# Introduction to Python Exercises

## L2 - Exercises

Debbie, Liv, Peter

27 January 2020

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

1.1 Open the python interpreter and read the the Zen of Python
( >>> import this)

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

1.1 Open the python interpreter and read the the Zen of Python
( >>> import this)

1.2 Create a list of five elements, select the first two elements of that list. Add the selected elements to the original list. Append the following element: "Chilindrina". Duplicate the list and count the number of elements!

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

1.1 Open the python interpreter and read the the Zen of Python
( >>> import this)

1.2 Create a list of five elements, select the first two elements of that list. Add the selected elements to the original list. Append the following element: "Chilindrina". Duplicate the list and count the number of elements!

1.3 Create an array of 16 elements, from 1 to 16. Reshape the array as a two dimensional array (2, 8). Calculate the sin for the first dimension and cosine for the second dimension. Plot the two dimensions.

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

1.1 Open the python interpreter and read the the Zen of Python
( >>> import this)

1.2 Create a list of five elements, select the first two elements of that list. Add the selected elements to the original list. Append the following element: "Chilindrina". Duplicate the list and count the number of elements!

1.3 Create an array of 16 elements, from 1 to 16. Reshape the array as a two dimensional array (2, 8). Calculate the sin for the first dimension and cosine for the second dimension. Plot the two dimensions.

1.4 Calculate a Fibonacci series of 20 elements using a while or a for loop.

# Exercise 1 - Familiarise with python

Run the basic commands introduced during the lecture and experience their effects.

1.1 Open the python interpreter and read the the Zen of Python ( >>> import this)

1.2 Create a list of five elements, select the first two elements of that list. Add the selected elements to the original list. Append the following element: "Chilindrina". Duplicate the list and count the number of elements!

1.3 Create an array of 16 elements, from 1 to 16. Reshape the array as a two dimensional array (2, 8). Calculate the sin for the first dimension and cosine for the second dimension. Plot the two dimensions.

1.4 Calculate a Fibonacci series of 20 elements using a while or a for loop.

1.5 Code the simple exponential decay model (L2, page 80), produce different solutions using different values for the time decay parameter, plot the results...

# Exercise 2 - Branching model

2.1 Code the 'branching' model, i.e.: $dn(t)/dt = b \cdot n(t)$ (L1, page 32)

# Exercise 2 - Branching model

2.1 Code the 'branching' model, i.e.: $dn(t)/dt = b \cdot n(t)$ (L1, page 32)

2.2 Use an initial number of branches of $n(t=0) = 10.0$ and a branching rate of $b = 0.25$ branches per day

# Exercise 2 - Branching model

2.1 Code the 'branching' model, i.e.: $dn(t)/dt = b \cdot n(t)$ (L1, page 32)

2.2 Use an initial number of branches of $n(t = 0) = 10.0$ and a branching rate of $b = 0.25$ branches per day

2.3 Generate a solution $n(t)$ over 101 evenly spaced time steps over 10 days

# Exercise 2 - Branching model

2.1 Code the 'branching' model, i.e.: $dn(t)/dt = b \cdot n(t)$ (L1, page 32)

2.2 Use an initial number of branches of $n(t = 0) = 10.0$ and a branching rate of $b = 0.25$ branches per day

2.3 Generate a solution $n(t)$ over 101 evenly spaced time steps over 10 days

2.4 Plot the results.

# Exercise 3 - Cat and mouse model

3.1 Code the 'cat and mouse' model, i.e.: $dn(t)/dt = b \cdot n(t) - d \cdot n(t) + m$ (L1, page 33)

# Exercise 3 - Cat and mouse model

3.1 Code the 'cat and mouse' model, i.e.: $dn(t)/dt = b \cdot n(t) - d \cdot n(t) + m$ (L1, page 33)

3.2 Use an initial number of mice of $n(t = 0) = 10.0$, a birth rate of 2 mice every 8 days, a cat eats 1 mouse every 2 days, and 3 new mice migrate every 10 days from nearby yards

# Exercise 3 - Cat and mouse model

3.1 Code the 'cat and mouse' model, i.e.: $dn(t)/dt = b \cdot n(t) - d \cdot n(t) + m$ (L1, page 33)

3.2 Use an initial number of mice of $n(t = 0) = 10.0$, a birth rate of 2 mice every 8 days, a cat eats 1 mouse every 2 days, and 3 new mice migrate every 10 days from nearby yards

3.3 Generate a solution $n(t)$ over 101 evenly spaced time steps over 10 days

# Exercise 3 - Cat and mouse model

3.1 Code the 'cat and mouse' model, i.e.: $dn(t)/dt = b \cdot n(t) - d \cdot n(t) + m$ (L1, page 33)

3.2 Use an initial number of mice of $n(t=0) = 10.0$, a birth rate of 2 mice every 8 days, a cat eats 1 mouse every 2 days, and 3 new mice migrate every 10 days from nearby yards

3.3 Generate a solution $n(t)$ over 101 evenly spaced time steps over 10 days

3.4 Plot the results.

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a \cdot c \cdot n(t) \cdot s(t)$ and $dn(t)/dt = +a \cdot c \cdot s(t) \cdot n(t)$ (L1 page 34)

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a{\cdot}c{\cdot}n(t){\cdot}s(t)$ and $dn(t)/dt = +a{\cdot}c{\cdot}s(t){\cdot}n(t)$ (L1 page 34)

4.2 In python considering: 10 contacts between carrier and healthy individuals every day and a 2.5 % probability that the flu is transmitted at every contact

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a \cdot c \cdot n(t) \cdot s(t)$ and $dn(t)/dt = +a \cdot c \cdot s(t) \cdot n(t)$ (L1 page 34)

4.2 In python considering: 10 contacts between carrier and healthy individuals every day and a 2.5 % probability that the flu is transmitted at every contact

4.3 Use an initial number of healthy individuals of $s(t = 0) = 95.0$, an initial number of carriers of $n(t = 0) = 5.0$

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a \cdot c \cdot n(t) \cdot s(t)$ and $dn(t)/dt = +a \cdot c \cdot s(t) \cdot n(t)$ (L1 page 34)

4.2 In python considering: 10 contacts between carrier and healthy individuals every day and a 2.5 % probability that the flu is transmitted at every contact

4.3 Use an initial number of healthy individuals of $s(t = 0) = 95.0$, an initial number of carriers of $n(t = 0) = 5.0$

4.4 Generate a solution over 101 evenly spaced time steps over 10 days

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a \cdot c \cdot n(t) \cdot s(t)$ and $dn(t)/dt = +a \cdot c \cdot s(t) \cdot n(t)$ (L1 page 34)

4.2 In python considering: 10 contacts between carrier and healthy individuals every day and a 2.5 % probability that the flu is transmitted at every contact

4.3 Use an initial number of healthy individuals of $s(t = 0) = 95.0$, an initial number of carriers of $n(t = 0) = 5.0$

4.4 Generate a solution over 101 evenly spaced time steps over 10 days

4.5 Plot the results

# Exercise 4 - Flu model

4.1 Code the 'flu' model, i.e.: $ds(t)/dt = -a \cdot c \cdot n(t) \cdot s(t)$ and $dn(t)/dt = +a \cdot c \cdot s(t) \cdot n(t)$ (L1 page 34)

4.2 In python considering: 10 contacts between carrier and healthy individuals every day and a 2.5 % probability that the flu is transmitted at every contact

4.3 Use an initial number of healthy individuals of $s(t = 0) = 95.0$, an initial number of carriers of $n(t = 0) = 5.0$

4.4 Generate a solution over 101 evenly spaced time steps over 10 days

4.5 Plot the results

4.6 Plot both $s(t)$ and $n(t)$ and $s(t) + n(t)$ (i.e. the total number of individuals) in the same panel; after how many days does the model reach steady-state?