

SECTION 3: PYTHON OBJECT AND DATA STRUCTURE BASICS, 2 hrs 2 mins, 33 parts

• 15/36 Print Formatting with Strings

- -> string formatting for printing
- -> injecting a variable into a string for printing
- -> **if a variable appears in a string name**
 - **string interpolation**
 - two methods
 - -> .format()
 - {}.format('something1','something2')
 - -> print('This is a string {}'.format('INSERTED'))
 - -> this returns This is a string INSERTED, so it's inserted the text inside the format into what is in the curly braces
 - -> if you have more than one string there, you can list 0, 1, 2 etc in the {}'s -> and then in the inserted
 - -> you can use variable names when you do this, e.g -
 - **print('The {q} {b} {f}'.format(f='fox', b='brown', q='quick'))**
 - This returns the quick brown fox
 - Floating formatting follows (rounding to dp's)
 - 100/777 is a long decimal result
 - -> **print("The result was {r}".format(r=result))**
 - -> result is the name of the variable which stores 100/777 (that decimal)
 - -> value:width.precision f
 - **r:1.3f <- this returns (in this example 3sf)**
 - -> you can have a lot of whitespace if you use r:1.5f
 - -> using a lot of significant figures etc
- -> **f-strings <- formatted string literals**
 - name = "Jose"
 - **print(f'Hello, his name is {name}')** <- this is an f string -> a formatted string literal, name is the name of the variable
 - -> name = "Sam"
 - -> age = 3
 - -> print(f'{name} is {age} years old.') <- **string interpolation is writing variables names into strings (and their values being printed in the result)**
 - -> there is also documentation and are more examples in the course notebooks