# String Methods

Save As: stringmethods_example.py

**REMINDER: STRINGS are text enclosed in "quotes"**

```
phrase = "Giraffe Academy"
print(phrase)
```

**OUTPUT:**

```
Giraffe Academy
```

# REMINDER: STRINGS on a new line (\n character)

```
phrase = “Giraffe\nAcademy”
print(phrase)
```

**OUTPUT:**

```
Giraffe
Academy
```

**REMINDER: Concatenating STRINGS use +**

```
phrase = “Giraffe Academy”
print(phrase + “ is cool”)
```

**OUTPUT:**

**Giraffe Academy is cool**

# Common STRING METHODS

| Method | phrase = “Giraffe Academy” | OUTPUT |
|---|---|---|
| lower( ) | print(phrase.lower( )) | giraffe academy |
| upper( ) | print(phrase.upper( )) | GIRAFFE ACADEMY |
| isupper( ) | print(phrase.isupper( )) | Checks to see if the string is uppercase, then returns a **True or False**. |
| | print(phrase.upper( ).isupper( )) | Combines 2 methods. First converts to uppercase, then checks to see if string is uppercase.<br>**Returns True or False** |
| len( ) | print(len(phrase)) | Returns the **length** of the string. |

| phrase = | G | i | r | a | f | f | e |  | A | c | a | d | e | m | y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [index] or position | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| To Access STRING ELEMENTS – use their (index or position) | |
|---|---|
| **phrase = "Giraffe Academy"** | **OUTPUT** |
| **print(phrase[0])** | **G** - Returns the string character at index[0]. |
| **print(phrase[8])** | **A** - Returns the string character at index[8]. |
| **print(phrase[20])** | **IndexError** – string out of range. |

# More String Methods

| Method | phrase = “Giraffe Academy” | OUTPUT |
|---|---|---|
| index( ) | print(phrase.index(“G”)) | 0 – Returns the position where the string is found. |
| index( ) | print(phrase.index(“z”)) | Returns ValueError: substring not found |

# More String Methods

| Method | phrase = “Giraffe Academy” | OUTPUT |
|---|---|---|
| replace( ) | print(phrase.replace(“Giraffe”, “Elephant”)) | Replaces Giraffe with Elephant<br>OUTPUT: Elephant Academy |
| | print(phrase.replace(“Gira”, “Elephant”)) | OUTPUT: Elephantffe Academy |
| | print(phrase.replace(“Z”)) | OUTPUT: Giraffe Academy |
| | print(phrase.replace(0)) | TypeError: (??) must be string |

# More String Methods

| Method | phrase = " Giraffe, Academy " | OUTPUT |
|---|---|---|
| strip( ) | print(phrase.strip( )) | Removes any whitespace at the beginning or the end of the string. |
| split( ) | print(phrase.split(",") ) | Splits a string into substrings based on the indicated separator. Useful searching through text files. |

**Go to w3schools.com Python Tutorial to view other String Methods

| To Access parts of the string – SLICE the string | | | | | | |
|---|---|---|---|---|---|---|
| fruit = | b | a | n | a | n | a |
| [index or position] | 0 | 1 | 2 | 3 | 4 | 5 |

| SLICING the String | |
|---|---|
| fruit = “banana” | OUTPUT |
| print(fruit[n:m]) | Start at n position, up to NOT including m position |
| print(fruit[0]) | b |
| print(fruit[0:3]) | ban |

| To Access parts of the string – SLICE the string | | | | | | |
|---|---|---|---|---|---|---|
| fruit = | b | a | n | a | n | a |
| [index or position] | 0 | 1 | 2 | 3 | 4 | 5 |
| | | | | -3 | -2 | -1 |

| NEGATIVE SLICING | |
|---|---|
| fruit = “banana” | OUTPUT |
| **print(fruit[n:m])** | Start at n position, up to NOT including m position |
| *To access the end of the string, the last position is **also** -1. | |
| **print(fruit[-3:-1])** | an |