Python: Variables & Data Objects

1405 Instructor: Ruiqing (Sam) Cao

Data Objects & Their Types

- A **data object** can be anything that is stored as data or information, for instance – a number, a piece of text, a sequence, and even a function

- Everything can be a data object, and has associated attributes (properties) and methods (actions)

- A **data type** is a fundamental characteristic of a data object (or a value)

Scalar Types: numeric, bool, NoneType

- **Scalar types** represent a single, atomic, and indivisible value

- A scalar data object is a *numeric* value (**`int`** or **`float`** type) or a *Boolean* value (**`bool`** type)
  - (Ex. `int`) `1`, `2`, `-3`; (Ex. `float`) `-3.14159`, `2.7182`
  - `bool` only has two values `True`, `False`

- **`NoneType`** is a special type of scalar object that only has one value `None` – it is used to represent a null or empty value

Casting (Between Scalar Types)

- **Casting** (or converting) a data object from one scalar type into another is allowed, and quite common in some cases

- Automatic casting from a *coarser* type to a *finer* type
  `bool` → **`int`** → **`float`**
  - For instance, `True+1+0.2` (becomes `1.0+1.0+0.2`) → `2.2`

- Explicit casting using *type conversion functions*
  - For instance, `float(3)` → `3.0`; `int(3.0)` → `3`; `bool(0.0)` → `False`
  - But be careful: accidental precision loss e.g., `int(2.9)` → `2`

Data Types: Scalar & Non-Scalar

Type	Examples	Description
Numeric	`10,3.14,-5`	Numbers such as integers (`int`) or decimals (`float`)
Boolean	`True,False`	Logical values, True or False (`bool`)
String	`"Hello",'world'`	Text data, enclosed in quotes (`str`)
List	`[1,2,3],["a","b"]`	A sequence of items that *can* be changed (`list`)
Tuple	`(1,2,3),("a","b")`	A sequence of items that *cannot* be changed (`tuple`)
Dictionary	`{"key":"value"}`	A collection of key-value pairs (`dict`)
Set	`{1,2,3},{"a","b"}`	An unordered collection of unique items (`set`)

Data Types: Scalar & Non-Scalar

Type	Examples	Description
Numeric	`10,3.14,-5`	Numbers such as integers (`int`) or decimals (`float`)
Boolean	`True,False`	Logical values, True or False (`bool`)
String	`"Hello",'world'`	Text data, enclosed in quotes (`str`)
List	`[1,2,3],["a","b"]`	A sequence of items that *can* be chang
Tuple	`(1,2,3),("a","b")`	A sequence of items that *cannot* be cha (`tuple`)
Dictionary	`{"key":"value"}`	A collection of key-value pairs (`dict`)
Set	`{1,2,3},{"a","b"}`	An unordered collection of unique item

Scalar Types

Ordered Non-Scalar

Unordered Non-Scalar

Variables, Values, & Data Objects

- A **variable** is a name that refers to a **value** stored in the computer's memory
- A value is a **data object**; a variable is not a data object, but the value stored in it is a data object
- A value stored in a variable can be *accessed* and *modified* by referencing the name of the variable

Variable & Value Stored in it

The glass (variable) holds **orange juice** (value)

The same glass (variable) holds **black coffee** (value)

```
glass = "orange juice"
print(glass)
```

Output: `orange juice`

```
glass = "black coffee"
print(glass)
```

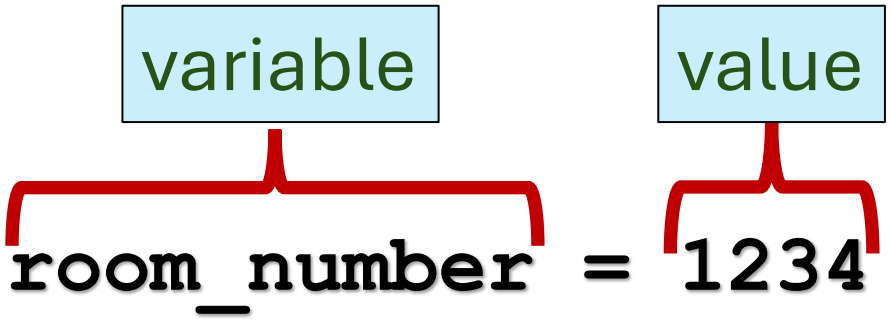
Output: `black coffee`

The same `print(glass)` prints different outputs because the value stored in `glass` has changed !

Assigning a Value to a Variable

- A value can be assigned to a variable using the ***assignment operator***, which is the equal sign (=) in Python

- After assigning a value to a variable, that value becomes stored in the variable



Assign (=) vs. Equal (==) Operators

Important: the assignment operator (assign) **=** *does not* judge if two values are equal, which the comparison operator (equal) **==** does

- The assignment operator (assign) `=` stores the specified value in the specified variable, and does not return any value

- The comparison operator (equal) `==` evaluates to `True` if two values are of the same value, and `False` if they are of different values

Data Types: Scalar & Non-Scalar

Type	Examples	Description
Numeric	`10,3.14,-5`	Numbers such as integers (`int`) or decimals (`float`)
Boolean	`True,False`	Logical values, True or False (`bool`)
String	`"Hello",'world'`	Text data, enclosed in quotes (`str`)
List	`[1,2,3],["a","b"]`	A sequence of items that *can* be chang
Tuple	`(1,2,3),("a","b")`	A sequence of items that *cannot* be cha (`tuple`)
Dictionary	`{"key":"value"}`	A collection of key-value pairs (`dict`)
Set	`{1,2,3},{"a","b"}`	An unordered collection of unique item

Scalar Types

Ordered Non-Scalar

Unordered Non-Scalar