Fundamental Data Types: Takeaways

Syntax

- The `type()` function can be used to figure out data type:

```
salary = 75000
print(type(salary))
```

- Python provides several functions for converting one type into another:

```
int("75000")        # string to integer
float("75000.00")   # string to float
str(75000)          # integer to string
```

- Using quotation marks to create a string:

```
library_name = "Scikit-learn"
```

- Concatenating two or more strings:

```
print('a' + 'b') # prints 'ab'
print('a' + 'b' + 'c') # prints 'abc'
```

- Escaping special characters:

```
'One data professional\'s advice is: "always comment your code".'
```

- Multi-line strings:

```
advice = '''One data professional's advice is: "always comment your code."
Another data professional's advice is: "read Scikit-learn's documentation."'''
```

Concepts

- There are four primitive types in Python: integers, decimals, strings, and Booleans.
- **Keywords** in Python are special words that cannot be used as variable names. `True`, `False` and `None` are examples of keywords.
- Python provides several functions for casting (or converting) one data type into another. This is important when we need to combine two variables, since they need to be the same type.
- Python also provides a special keyword, `None`, to indicate that data is missing.

Resources

- More info on Python's built-in Data Types
- More on type casting in Python
- More string concatenation examples