What is a Variable?

- Python allows us to name objects so that we can refer to them later.

- A variable name is a location in memory which we can refer to by an identifier, and in which a data value that can be changed is stored.

Identifiers

- an identifier must start with a letter or underscore, and be followed by zero or more letters

  (A-Z, a-z), digits (0-9), or underscores

- VALID

```
age_of_dog          taxRateY2K
PrintHeading  ageOfHorse
```

- NOT VALID (Why?)

```
age#                    2000TaxRate
Age-Of-Cat
```

Identifiers

- Identifiers are case sensitive, e.g. age is not the same as Age
- Reserved words (keywords) have special meaning in Python and cannot be used as programmer-defined identifiers
  - Table 1-2, p. 17
- Variable name should reflect its use
- Variables can reference different values while program is running

Assignment Operator Syntax

```
Variable  =  Expression
```

First, Expression on the right is evaluated.

Then the resulting value is stored in the memory location of the variable on the left, which we can reference later on our program

Example

```
>>> pi = 3.14
>>> pi
>>> radius = 8.0
>>> height = 16
>>> baseArea = pi * radius **2
# comment
>>> baseArea
>>> cylinderVolume = baseArea * height
>>> cylinderVolume
```

Exercise

value of

a	b	c	d	idx

```
>>> a = 10
>>> b = 20
>>> a = b
----------------------
>>> a = 10
>>> b = 20
>>> c = a * b
>>> d = a + b
----------------------
>>> idx = 0
>>> idx = idx + 1
>>> idx = idx + 2
```

Script Mode

- Statements entered in interactive mode are not saved as a program

- To have a program use script mode
  - Save a set of Python statements in a file
  - The filename should have the .py extension
  - To run the file, or script, type

    `python` *`filename`*

    at the operating system command line or open the file and select Run -> Run Module

Demo

- Open Python shell
  - File -> New File
  - Enter

```
pi = 3.14
radius = 8.0
baseArea = pi * radius **2
print(baseArea)
```

  - File -> Save As
    - Navigate to your H: drive
    - Name the file myFirstProg.py
  - Run -> Run Module
  - Program output is written to ???
  - Help -> Python docs

What is a string?

- Everything in Python is an object; objects could be of different types

- A string is simply a sequence of characters (any characters)
  - In Python an object is a string, if it is surrounded by single or double quotes

```
>>> "hello"
??
>>> 'hello'
??
>>> howdy = "Howdy TCSS 142"
>>> howdy
??
>>> howdy2 = “You rock!”
>>> howdy2
??
```

Concatenation

- To append strings together, use +

```
>>> howdy + howdy2
??
>>> message = howdy + '. ' + howdy2
>>> message
??
```

Output

- Function: piece of prewritten code that performs an operation
- `print` function: displays output on the screen
- Argument: data given to a function
  - Example: data that is printed to screen

```
print(baseArea)
print(13)
print(2 * 3 + 4)
print("baseArea")
print('2 * 3 + 4')
```

Exercise

- First evaluate on a piece of paper, then run in Python

```
print('hello', 'tcss', '142')
print('hello', 'tcss', '142', sep = ',')
print('hello', 'tcss', '142', sep = '\t')
print("hello 'tcss' 142")
print("hello "tcss" 142")
print("hello \"tcss\" 142")
print("hello tcss " + 142)
```

Example

- What would be the output of the following program?

```
# first verse
print("Bawitdaba", end = ' ' )
print("da bang a dang diggy diggy diggy")
print()

# second verse
print("said the boogy")
print("said up jump the boogy")
```

Exercise

- Write and save a program `output.py` to your H:drive. The program prints the following message – line-by-line

The “” represents an empty string.
Backslash character \ causes an "escape" from the ‘normal’ way characters are interpreted by the compiler.

And # represents comments.

Receipt Question

Download the receipt program from Canvas and improve it using variables.

```
# Calculate total owed, assuming 8% tax / 15% tip

print("Subtotal:", end=' ');
print(38 + 40 + 30);

print("Tax:", end=' ');
print((38 + 40 + 30) * .08);

print("Tip:", end=' ');
print((38 + 40 + 30) * .15);

print("Total:" , end=' ');
print(38 + 40 + 30 + (38 + 40 + 30) * .15 + (38 + 40 +
  30) * .08);
```

Receipt Answer

```
subtotal = 38 + 40 + 30
tax = subtotal * .08
tip = subtotal * .15
total = subtotal + tax + tip

print("Subtotal:", end =' ')
print(subtotal)

print("Tax: ", tax)

print("Tip: ", tip)

print("Total: " + str(total))
```

Last Slide ☺

- There is reading / quiz to be completed before the next class meeting
  - Read chapter 2 from your textbook and take the associated quiz
- There is reading / quiz to be completed within the next 2 weeks
  - Read chapter 1 from your textbook and take the associated quiz
- Class ends at 17:10

Number and field formatting

- Using built-in `format` function
  - Two arguments:
    - Numeric value to be formatted
    - Format specifier
  - Returns string containing formatted number
  - Format specifier typically includes precision and data type

- As a lab exercise

- Formatting does NOT change variable value – formats for display purposes only!!!