Lecture 2: Variables, Types, Assignments

Interesting facts about Python

- Fastest growing programming language
- Widely used in Data Science
- Demand for Python developers is high and growing
- All the big names use Python: Google, Facebook, IBM, Instagram, Dropbox etc.
- average salary of Python developer is $100,000+ in the US

Compare a Java program and a Python program

```java
package com.company;

public class Main {

    public static void main(String[] args) {
    // write your code here
        System.out.println("Hello World!");
    }
}
```

```python
print("Hello World!")
```

Notes about your first program: part1

- No need to write a class like Java
- No need to write a function like Java
- No need for semicolon

- Use the print function, pass the text as an argument, and that’s it

Notes about your first program: part2

- You can however write a class like Java
- You can write a method within the class
- You can just write a function

- However, for most things, Python tries to be as simple as possible
- Focuses on readability
- Uses indentation to highlight blocks of code

Variables

- Variables store data such as numbers and letters.
  - Think of them as places to store data.
  - They are implemented as memory locations.
- The data stored by a variable is called its value.
- The value is stored in the memory location.
- Its value can be changed.

Variables

Examples:

- x = 10
- y = 20
- name = “John”
- answer = True
- x = “John”

Variables and Values

Variables

- numberOfBaskets
- eggsPerBasket
- totalEggs

Assigning values

- eggsPerBasket = 6
- eggsPerBasket = eggsPerBasket - 2
- totalEggs = 12

Naming and Declaring Variables

- Choose names that are helpful such as **count** or **speed**, but not **c** or **s**.
- When you declare a variable, you provide its **name**.
  - numberOfBaskets, eggsPerBasket;
- Unlike many other languages, you don't need to declare the type of the variable before using it
- ***No need to declare a variable before using it***
- Create variables as you need

Static vs Dynamic Typing: Part1

- Types of variable or expression determined from source at “compile time”
- Some may be declared explicitly
- Others inferred implicitly from context
- Examples: Java, Scala, Haskell

source of this slide: https://john.cs.olemiss.edu/~hcc/csci556/Py3MPP/Ch05/TypeConceptsSlides.html#(8)

Static vs Dynamic Typing: Part2

- Dynamically typed language:
  - type of variable or expression cannot be determined at “compile time” – checked at runtime
- Examples: Lisp, Python, JavaScript, Lua
- Most languages mix static and dynamic
- Java uses declared types

source of this slide: https://john.cs.olemiss.edu/~hcc/csci556/Py3MPP/Ch05/TypeConceptsSlides.html#(8)

Naming variables

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

Naming variables

- Identifiers may not contain any spaces, dots (.), asterisks (*), or other characters:
  - **7-11** (not allowed)
  - **yourName.com** (not allowed)
  - **util.*** (not allowed)
  - **%hello** (not allowed)
- names can be arbitrarily long.
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Keywords or Reserved Words

- Words such as **if** are called *keywords* or *reserved* words and have special, predefined meanings.
  - Cannot be used as identifiers.

Reserved Keywords in Python

and	except	lambda	with
as	finally	nonlocal	while
assert	false	None	yield
break	for	not	
class	from	or	
continue	global	pass	
def	if	raise	
del	import	return	
elif	in	True	
else	is	try	

Naming Conventions

- Class names begin with an uppercase letter
- Built-in types start with lowercase
  - int, str, unicode, float etc.
- If variable name has multiple words, use either **snake_case** or **camelCase**
  - speed_of_light (snake_case)
  - speedOfLight (camelCase)

Examples of Primitive Values

- Integer types
  - 0 -1 365 12000
- Floating-point types
  - 0.99 -22.8 3.14159 5.0
- Character type
  - 'a' 'A' '#' ' '
- Boolean type
  - True False
- Since Python is dynamically typed, the type of the variable will be determined at run-time depending on the value

Assignment Statements

- An assignment statement is used to assign a value to a variable.
  - answer = 42
- The "equal sign" is called the assignment operator.
- We say, "The variable named **answer** is assigned a value of 42," or more simply, "**answer** is assigned 42."

Assignment Examples

- amount = 3.99
- firstInitial = 'W'
- score = numberOfCards + handicap
- eggsPerBasket = eggsPerBasket - 2

```python
print("Hello World!")

name = "John"
age = 23

salaryPerHour = 12.5

print(name)
print(age)
print(salaryPerHour)

salaryPerHour = int(salaryPerHour)
print(salaryPerHour)

salaryPerHour = salaryPerHour + 0.5
print(salaryPerHour)
```

```
Hello World!
John
23
12.5
12
12.5
```

```
: print("Hello World!")

name = "John"
age = 23

salaryPerHour = 12.5

print(name)
print(age)
print(salaryPerHour)

salaryPerHour = int(salaryPerHour)
print(salaryPerHour)

salaryPerHour = salaryPerHour + 0.5
print(salaryPerHour)

salaryPerHour = str(salaryPerHour)
salaryPerHour = salaryPerHour + 2
print(salaryPerHour)
```

```
Hello World!
John
23
12.5
12
12.5

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-614c447ac4c6> in <module>
     17
     18 salaryPerHour = str(salaryPerHour)
---> 19 salaryPerHour = salaryPerHour + 2
     20 print(salaryPerHour)

TypeError: can only concatenate str (not "int") to str
```