

CS 230
Programming Languages

10 / 13 / 2022

Instructor: Michael Eckmann

Today's Topics

- Questions? / Comments?
- An overview of Python (some of which you may not know)
-

Python

- Reliability – e.g. has array bounds checking, try/catch exceptions (like Java)
- Readability – by design **requires** indentation which formats the source appropriately
- Easy entry into learning the language (and this can be gradual) --- doesn't require creating a class (like Java does), ...
- Many useful libraries available for a variety of purposes

Python

- Python uses late binding / dynamic typing --- that is types of data determined at run-time vs. statically typed languages (e.g. Java) where types can be determined at compile-time
- Variables do not have a type / storage, instead they point to a value (data) which has a type. A variable may point to different types of data over time.
- Integers, floats and strings are all immutable
 - Makes it seem that these are passed by value since an attempt to modify a string or number through a parameter fails to change the value in the caller, but this is caused by the immutability of the values not by how Python does parameter passing

Python

- Code formatting
 - Python requires blocks of code to be either indented using tab(s) or space(s) your choice --- but do not interchange
 - it is visually impossible to tell whether a tab or spaces are used
 - Readability by design

Python

- Some data types:

int = integer

float = floating point number

complex = complex number

str = string

Python

- Operators
- `+, -, *, /, //, **, %` (exponentiation), `%` (modulus)
- integer division - truncates the result

e.g.

`1//2` results in `0`

`1/2` results in `0.5`

no limit on size of an integer

e.g. `21 ** 100`

Python

Operators

all operators on last slide work on ints and floats

overflow problems with floats (limited precision)

e.g. try `100 ** 20000.0` vs. `100 ** 20000`

support for complex numbers: `1j` is square root of `-1`

Python

Importing modules

`import math` vs. `from math import *`

first one needs `math.` preceding all calls

second does not, but overwrites any functions

that are named the same as in `math` module

e.g. try `math.sqrt(2)` vs. `sqrt(2)`

Python

help()

math

math.sqrt

Enter to exit help

exit() to exit the python interactive interpreter

Python

type conversion use type name with value to be converted inside parens

e.g.

`int(5.7)` --- float to int

`float('2.1')` --- str to float

`str(18)` --- int to str

`round(4.3)` ---- results in 4

`round(4.5)` ---- also results in 4

`round(4.500001)` ---- results in 5

Python

variables

names can be any length containing letters, digits, underscore,
can't start with a digit,

can't be a python keyword

case sensitive: TAX is different from Tax

assignment statements with = (the assignment operator)

assignments don't make copies, they just label/point a variable
to a value

numbers and strings are immutable

Python

- Variables
 - No concept of a variable declaration
 - Variables don't have a type, but values do
 - A variable can point to different types of values over time

Python

- Can use `type(var_name)` to see what type of the data item `var_name` is pointing to. Also can do `type(value)`
- `<`, `>`, `<=`, `>=`, `==`, `!=` - comparison operators, can be chained
- `and`, `or` – short circuit, evaluate to the operand that determines results
- Indentation – Python style guide says 4 spaces, no tabs, per indentation level.

Python

- Show some things can do with mutable lists
- List slices using :
- Negative indices
- append and remove
- len()

Python

- Let's cause an exception:
- `int('abc')`

```
try:  
    # statements to do that may cause ValueError  
except ValueError as err:  
    # statements to do if exception ValueError happens
```