CoGrammar

The String and Numerical Data Types

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

Data Science Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: Open Class Questions

Data Science Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support

- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting

- We would love your **feedback** on lectures: Feedback on Lectures

Lecture Objectives

- Learning how to store and manipulate text using the String data type.
- Exploring the different types of numbers used in the Python programming language.

CoGrammar

What are Strings?

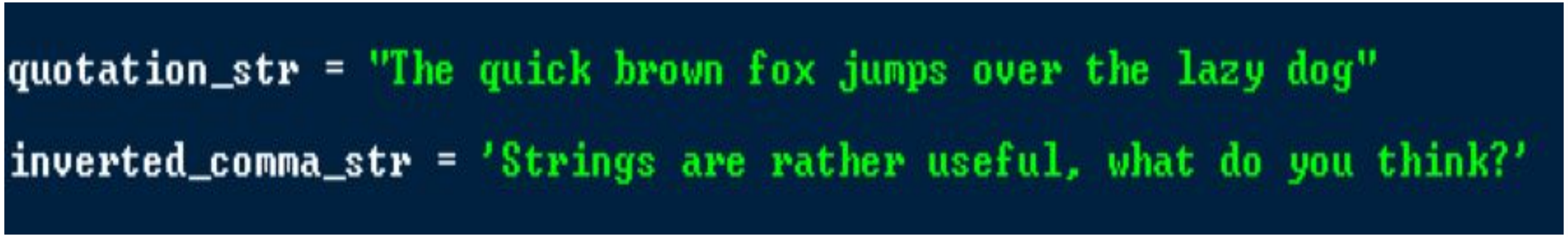
★ Strings are essentially any data made up of a sequence of letters or other characters.

★ Simply put, strings are just characters that have been “strung” together.

The String Data Type

★ Strings in Python are detected by quotation marks (“”) or inverted commas (‘’)

★ Example :



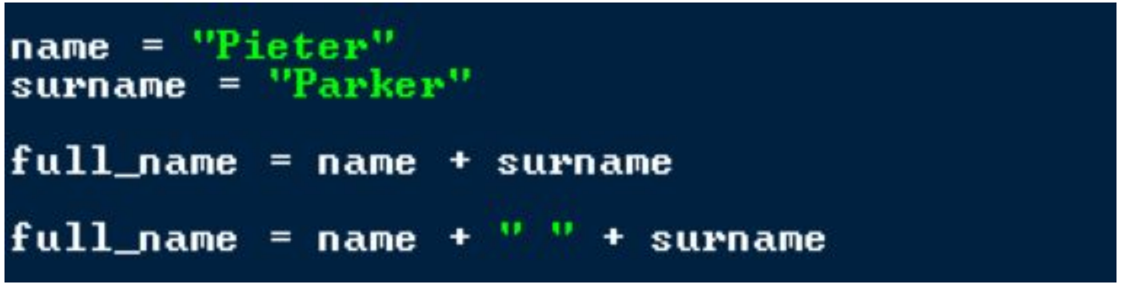
```
quotation_str = "The quick brown fox jumps over the lazy dog"

inverted_comma_str = 'Strings are rather useful, what do you think?'
```

CoGrammar

Concatenation of Strings

★ Strings can be added to one another. This is referred to as concatenation.

★ Example :



```
name = "Pieter"
surname = "Parker"

full_name = name + surname

full_name = name + " " + surname
```

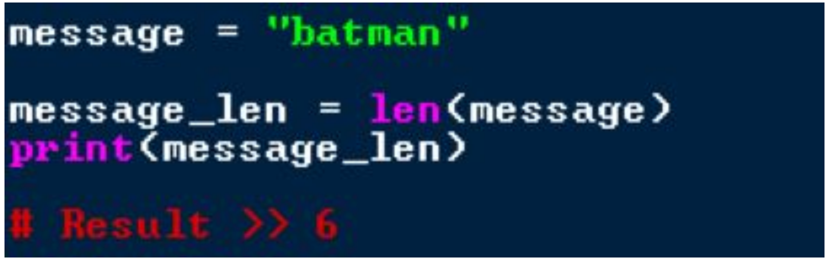
String Methods

- ★ **String methods are ways to express and action in programming.**
  - ○ **Within the brackets of the method are its arguments.**
  - ○ **Arguments are extra information given to the method.**

len()

★ **The len() method will simply output the length value of a string.**

★ **Example:**



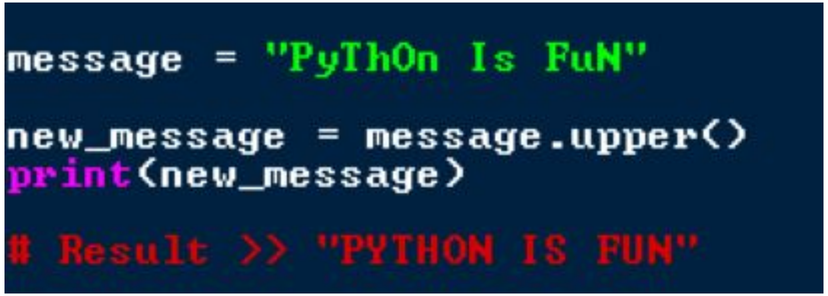
```
message = "batman"

message_len = len(message)
print(message_len)

# Result >> 6
```

CoGrammar

upper()

★ The upper() method will take a string and convert all the characters to uppercase.

★ Example:

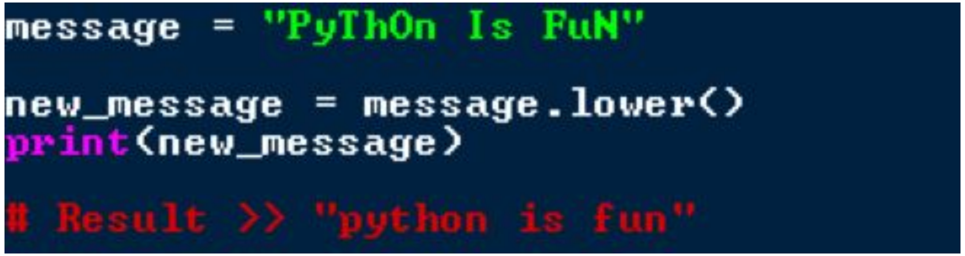


```
message = "PyThOn Is FuN"

new_message = message.upper()
print(new_message)

# Result >> "PYTHON IS FUN"
```

CoGrammar

lower()

- ★ **The lower() method will take a string and convert all the characters to lowercase.**
- ★ **Example:**

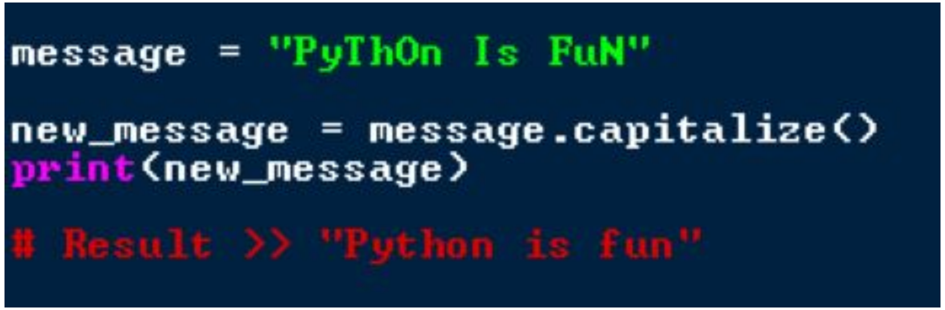


```
message = "PyThOn Is FuN"

new_message = message.lower()
print(new_message)

# Result >> "python is fun"
```

capitalize()

- ★ The capitalize() method will take a string and convert the first letter to uppercase and the rest of the characters to lowercase, should there be any other uppercase characters.

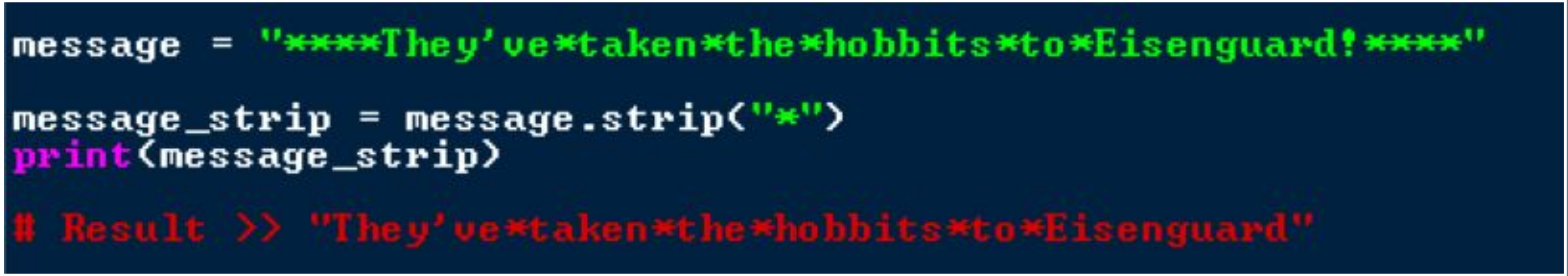
- ★ Example:



```
message = "PyThOn Is FuN"

new_message = message.capitalize()
print(new_message)

# Result >> "Python is fun"
```

strip()

- ★ The strip() method will remove a symbol from a string.
- ★ Keep in mind that strip() will only remove from the ends of a string.
- ★ Example:



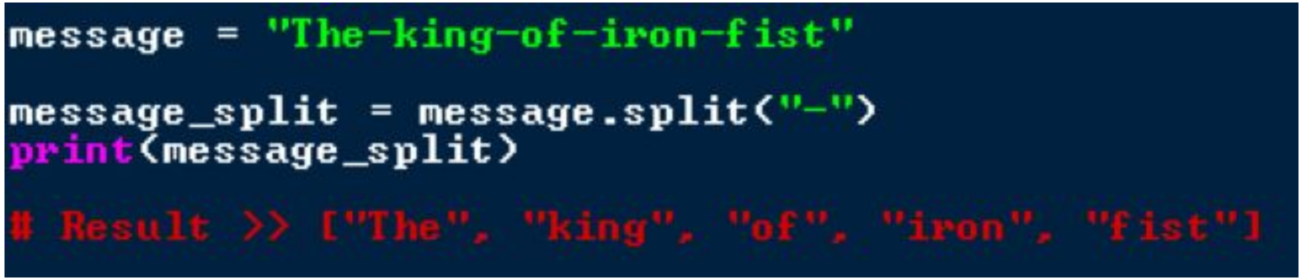
```
message = "****They've*taken*the*hobbits*to*Eisenguard!****"

message_strip = message.strip("*")
print(message_strip)

# Result >> "They've*taken*the*hobbits*to*Eisenguard"
```

CoGrammar

split()

- ★ The split() method, will split a string by a symbol. However, once the split occurs the string will then be placed in what’s called a list, which can be indexed.

- ★ Example:



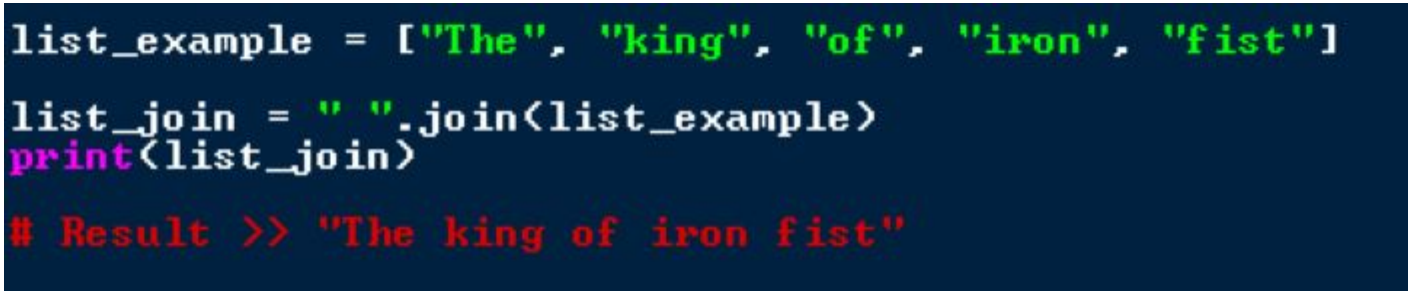
```
message = "The-king-of-iron-fist"

message_split = message.split("-")
print(message_split)

# Result >> ["The", "king", "of", "iron", "fist"]
```

CoGrammar

join()

★ The join() method will take a list of strings, and concatenate them to form one string.

★ Example:



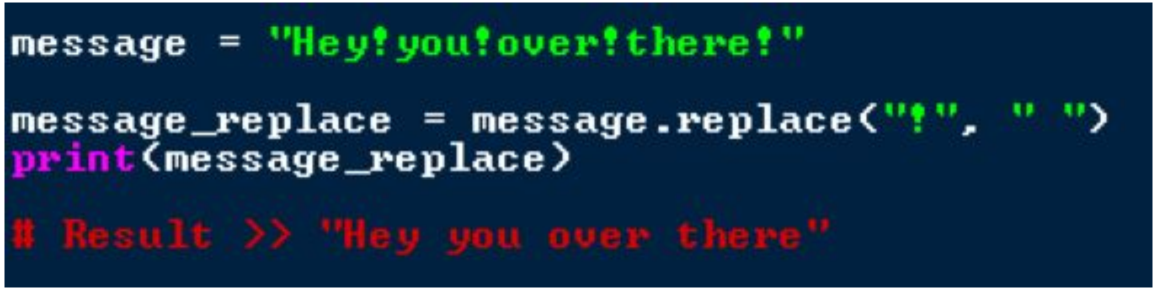
```
list_example = ["The", "king", "of", "iron", "fist"]

list_join = " ".join(list_example)
print(list_join)

# Result >> "The king of iron fist"
```

CoGrammar

replace()

★ The replace() method will replace any specified character in a string with a new one. Keep in mind that replace() requires two arguments to function. First to identify what to replace, and second to identify what to replace it with.

★ Example:

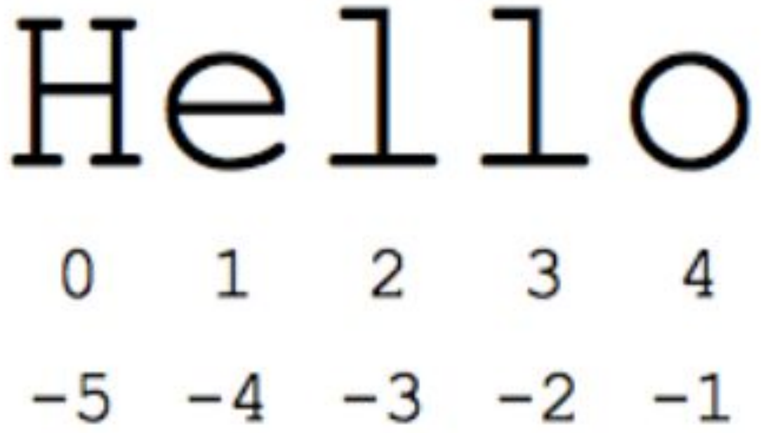


```
message = "Hey!you!over!there!"

message_replace = message.replace("!", " ")
print(message_replace)

# Result >> "Hey you over there"
```

CoGrammar

Indexing

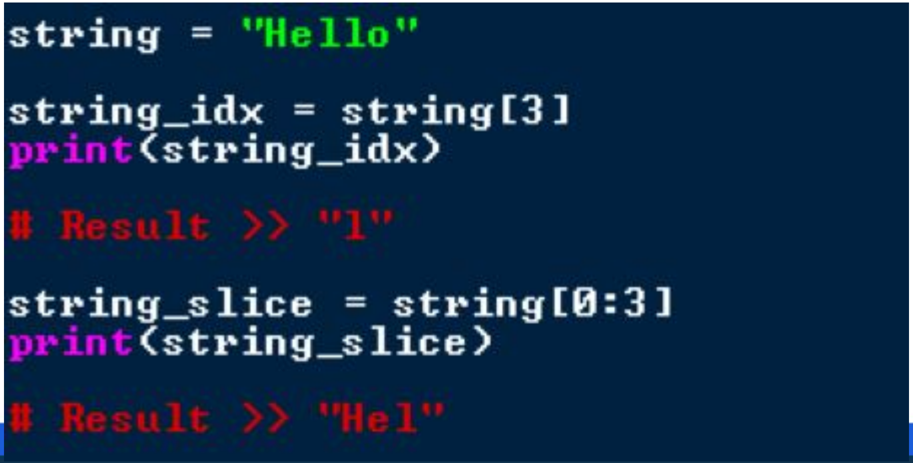
★ Strings are basically a list of characters. An example would be “Hello”, which consists of the characters H+e+l+l+o.



CoGrammar

String Slicing

★ String slicing is a way of extracting multiple characters from a string based on their index position.
★ Important to remember that this is done character by character, not word by word.
★ Example:



```
string = "Hello"

string_idx = string[3]
print(string_idx)

# Result >> "l"

string_slice = string[0:3]
print(string_slice)

# Result >> "Hel"
```

Escape Characters

★ Python uses the backslash (\) as an escape character.

★ The backslash is used as a marker to inform the compiler that the next character has a special use/meaning.

★ The backslash combined with specific other characters is known as an escape character.

Escape Characters

- ★ **Some useful escape characters:**
    - **\n - New line**
    - **\t - Tab Space**

- ★ **The escape character can also be used for quoting in a string.**

- ★ **By placing a backslash in front of a quotation mark, you can tell the compiler to avoid terminating the string.**

Let's Breathe

Let's take a small break before moving on to the next topic.

CoGrammar

CoGrammar

Numbers in Python

- ★ Here we will speak of 3 types of numbers used in Python:
  - Integers : whole numbers that are either positive or negative :
    - E.g. -32, 0, 600, 138227, etc.
  - Floats : decimal numbers that are also either positive or negative:
    - E.g. 6.2, -27.157, 33.3333, etc.
  - Complex : numbers that have a real and imaginary part, both of which are floats.

CoGrammar

Declaring Numeric Variables

★ Python is able to determine what data type a variable is based on the data’s characteristics:

★ num_one = 7 → no decimal point, no quotation marks, meaning it has to be an integer.

★ avg_grade = 8.3 → decimal point, no quotation marks, meaning it has to be float.

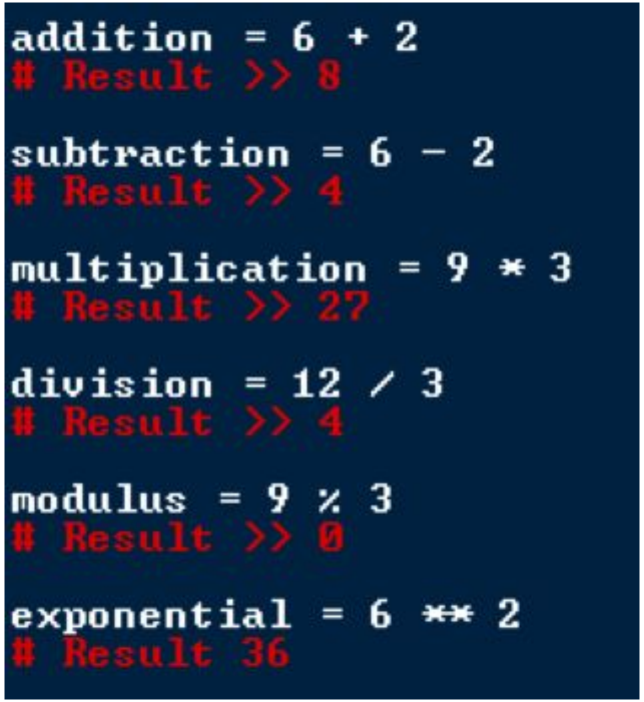
Arithmetic Operations

Similarly, with real world mathematics, we are able to apply maths to our numeric variables.

However, note that Python has a different way of interpreting the operation symbol, meaning that multiplication in Python is not written as ‘x’. The same applies for division and exponents.

CoGrammar

Arithmetic Operations



```
addition = 6 + 2
# Result >> 8

subtraction = 6 - 2
# Result >> 4

multiplication = 9 * 3
# Result >> 27

division = 12 / 3
# Result >> 4

modulus = 9 % 3
# Result >> 0

exponential = 6 ** 2
# Result 36
```

CoGrammar

Casting Data Types

★ **In Python, we can convert variables into other data types should we need to. This is known as casting.**

- **Cast to String → str()**
- **Cast to Integer → int()**
- **Cast to Float → float()**

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

CoGrammar

Thank you for joining!