CS 1112: Introduction To Programming

Python Basics; Variables; Operations; Type Casting

Dr. Nada Basit // basit[*at*]Virginia[*dot*]edu

Spring 2024

Friendly Reminders

- Your ***safety*** and ***comfort*** is important!
  - If you choose to wear a mask you are welcome to do so
  - *We will interpret wearing a mask as being considerate and caring of others in the classroom (not that you are sick), and realize that some may choose to mask to remain distanced*
- Be an ***active*** participant in your learning!
  You're welcome and ***encouraged*** to ask questions during class!
- If you feel ***unwell***, or think you are, please stay home
  - *We will work with you!*
  - Get some rest ☺
  - View the recorded lectures – *please allow 24-48 hours to post*
  - *Contact us!*

Reminder: Syllabus Quiz

Don't forget to take the Syllabuzz Quizz!

- This quiz is *Mandatory!*
- Take this quiz ***individually***. Absolutely no collaboration permitted.
- Must get **100%** (12 points) to stay in the course!
  - *May take it as many times as needed.*
  - Review the detailed Syllabus
  - This quiz is ***open-book***
- ***Where?:*** *"Assignments" tab > "Syllabus Quiz (Required)";* **or** *"Quizzes" tab*
- ***Deadline:*** **January 31, 2024 @ 11:00pm**. *Take it early!*

- Office hours officially start next week: Jan. 29
- My office hours – location: Rice 405
- TA office hours – location: Thornton Stacks
  (if an alternate location is chosen, we'll let you know!)

CS 1112 Pledge!

Taking this pledge is mandatory for our class to have a community of trust

★ Google form: https://forms.gle/eksrS9pcinGLYNKu7 (try this first)

★ Microsoft form: https://forms.office.com/r/wVdPysWRN6

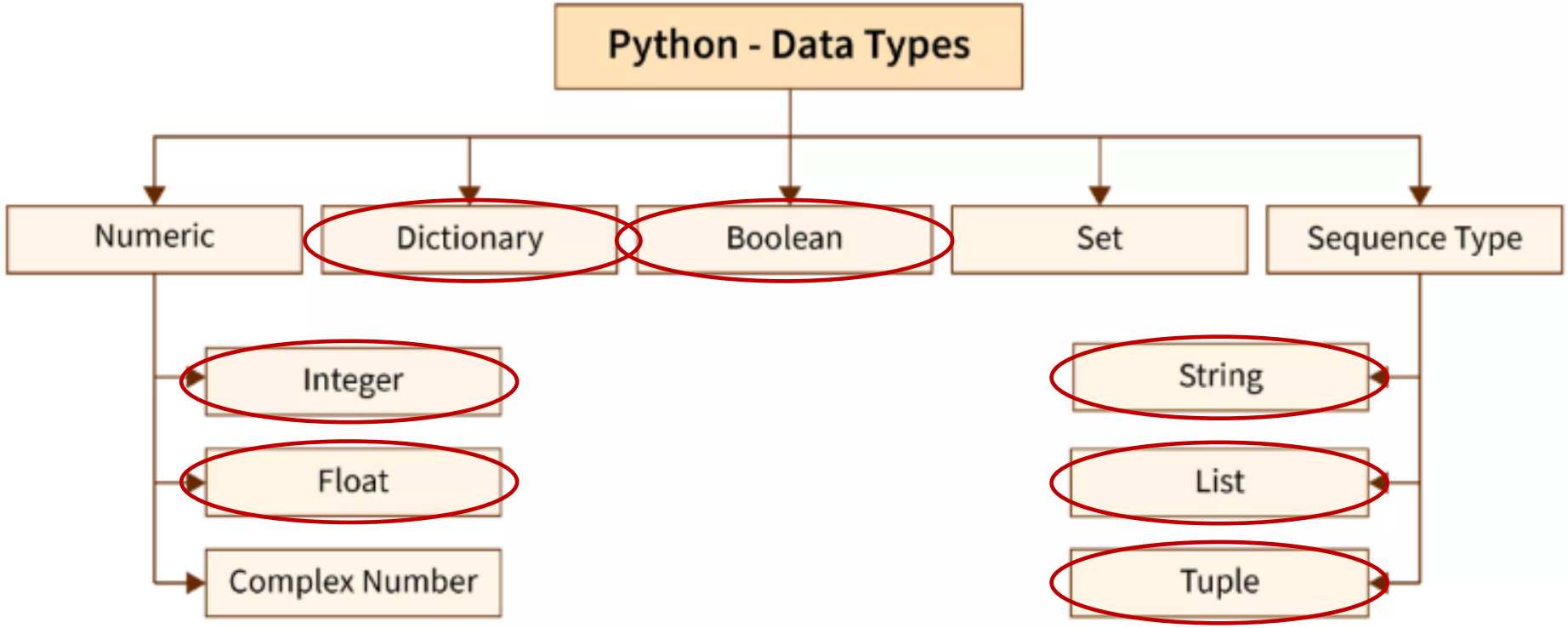
[Please submit your pledge at only ONE of the above links!]

I gratefully acknowledge Prof. J. Cohoon for introducing the course pledge in this course!

Quick & Fun Survey Questions

Get to know your peers! ☺

When you listen to music, do you listen to the rhythm or the lyrics?



© Scaler, topic on data types

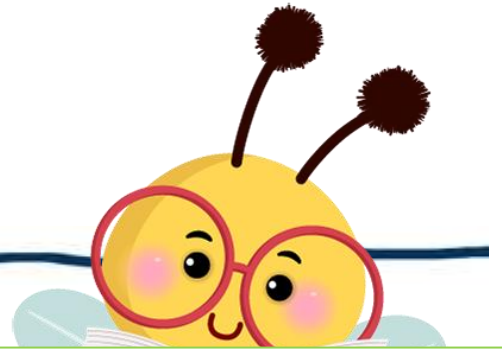
Data Types

Variables and Variable Types

- Variable names
  - Lower case *(all characters)*
  - Snake case (e.g., `prime_number`, or `int_quotient`)
- Variable Types

Variable Type	Description (example)
Integer (int)	Whole number (77)
Float (float)	*(Floating-Point number)* Positive or negative numbers with a decimal point (8.16, -2.34)
String (str)	A sequence of characters. Can be delimited using either single or double quotes (“hello”)
Boolean (bool)	Objects of Boolean type may have one of two values (True or False)
List	An *ordered* collection of objects (like an array in other languages) (lst = [‘cat’, ‘rabbit’, ‘dog’])
Tuple	An *ordered*, *immutable*, collection of objects (tpl = (‘beetroot’, ‘cauliflower’, ‘fennel’, ‘garlic’))
Dictionary	A collection of key-values pairs, with no duplicate keys

Variables and Variable Types



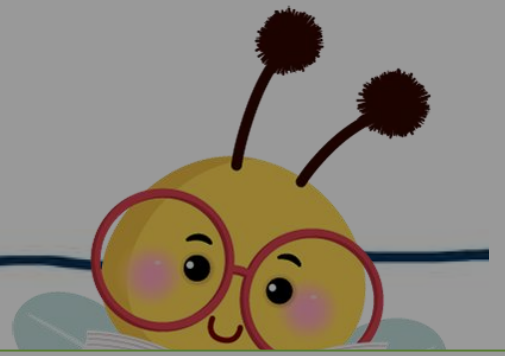
- Variable names
  - Lower case
  - Snake case (e.g., `prime_number`, or `int_quotient`)
- Variable Types

Example of a dictionary:

```
this_dictionary = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
```

Variable Type	Description (example)
Integer (int)	Whole number (77)
Float (float)	*(Floating-Point number)* Positive or negative numbers with a decimal point (8.16, -2.34)
String (str)	A sequence of characters. Can be delimited using either single or double quotes ("hello")
Boolean (bool)	Objects of Boolean type may have one of two values (True or False)
List	An *ordered* collection of objects (like an array in other languages) (lst = ['cat', 'rabbit', 'dog'])
Tuple	An *ordered*, *immutable*, collection of objects (tpl = ('beetroot', 'cauliflower', 'fennel', 'garlic'))
Dictionary	A collection of key-values pairs, with no duplicate keys

Variables and Variable Types



- Variable names
  - Lower case
  - Snake case (e.g., `prim`
- Variable Types

Example of a dictionary:

```
this_dictionary = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
```

See example of variables and data types in today's Python script!

Variable Type	
Integer (int)	Whole n
Float (float)	*(Floating-Point number)* Positive or negative numbers with a decimal point (8.16, -2.34)
String (str)	A sequence of characters. Can be delimited using either single or double quotes ("hello")
Boolean (bool)	Objects of Boolean type may have one of two values (True or False)
List	An *ordered* collection of objects (like an array in other languages) (lst = ['cat', 'rabbit', 'dog'])
Tuple	An *ordered*, *immutable*, collection of objects (tpl = ('beetroot', 'cauliflower', 'fennel', 'garlic'))
Dictionary	A collection of key-values pairs, with no duplicate keys

Arithmetic Operations:

```
# Calculations
total = a + b             # simple addition
difference = a - b        # simple subtraction
product = a * b           # multiplication
dec_quotient = a / b      # normal division : result is a float
int_quotient = a // b     # integer division : result is an integer (cuts off the decimal portion)
power = a ** b            # power : a^b == a**b
remainder = a % b         # modulus operator : result is the remainder value after a/b
```

Example:
a = 3
b = 2

```
# Print results
print( "a + b: ", total)
print( "a - b: ", difference)
print( "a * b: ", product)
print( "a / b: ", dec_quotient)
print( "a // b:", int_quotient)
print( "a ** b:", power)
print( "a % b: ", remainder)
```

Boolean Operations:

```
# Boolean Operations
x = True
y = False
and_result = x and y
or_result = x or y
not_result = not x
# Print results
print(and_result)
print(or_result)
print(not_result)
```

Let's see how we might solve a larger problem in today's Python script!

[Rounding a floating-point number]

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth Tables

```
# Boolean Operations
x = True
y = False
and_result = x and y
or_result = x or y
not_result = not x
# Print results
print(and_result)
print(or_result)
print(not_result)
```

Type Casting

Most common use case for type casting relates to **user input**, especially data that is numerical that comes in as a string and needs to be converted

- Type-casting is a method with which we can switch (*convert*) a variable's **type** to another
- i = 1
  f = 2.9
  s = "33"
  st = "4.5"
  b = "True"
- Conversions:

		Result:	
• float to int:	`int(f)`	`2`	
• int to float:	`float(i)`	`1.0`	
• str to int:	`int(s)`	`33`	[*invalid*: int("hello")] **!!**
• str to float:	`float(st)`	`4.5`	
• str to bool:	`bool(b)`	`True`	[*always True*: bool("hello") = True] **!!**

Python Demonstration

Variables and variable **Types**

Arithmetic and Boolean **Operations**

Type Casting with Variables

In-Class "lab" Activity: Vocabulary Worksheet

In-Class "lab" Activity!

Vocabulary Worksheet

- In pairs work on the vocabulary worksheet
- **Match** each term on the LEFT with its definition or code snippet (that represents the term) on the RIGHT
- *No choice is used more than once!*

Remember to check-in with a TA before leaving class today!

Notes/Reminders…

Reminder: CS Laptop Loaner Program

- This course requires students to have a **laptop**
- I realize that not everybody might have one (nor necessarily need one for their desired major / path…)
- If you do not have a laptop for any reason… ***not to worry!***
- The CS department's Systems staff has a notebook / laptop loaner program and will be able to loan you a notebook / laptop computer for the duration of the semester if you don't have one or if you cannot afford one.
  - Also available if your laptop is broken and under repair, we can arrange for you to receive a loaner laptop for a week or two until your own laptop is fixed

Interested? Link: https://www.cs.virginia.edu/wiki/doku.php?id=cs_laptop_loaner

I am happy to be your sponsor. Please let me know.

Tools: Piazza

- We will use **Piazza** in the following way:
  - Website: https://piazza.com/ [Linked through Canvas]
  - Piazza is a great tool for asking questions about **course content**, **policies**, or getting help on **homework** assignments
  - While you are waiting for an answer, see if there's an answer you can provide to someone else's question. We're all in this together! CS is a team sport! ☺
  - TAs will monitor and answer questions throughout the semester
  - Not a means to help you debug your code! (See more below)

It is very important to remember the following:

- **Do not post** complete or partial code solutions (for Homework) on Piazza when seeking answers to your question unless it is in a **PRIVATE** post
- **Do not post** complete or partial quiz solutions (code or short-answer) when seeking answers to your question unless it is in a **PRIVATE** post

Tools: Gradescope

- We will use **Gradescope** in the following way:
  - Website: https://www.gradescope.com/
  - Linked through Canvas
  - Homework assignments will be **submitted**
    - Most programming assignments are autograded
    - Some aspects of programming assignments may be manually graded