



Lisbon School  
of Economics  
& Management  
Universidade de Lisboa



Carlos J. Costa

# STRINGS

# Strings

- Sequence of characters
- String literals in python are surrounded by either
  - single quotation marks ('hello') or
  - double quotation marks ("hello").

# Strings

- String is an array:

```
a = "Hello, World!"  
print(a[1])
```

# Slicing

- Return a range of characters by using the slice syntax.

```
a = "Hello, World!"  
print(a[2:5])
```

- Negative indexes to start the slice from the end of the string:

```
b = "Hello, World!"  
print(b[-5:-2])
```

# Methods

- `capitalize()` Converts the first character to upper case
- `casefold()` Converts string into lower case
- `center()` Returns a centered string
- `count()` Returns the number of times a specified value occurs in a string
- `encode()` Returns an encoded version of the string
- `endswith()` Returns true if the string ends with the specified value
- `expandtabs()` Sets the tab size of the string
- `find()` Searches the string for a specified value and returns the position of where it was found
- `format()` Formats specified values in a string
- `format_map()` Formats specified values in a string
- `index()` Searches the string for a specified value and returns the position of where it was found
- `isalnum()` Returns True if all characters in the string are alphanumeric
- `isalpha()` Returns True if all characters in the string are in the alphabet
- `isdecimal()` Returns True if all characters in the string are decimals
- `isdigit()` Returns True if all characters in the string are digits
- `isidentifier()` Returns True if the string is an identifier
- `islower()` Returns True if all characters in the string are lower case
- `isnumeric()` Returns True if all characters in the string are numeric
- `isprintable()` Returns True if all characters in the string are printable
- `isspace()` Returns True if all characters in the string are whitespaces
- `istitle()` Returns True if the string follows the rules of a title
- `isupper()` Returns True if all characters in the string are upper case
- `join()` Joins the elements of an iterable to the end of the string

# Methods

- `ljust()` Returns a left justified version of the string
- `lower()` Converts a string into lower case
- `lstrip()` Returns a left trim version of the string
- `maketrans()` Returns a translation table to be used in translations
- `partition()` Returns a tuple where the string is parted into three parts
- `replace()` Returns a string where a specified value is replaced with a specified value
- `rfind()` Searches the string for a specified value and returns the last position of where it was found
- `rindex()` Searches the string for a specified value and returns the last position of where it was found
- `rjust()` Returns a right justified version of the string
- `rpartition()` Returns a tuple where the string is parted into three parts
- `rsplit()` Splits the string at the specified separator, and returns a list
- `rstrip()` Returns a right trim version of the string
- `split()` Splits the string at the specified separator, and returns a list
- `splitlines()` Splits the string at line breaks and returns a list
- `startswith()` Returns true if the string starts with the specified value
- `strip()` Returns a trimmed version of the string
- `swapcase()` Swaps cases, lower case becomes upper case and vice versa
- `title()` Converts the first character of each word to upper case
- `translate()` Returns a translated string
- `upper()` Converts a string into upper case
- `zfill()` Fills the string with a specified number of 0 values at the beginning

# Format

- **Example with just one parameter**

```
age = 22
txt = "My name is John, and I am {}"
print(txt.format(age))
```

- **More than one**

```
name = 'Mark'
age = 22
txt = "My name is {}, and I am {}"
print(txt.format(name, age))
```

# find

- The `find()` method finds the first occurrence of the specified value.

```
x = "like looking for a needle in a  
haystack"  
x = txt.find("needle")  
print(x)
```

- What will happen if I want to find a “pin”?



# replace

- The `replace()` method replaces a specified phrase with another specified phrase.

```
txt = "We should eat sprouts"  
x = txt.replace("sprouts", "hamburgers")  
print(x)
```

# Split

- Split a string into a list where each word is a list item:

```
txt = "ISEG is the best"  
x = txt.split()  
print(x)
```

# strip

- Remove spaces at the beginning and at the end of the string:

```
txt = "    banana    "  
x = txt.strip()
```

# Exercises

Consider the following expression

```
text="He forgot the medication."
```

- Print the length of the string (use the len function).
- Get the first character of the string text.
- Get the characters from index 1 to index 3
- Return the string without any whitespace at the beginning or the end.
- Convert the value of txt to upper case .
- Convert the value of txt to lower case.
- Replace the character c with a t.
- Convert the string into a list of words.
- Conclude the following code using the format method

```
name ="carlos"
```

```
age = 21
```

```
text = "My name is {}, and I am {}"
```