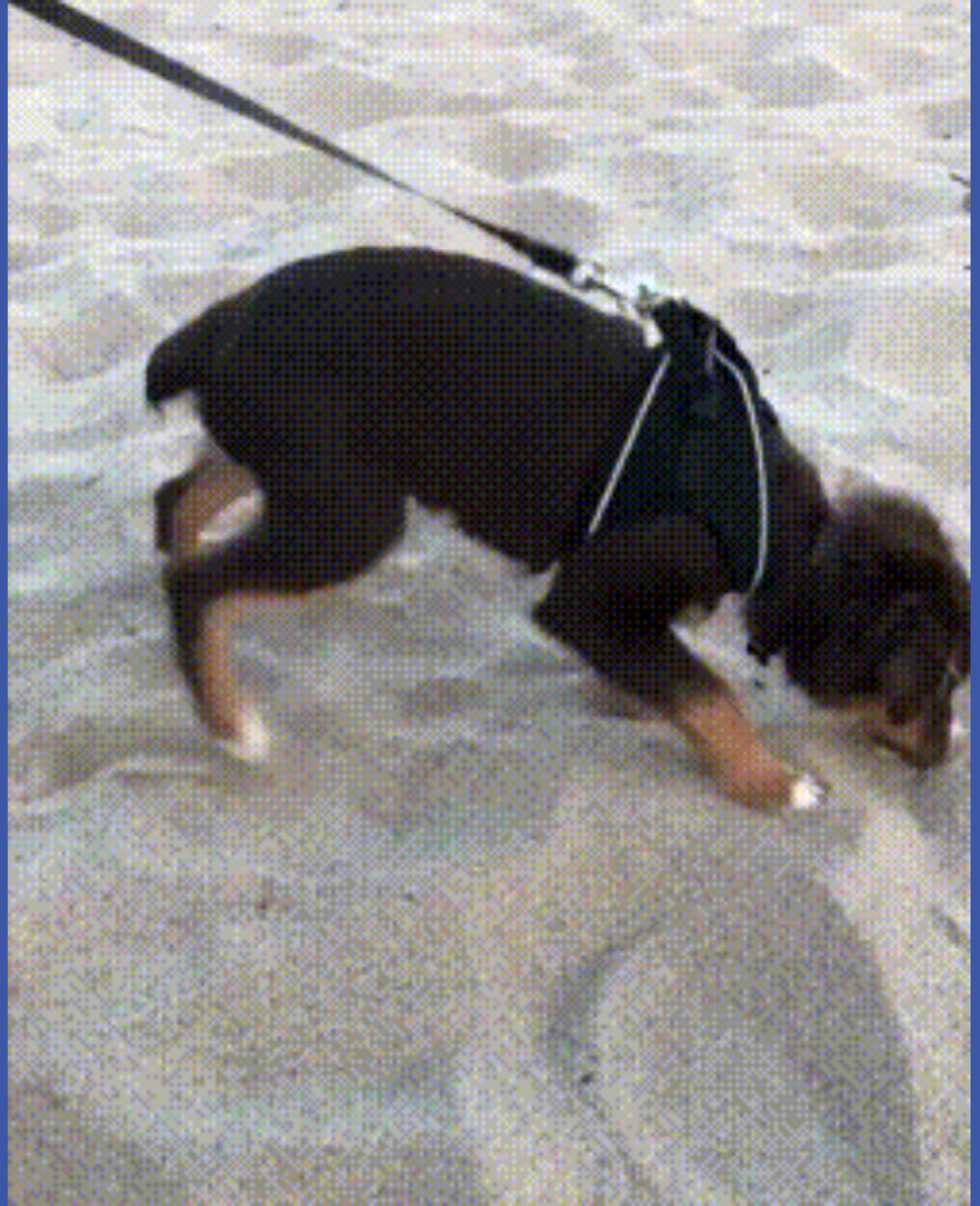
Data Science Fundamentals
Lesson 4

Hay Kranen

Wednesday October 3rd, 2018



Today's programme

Recap
Advanced usage of the for loop
Multidimensional lists
Exercise
Slicing
Exercise
Files and the CSV format
Exercise
Lunch break

RECAP

```
#Creating an empty list
names =[]



#Add names to the list
names.append("Rob")
names.append("David")
names.append("Nick")
```

```
#create a list that can be appended and loop
friends =["Peter", "Jane", "Simon"]
friends.append("Annie")
friends.append("Luke")
```

```
["Rob", "David", "Nick"]
["Peter", "Jane", "Simon", "Annie", "Luke"]
```

```
#ask the user for their name and add it to the list. .title() makes the first character an uppercase
names.append(input("What's your name? ").title())
```

```
#adding the input of the snacks into the empty friends snack list
friends_snack.append(input(names + " (" + str(len(names)) + ")" + " What's your favorite snack?"))
```

```
snack = input(name + ", what is your favorite snack?")
```

```
# ask each person personally what their favorite snack is
snack = input("What is your favourite snack " + name + "? ")
```

```
for name in names:
```

```
for name in friends:
```

```
for names in friends_names:
```

```
for word in sentence_words:
```

```
for zincheck in zinlijst:
```

```
for word in sentence:
```

```
for woord in woorden:
```

```
for word in words:
```

STYLE NOTES

Keep everything in English

List should be plural, items in a list singular

```
for word in words:
    print(word)

for item in words:
    print(item)
```

```
#print(words[index])
if words[index] in wrong:
        #print("error")
        if words[index] == "internet":
                words[index] = "papier"
        elif words[index] == "computer":
                words[index] = "printer"
        elif words[index] == "laptop":
                words[index] = "desktop"
        elif words[index] == "apple":
                words[index] = "pear"
        elif words[
                wor
        elif words[
                wor
        elif words[
                wor
        elif words[
                wor
        else:
                words[index] = words[index]
```

```
if word in wrong:
    wrong_index = wrong.index(word)
    words[index] = good_words[wrong_index]
```

```
#compare, warn and exchange
for word in sentence.split(" "):
        if word in badwords:
                index = badwords.index(word)
                check = goodwords[index]
                print("***" + word + " is spelt wrong. " + word + " should be " + check)
                new_sentence.append(check)
        else:
                new_sentence.append(word)
```

ADVANCED USAGE OF THE FOR LOOP

Advanced use of the `for` statement

In [2]:
```
# There are two more useful tricks you can do with the for statement, and both involve the index o
# As we saw earlier we can manually track the index of a for loop
friends = ["Barrie", "Tinus", "Hans"]

index = 0
for friend in friends:
    print(friend + " is friend number " + str(index))
    index = index + 1 # Make sure we update this whenever the loop ends
```

```
Barrie is friend number 0
Tinus is friend number 1
Hans is friend number 2
```

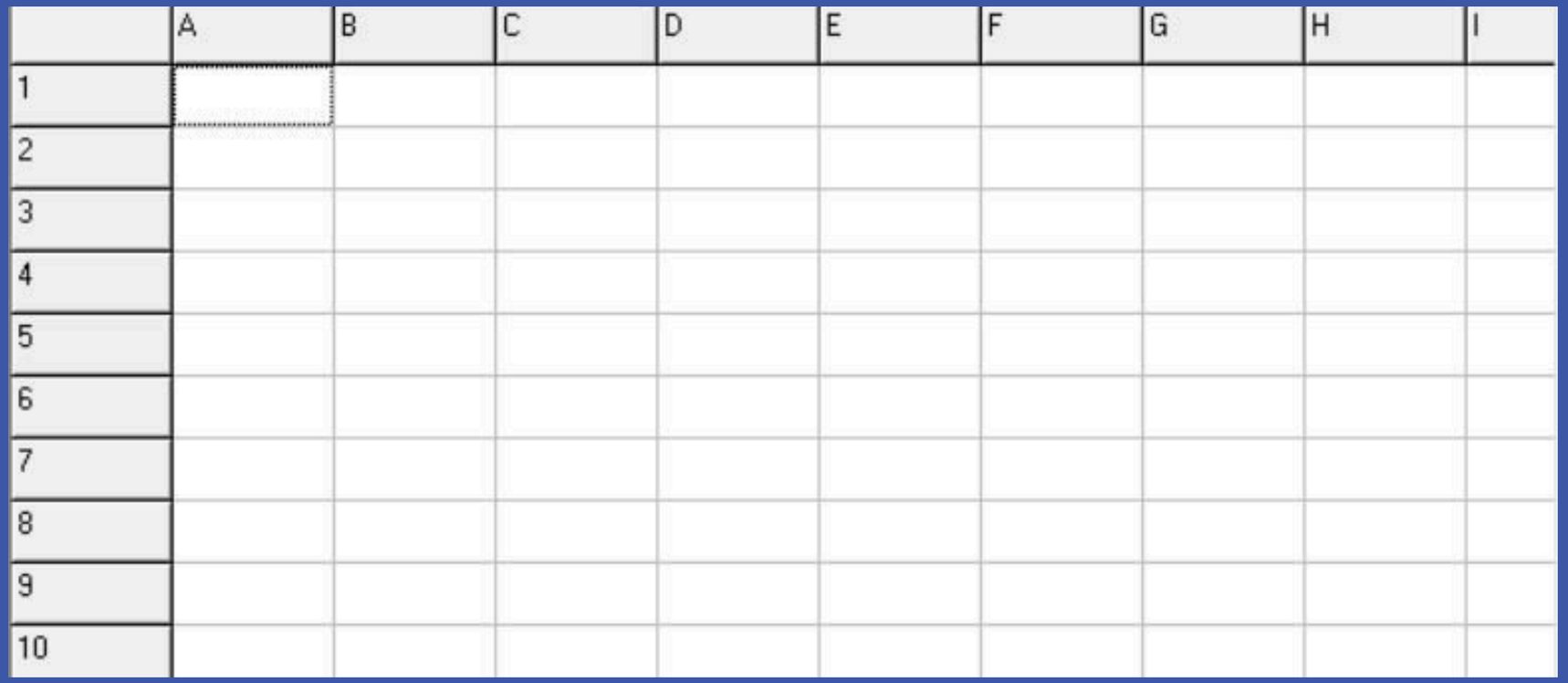
In [3]:
```
# However, Python also provides a method that makes this easier, it's the enumerate method
for index, friend in enumerate(friends):
    print(friend + " is friend number " + str(index))
```

```
Barrie is friend number 0
Tinus is friend number 1
Hans is friend number 2
```

In [4]:
```
# When you just want to count numbers instead of looping over a list you can use the range()
# function, this counts from one number to another, the default starting by zero
for number in range(5):
    print("Number " + str(number))
```

```
Number 0
Number 1
```

Multidimensional Lists



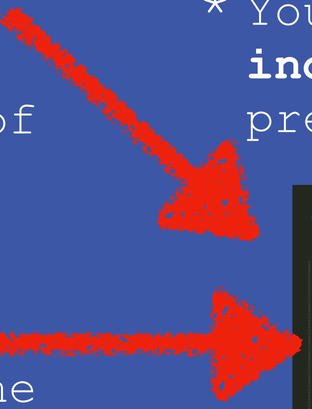
Multidimensional snacknames

Take your 'snacknames' program you wrote earlier and convert it to a multidimensional list. If you haven't finished your program yet, use the snacknames.py file in the Github repo.

* Create a multidimensional list that has the names of three friends
* Loop over this list and print the name of the friend and the length of his/her name
* In the same loop, ask for the favourite snack of this friend
* Add that snack to the list
* When the loop is done, loop over the list again and now print the name of your friend and what they like

Tips

* You only need **one** list variable in your program
* The **append()** method is handy here
* Indexing works the same in multidimensional lists, there's nothing special about multidimensional lists, it's just another value.
* You don't need to manually keep an **index** variable like you did previously.



```
friends = [
    ["Tinus", "Twix"],
    ["Barrie", "Oreo"],
    ["Hans", "Pie"]
]

print(friends[1][1]) # 'Barrie'
```

SLICING

Sentence slicer

Write a program that asks for a sentence and then shows the middle part of the sentence (which starts at 25% and ends at 75%).
E.g. ‘Hello, students’ -> ‘o, stud’

* Ask for a sentence
* Based on the length of the sentence, calculate the start and end character of the sliced sentence
* Create a new string that is sliced with those numbers
* Print it

Advanced exercises

* Convert the sentence to a list, **split()** by the space character
* Do the same thing as you did with the string: slice it from 25% to 75%
* **join** the new list together with the space character and print to the user
* Note how the two results differ

Tips for the basic exercise

* You will need the **len()** function
* You will probably need the **round()** function
* Remember that a percentage is just a fraction, or a floating point number

Files and the CSV format

CSV reader

Write a program that loads and nicely displays either the **footballers.csv** or **paintings.csv** file from the Github repo.

* Open the CSV file using the **open()** function
* Loop over the lines of the file using a **for** loop
* Split every line by **comma (“,”)**
* Assign the different list items to variables
* Print a nice sentence like “Interchange is a painting by Willem de Kooning and was sold for 300 million dollars”
* Close the file using the **close()** method

Advanced exercises

* Instead of printing the sentences, write them to a new text file
* Try formatting the prices to a number with the appropriate number of zeroes
* Ask for an extra footballer or painting and add that to the csv file

Tips for the basic exercise

* Look at the **examples-2 Notebook** for more examples
* Make sure your Python file and the csv file are in the same directory
* Use the **open()** function with a string like this: **open(“footballers.csv”)**
* A file is just like a list, you can iterate using the **in** operator
* You will need to use the **split()** string method to transform a line to a list
* Note that you can assign an indexed list item to any variable for easy reference
* Make sure to **close()** your file