Exercises - Part 3

October 19, 2020

0.0.1 Comparison Operators

```
a = 20
b = 30
print(a > b)
```

Exercises:

1. Assign the variables `a = 20` and `b = 30`
2. Print if `a` is equal to `b`
3. Print if `a` is not equal to `b`
4. Print if `a` is larger than `b`
5. Print if `a` is smaller than `b`
6. Print if `a` is larger than or equal to `b`
7. Print if `a` is smaller than or equal to `b`

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

[ ]:
```
# Exercise 3
```

[ ]:
```
# Exercise 4
```

[ ]:
```
# Exercise 5
```

[ ]:
```
# Exercise 6
```

[ ]:
```
# Exercise 7
```

0.0.2 If / Else

```
# Example
if temperature > 85:
    print("It is hot")
else:
    print("It is nice")
```

Exercises:

1. Assign the variable `score = int(input("What was your (percentage) score? "))`. Enter 80 in the prompt when you run the cell.
2. A passing grade is 70% and above. Write a program with a `if` and `else` statement to check if a student has passed. If so, it should print `"Passed"`, else it should print `"Failed"`
3. Write a program that uses if/else statements to convert a percentage score to a letter grade (see the table above). You will need multiple `elif` statements to include all the options. The output should be a print statement: `Your score of 99 corresponds with the letter grade A`
4. Test if the program works for 95%, 72% and 60%

Percentage	Letter Grade
90%–100%	A
80%–89%	B
70%–79%	C
60%–69%	D
< 60%	F

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

[ ]:
```
# Exercise 3
```

[ ]:
```
# Exercise 4
```

0.0.3 Logical and membership operators

Tickets for the theme park "RollerCoasterLand" normally cost $25. They also have promotions: - Free for children under 13 - Free for elderly of 65 years and older

Exercise: 1. Create a program to calculate and print the price of your ticket. 2. Test your program for various user inputs and check if it works correctly

Extra challenge: Add another promotion: 40% discount for "season pass" holders. Create a new variable `is_season_pass_holder` and add logic to your program.

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

[ ]:
```
# Extra challenge
```

0.0.4 For Loops

```
# Example
ingredients = ["egg", "bacon", "sausage", "spam"]
```

```
for item in ingredients:
    print(item)
```

1. Assign the variable `planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]`
2. Create a for loop to print every planet to the screen (one by one, on seperate lines)
3. We are only interested in *other* planets. So create a for loop again that prints all the planets, except for "Earth" (hint: use a `if` statement and the `continue` statement)

Extra challenge: create a loop that prints all the planets, but now with their respective place number in the list. So the expected output is:

```
1 Mercury
2 Venus
3 Earth
...etc
```

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

[ ]:
```
# Exercise 3
```

[ ]:
```
# Extra challenge
```

0.0.5 While Loops

```
# Example
x = 0
while x < 5:
    print(x)
    x = x + 1
```

Exercise: 1. Create a `while` loop that prints `even` numbers from 0 to 10 (so 0, 2, 4, 6, etc.) 2. Create a `while` loop that prints `even` numbers to screen, but stops (`breaks`) after 4

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

0.0.6 Range

```
# Example range
for n in range(5):
    print(n)
```

Exercises: 1. Create a loop with the `range()` function that prints the numbers from 0 to 10 2. Create a loop with the `range()` function that prints the `even` numbers from 10 to 20

Documentation: https://docs.python.org/3/library/stdtypes.html#range

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

0.0.7 Enumerate

```
# Example
ingredients = ["egg", "bacon", "sausage", "spam"]

for i, item in enumerate(ingredients, start=0):
    print(i, item)
```

Exercises: 1. Assign the variable `databases = ["MySQL", "PostgreSQL", "MongoDB", "Redis"]` 2. Use the `enumerate()` function to print the databases with their respective position in the list.

```
# Desired output
1 MySQL
2 PostgreSQL
3 MongoDB
4 Redis
```

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

0.0.8 Loop over Dictionaries

```
# Example
grades = {"French": 75, "English": 95, "Math": 100, "Biology": 80}
for key, value in grades.items():
    print(key, value)
```

Exercise: 1. Assign the variable `person = {"name": "Alice", "age": 42, "email": "alice@gmail.com", "country": "U.K."}` 2. Create a loop using the `.items()` method to print all the key/value pairs. The output should look as follows:

```
# Desired output
name: Alice
age: 42
email: alice@gmail.com
country: U.K.
```

Extra challenge: Search the internet on how to print the items (justified) with padding, so it looks a little nicer:

```
# Desired output
name      : Alice
age       : 42
email     : alice@gmail.com
country   : U.K.
```

[ ]:
```
# Exercise 1
```

[ ]:
```
# Exercise 2
```

[ ]:
```
# Extra challenge
```