

Youtube video 1: Best of Psych Series

Friday, November 12, 2021 12:00 PM

Shot 1:

- video screen: Quick scroll through of Google Colab document for Psych Python intro
- Voice:
 - Introduction from the Brock University Digital Scholarship lab on using the MBTI 16 personality theory to learn how to create variables, conditionals, math, and loops in Python.
 - Remind viewers that they can pause the video or hit the j k to rewind 10 seconds if the video is too fast paced.

Shot 2:

- video screen: scrolling to the 'molecules' image that shows the cognitive functions, and scrolling down to the de
- scriptions image of the types
- Voice: background information on the 16 types

Shot 3:

- Video screen: navigating to a blank Google colab document
- Voice: describing how viewers can follow along with the video by launching a blank google colab document.

Shot 4:

- Video screen: title - colorful title slide with the word variables written in a large font
- Voice: none

Shot 5:

- Video screen: code cell that includes variables examples
 -

▼ Examples of Variables using MBTI 16 personalities

```
[2] 1 #Example variable 1
    2 mbti_type_I_think_you_are = "estj"
    3
    4 #Example variable 2
    5 mbti_type_you_think_you_are = input("what is your type? ")
    6
    7 #Example variable 3 - List
    8 mbti_types = ["isfp", "istp", "infp", "infj", "intj", "intp", "entp", "esfj", "isfj", "estj", "estp", "entj", "enfj", "esfp", "enfp", "istj"]
    9
   10 #Example variable 4 - Integer
   11 number_of_mbti = 16
   12
   13 #Example variable 4 - Real number
   14 infj_prevalence_decimal_value = 0.9
```

what is your type? infj

- Voice: explanation of each variable, and hit run, and the fake participant types in their type

Shot 6:

- Video screen: title screen: Using some pre-made functions
- Voice: none

Shot 7:

- Video screen:

▼ Examples of using a pre-made function: print()

```
1 #Example of the print function so that we can see the variables we just created
2 print(mbti_type_you_think_you_are)
3
4 #Example of the print function 2
5 print("Hello " + mbti_type_you_think_you_are + "!")
6
7 #Example of the print function 3
8 print("There are" , number_of_mbti , " personalities")
```

infj
Hello infj!
There are 16 personalities

- Voice: description of the print function and input function

Shot 8:

- Video screen: title screen: Using some Conditionals
- Voice: None

Shot 9:

- Video screen:

Examples of Conditionals

```
1 if mbti_type_I_think_you_are == "estj" and mbti_type_you_think_you_are == "estj":
2     print("I guessed your type right!")
3 else:
4     print("I guessed your type wrong!")
```

I guessed your type wrong!

- Voice: brief description of conditionals purpose, 'if' 'else', '==' sign.

Shot 10:

- Video screen: Review of List variable, intro to Dictionary variable
- Voice: none

Shot 11:

- Video screen:

Review of List variable, and intro dictionary variable

```
1 #Review of what a list variable looks like
2 mbti_types = ["isfp", "istp", "infp", "infj", "intj", "intp", "entp", "esfj", "isfj", "estj", "estp", "entj", "enfj", "esfp"]
3
4 #Now, an example of what a dictionary variable looks like
5 mbti_descriptions = {"isfp": "fi, se, ni, te",
6                      "istp": "ti, se, ni, fe",
7                      "infp": "fi, ne, si, te",
8                      "infj": "ni, fe, ti, se",
9                      "intj": "ni, te, fi, se",
10                     "intp": "ti, ne, si, fe",
11                     "entp": "ne, ti, fe, si",
12                     "esfj": "fe, si, ne, ti",
13                     "isfj": "si, fe, ti, ne",
14                     "estj": "te, si, ne, fi",
15                     "estp": "se, ti, fe, ni",
16                     "entj": "te, ni, se, fi",
17                     "enfj": "fe, ni, se, ti",
18                     "esfp": "se, fi, te, ni",
19                     "enfp": "ne, fi, te, si",
20                     "istj": "si, te, fi, ne"}
21
22 #A second useful example of a dictionary
23 mbti_percentage_population = {"isfp": 9,
24                              "istp": 5,
25                              "infp": 4,
26                              "infj": 2,
27                              "intj": 2,
28                              "intp": 3,
29                              "entp": 3,
30                              "esfj": 12,
31                              "isfj": 14,
32                              "estj": 9,
33                              "estp": 4,
34                              "entj": 2,
35                              "enfj": 3,
36                              "esfp": 9,
37                              "enfp": 8,
38                              "istj": 12}
```

- Voice: explanation of the difference between lists and dictionaries, and the uses of dictionaries.

Shot 12:

- Video screen: title: printing the mbti description dictionary to the screen
- Voice: none

Shot 13:

- Video screen:

- We can print a dictionary in a very similar way to other variables like we saw earlier.

```
[ ] 1 print(mbti_descriptions)
```

- You can do some interesting things with dictionaries, (we'll see more as time goes on)
- For example you can print out different parts of the dictionary by adding in a key

○ [] 1 print(mbti_descriptions["isfp"])

- Voice: explanation of how to print a whole dictionary and part of a dictionary to the screen.

Shot 14:

- Video screen: title – printing the mbti percentage population dictionary to the screen
- Voice: none

Shot 15:

- Video screen:

○

✓ [48] 1 print(mbti_percentage_population)

```
{'isfp': 9, 'istp': 5, 'infp': 4, 'infj': 2, 'intp': 3, 'entp': 3, 'esfj': 12, 'isfj': 14, 'estj': 9, 'estp': 4, 'entj':
```

○

✓ 1 print(mbti_percentage_population["isfp"])

9

- Voice: another example of printing a dictionary to the screen

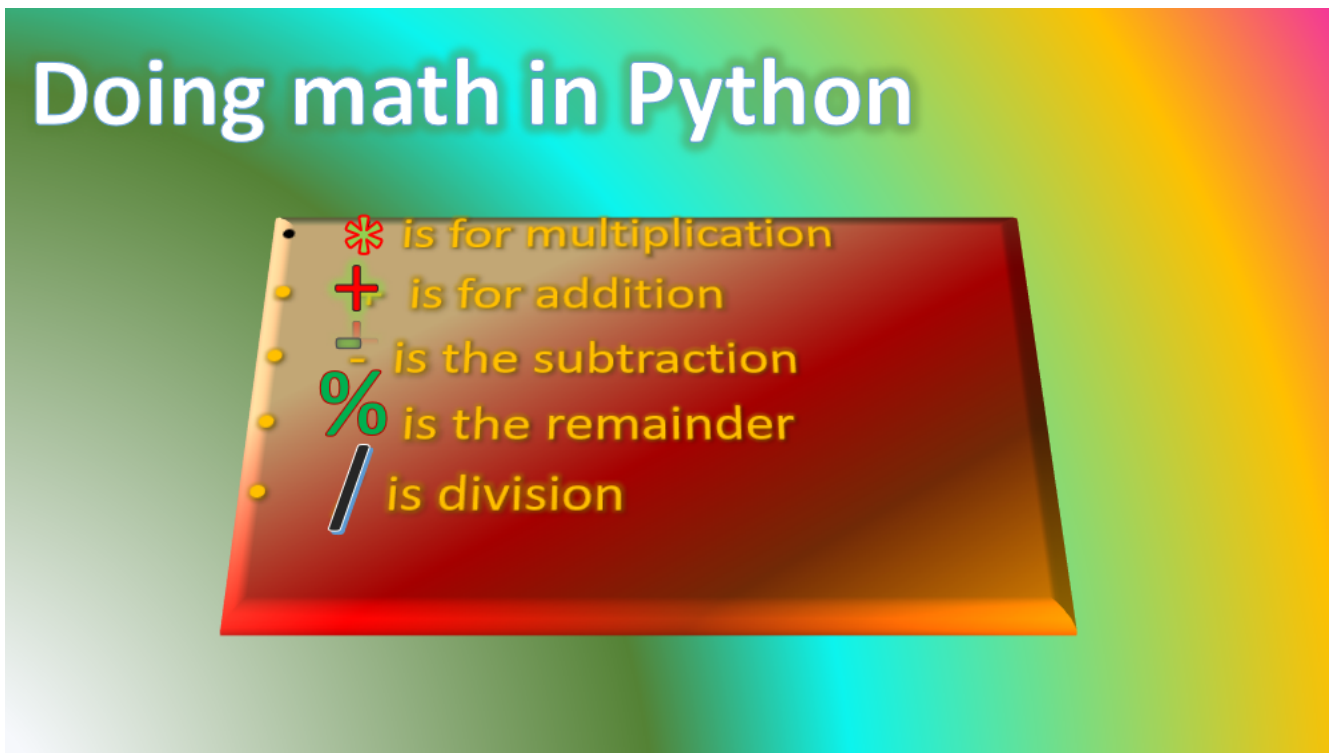
Shot 16:

- Video screen: title – Reasons to do Math in Python
- Voice: none

Shot 17:

- Video screen:

○



- Voice: Explanation of all the math symbols

Shot 18:

- Video screen:

○

▼ Doing Math in Python

```
1 #Let's find the total percentage of sensors and intuitives in the population
2 intuitives = mbti_percentage_population['infp'] + \
3             mbti_percentage_population['infj'] + \
4             mbti_percentage_population['entj'] + \
5             mbti_percentage_population['intj'] + \
6             mbti_percentage_population['enfj'] + \
7             mbti_percentage_population['enfp'] + \
8             mbti_percentage_population['entp'] + \
9             mbti_percentage_population['intp']
```

```

10
11 print("intuitives:")
12 print(intuitives)
13
14 sensors = 100 - intuitives
15
16 print("sensors:")
17 print(sensors)
18 print()
19 #Let's find the total percentage of judges (J) and Perceivers (P) in the population
20 judges = mbti_percentage_population["isfj"] + \
21         mbti_percentage_population["infj"] + \
22         mbti_percentage_population["entj"] + \
23         mbti_percentage_population["intj"] + \
24         mbti_percentage_population["enfj"] + \
25         mbti_percentage_population["esfj"] + \
26         mbti_percentage_population["estj"] + \
27         mbti_percentage_population["istj"]
28
29 print("judgers:")
30 print(judgers)
31
32 perceivers = 100 - judges
33
34 print("perceivers:")
35 print(perceivers)

```

- Voice
 - Examples of using addition, subtraction, the forward slash, creating new variables, and using the print function to skip a blank line (in line 18)

Shot 19:

- Video screen:

```

1 #Example of using the division symbol: the average percentage of a sensor and intuitive found in the population
2 avgintuitive = intuitives / 8
3 print(avgintuitive)
4 avgsensor = sensors / 8
5 print(avgsensor)
6
7 #Turning them back into their total percentage of the population, using multiplication
8 print()
9 print(avgintuitive * 8)
10 print(avgsensor * 8)

```

3.375
9.125

27.0
73.0

- Voice: example of using division and multiplication symbols

Shot 20:

Video screen: Loops
Voice: none

Shot 21:

- Video screen:
 - Printing items to the screen using Loops

```

1 #Review of one way to print out the contents a list
2 mbti_types = ['isfp', 'istp', 'infp', 'infj', 'intj', 'intp', 'entp', 'esfp', 'isfj', 'estj', 'estp', 'entj', 'enfj', 'esfp', 'enfp', 'istj']
3
4 print(mbti_types)
5 #Create a space between the two versions using an empty print function
6 print()
7 #What happens when you print the contents of a list using a loop
8 for personality in mbti_types:
9     print(personality)

```

['isfp', 'istp', 'infp', 'infj', 'intj', 'intp', 'entp', 'esfp', 'isfj', 'estj', 'estp', 'entj', 'enfj', 'esfp', 'enfp', 'istj']

isfp
istp
infp
infj
intj
intp
entp
esfp
isfj
estj
estp
entj
enfj
esfp
enfp
istj

- Voice: Comparing printing lists out to the screen using just the print function, and then adding a loop statement so that it prints the items one below the other and without punctuation.

Shot 22:

- Video screen: Counting with Loops
- Voice: none

Shot 23

- Video screen:
 -

○

- Voice: Describing how you can use a loop to count items in a list

Shot 24

- Video Screen:

○

▼ Dictionaries and Loops

- Watch what happens when we try to do a loop through a dictionary

```

1 for personality in mbti_percentage_population:
2     print(personality)

isfp
istp
infp
infj
intj
intp
entp
esfj
isfj
estj
estp
entj
enfj
esfp
enfp
istj

```

- Since dictionaries are more complex, there are many ways to interact with them
- We add .values() to get the values of the dictionary
- We add .keys() to get the keys of the dictionary
- Voice: Description of how we need to add .values() or .keys() in order to get what we want from the dictionary during a loop.

Shot 25:

- Video screen:
 - Since dictionaries are more complex, there are many ways to interact with them
 - We add .values() to get the values of the dictionary
 - We add .keys() to get the keys of the dictionary

```

[8] 1 #print all the items in our Dictionary
    2
    3 for percentage in mbti_percentage_population.values():
    4     print(percentage)

```

```

1 # print out the keys in our Dictionary
2
3 for name in mbti_percentage_population.keys():
4     print(name)

```

- Voice: description of how to either print out the values of a dictionary or keys using a loop.

Shot 26:

- Video screen: title - More uses of conditionals
- Voice: none

Shot 26

- Video screen:

▼ Conditionals

- When we want to run a piece of code only if something is true
- We can compare with the following:

```

==
>=
<=
<

```

```
>
!=
```

- Voice: We briefly saw how to use the '==' sign, but other signs include the ones here (provide a description of each)

Shot 27

- Video screen
 - Let's pretend that anything less than 4 percent of the population is rare. Let's make a conditional that prints out if an **intp** is rare

```
1 if mbti_percentage_population["intp"] < 4:
2
3     print("Rare Personality Type")
4 else:
5     print("Common Personality")
```

Rare Personality Type

- Voice: Description of how one could apply the '<' symbol to our dictionary

Shot 28

- Video screen – title - Combining boolean to create a mini mbti quiz
- Voice – none

Shot 29

- Video screen
 - Combining with boolean
 - Sometimes we want to combine conditionals because we want to test more complicated things
 - We can do this with the following:
 - and
 - or
 - not
- Voice: description of the different Booleans

Shot 30

- Video screen

```
[ ] 1 Question_1 = input("Type a, b, or c, to answer the following question: I try not to draw attention to myself. a-strongly agree, b-neutral, c-strongly disagree")
2
3 Question_2 = input("Type a, b, or c, to answer the following question: I have a vivid imagination. a-strongly agree, b-neutral, c-strongly disagree")
4
5 if Question_1 == "a" and Question_2 == "a":
6     print("Because you answered 'a' for questions 1 and 2,")
7     print("you're either an infp or intp!")
8 if Question_1 == "a" and Question_2 == "b":
9     print("Because you answered 'a' for Q1 and 'b' for Q2,")
10    print("you're either an isfp or an istp!")
11 # There aren't any possibilities for Q1 = a and Q2 = c in this mini test
12 # There aren't any possibilities for Q1 = b and Q2 = b in this mini test
13 if Question_1 == "b" and Question_2 == "a":
14     print("Because you answered 'b' for Q1 and 'a' for Q2,")
15     print("you're either an infj or intj!")
16 if Question_1 == "b" and Question_2 == "c":
17     print("Because you answered 'b' for Q1 and 'c' for Q2,")
18     print("you're either an isfj or istj!")
19 if Question_1 == "c" and Question_2 == "a":
20     print("Because you answered 'c' for Q1 and 'a' for Q2,")
21     print("you're either an entj, enfj, enfj, or an entp!")
22 if Question_1 == "c" and Question_2 == "b":
23     print("Because you answered 'c' for Q1 and 'b' for Q2,")
24     print("you're either an esfj or estj!")
25 if Question_1 == "c" and Question_2 == "c":
26     print("Because you answered 'c' for Q1 and 'c' for Q2,")
27     print("you're either an esfp or estp!")
```

- Voice: description of how combining booleans, print functions, conditionals, variables, and input function can create the mbti quiz.

Shot 31

- Video screen: title – Creating your own functions
- Voice: none

Shot 32

- Voice: describing how to make the following function
- Video screen:
 - The function below will change the values to decimals from percentage
 - Statements can get pretty long if you call lots of functions in them
 - Here we find the isfp's percentage converted to decimal and print it

```
1 def perc_to_dec(perc):
2     return perc / 100
3
4 print(perc_to_dec(mbti_percentage_population["isfp"]))
```