notes

June 5, 2025

1 Exam Notes

1.1 BASICS

Variables and Types:

• Purpose and use of int vs. float vs. string, conversion between types, how operators perform differently on different types (e.g. +)

[7]:
```
#integers
int_a = 10
int_b = 3

print(int_a + int_b)
print(int_a/int_b)
```

[8]:
```
#floats
float_a = 10.0
float_b = 3.0
print(float_a+float_b)
print(float_a/float_b) #note due to binary decimals are not precise, this can be␣
 ↪tested
print(f'{float_a/float_b:.0%}') #in an f-string use :.2f to do 2 decimal places␣
 ↪decimal
#                                          use :.2% to do 2 decimal place␣
 ↪percentage
```

[9]:
```
#strings
string_a = 'string a'
string_b = 'string b'
print(string_a + string_b)
print(string_a[1:4]) #note can slice strings, strings are arrays
```

Basic input and output in Python:

• Gather user input using "input" and format output using "print" and f-strings

[10]:
```
#input_a = input("what number do you want?")
#print(type(input_a)) #note this is always a string
#print(type(float(input_a))) # convert to float float()
```

```
#print(type(int(input_a))) # convert to integer int()
```

Basic input and output in Python:

• Basic Mathematical operators (e.g. + - * / ** // %) and built-in functions (e.g. abs, round and max etc.)

[11]:
```
print(2**4)
print(10//3)
print(10%3)
print(abs(-1))
print(round(2.5)) #floating point error prints 2, rounds normally though
print(max([1,2,3,4]))
print(min([1,2,3,4]))
```

1.2 DATA STRUCTURES

Data Structures in Python:

- Lists
- Dictionaries
- Sets
- Tuples

1.2.1 Lists

- Creating lists, indexing, append, insert, remove/pop, index and in
- Using lists to store/manipulate data, other functions including len, min, max, sum and sorted

[12]:
```
list = [1,2,3,4, 'tangerine']
print(list)

#indices
print(list[0:0]) #note non-inclusive
print(list[0:8]) #note can be expansive
print(list[:8]) #slices (certain amount of list taken)
print(list[:-2]) #can use negative indices to make a set with a given amount␣
 ↪less than original
```

Managing items in a list

Adding data

[13]:
```
extras = ['fruit', 1,2]
list_new=list.copy()
print(list_new.append(extras))
print(list_new.extend(extras))
print(list_new.insert(1,'banana'))
#note none of these print desired results (prints 'none') due to logical errors
#however the list is being modified
```