Foundations of Programming

Identifiers, variables, data and functions

Learning outcomes/key ideas

- Describe (and change) the type of data in Python
- Store data in variables using assignment statements
- Describe the difference between printing and returning values
- Define and call functions in Python

Nice reference: https://www.w3schools.com/python/default.asp

What is an identifier?

- The name of something.
- The name of a variable, a function or class.
- Composed of letters, numbers and/or underscores starting with a letter or underscore.
- Case matters: ABC is different than abc
- Name selection should give context to its use (VERY IMPORTANT)

What is a keyword?

- A word in the language that has a defined meaning.
- Can’t be an identifier.
- Case matters:
  - “if” is a keyword
  - “IF” is an identifier

What is a statement?

- One line of code
- Written in a computer language (Python)
- Has a purpose:
  - Ask the computer to give a value to a variable
  - Ask the computer to evaluate an expression

What is a variable?

- A name associated with a value
- A name of a storage location that holds a value
- An association of a name of a storage location that holds a certain type of data.
    - xyz = 123   # “int”eger variable xyz holding an initial value (initialization) of 123
    - abc = 2.3   # float variable abc holding an initial value of 2.3
    - is_it = True   # “bool”ean variable is_it holding an initial value of True
    - message = “Hello World”   # “str”ing variable holding an initial value of “Hello World”
- You can update its value:
    - xyz = xyz + 456   # assignment...xyz gets prior value plus 456

What is a Function?

- A series of statements grouped together as a unit.
- Grouping has a purpose
- Grouping is named for that purpose
- The function is “called” to execute those statements
- The function often produces a result that is “return”ed to the caller.
- The caller often captures the return result and stores into a variable.
- Sometimes input is needed for the Function to produce that result.
- Inputs are called: parameters (sometimes called arguments)
- Functions are sometimes called: methods
- Code inside the function is called “the function body”
- A function body defines a “scope”
- Code inside function can be reused without retyping the code.

Functions:

- Used as a way to organize execution of a related group of statements to act as a unit to perform a task.
- Function call performs that task.
- Function definition includes:
    - name, parameters, function body - code

```
def add (first, second):

    """ adds parameters and send back result """

    return first + second          # adds parameters and send back result
```

- Indenting defines a “scope”
- “first” and “second” are only known within the “add” function body

Functions:

```
def add (first, second):
        """ adds parameters and send back result """
        return first + second          # adds parameters and send back result

def add_print (first, second):
        """ adds parameters and prints result """
        print (first + second)         # adds parameters and prints result

result = add (1,2)                     # function call with assignment
# func call:  does above work?         A. Yes  B. No   C. Maybe     D. Sometimes
# What is displayed?                   A. 3     B.  0    C.  Nothing  D.  1 + 2    E.  Error
# What is value of result?             A.  3    B.  0     C.  None     D.  Other   E.  Error

result = add_print (1,2)               # function call with assignment
# func call:  does above work?         A. Yes  B. No   C. Maybe    D. Sometimes
# What is displayed?                   A. 3     B.  0    C.  Nothing  D.  1 + 2    E.  Error
# What is value of result?             A.  3    B.  0    C.  None      D.  Other   E.  Error
```

Strings

Any sequence of characters between "   " or ' '   # use double or single quotes

Strings are immutable (don’t change, are not variable).

`Strings` – examples

Which of these are Strings?

1. `" a "`
2. `"abc"`
3. `abc`
4. `"5"`
5. `""`
6. `"5 + 5"`
7. `5`

A: 1, 2, and 3 only
B: Everything but 3
C: Everything but 3 and 7
D: Everything but 4, 6, and 7
E: None of the above

What is an expression?

- A series of terms and operators that evaluates to a result.
- Can’t be a “LHS” (left hand side of assignment)
- Example:

```
variable = 1 + 3        # variable is LHS
    # and 1 + 3 is the expression
1 + 3 = variable        # does that work?
```

A. Yes   B. No

What can a “term” be?

- Literal: 1, 25, 1.3   (use that value, literally)
- Variable: xyz, abc
- Function call: some_function (1, 2)

# Python uses underscores in identifier names

Changing Types (aka “type casting”)

Ex: print (int (3.5))

A. 3.5 B. 3 C. 0 C. Other D. Error

Ex: print (float (10))

A. 10 B. 10.0 C. 0.0 C. Other D. Error

Ex: print (str (5) + str (5))

A. 10 B. 5+5 C. 55 C. Other D. Error

```
#  Alternatively:  Assign to a variable without printing
        var = int (3.5)
        var2 = float (10)
        var3 = str(5) + str (5)
```