

## AI1072: Machine Learning, exercise sheet 2a

Generally, you should use a separate .py file for each exercise. From the command line, you can execute these files (that you create with any editor, e.g. gedit) by invoking:

```
python3 <filename>
```

When executing files in this way, Python prints nothing except when you tell it to, with `print()`. So be sure to print out each results explicitly.

### 1 Functions

Create a function `fak(x)` that computes  $X!$  recursively by observing that  $0! = 1$  and  $x! = x(x-1)!$ . Call that function with arguments from 1 to 6.

### 2 List creation

Create lists with the following properties, choose names like `ex1_1`, `2`, `3` for them:

1. numbers from 0 to 10 that are even
2. numbers from 0 to 100 that can be divided by 15 (use `%`, the modulus operator and list comprehension)
3. odd numbers from 15 to 1
4. the string "xx", repeated 5 times
5. the string "stringX" repeated 10 times, where X goes from 5 to 14. Use the builtin function `str()` to convert numbers to strings and the fact that strings can be concatenated using the "+" operator
6. a list with the items "1", 2, 3.0, 4
7. all the numbers from 0 to 99 that contain the cypher "3". You may use the method `find()` that all strings possess to look for a substring. If it is found, the start index is returned, otherwise -1.

### 3 List manipulation and slicing

Create a list with numbers from 0 to 49 and...

1. create and call a function that returns its first element and the remaining list
2. create and call a function that returns the sum, difference and modulus of the first and last element

3. create and call a function that sums up the list and returns the result!
4. create and call a function that returns the list elements at odd index positions
5. create and call a function that copies out the list elements from the last element to the second one (excluding the first, ie the one at index 0)
6. create and call a function that returns an inverted list, i.e., that starts at the end and includes all elements including the first
7. create and call a function that returns a list, of the same length as the argument, that has 1 wherever the argument is odd, and 0 elsewhere. Hint: use list comprehension!

## 4 Dictionary creation and manipulation

First, create an empty dictionary  $D$ .

1. Fill  $D$  with key-value pairs of the form 1:"1", 2:"2", ... up until 10:"10". Use a for loop for this!
2. Dictionaries are iterables: obtain the iterator object of  $D$  and print out the first 5 elements of the iterable represented by the iterator. Is it the keys or the values of  $D$ ?
3. write a function  $dget(D, k)$  which returns True if a key  $k$  is present (in the sense of  $==$ ) in the keys of a dictionary  $D$ , and False if not. For this, you can use the keyword expression "x in I" which returns True if a value  $x$  exists in an iterable  $I$  (in the sense of  $==$ ), and False otherwise. Keep in mind the previous exercise and test your function on the  $D$  that you created previously!
4. write a function  $dget2(D, k)$  which returns True if a key  $k$  is present (in the sense of "is") in a dictionary  $D$ , and False if not. There are no shortcuts for this, so use a for loop or similar constructs.

## 5 Iteration

Create a function  $pr(x, i)$  that prints out the first  $i$  elements of an iterable  $x$ . Test this function with a tuple, a list, a string, a dictionary and an iterator given by `range()` as examples of iterables. What happens when  $i$  is larger than the number of elements in the iterable?