

# Lessons 06 & 07

## Asg 6.1

### The Absolute Basics

&

### Types of Operators, Comments in Python, Formatting Print Statements

#### Question #1:: Short Answers:

- a. Instructions written in source code are called \_\_\_\_\_.
- b. The preferred indentation method in Python is \_\_\_\_\_.
- c. There are three special types of comments in Python. These are \_\_\_\_\_ , \_\_\_\_\_ , and \_\_\_\_\_.
- d. Is it more accurate to say that variables in Python are containers or labels?
- e. In Python, a combination of operators and operands (variables and values) that represent a single result value is called a/an \_\_\_\_\_.
- f. True or False: (If False, explain why the statement is False.) *Arithmetic operators involving integers always return an integer.*
- g. True or False: (If False, explain why the statement is False.) *In traditional integer division the result gets rounded.*

- h. What Python basic data type defines a single special data object that is used to represent an empty value?
- i. The result of the `input()` function always outputs as a \_\_\_\_\_. So if you want to use that output as a number in a calculation, you must first do a \_\_\_\_\_.
- j. What are the different components of a conversion specifier in a formatted print statement?
- k. What is another way that you could write code that would accomplish the same thing as the statement `x = x + y`?
- l. If you place the `>` formatting type inside placeholders in a formatted print statement what will be the result?
- m. If you place the `<` formatting type inside placeholders in a formatted print statement what will be the result?
- n. What do the `\n` characters do?
- o. What do the `\t` characters do?

## Answers for Question #1

- a. Statements
- b. 4 Spaces
- c. Block, inline, docstrings
- d. Label
- e. Expression
- f. False, dividing an integer by an integer returns a float
- g. False. The answer gets truncated
- h. None
- i. string, type conversion
- j. %,Flags, width, precision, type
- k. `x+=y`
- l. right aligns the text
- m. left aligns the text
- n. Creates a new line
- o. Adds a tab

**Question #2:** Found below is the first half this week's AP Top 25 college football teams.

**Instructions:**

Write code that uses a **\*\*multi-line statement\*\*** to display the top 12 teams in this week's rankings. ![Rankings-2.JPG](attachment:Rankings-2.JPG)

```
In [6]: teamNames = ["Alabama", "Georgia", "Oklahoma", "Oregon", "Iowa", "Clemson", "Texas A&M",
print(f"The top 12 teams of the week 1 are:\n\n")
counter = 0
for team in teamNames:
    print(f"The number {counter+1} ranked team of week 1 is: {teamNames[counter]}\n")
    counter +=1
```

The top 12 teams of the week 1 are:

The number 1 ranked team of week 1 is: Alabama

The number 2 ranked team of week 1 is: Georgia

The number 3 ranked team of week 1 is: Oklahoma

The number 4 ranked team of week 1 is: Oregon

The number 5 ranked team of week 1 is: Iowa

The number 6 ranked team of week 1 is: Clemson

The number 7 ranked team of week 1 is: Texas A&M

The number 8 ranked team of week 1 is: Cincinnati

The number 9 ranked team of week 1 is: Ohio State

The number 10 ranked team of week 1 is: Penn State

The number 11 ranked team of week 1 is: Florida

The number 12 ranked team of week 1 is: Notre Dame

**Question #3: Instructions:**

Use %-Formatting to print out the statement found below.

To do this, create the following variables:

newspaper, abbreviation, world\_circulation, and usa\_circulation.

Then use those variables in a tuple to insert their values into the formatted string.

***The New York Times (abbreviated N.Y.T.) is ranked 18th in the world by circulation and 3rd in the U.S.***

```
In [7]: newspaper = "The New York Times"
        abbreviation = "N.Y.T."
        world_circulation = "18th"
        usa_circulation = "3rd"

        print("%s (abbreviated %s) is ranked %s in the world by circulation and %s in the U.S.")
```

The New York Times (abbreviated N.Y.T.) is ranked 18th in the world by circulation and 3rd in the U.S.

#### Question #4: Instructions

Use %-Formatting along with the conversion types d, s, and f to print out the statement found below.

***10 MLB baseball players have batting averages higher than 0.325.***

Note: Use the s conversion type in conjunction with MLB.

```
In [8]: print("%d %s baseball players have batting averages higher than %.3f." %(10, "MLB", 0.325))

10 MLB baseball players have batting averages higher than 0.325.
```

**Question #5: Instructions:** For Parts a - c: Use the str.format() approach to print out the statements found in the table below. Also, make sure to follow the other guidelines that are specified in the table. ![car%20purchases.PNG](attachment:car%20purchases.PNG)

```
In [9]: # Use this cell for Part a
        name = "Sally"
        car = "2021 red Honda Civic"
        price = "$21700"
        print("{} bought a {} for {}".format(name, car, price))
```

Sally bought a 2021 red Honda Civic for \$21700

```
In [10]: # Use this cell for Part b
        person = {'name': 'Ed', 'car': '2020 blue Ford Focus', 'price': '$17980'}
        print("{name} bought a {car} for {price}.".format(name=person['name'], car=person['car'], price=person['price']))
```

Ed bought a 2020 blue Ford Focus for \$17980.

```
In [11]: # Use this cell for Part c
        name = "Ann"
        car = "2019 green Toyota Corolla"
```

```
price = "$187210"
print("{} bought a {} for {}".format(name, car, price))
```

Ann bought a 2019 green Toyota Corolla for \$187210

**Question #6::** Step 1: Do a search to find at least 3 advantages of (or reasons for) including comments within a program Step 2: Use two different methods for adding multiline comments within a code cell to list the advantages of (or reasons for) including comments within a program that you found in Step 1 above. Make sure to name the method right after the 3 dots in the top line of the code cell provided below.

```
In [12]: # Method #1 for Adding Multiline Comments using print function and \n
print(" 1. Commented code is more readable to an unknown person reading your code for the first time.
```

```
2. Commenting helps find bugs faster.
3. By commenting, you can write notes to yourself about the previous day of coding.
```

```
In [13]: # Method #2 for Adding Multiline Comments using f string formatting
print(f' 1. Commented code is more readable to an unknown person reading your code for the first time.
      f' 2. Commenting helps find bugs faster.\n'
      f' 3. By commenting, you can write notes to yourself about the previous day of coding.
```

```
1. Commented code is more readable to an unknown person reading your code for the first time.
2. Commenting helps find bugs faster.
3. By commenting, you can write notes to yourself about the previous day of coding.
```

**Question #7::** Use the f-strings approach to write code that will print out (exactly as they appear), the two statements found below:

1) *In her pregame speech, Wisconsin's coach said "The #1 thing is that we have to play really hard in tonight's game".* 2) *In his halftime speech, Northwestern's coach said "We've overcome more adversity than this in the past!"*

Include variable assignments for the following variables in your code:

gender\_pronoun, speech\_type, university, and sport

```
In [14]: gender_pronoun = "her"
speech_type = "pregame"
university = "Wisconsin"
print(f"In {gender_pronoun} {speech_type}, {university}'s coach said \"The #1 thing is that we have to play really hard in tonight's game\"")
```

In her pregame, Wisconsin's coach said "The #1 thing is that we have to play really hard in tonight's game"

```
In [15]: # Use this cell for statement 2
gender_pronoun = "his"
```

```
speech_type = "halftime"
university = "Northwestern"
print(f"In {gender_pronoun} speech, {university}'s coach said \"We've overcome more adv
```

In his speech, Northwestern's coach said "We've overcome more adversity than this in the past"

**Question #8:** Use the f-strings approach to write code that will print out the statement found below:

*Mr. Cordell said, "The solution set to Question #6 is  $\{x | x > 5\}$ ."*

```
In [16]: questionNum = "6"
equation = "{x|x > 5}"
teacherName = "Mr.Cordell"
print(f"{teacherName} said, \"The solution set to Question #{questionNum} is {equation}
```

Mr.Cordell said, "The solution set to Question #6 is  $\{x|x > 5\}$ ."

**Question #9:** Use a multiline f-string to print out the information found in the table below. Make sure that all of the output is arranged in very orderly columns and that decimals are formatted properly. (Hint: refer back to the information on conversion specifiers and conversion types to help you with this.)

!exc-2.PNG(attachment:exc-2.PNG)

```
In [17]: print("student      GPA      course      block      teacher\n ")
print(f'John Smith    3.817  Chemistry    1          Harder\n'
      'Jane Doe      3.982  Calculus     2          Vraney\n'
      'Sally Jones    3.62   AP US History 3          Coffey\n'
      'Ed Baker       2.98   Spanish      4          Lentz\n'
      )
```

student	GPA	course	block	teacher
John Smith	3.817	Chemistry	1	Harder
Jane Doe	3.982	Calculus	2	Vraney
Sally Jones	3.62	AP US History	3	Coffey
Ed Baker	2.98	Spanish	4	Lentz

**Question #10:** Step #1: Write code that asks the user for their dog's name and then asks the user for their dog's age in years.

Step #2: Write a formatted print statement to print the dog's name and age in dog years (to the nearest tenth).

You will need to use the formula below to calculate the age of the dog in dog years.

```
dog_years = (10.5 * 2) + (4 * (age - 2))
```

Run your code for a dog named Charlie that is 10.5 years old. If you have written the proper code, you should get the following output:

```
What is your dog's name? charlie
How old (in years) is your dog? 10.5
Charlie is 55.0 years old in dog years.
```

```
In [18]: dog_name = input("What is your dog's name? ")
capital_first_name = dog_name.capitalize()
dog_age = float(input("How old (in years) is your dog? "))
dog_years = (10.5 * 2) + (4 * (dog_age - 2))

print(f"{capital_first_name} is {dog_years} old in dog years.")
```

```
What is your dog's name? charlie
How old (in years) is your dog? 10.5
Charlie is 55.0 old in dog years.
```