09-09 (Handout)

1 Retrieval

2 Input and Output Functions (POGIL, 15 min)

You can use *functions* to perform specific operations. Some functions require values, known as *arguments*, to perform their operation. Functions may also *return* a result. For example:

```
name = input("What's your name? ")
```

`input` is a function, `"What's your name? "` is an argument, and the return value (typed by the user) is stored in `name`.

The following table shows additional examples of functions. They were written by a scientist to set up an experiment.

Do not type anything yet! Read the questions first!

Python code	Shell output
`input("enter the mass in grams: ")`	
`mass = input("enter another mass in grams: ")`	
`mass`	
`unit = input("enter the units for mass: ")`	
`print(mass, unit)`	
`print(mass / 2)`	
`ten = 10`	
`print(ten / 2)`	

Python code	Shell output
`abs(-1)`	
`abs(-1 * ten)`	

List the names of the three functions used in the examples.

What are the arguments of the first use of the `print` function?

Type each line of code in a Python Shell, one line at a time, and write the corresponding output (if observed) in the right column of the table. If an error occurs, write what type of error it was (i.e., the first word of the last line of the error message).

Place an asterisk (*) next to any output for which you were surprised, and note what was unexpected about the output. Don't worry yet about *understanding* any strange output you may see; we will discuss what it all means by the end of class.

Which function delayed execution until additional input was entered?

Which term, *user* or *programmer*, best defines the role of the person who entered the additional input? Explain.

Based on the Shell output, what does the word `mass` represent, and how did it get its value?

What does the word `ten` represent, and how did it get its value?

Do the values of `mass` and `ten` both represent a number? Explain why or why not.

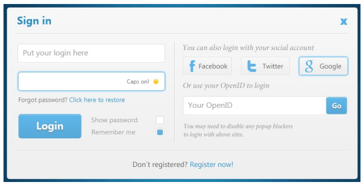
3 Input and Output

3.1

- Programming is nothing without the design of an **interface**!
  - I have to be able to **input** data in the program, and
  - I have to be able to get results (**output**) from the program.

3.2 Graphical input/output

- Also called Graphical User Interface (GUI) - we'll study it in Unit 10
- Kind of mimics the way we use mechanical input and output
- Traditionally, WIMP (Windows, Icons, Menus and Pointers)



You can also login with your social account

Or use your OpenID to login

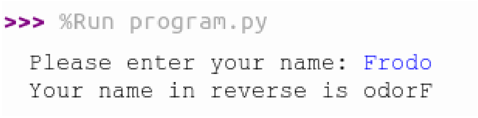
You may need to disable any popup blockers to login with above sites.

3.3 Text input/output

- Even simpler, however, it is a good start for programming!

```
name = input("Please enter your name:")
reverse = name[::-1]
print("Your name in reverse is", reverse)
```

The command-line interface will ask for input from our keyboard, and then:



```
>>> %Run program.py

  Please enter your name: Frodo
  Your name in reverse is odorF
```

4 Python text output: `print()`

4.1

- Put what you want to print between the parentheses: `print("Hello World")`
- If you want to jump to a new line, use `\n`: `print("Hello\nWorld")`
- You can also pass multiple arguments by separating them with commas: `print("x has the value:", x, "\nand y has the value:", y)`

5 Python text input: `input()`

5.1

- The command waits until the user types some text in the command-line interface and finishes with ENTER
- The term `input()` “turns” into the text entered, and is **ALWAYS** an object the type `string`!
- Thus, it needs to be saved into a variable: `x = input()`
  - After the user types “Hi”, for example, it is as if: `x = "Hi"`
- You can customize an input message by passing a string:

```
x = input("Please enter your name: ")
```

5.2 Input of numeric values

- Now, suppose we want to calculate the sum of two numbers:

```
x = input("Please enter first number: ")
y = input("Please enter second number: ")
z = x + y
print("The sum is", z)
```

What happened???

6 Converting string to number types

6.1

- You can convert a string to a number using the methods `int()` and `float()`
    - The string that goes inside the parentheses (which we call the “argument” of the method) will be turned to an integer/float

```
xstring = input("Please enter your age: ")
x = int(xstring)
print("Your age is ", x)
```

- Just make things shorter by chaining one method into another!

```
x = int(input("Please enter your age: "))
print("Your age is ", x)
```