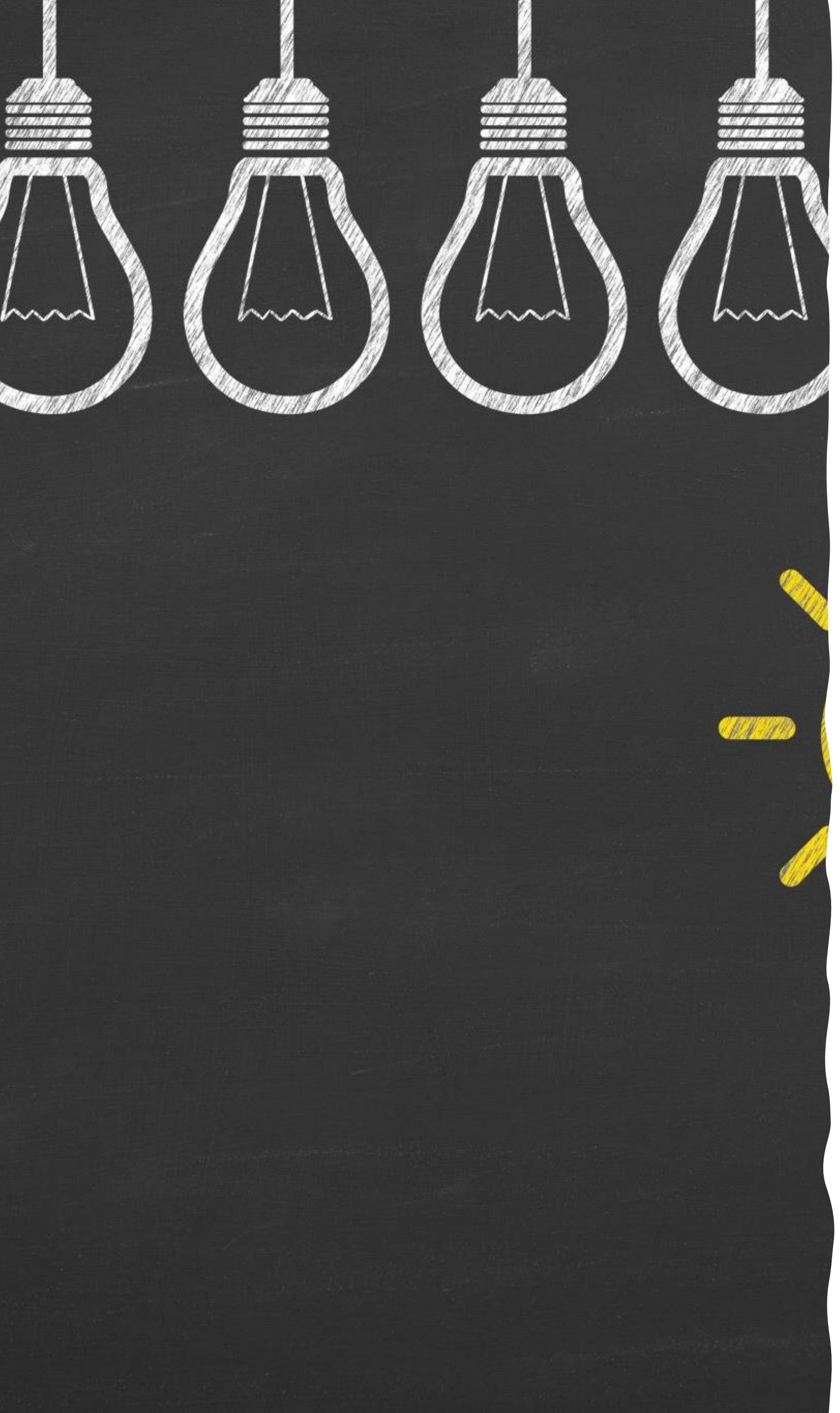

CMSC 105 Elementary Programming

Acknowledgement: These slides are adapted from slides provided with “Introduction to Programming Using Python, Liang (Pearson 2013)” and slides shared by Dr. Jory Denny

Math, Strings, and Objects

Math Module

- Python provides many useful mathematics methods in its Math module for performing common mathematical functions.

Examples of Math Module

```
max(2, 3, 4) # Returns a maximum number : in
this case 4
min(2, 3, 4) # Returns a minimum number : in
this case 2
round(3.51)  # Rounds to its nearest integer
round(3.4)   # Rounds to its nearest integer
abs(-3)      # Returns the absolute value
pow(2, 3)    # Same as 2 ** 3
```

The math Functions

Function	Description	Example
fabs(x)	Returns the absolute value of the argument.	fabs(-2) is 2
ceil(x)	Rounds x up to its nearest integer and returns this integer.	ceil(2.1) is 3 ceil(-2.1) is -2
floor(x)	Rounds x down to its nearest integer and returns this integer.	floor(2.1) is 2 floor(-2.1) is -3
exp(x)	Returns the exponential function of x (e^x).	exp(1) is 2.71828
log(x)	Returns the natural logarithm of x.	log(2.71828) is 1.0
log(x, base)	Returns the logarithm of x for the specified base.	log10(10, 10) is 1
sqrt(x)	Returns the square root of x.	sqrt(4.0) is 2
sin(x)	Returns the sine of x. x represents an angle in radians.	sin(3.14159 / 2) is 1 sin(3.14159) is 0
asin(x)	Returns the angle in radians for the inverse of sine.	asin(1.0) is 1.57 asin(0.5) is 0.523599
cos(x)	Returns the cosine of x. x represents an angle in radians.	cos(3.14159 / 2) is 0 cos(3.14159) is -1
acos(x)	Returns the angle in radians for the inverse of cosine.	acos(1.0) is 0 acos(0.5) is 1.0472
tan(x)	Returns the tangent of x. x represents an angle in radians.	tan(3.14159 / 4) is 1 tan(0.0) is 0
fmod(x, y)	Returns the remainder of x/y as double.	fmod(2.4, 1.3) is 1.1
degrees(x)	Converts angle x from radians to degrees	degrees(1.57) is 90
radians(x)	Converts angle x from degrees to radians	radians(90) is 1.57

Strings and Characters

- A string is a sequence of characters. String literals can be enclosed in matching single quotes (') or double quotes ("). Python does not have a data type for characters. A single-character string represents a character.

- Strings have many methods to use to manipulate their data

```
Letter  = 'A'               # Same as letter = "A"
numChar = '4'               # Same as numChar = "4"
message = "Good morning"    # Same as message = 'Good morning'
```

The String Concatenation Operator

- You can use the + operator add two numbers. The + operator can also be used to concatenate (combine) two strings. Here are some examples:

```
message = "Welcome " + "to " + "Python"
```

Python Escape Sequences

Code	Result
\'	Single Quote
\\	Backslash
\n	New Line
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed

Example:

```
>>> print('I am learning about the \" Math and String functions\" \n I will practice them today’)

I am learning about the " Math and String functions"
 I will practice them today
```

Reading Strings from the Console

- To read a string from the console, use the input function. For example, the following code reads three strings from the keyboard:

```
s = input("Enter a string: ")
print("s is " + s)
```

Striping beginning and ending Whitespace Characters

- Another useful string method is strip(), which can be used to strip the whitespace characters from the both ends of a string.

```
s = "\t     Welcome      \n"
s1 = s.strip() # Invoke the strip method, s1 stores
'Welcome'
```

Methods

- **Methods** are subroutines that we would like to (re)use again and again in code
- For example, would you like a method to compute $\sqrt{x}$ or write a lengthy algorithm every time you wish to use it?
- Python provides many useful methods. Some we have seen:
  - **`print`**`()`, **`input`**`()`, **`round`**`()`

Interpreting Functions/methods

- Consider the following from the **Math** library:
  sqrt(x)
- sqrt is an *identifier*, i.e., a name, for this method
- x is called a **parameter**, or an **argument**. This is the *input* to the method.
- Methods can optionally *output* data too, in this case it will output a number.
- In a few weeks, we will learn to write our own methods. For now, we need to know how to use them.

Introduction to Objects and Methods

- In Python, all data—including numbers and strings—are actually objects.

- An object is an entity. Each object has an id and a type. Objects of the same kind have the same type. You can use the id function and type function to get these information for an object.

Strings

\n : end the current line of text and start a new one

```
>>> print(‘Hi there!\nHow are you\nI\'m fine')
Hi there!
How are you
I'm fine
```

Since the symbol ' has a special meaning in python (it denotes a string) So a backslash (\) is used in the middle of a string that tells Python that the next character will have special meaning.

\t :skip to the next “tab stop” in the text. This allows output in columns

```
>>> print('Hello\tHow are you\tI\'m fine')
Hello          How are you  I'm fine
```

```
>>> print('C:\name_folder’)
```

New line character

What will be the output?

```
C:
ame_folder
```

Print Examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")

x	x
xx	xx
xxx	xxx
```

Print Examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")
```

```
x  →  x
xx    xx
xxx   xxx
```

Print examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")
```

```
x	x
xx	xx
xxx	xxx
```

Print examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")

x	x
xx	xx
xxx	xxx
```

Print examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")
```

```
x	x
xx	xx
xxx	xxx
```

Print examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")
```

```
x       x
xx      xx
xxx→xxx
```

Print examples

```
>>>print("x\tx\nxx\txx\nxxx\txxx\n")

x	x
xx	xx
xxx	xxx

```

String Concatenation Example

```
>>> string1="We are‟
>>> string2="Learning"
>>> string3=string1+" "+string2+" "+"python"
>>> string3
'We are Learning python'
```

String Operations

Indexing: Strings can be indexed, with the first character having index 0

0	1	2	3	4	5	6
T	u	e	s	d	a	y

```
>>> word_1 = "Tuesday"

>>> word_1[0]

'T’

>>> word_1[5]

'a'
```

Slicing Operator [start : end]

The **slicing** operator returns a slice of the string using the syntax s[start : end]. The slice is a substring from index start to index end – 1 .

0	1	2	3	4	5	6
T	u	e	s	d	a	y

```
>>> word_1[2:5]
```

'esd'

Characters from positions 2 (included) to position 5 (excluded)

Slicing Example

```
>>> s='Tuesday’

>>> s[:6]
'Tuesda'
```

Slice string from index 0 (start) to index 6 – 1 (end -1)

```
>>> s[3:]
'sday'
```

Slice string from index 3 (start) to end of string

```
>>> s[1:-1]
'uesda'
```

Slice string from index 1 (start) to last but 1 (-1 from end) index of string

len() function

- Length of a string is defined as the number of characters in a string

- Example:

```
>>> word_1 = "Tuesday"

>>> len(word_1)

7
```

There are 7 characters in the string

count() function

- count() function returns the number of times a character or sequence of characters appear in a string.

- Example:

```
>>> word_1="It is going to rain"
```

```
>>> word_1.count('g')
```

```
2
```

‘g’ appears 2 times in word_1

```
>>> word_1="I have to go. I will go and check."
```

```
>>> word_1.count('go')
```

```
2
```

‘go’ appears 2 times in word_1

Searching for Substrings

- find(s1): int
  - Returns the lowest index where s1 starts in this string, or -1 if s1 is not found in this string

- Example:

Lowest index of ‘e’ is 1

```
>>> s="We are learning python“
>>> s.find('e')
1
```

upper() function

- The upper() function converts all the characters in a string to uppercase

Example:

```
>>> word_1="It is going to rain"

>>> word_1.upper()

'IT IS GOING TO RAIN'
```

lower() function

- The lower() function converts all the characters in a string to lowercase

Example:

```
>>> word_1="It is going to rain"
>>> word_1.lower()

'it is going to rain'
```

Repetition Operator *

- As the name suggests, repetition operator is used to repeat a string
- Example:

```
>>> string1_value = 'Hi’
>>> string2_value = string1_value * 5
>>> string2_value
'HiHiHiHiHi'
```

$\rightarrow$ Repeat string1_value 5 times

Exercise

1. Given 3 numbers 10, 15, 20. Write commands (math functions) to find maximum and minimum values

2. Write a program that reads 2 positive numbers (say, num1 and num2) and displays the following:
   - $num1^{num2}$
   - $\sqrt{num1} + \sqrt{num2}$

Use math library functions to solve the above problems.

Exercise

3. Given the string below,

    s = “Welcome”

Write commands in python that would do the following:

- Find the index (first occurrence) of the character “e” in the string)
- Find the index of the string “elc” in the input string
- Also display the input string 5 times using repetition operator

Thank you!
Questions?