to	not **=**	not **is**
!=	is not equal to		
>	is greater than		
<	is less than		
>=	is greater than or equal to		
<=	is less than or equal to		

Boolean & if-statement

- If – statement

  if …   (True, go in to the if statement and execute code in it; False, do not go in to the if statement):

  code

- If-elif-statements

  if there are more than one condition, use **elif** to check if other conditions are true

  If any of these is true, then rest of **elif** blocks will not be evaluated.

- If-if-if..-statements

  multiple if statements, will check all if statements one-by-one and evaluate each if statement individually

String immutable

- * can also be used for string

  ‘Hello’ * 3 (‘HelloHelloHello’)

- len() function

  will count the number of character/item in a string/list

  (whitespace(s) will also be counted for strings)

- count( ) method [slides Unit 2 page 62]

  sentence.count(‘ ’)

- endswith & startswith method [slides Unit 2 page 62]

for loop

- Function: use for- loop whenever you want to visit all the elements in the list or string one-by-one.
- index:

  KEEP IN MIND! Index in string or list begins with 0
- Ability to trace the code [see it in unit 3 slide page21-28]

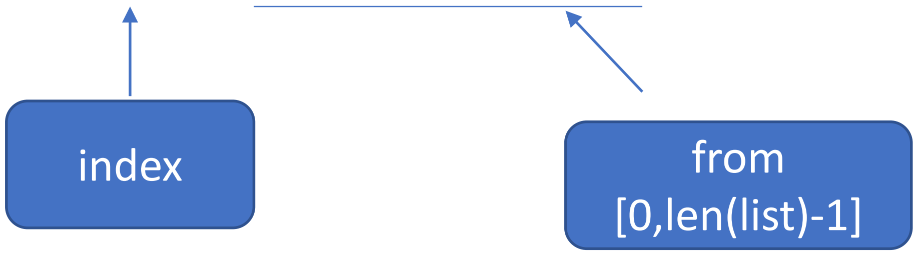
  use debugger to practice this ability
- Range

  range(3) #means 0,1,2

  range(1,4) #means 1,2,3
-   range(1,6,2) #means 1,3,5
- Read the unit 3 slides from page 38-50, if you are not familiar with above concepts.

for loop

- **for** i **in range**(len(list1)):



index

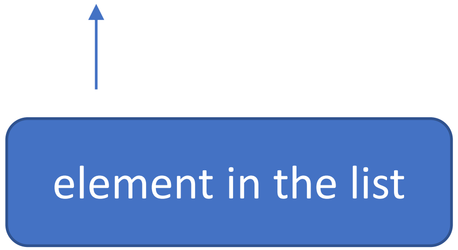
from [0,len(list)-1]

....

....

Use list[i] to represent every element in the list

- **for** i **in** list1:



element in the list

....

....

Use i to represent every element in the list

for loop

- Normally, the code in the for loop is to update the value of variables

so, you might want to write down the return statement outside the for loop.

Stepping through the worksheet

- for i in range(2*k, len(a), k):

  # means start from 2*k end at len(a)-1   and jump step size is k

Read unit3 slides page 56 →

List_name[ ]   string_name[ ]

- If you are extracting value from a list you need to use list_name[index]

list_name[index] helps you to extract the value of the certain index of the list.

we also need to return the value by using the right datatype.

Let's say x = 'string’

print(x[0]) will give you 's'

Let's say y = ['s','t','r','i','n','g’]

print(y[0]) will give you 's'

```
r=[‘stony’, ‘brook’, ‘university’]
print(r[1])
# will give you    ‘brook’
```

```
h = [ [1, 3, 7], [2, 6, 8, 4], [5, 2] ]
print (h[1])
# will give you [2,6,8,4]
```

List ------- a box to store objects

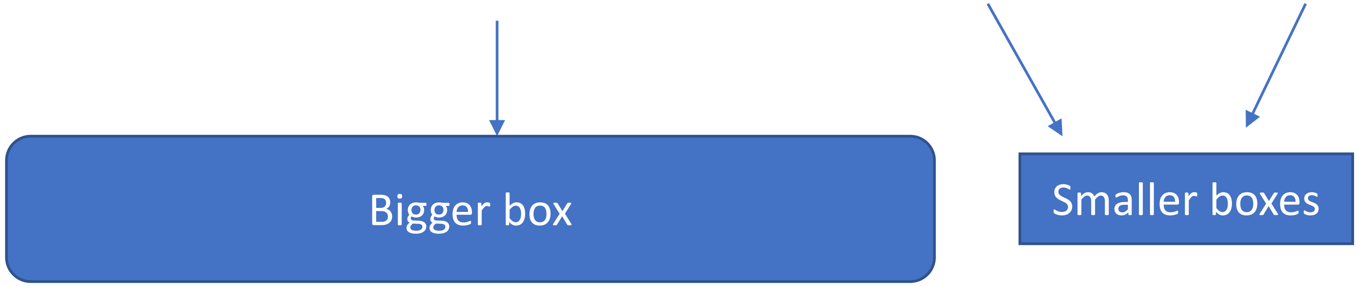
- List of Strings

```
list1 = [‘abc’, ‘def’, ‘ghijk’]
len(list1)   # return the value 3
```

- List of lists

Consider sublists as small boxes that packed in a bigger box(list)

```
list1 =[  [98, 99, 100], [67, 97, 88], [86, 76, 92] ]
```



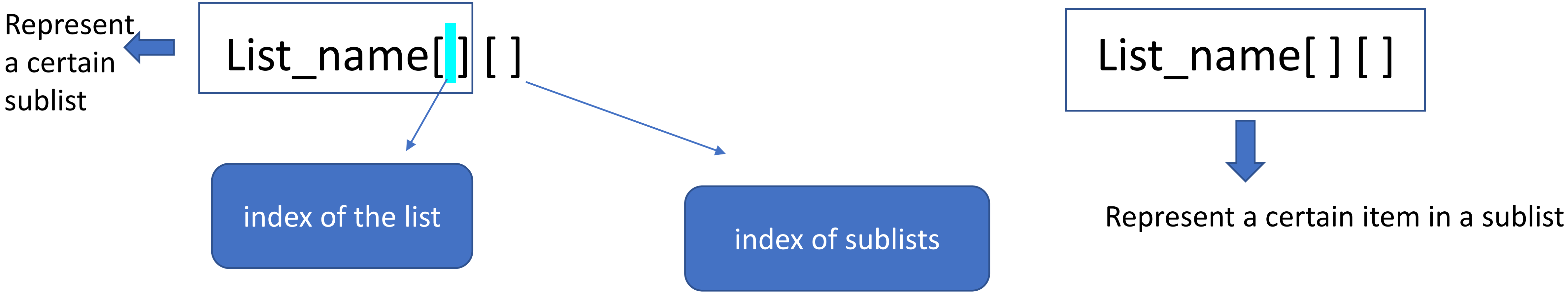
Bigger box

Smaller boxes

List_name[ ][ ]   (read unit3 slides page116→)

- List_name[ ] [ ] is the way that help you to extract value from a sublist.
- Usually, we will use nested for loop to solve those problem.

  (read Unit 3 slide page 116 → for more details)



Represent a certain sublist

List_name[ ] [ ]

index of the list

index of sublists

List_name[ ] [ ]

Represent a certain item in a sublist

How to create a list

```
def non_null(a):
        res = [ ]    → create an empty list for adding something into it later
```

Add something into a list

- Use ‘+=‘

```
res = [ ]
res += ‘stony’
# res will be [‘stony’]
```

- Use append method

```
res = [ ]
res.append(‘stony’)
# res will be [‘stony’]
```

Add something into a string

- Use ‘+=‘

```
res=““
res += “Hello”
# res will be “Hello”
```

While loop

- Live on condition expression

  Python evaluates the Boolean expression next to the keyword **while**

- if the expression is true, go into the while loop, and execute code in the loop.
- Index:

  index in while loop must be initialized before the loop.

  Also, you need to update the index by yourself, just by adding 1 to the index at the end of the loop body. Otherwise, it will be a infinite loop.

- In detail? Read unit4 slides page 15

Read unit4 slides for:

- Linear search
- Insertion sort
- Nested loops
- While loop
- Differences between for loop and while loop

For loop VS. while loop

- Function:

➢For loop : if you want to visit every items in the list or string one-by-one.
Normally, you know how many times you need to execute the loop

➢While loop: if you want to test a condition to decide to enter the loop or not.
Normally, you have no idea how many times you need to execute

For loop VS. while loop

- How to stop the loop?

for loop:

use index and range.

if index is out of the range, we will never enter the loop again

use item **in** list_name:

after visit every item in list/string, we will never enter the loop again

While loop:

use Boolean expression

if Boolean expression false, we will never enter the loop again

Index in for loop & while loop

- In for loop, index automatically add 1, after we execute for loop.

  It is invisible. You do not need to update the index by yourself

- In while loop, you need to initialize the index before the loop, and you need to update the index by yourself at the end of the loop body.

- Read loop code in slides. It helps a lot.

Ability to Trace Execution:

You need to have the ability of tracing execution of code.

If you are not familiar with it, read though the Trace Execution parts in slides, or use debugger in Pycham to help you to learn how to trace the code.

✓This ability is important. You need this ability to read nested for loop and nested if – statements with a clear logic.

Good luck ☺