Input and Arithmetic

Fall 2019

Input

- You can get input from the user using another built in Python function - input()
- Input is used much in the same vein as print
- In order to use the input, you must assign it to a variable
  - Example - name = input(“What is your name? “)
    - This will store whatever you type and enter into the variable “name”
    - You can then use this value in your program

```
# Asks the user to input a number
# stores that number to a variable
number = input("Enter a whole number: ")
```

```
=== RESTART: /home/kor
Enter a whole number:
```

```
=== RESTART: /home/kort/M
Enter a whole number: 6
```

Input

- Input will default store items as strings
- If you want to have an item stored as a different data type you have to convert it
  - Conversion happens through more built it Python functions
    - int() - converts an item to an int
    - float() - convert an item to a floating point
    - str() - convert an item to a string
  - Conversion introduces a potential for errors
    - Strings (that do not contain just a number; such as “hello” instead “2”; decimals will cause errors unless converted to a float first) sent to int() will cause an error
    - Strings (that do not contain just a number and decimal; such as “hello” instead “2” or “2.0”) sent to float() will cause an error

Conversion

- Conversion can happen two ways
  - Input can be converted directly and stored into a variable
    - Example. number = int(input(“Enter a number: “))
  - Variables can also be converted
    - Example. number = input(“Enter a number: “)
    - number = int(number)

```
# Asks the user to input a number
# stores that number to a variable
number = input("Enter a whole number: ")
print()

intNumber = int(number)

# shows output
print("The integer value is", intNumber)
```

Syntax Warning

- Make *absolutely sure* your parentheses match up
  - input() requires its own set of parentheses
  - If you put your input() inside of a conversion (int() or float()) you need to make sure input() has a complete set of parenthese
  - int() and float() *also* require a set of parentheses
- Every opening parenthesis must have a closing parenthesis
- Everything you're passing to a built-in function must be contained *inside* the parentheses

Arithmetic

Table 2-3 Python math operators

Symbol	Operation	Description
+	Addition	Adds two numbers
-	Subtraction	Subtracts one number from another
*	Multiplication	Multiplies one number by another
/	Division	Divides one number by another and gives the result as a floating-point number
//	Integer division	Divides one number by another and gives the result as an integer
%	Remainder	Divides one number by another and gives the remainder
**	Exponent	Raises a number to a power

Arithmetic

Table 2-6 Algebraic expressions

Algebraic Expression	Operation Being Performed	Programming Expression
$6B$	6 times B	`6 * B`
$(3)(12)$	3 times 12	`3 * 12`
$4xy$	4 times x times y	`4 * x * y`

Table 2-7 Algebraic and programming expressions

Algebraic Expression	Python Statement
$y = 3\frac{x}{2}$	`y = 3 * x / 2`
$z = 3bc + 4$	`z = 3 * b * c + 4`
$a = \frac{x + 2}{b - 1}$	`a = (x + 2) / (b - 1)`

Arithmetic Notes:

- Math in programming follows the order of operations
  - To separate out a section to complete outside of the order of operations, use parentheses (as you would on a calculator)
- When receiving input from a user to use in math, you must convert that to an int or float *first*
  - Trying to send a string to a math function will result in an error

```
# gets input from the user
firstNumber = int(input("Enter a whole number: "))
secondNumber = int(input("Enter another whole number: "))
print()

# adds those numbers together
numberSum = firstNumber + secondNumber
```

Notes:

- With arithmetic (and *all* variables in Python) make sure variable assignment statements always reads as variable = value
  - x = 2 + 2 is valid
  - 2 + 2 = x is *invalid*
- Make sure there's only *one* variable per assignment statement and that *all* calculations appear on the right side of the assignment statement
  - y = (x + 1) * 3 is valid
  - y / 3 = x + 1 is *invalid*