**CRUCIAL PYTHON CONCEPTS NEEDED FOR THE ENTIRE COURSE**

Print and **keep handy** for reference.

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCEPT** | **PAGES** | **WHAT IT IS OR WHY YOU SHOULD KNOW ABOUT IT** |  |
| IDLE | 23, 803 | Integrated Development and Learning Environment, the program that comes with Python.  Its **script** window is used for writing programs.  Its **interactive** window is for testing Python statements and displaying program output. |  |
| Program | various | A series of instructions inside a computer that perform some task.  Program operations include data **input**, **processing** and **output**.  Python programs contain statements using language keywords and functions.  You will learn the keywords and many functions as the course progresses. |  |
| Data | various | The **items of input** a program needs to store in **memory** and process to do its job.  A Human Resources program’s data would include names, jobs and pay rates.  A cashier program’s data would include item descriptions, prices and quantities. |  |
| Variable | 40-48 | An **individual, named item** of data stored in a program.  Must be **named** properly so it can be used in the program.  See pages 43-44 for the **rules for naming** Python variables. |  |
| Function | various | A function is a named block of code that accomplishes some task.  Python has many built-in functions such as…  The **print()** function: used for displaying output on the screen. The i**nput()** function: enables the user to **input data from the keyboard**. Later, you will create custom functions to perform specified tasks. |  |
| String | 37-38, 48, 65-67 | A **non-numeric data type** that cannot be used for math purposes.  Used for names, addresses, titles etc.  Must be enclosed **within quotes** and will be colored **green** in IDLE.  Quotes can be single (using apostrophes), regular double quotes, or even triple quotes.  Strings can be joined (**concatenated**) with **+ sign** to make a larger string. |  |
| int | 47,51 | A numeric data type for **integers** (numbers without decimals). |  |
| float | 47,51 | A numeric data type for numbers with **decimals**. |  |
| Arithmetic operators | 54-61 | / is **division**  12 / 5 = 2.4  // is **integer division**  12 // 5 = 2  % gives **remainder** after division  25 % 7 = 4  \*\* is for **exponents**  2 \*\* 4 = 7 |  |
| Comments | 39-40 | Entries in a program that begin with a #. They explain or **document** what code is doing, but **don’t execute**.  They are colored **red** in IDLE. |  |
| Variable  Assignment | 41-48, 49-53 | General Form: variable **=** value, expression, or function **that returns a value**. See input() function below for variable assignment from the keyboard. |  |
| print() function | 36-38, 67-70 | Outputs the **argument(s)** inside parentheses to the screen, then generates a **newline.** The newline can be altered with the **end** operator (see below).  A **comma** separator between arguments outputs a **space** between them, but this can be changed with **sep** (see next). |  |
| sep = ‘string’ | 68-69 | Cancels the **space** a comma generates between print items, replacing it with the specified string. With an empty string, the space is deleted. |  |
| end = ‘string’ | 67-68 | Cancels a print function’s **newline**, replacing it with the specified string. The output from a following print() will be on the **same** line. |  |
| input(‘string’) function |  | For user entry of a data item **from the keyboard**. A “prompt string” is specified inside the ( ) so the user knows what to enter.  **var = input(‘prompt string’)** # what user enters is **assigned to var variable** as a **string**. The input() function by itself **returns a string**.  **Embed** the input() function inside an **int()** function to **return an int**.  **Embed** the input() function inside a **float()** function to **return a float**. |  |
| int() function |  | Converts a string variable containing an integer into an **int** variable. |  |
| float() function |  | Converts a string variable containing a number into a **float** variable. |  |
| f-strings | 70-78 | Enables total control of output with variables or expressions inside **{ } placeholders**.  { } can contain optional **designators** to **format** and **align** output.  **Errattum** bottom of page 71: should be {**variable or expression** : format-specifier}. **{ }** is the placeholder.  See **page 78** for the **order** of designators inside placeholders.  NOTE: you may find other ways to format output online, but **this course requires f-strings**. |  |
| Constants | 80 | A constant is an **assigned** value (numeric or string) that **does not change**.  Use constant names in **all upper case** with **underscores between words**, if applicable.  Examples:  SALES\_TAX\_RATE = 0.07 COLLEGE = “SPC” |  |