CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

# Substrings and String Formatting

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

# What will we Cover?

- Using the string data type
- Substrings
- String formatting

# Substrings

- `Str_name[start:end:step]` – used to extract a subset of the original string
  - `start` – starting `index` of the substring (default is `0`)
  - `end` – stopping `index+1` (default is `len(str_name)`)
  - `step` – index increment (default is `1`)

| Example | Result |
| --- | --- |
| `str1 = "Hello World!"`<br>`sub_str = str1[0:11:2]`<br>`print(sub_str)` | `"HloWrd"` |

# Examples I

## Default end and step

```
sub_str2 = "Python"[4:]
print(sub_str2)
```

```
on
```

## Default start and step

```
sub_str1 = "Python"[:2]
print(sub_str1)
```

```
Py
```

## Default start and end

```
sub_str3 = "Mississippi"[::3]
print(sub_str3)
```

```
Mssp
```

## Default step

```
sub_str4 = "Mississippi"[3:6]
print(sub_str4)
```

```
sis
```

CITY
UNIVERSITY OF LONDON
EST 1894

# Placeholder

- `format()` method and curly brackets `{}` are used to create a placeholder in a string
- The placeholder can then be replaced with a different value of type string

| Example | Result |
| --- | --- |
| `city = "London"`<br>`str1 = "Welcome to {}"`<br>`print(str1.format(city))` | `"Welcome to London"` |

# Placeholder Identifiers

- If multiple placeholders are used for one string, they can be identified using
  - Named indexes
  - Numbered indexes

| Example | Result |
| --- | --- |
| `Print("{name} is {age}".format(name = "John", age = 36))` | `"John is 36"` |
| `Print("{0} is {1}".format("John", 36))` | `"John is 36"` |

# Examples II

```
str1 = input("Please enter your name: ")
print("{0}, your name is {0}".format(str1))
```

```
Please enter your name: Helen
Helen, your name is Helen
```

```
int1 = int(input("Please enter a number: "))
int2 = int(input("Please enter a number: "))
print("{i1} + {i2} = {sum}".format(i2 = int2, i1 = int1, sum = int2+int1))
```

```
Please enter a number: 2
Please enter a number: 3
2 + 3 = 5
```

CITY UNIVERSITY OF LONDON EST 1894

# String Alignment I

- : operator followed by an integer is used within a placeholder to add a specified amount of extra white spaces

| Example | Result |
| --- | --- |
| `str2 = "Hello{:3}World!"`<br>`print(str2.format(""))` | `Hello   World!` |
| `str3 = "Hello World{:5}!"`<br>`print(str3.format(""))` | `Hello World     !` |

# String Alignment II

- `<` , `^` , `>` operators are used with the `:` operator to align the text in the placeholder to the left, centre or right respectively

| Example | Result |
|---|---|
| `Str4 = "a{:>4}c".format("b")` | `a  bc` |
| `Str5 = "a{:^4}c".format("b")` | `a b  c` |
| `Str6 = "a{:<4}c".format("b")` | `ab  c` |

# Examples III

Align together, centrally and apart

```
print("|{hi:<15}|{hi:>15}|".format(hi = "Hello there"))
```

```
|Hello there    |    Hello there|
```

```
print("|{hi:^15}|{hi:^15}|".format(hi = "Hello there"))
```

```
|  Hello there  |  Hello there  |
```

```
print("|{hi:>15}|{hi:<15}|".format(hi = "Hello there"))
```

```
|    Hello there|Hello there    |
```

# Try It Yourself

Write a program which reads a letter from the user and determines whether it is a vowel or a consonant.

Implement a check which ensures that the input contains letters only and no digits or other symbols.