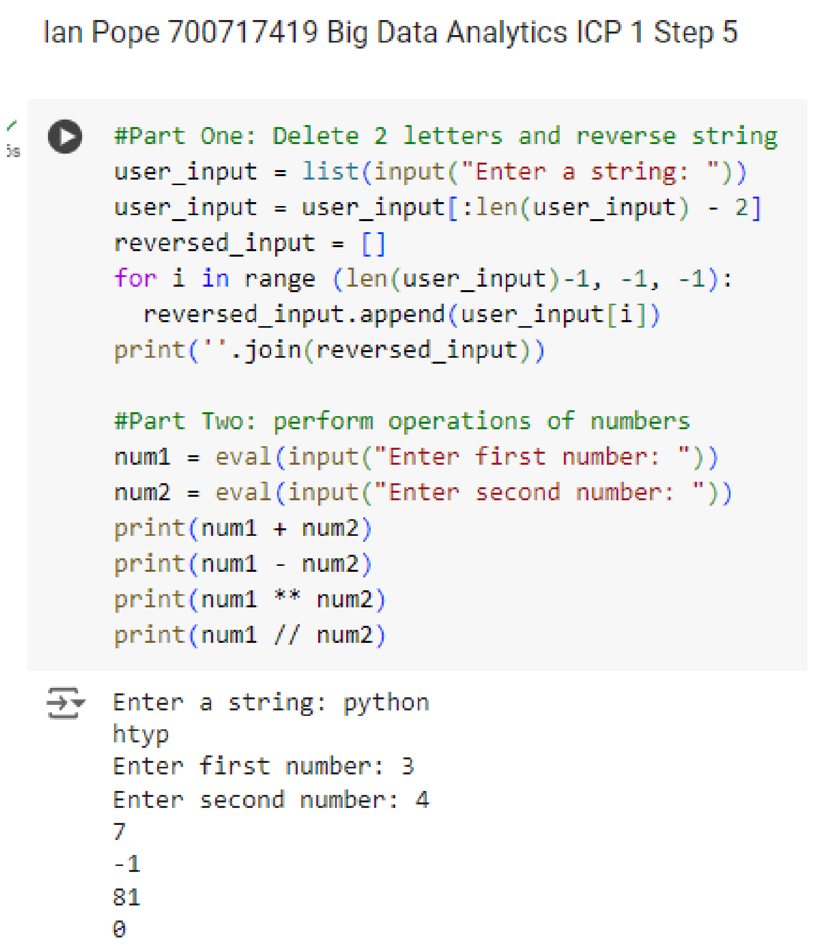
Ian Pope, 700717419, DSA 5620, Big Data Analytics, ICP 1

YouTube link: https://youtu.be/DwyM7uOS0iU

Part 5:



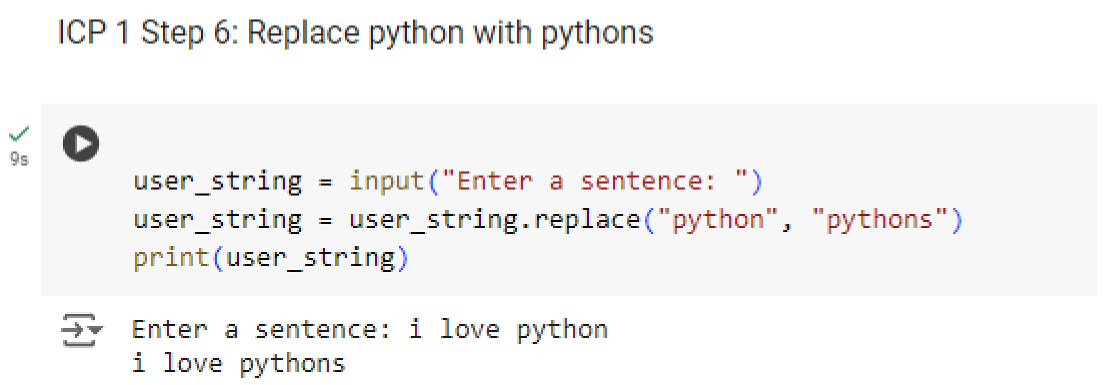
Ian Pope 700717419 Big Data Analytics ICP 1 Step 5

```
#Part One: Delete 2 letters and reverse string
user_input = list(input("Enter a string: "))
user_input = user_input[:len(user_input) - 2]
reversed_input = []
for i in range (len(user_input)-1, -1, -1):
  reversed_input.append(user_input[i])
print(''.join(reversed_input))

#Part Two: perform operations of numbers
num1 = eval(input("Enter first number: "))
num2 = eval(input("Enter second number: "))
print(num1 + num2)
print(num1 - num2)
print(num1 ** num2)
print(num1 // num2)
```

```
Enter a string: python
htyp
Enter first number: 3
Enter second number: 4
7
-1
81
0
```

Part 6:

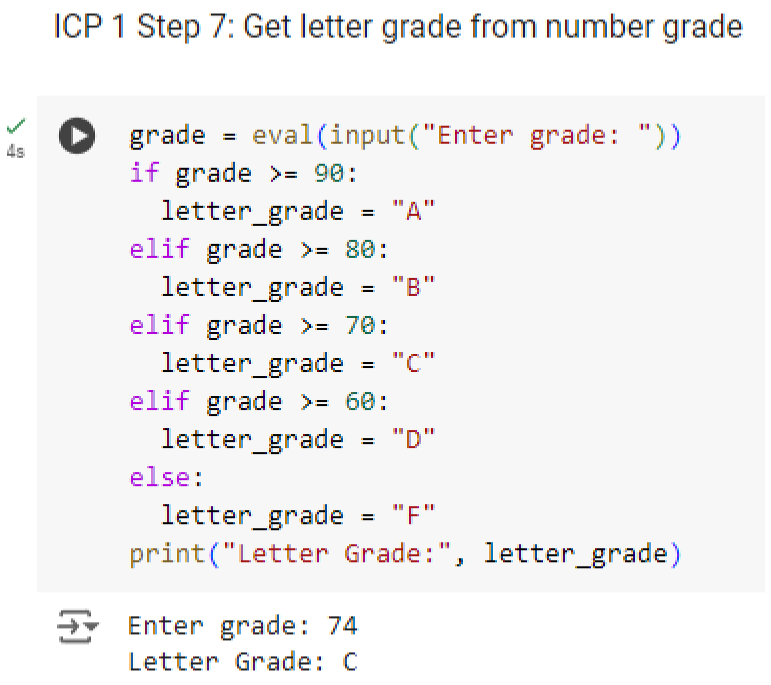


ICP 1 Step 6: Replace python with pythons

```
user_string = input("Enter a sentence: ")
user_string = user_string.replace("python", "pythons")
print(user_string)
```

```
Enter a sentence: i love python
i love pythons
```

Part 7:

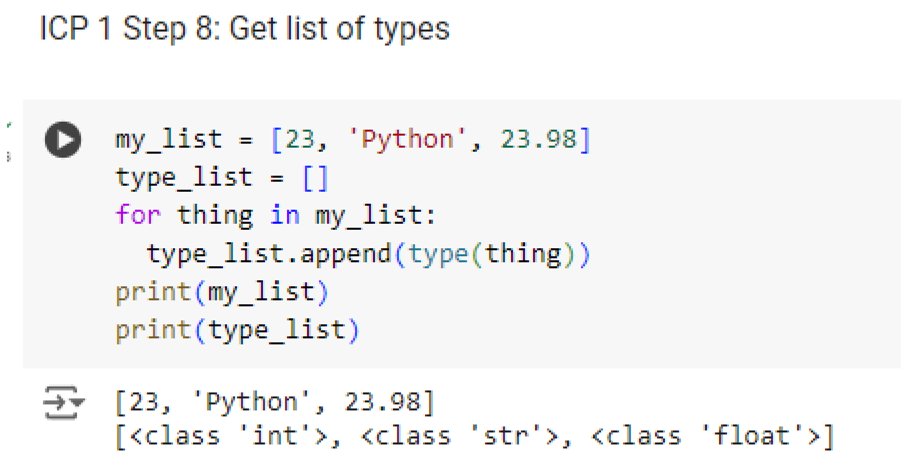


ICP 1 Step 7: Get letter grade from number grade

```
grade = eval(input("Enter grade: "))
if grade >= 90:
  letter_grade = "A"
elif grade >= 80:
  letter_grade = "B"
elif grade >= 70:
  letter_grade = "C"
elif grade >= 60:
  letter_grade = "D"
else:
  letter_grade = "F"
print("Letter Grade:", letter_grade)
```

```
Enter grade: 74
Letter Grade: C
```

Part 8:



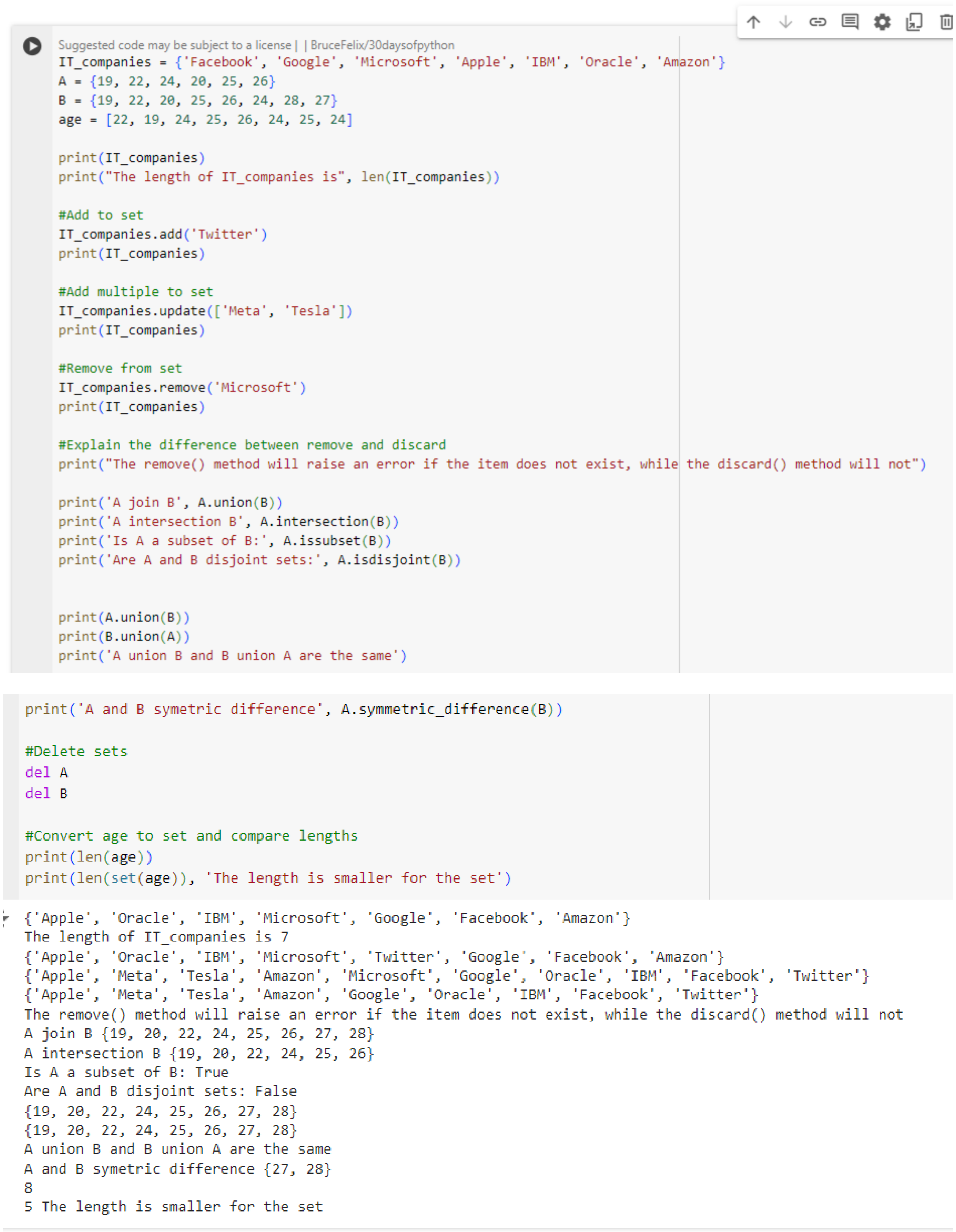
ICP 1 Step 8: Get list of types

```
my_list = [23, 'Python', 23.98]
type_list = []
for thing in my_list:
  type_list.append(type(thing))
print(my_list)
print(type_list)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

Part 9:

ICP 1 Step 9: Sets



Suggested code may be subject to a license | | BruceFelix/30daysofpython

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

print(IT_companies)
print("The length of IT_companies is", len(IT_companies))

#Add to set
IT_companies.add('Twitter')
print(IT_companies)

#Add multiple to set
IT_companies.update(['Meta', 'Tesla'])
print(IT_companies)

#Remove from set
IT_companies.remove('Microsoft')
print(IT_companies)

#Explain the difference between remove and discard
print("The remove() method will raise an error if the item does not exist, while the discard() method will not")

print('A join B', A.union(B))
print('A intersection B', A.intersection(B))
print('Is A a subset of B:', A.issubset(B))
print('Are A and B disjoint sets:', A.isdisjoint(B))


print(A.union(B))
print(B.union(A))
print('A union B and B union A are the same')
```

```python
print('A and B symetric difference', A.symmetric_difference(B))

#Delete sets
del A
del B

#Convert age to set and compare lengths
print(len(age))
print(len(set(age)), 'The length is smaller for the set')
```

```
{'Apple', 'Oracle', 'IBM', 'Microsoft', 'Google', 'Facebook', 'Amazon'}
The length of IT_companies is 7
{'Apple', 'Oracle', 'IBM', 'Microsoft', 'Twitter', 'Google', 'Facebook', 'Amazon'}
{'Apple', 'Meta', 'Tesla', 'Amazon', 'Microsoft', 'Google', 'Oracle', 'IBM', 'Facebook', 'Twitter'}
{'Apple', 'Meta', 'Tesla', 'Amazon', 'Google', 'Oracle', 'IBM', 'Facebook', 'Twitter'}
The remove() method will raise an error if the item does not exist, while the discard() method will not
A join B {19, 20, 22, 24, 25, 26, 27, 28}
A intersection B {19, 20, 22, 24, 25, 26}
Is A a subset of B: True
Are A and B disjoint sets: False
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26, 27, 28}
A union B and B union A are the same
A and B symetric difference {27, 28}
8
5 The length is smaller for the set
```