



## Unit Testing

- Take your previous function `german_polite_form` and save it in the file `german.py`
- Create the file `test_german.py`
- Write one test that verifies that Conchita Wurst gets addressed correctly
  - You only need to import your `german_polite_form` file and create a function for the test starting with `test_`
  - Use `assert` to test your assumption
- Write one test that verifies that using the number 3 does *not* work
  - Use the `try ... except` construct in the unit test to verify that `german_polite_form` raises the expected exception.

## Reading documentation

Download the Python module (file) `animal.py`, but **do not open it yet**. The module contains a class called `Animal`. It is a little different than the `Animal` class we created in class on Friday.

1. Print the module's documentation by writing:

```
import animal
```

```
print(help(animal.Animal))
```

2. Answer the following questions just by reading the docstring:
  - A. What are the attributes (properties) of the class?
  - B. How many methods does the class have?
  - C. What happens if you call `.says('Moooooo')`?
  - D. Verify your expectation by creating an `Animal` and asking it to `.says('Moooooo')`.

# Reading an Excel file with `openpyxl`

1. Open the Befolknings data in file `befkbhalderstatkode_small.xlsx` using `openpyxl` .

```
openpyxl.load_workbook(filename)
```

2. Store the sheet object in a variable

- Remember you can get a sheet by doing:

```
workbook["Sheet1"]
```

- What is type of the variable?

3. Using the `[]` notation of the sheet, get the cell `C3`

4. Can you use the range-notation `:from lists` to get the cells from `A1` to `C3` ?

# Descriptive Statistics with an Excel file and `openpyxl`

Write a small CLI program that counts how many people from a certain nation lived in Copenhagen in 2015.

- Call your program `count_nationalities.py`
- Let it read data from the Excel file `befkbhalderstatkode_small.xlsx`
- Let it consume an argument from the command-line, which says which nationality you want to count. For example, running `count_nationalities.py` with the CLI argument `5126` -the country code for Belgien- should print `273` :

```
$ python count_nationalities.py 5126
273
```

- *Hint:* Use a `for` loop iterating over each row in the spreadsheet.
- *Hint:* When the value of the element in first cell of the row is `2015` and the value of the fourth cell of the row is the given country code then you want to use the value of the last cell of the row for counting.



## Unit Testing

- Take your previous function `german_polite_form` and save it in the file `german.py`
- Create the file `test_german.py`
- Write one test that verifies that `Conchita Wurst` gets addressed correctly
  - You only need to import your `german_polite_form` file and create a function for the test starting with `test_`
  - Use `assert` to test your assumption
- Write one test that verifies that using the number 3 does *not* work
  - Use the `try ... except` construct in the unit test to verify that `german_polite_form` raises the expected exception.

## Reading documentation

Download the Python module (file) `animal.py` , but **do not open it yet**. The module contains a class called `Animal` . It is a little different than the `Animal` class we created in class on Friday.

1. Print the module's documentation by writing:

```
import animal
```

```
print(help(animal.Animal))
```

2. Answer the following questions just by reading the docstring:
  - A. What are the attributes (properties) of the class?
  - B. How many methods does the class have?
  - C. What happens if you call `.says('Mooooooo')` ?
  - D. Verify your expectation by creating an `Animal` and asking it to `.says('Mooooooo')` ."

# Reading an Excel file with `openpyxl`

1. Open the Befolkings data in file `befkbhalderstatkode_small.xlsx` using `openpyxl`.

```
openpyxl.load_workbook(filename)
```

2. Store the sheet object in a variable

- Remember you can get a sheet by doing:

```
workbook["Sheet1"]
```

- What is type of the variable?

3. Using the `[]` notation of the sheet, get the cell `C3`

4. Can you use the range-notation `:from lists` to get the cells from `A1` to `C3` ?

# Descriptive Statistics with an Excel file and `openpyxl`

Write a small CLI program that counts how many people from a certain nation lived in Copenhagen in 2015.

- Call your program `count_nationalities.py`
- Let it read data from the Excel file `befkbhalderstatkode_small.xlsx`
- Let it consume an argument from the command-line, which says which nationality you want to count. For example, running `count_nationalities.py` with the CLI argument `5126` -the country code for Belgien- should print `273` :

```
$ python count_nationalities.py 5126
273
```

- *Hint:* Use a `for` loop iterating over each row in the spreadsheet.
- *Hint:* When the value of the element in first cell of the row is `2015` and the value of the fourth cell of the row is the given country code then you want to use the value of the last cell of the row for counting.