Fairfield UNIVERSITY

CSPS 1101
Introduction to Computing

School of Engineering & Computing
Dept. of Computer Science & Engineering
Dr. Sidike Paheding

Agenda

- Call function, print()
- Variables
- Data Types
- Assignment
- Reading assignment: Ch 1 & 2: pages 28 - 43

The print() function

Function	Description
`print([data])`	Prints the data argument to the console followed by a new line character. If the call doesn't include a data argument, this function prints a blank line to the console.

A script with three statements:

- print("Hello out there!")
- print()
- print("Goodbye!")

The print() function

- Printing Multiple Items:
- You can print multiple items by separating them with commas.

```
print("The answer is", 42)
```

```
The answer is 42
```

The print() function

- Using String Formatting:
- You can format strings using f-strings:

```
name = "Alice"
age = 30
print(f"{name} is {age} years old.")
```

Alice is 30 years old.

The print() function

- Print with Custom Separator:
- You can specify a custom separator between items using the ‘sep’ parameter.

```
print("apple", "banana", "cherry", sep=" | ")
```

apple | banana | cherry

The print() function

- Print with Custom End:
- By default, print() ends with a newline (\n). You can change this using the ‘end’ parameter.

```
print("Hello", end=", ")
print("World!")
```

Hello, World!

The print() function

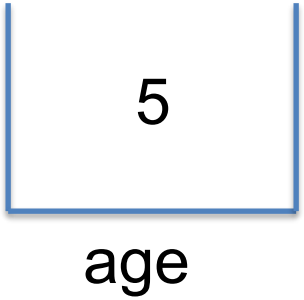
- Print with Formatted Strings (Using .format())

```
language = "Python"
version = 3
print("I am learning {} version{}.".format(language, version))
```

I am learning Python version 3.

Variables

- A variable is a “box” (i.e. storage location) with a label
- The label should be meaningful in natural language

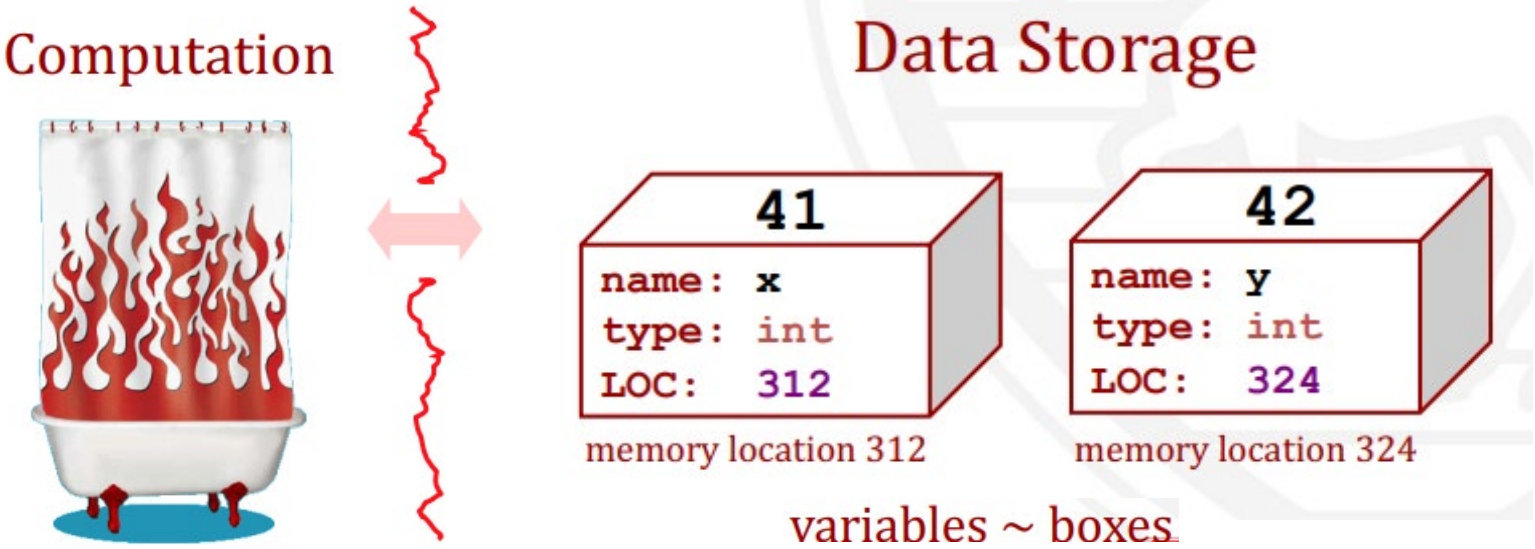


Inside the machine...

What's happening in python:

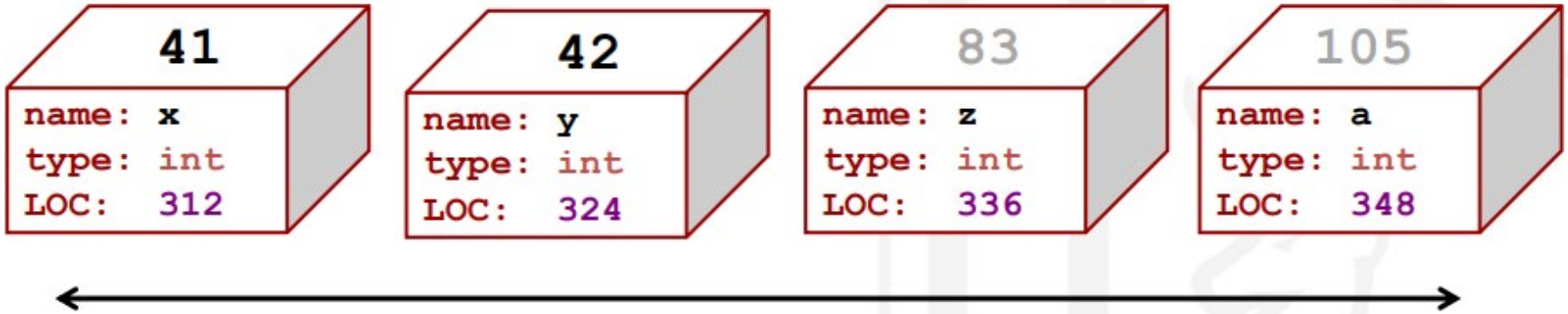
```
x = 41
y = x + 1
```

What is happening behind the scenes:



Memory!

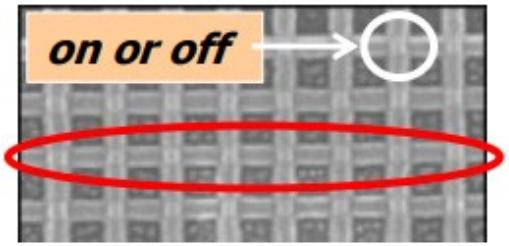
Random Access Memory



a big list of boxes, each with a name, type, location, and value

512 MB of memory

Wow! Who knew they make memory out of wicker?



bit = 1 "bucket" of charge

byte = 8 bits

Fairfield UNIVERSITY

Python's data types

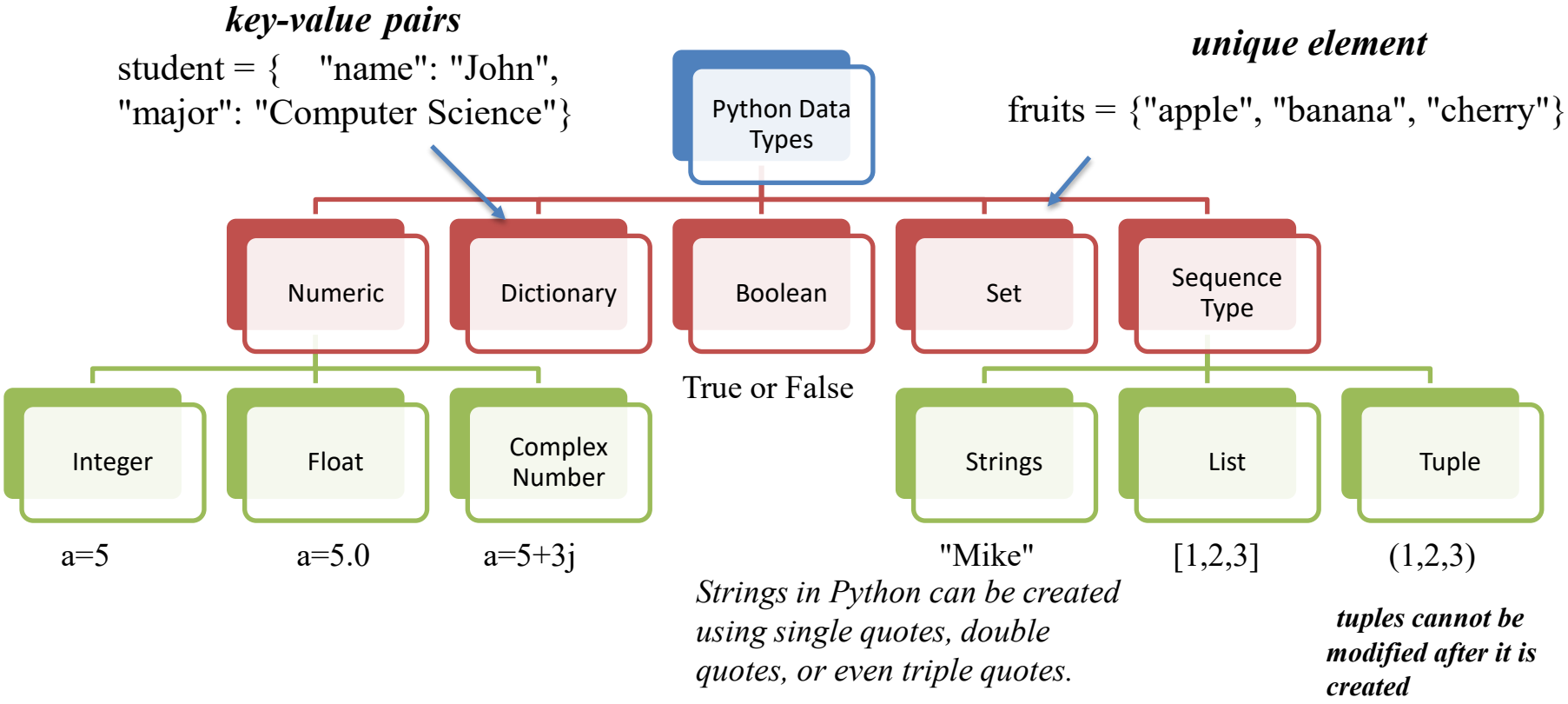
- Python Data types are the classification or categorization of data items.
- Since everything is an object in Python programming, Python data types are classes and variables are instances (objects) of these classes.
- To define the values of various data types of Python and check their data types we use the type() function.

Python's data types

- The following are the standard or built-in data types in Python:

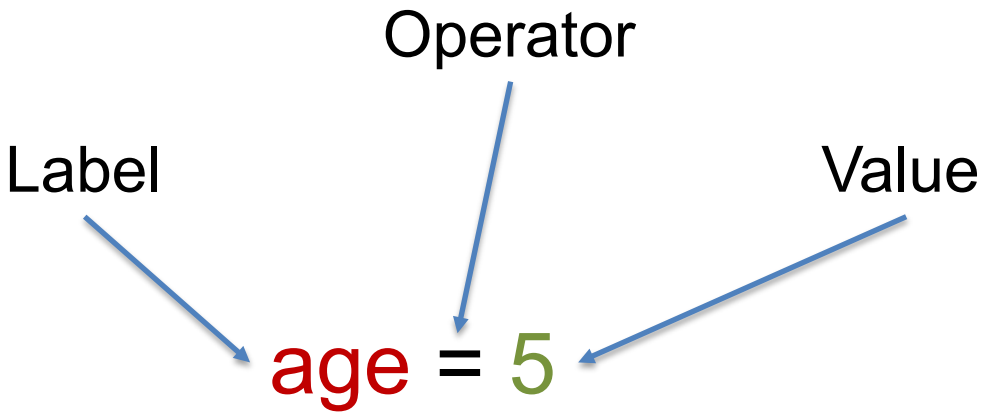
1) Numeric
2) Sequence Type
3) Boolean
4) Set
5) Dictionary
6) Binary Types (memoryview, bytearray, bytes)

Python's data types



Variables (cont'd)

- Syntax: Assignment statement – creates a new variable and gives it a value.



Variables Naming Convention

Naming Restrictions:

- should start with an alphabetic character (A -Z) or an underscore (_)
- no special characters: ., @, $, or %
- Can't be a keyword
- Python is case sensitive!
  - var1 is different than Var1

Python keywords

- The following identifiers are used as reserved words, or keywords of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

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

Source: https://docs.python.org/3/reference/lexical_analysis.html#keywords

naming styles for variables

- Start all variable names with a lowercase letter.
- Use underscore notation or camel case.
  - variable_name # underscore notation
  - variableName # camel case
- Use meaningful names that are easy to remember.
- Don’t use the names of built-in functions, such as print().

Variables (cont'd)

- Just "assign" a value to a variable (no declaration needed)
  - var1 = 10
  - var2 = "hello"
- Updating Variables (Reassignment)
- I can write the following:

```
var1 = 10
var1 = var1 + 1
Or
var1 = var1 - 5
```

- What is the output of the first addition?
- We can use a variable on both sides of "="

Variables (cont'd)

- Code that initializes variables
  - `first_name = "Mike" # sets first_name to a str of "Mike"`
  - `A = 3 # sets A to an int of 3`
  - `B = 5 # sets B to an int of 5`
- Code that assigns new values to the variables above
  - `first_name = "Joel" # sets first_name to a str of "Joel"`
  - `A = 10 # sets A to an int of 10`
  - `A = B # sets B to an int of 5`
  - `A = "15" # sets A to a str of "15"`
- Code that assigns values to two variables
  - `Aa = 3, Ee = 35`
- Code that causes an error due to incorrect case
  - `Aa = ee # NameError: 'ee' is not defined`

Variables (cont'd) Summary

- A variable can change, or vary, as code executes.
- A data type defines the type of data for a value.
- An assignment statement uses the equals sign (=) to assign a value to a variable. The value can be a literal value, another variable, or an expression like the arithmetic expressions in the next figure.
- To initialize a variable, you assign an initial value to it. Then, you can assign different values later in the program.
- You can assign a value of any data type to a variable, even if that variable has previously been assigned a value of a different data type.

Variables (cont'd)

- Let's try some samples!

1)	var1 = "hi" var2 = 5 *What is the value of var2 ?*
2)	var1 = "hello" var1 = 10 *What is the value of var1 ?*
3)	var1 = 10 var2 = var1 *What is the value of var2 ?*
4)	*var1 = var2 = var3 = 300* *What is the value of var1, var2, and var3 ?*

Output functions

- age = 50
- print(age)
- print(age + 10)

```
# Assigning values to variables
first_name = "John"
last_name = "Doe"

# Combining strings and variables
full_name = first_name + " " +
last_name

# Printing the combined string
print("Full Name:", full_name)
```

Class Assignment I

1. Use basic print() to display a message as “Python is fun”.
2. You have several pieces of information stored in separate variables, such as the words "Python", "is", "a", "powerful", "language", and you need to display them together in a single line.
3. You are building a user interface where you need to print a list of colors, but they should be separated by semicolon for readability. [Use the ‘sep’]
4. In a console-based game, you want to print a loading message followed immediately by "Complete!" on the same line, to simulate a progress indicator. [hint: use ‘end’]
5. You are logging activity in your program, such as how many hours a user spent on a task, and you need to format this information neatly in a log message. [hint: use ‘.format()’]

Class Assignment II

- Find out data types of the following:

1) x = "Hello World"
2) x = 50
3) x = 60.5
4) x = 3j
5) x = ["geeks", "for", "geeks"]
6) x = ("geeks", "for", "geeks")
7) x = range(10)
8) x = {"name": "Suraj", "age": 24}
9) x = {"geeks", "for", "geeks"}
10) x = True
11) x = b"Geeks"

Class Assignment III

1) Write a Python program to swap two numbers without using a temporary variable.
2) Write a Python program to swap three variables in Python (e.g., a, b, and c) such that a becomes b, b becomes c, and c becomes a?

Fairfield UNIVERSITY

Thank you for participating in CPSC 1101 - Intro to Computing.

Are there any questions?

spaheding@fairfield.edu