Python II

Programmierkurs 2 Data Science
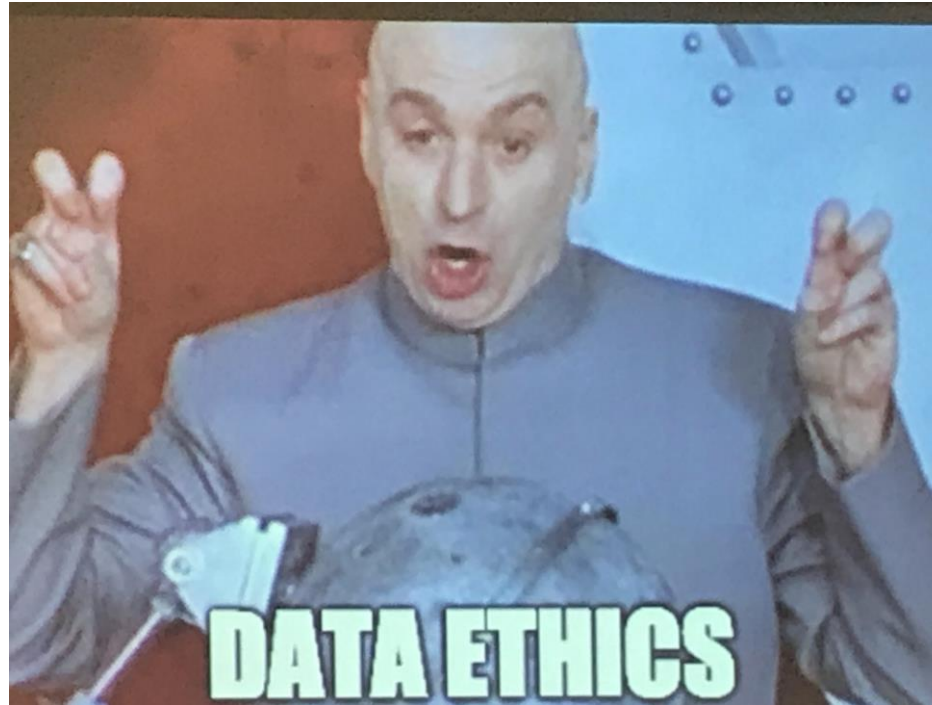## WS23/24

Leonard Traeger
M. Sc. Information Systems
leonard.traeger@fh-dortmund.de

Learning Goals Python II

- **Explain** the key concepts of Data Literacy and **demonstrate** its application.

- **List** components/steps of a data science life cycle framework and **outline** characteristics in which models differ in comparison.

- **Outline** the differences between the four container types in Python and **argue** which to use given an example.

- **Demonstrate** how index and values can be accessed, sliced, and iterated over respective containers.

- **Write** and define simple functions with single, multiple, and arbitrary numbers of variables as input.

- **Explain** Map, Filter, and Reduce functionalities and **provide examplarly** use-cases for each.

- **Apply** simple functions on containers via loop and list comprehensions with filtering.

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Ethics: should you care?

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Ethics (cont.)

A US art installation that will let people control a paintballing robot in a mock art gallery has been condemned by the firm that made the robo-dog.

Boston Dynamics criticised the project, calling it a "provocative use" of its quadruped robot, Spot.

It warned that if the "spectacle" goes ahead, Spot's warranty might be voided, meaning it could not be updated.

The group behind it, MSCHF, argues that Spot or robots like it will probably be used for military applications.

The group is known for creating viral stunts, stories and products.

The project, entitled Spot's Rampage, is due to start at 13:00 EST (18:00 GMT) on Wednesday, and will let people "remotely control a Spot robot" via a website.

https://www.bbc.com/news/technology-56182268

Forbes / Tech

FEB 16, 2012 @ 11:02 AM   2,998,353 VIEWS

How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

Kashmir Hill
FORBES STAFF ✔
Welcome to The Not-So Private Parts where technology & privacy collide

FULL BIO ›

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target TGT -0.43% , for example, has figured out how to data-mine its way into your womb, to figure out whether you have a baby on the way long before you need to start buying diapers.

Charles Duhigg outlines in the New York Times how Target tries to hook parents-to-be at that crucial moment before they turn into rampant — and loyal — buyers of all things pastel, plastic, and miniature. He talked to Target statistician Andrew Pole — before Target freaked out and cut off all communications — about the clues to a customer's impending bundle of joy. Target assigns every customer a Guest ID number, tied to

Target has got you in its aim

http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/#b228dae34c62, Retrieved 6/16/2016

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Ethics (cont.)

Ethical issues arise in practical applications, particularly when using **personal data.**

Data Science can lead to **impactful decisions** for **human beings**.

There are many important concerns about **fairness, privacy, security, abuse**...
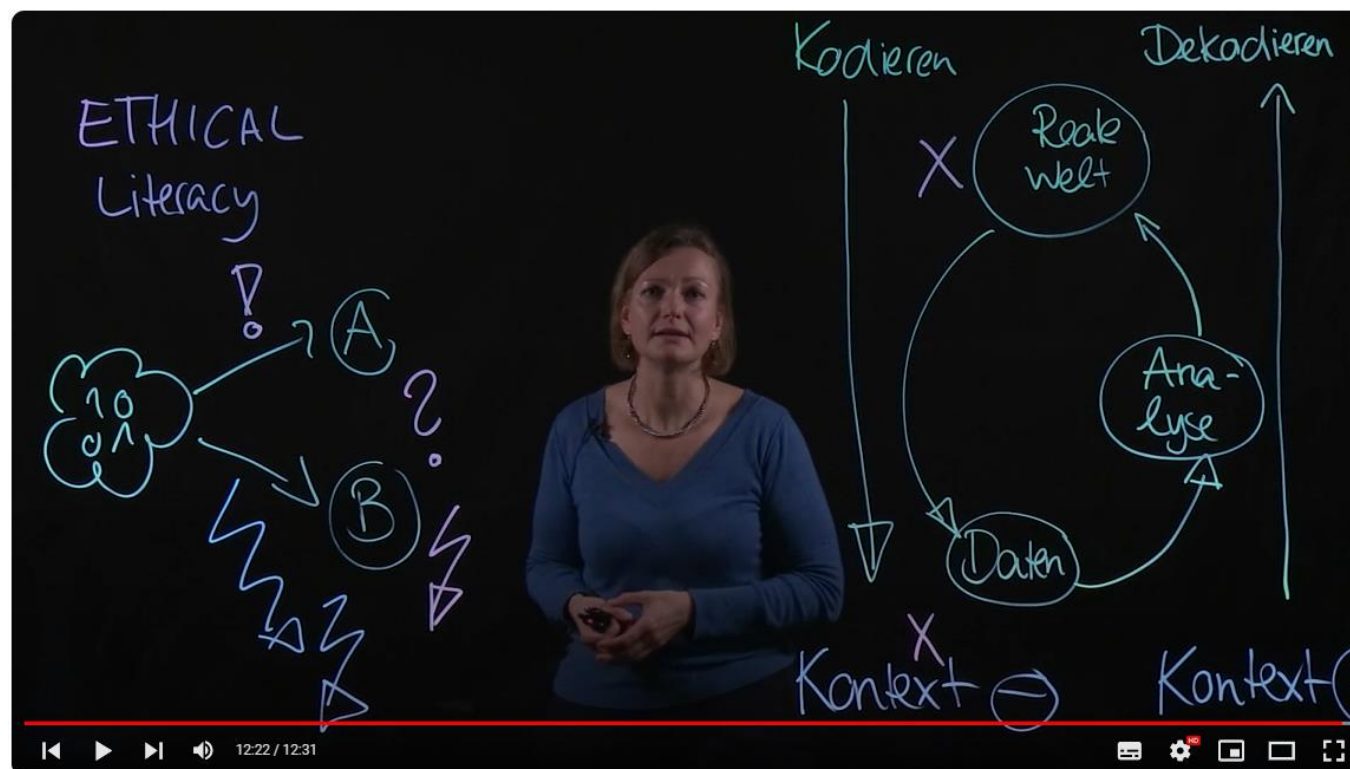
Can't we just anonymize the data?

- Anonymization is difficult.

- 85% of Americans can be identified from zip code, birth date and sex.

- The more dimensions / attributes / columns, the quicker you become unique!

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Some Common Principles in DS/AI Ethics

- Ensure safety

- Ensure fairness

- Respect privacy

- Promote collaboration

- Provide transparency

- Limit harmful uses

- Establish accountability

- Uphold human rights and values

- Reflect diversity / inclusion

- Avoid concentration of power

- Achknolwedge legal/policy implications

Programmierkurs 2 Data Science: Python II

**Technology
Arts Sciences
TH Köln**

Ethical and Data Literacy

Programmierkurs 2 Data Science: Python II

Data Literacy

...is the ability to **collect**, **manage**, **evaluate**, and **apply** data in a **critical way**.

- Competency of the 21st century which enables to act in a digitized world.

- *Systematical approach to turn data into knowledge.*

- *Data Science needs to be **planned** to consciously use and question it in the **respective context**.*

Skill categorization: conceptional (blue), core (green), advanced (red)

Conceptual Framework	Introduction to Data
Data Collection	Data Discovery and Collection
	Evaluating and Ensuring Quality of Data and Sources
Data Management	Data Organization
	Data Manipulation
	Data Conversion
	Metadata Creation and Use
	Data Curation, Security and Re-Use
	Data Preservation
Data Evaluation	Data Tools
	Basic Data Analytics
	Data Interpretation (Understanding Data)
	Identifying Problems Using
	Data Visualization
	Presenting Data (Verbally)
	Data Driven Decisions Making (DDDM)
Data Application	Critical Thinking
	Data Culture
	Data Ethics
	Data Citation
	Data Sharing
	Evaluating Decisions based on Data

Abbildung 1: Data-Literacy-Kompetenzen nach Ridsdale et al. (2015)

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Data Literacy (cont.)

1. What do I want to do with data?
   - Data and its analysis are not an end in themselves (Selbstzweck).
   - *Target a concrete use case or application.*

2. What can I do with data?
   - The technical and methodological possibilities play a crucial role.
   - *Become aware of your capabilities.*

3. What am I allowed to do with data?
   - Legal regulations governing the use of data.
   - *Consider what you are allowed or at least not allowed to do.*

4. What should I do with data?
   - Data is a valuable resource which can create, beyond legally permitted actions, something good for society.
   - *Consider your personal and societal benefit of your application.*

https://ki-campus.org/sites/default/files/2021-10/data-literacy-charta_v1_2.pdf

Programmierkurs 2 Data Science: Python II

Think-Pair-Share #1
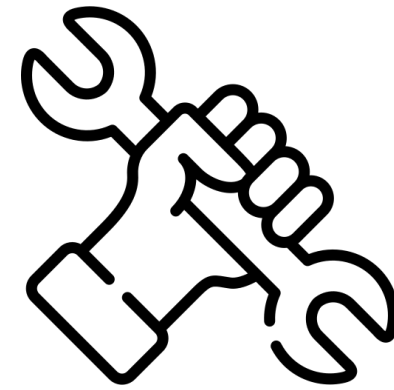
Suppose you have access to a data set from a **large multi-million-$$$ fast-food chain**.

The data set provides information about the personal information of
customers (age, weight, eating behaviour, …)
purchases (date, price, meals, …)
meal information (calories, sugar, fat, …).

Discuss with your neighbouring peers (group of two to three):

1) What do I want to do with data?

2) What can I do with data?
Imagine your technical and methodological capabilities as a multi-million-$$$ company.

3) What am I allowed to do with data?

4) What should I do with data?

Think-Pair-Share #1
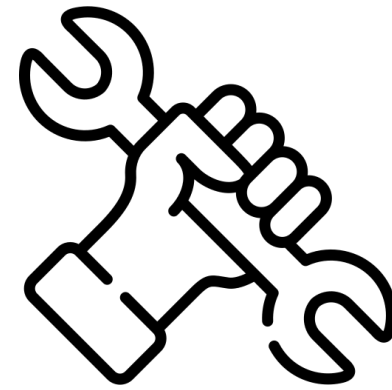
1) What do I want to do with data?
   - *Gesünderes Essen, Verschwendung verhindern, Integration fremder Daten, Neue Zielgruppen erschließen, Wartezeiten verkürzen, Expansion, Profit $$$, Neue Rezeptur*

2) What can I do with data?
   - *Infrastruktur, Analyse, Visualisieren (Tableu, Python), Prognose (KI, Beratung)*

3) What am I allowed to do with data?
   - *Persönliche Daten nicht verkaufen (DSGVO), Targeting*

4) What should I do with data?
   - *Zutaten transparent darstellen, vor Risiken warnen*

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

What's next?

We know what we

- want

- can

- are allowed to, and

- should do.

But how do we start a Data Science project?

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Data Science Life Cycle

- Iterative set of data science components to plan and deliver a project.

- Differences in
  - Non-linearity versus linearity.
  - Several smaller steps versus larger comprehensive phases.
  - Data-centric versus business-understanding.

Every data science project and team are different;
you must adapt your own version.

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Knowledge Discovery in Databases
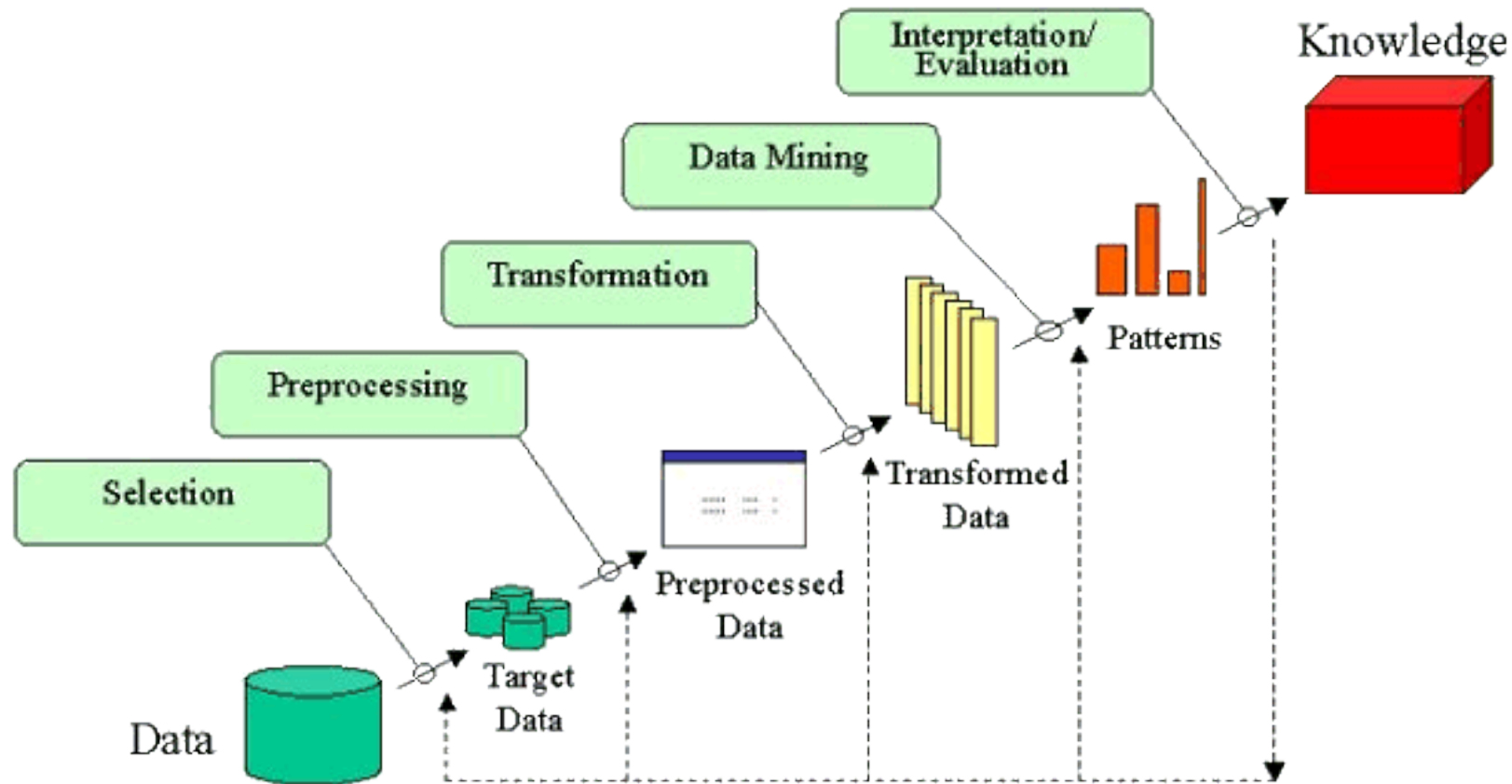
KDD, Fayyad et al., 1996: *https://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html*

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

CRISP-DM
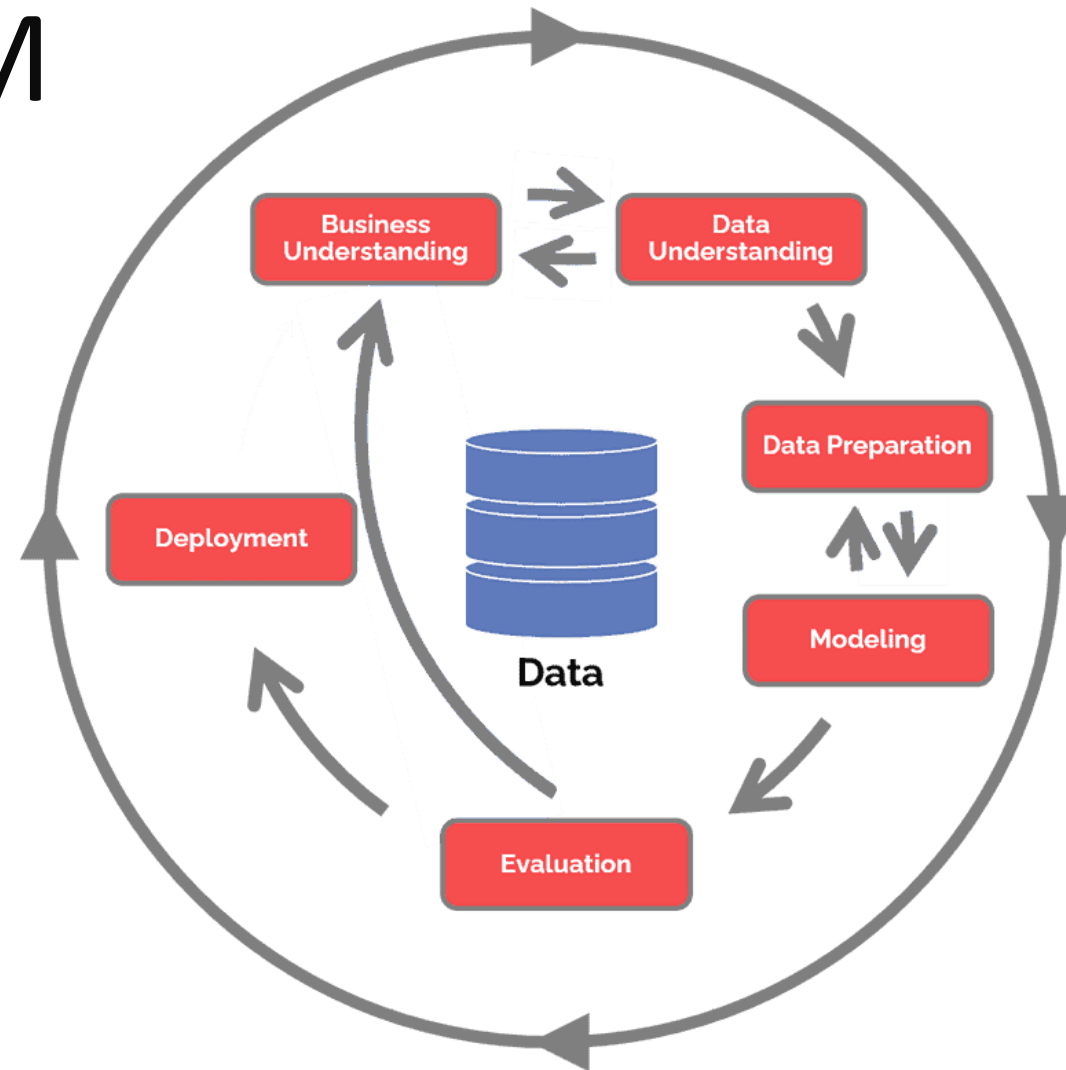
Cross-industry standard process for data mining, DS Process Allieance, 1999: *https://www.datascience-pm.com/crisp-dm-2/*

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences
TH Köln

OSEMN

Data Science Process

OBTAIN	SCRUB	EXPLORE	MODEL	INTERPRET
O	**S**	**E**	**M**	**N**
Gather data from relevant sources	Clean data to formats that machine understands	Find significant patterns and trends using statistical methods	Construct models to predict and forecast	Put the results into good use

Originally by Hilary Mason and Chris Wiggins
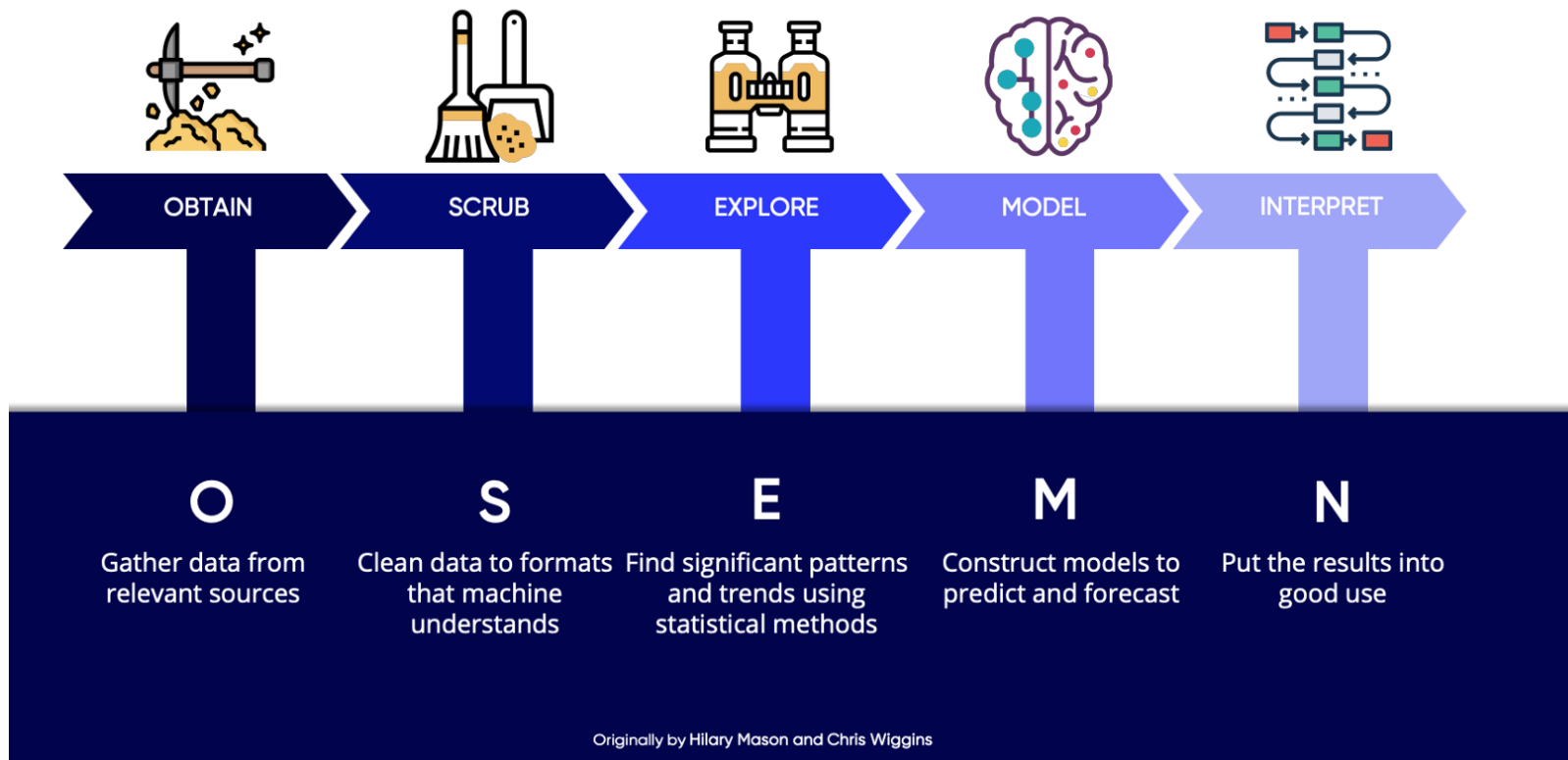
OSEMN: Mason and Wiggins, 2010: https://www.datascience-pm.com/osemn/
Image source: https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences
TH Köln

Transformation Pipeline

1st • Acquire Data

2nd • Extract Information

3rd • Gather Knowledge

4th • Transfer Knowledge Into Actions

Programmierkurs 2 Data Science: Python II

**Technology
Arts Sciences
TH Köln**
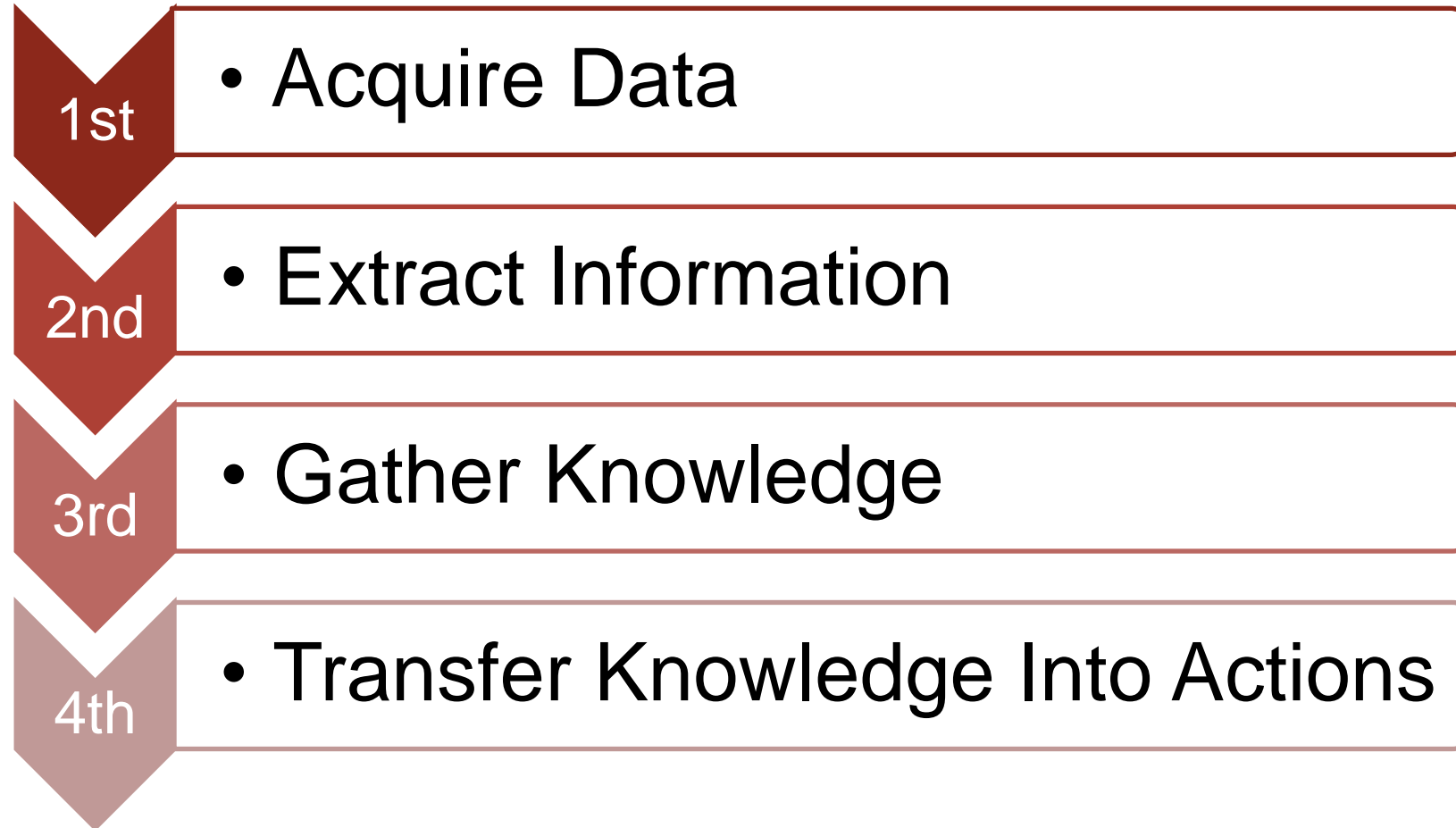
...back to **Data Science Literacy**

Programmierkurs 2 Data Science: Python II

Comparison and Logical Operators

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal
!=	not equal
and	both must be true
or	one or both must be true
not	reverses the truth value

Technology
Arts Sciences
TH Köln

Control Statement

- Example: Test if a is greater than b

- Python's `elif` is C's `else if`

```python
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
elif a > 201 or not( b > 33):
  print("something else")
else:
  print("a is greater than b")
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln
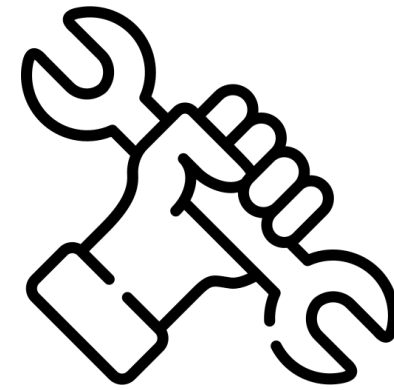
Training #1

1. Generate a random number representing the number of points a student scored in an exam between zero and 116.
   You can import the random int generator with
   ```
   from random import randint as ri
   ```
   and generate random numbers with
   ```
   ri(start_range, end_range).
   ```

2. Write a control statement that outputs a final grade given the randomly generated number. Distinct only between round grades (1,2,3,4, and failed) in this exercise.

Punkte	Note	
116 - 94,9 %	1,0	
<94,9 - 89,5 %	1,3	1
<89,5 - 84,3 %	1,7	
<84,3 - 79,0 %	2,0	
<79,0 - 73,7 %	2,3	2
<73,7 - 68,2 %	2,7	
<68,2 - 63,1 %	3,0	
<63,1 - 57,9 %	3,3	3
<57,9 - 52,6 %	3,7	
<52,6 - 50,0 %	4,0	4
< 50,0 %	n.b.	F

Technology
Arts Sciences
TH Köln

Containers / Collections

- How can we efficiently store data elements in Python?

- How can we efficiently access these?

		Ordered	Changeable	Indexed	Duplicates
List	[ ]	Yes	Yes	Yes	Yes
Tuple	( )	Yes	No	Yes	Yes
Set	{ }	No	Yes	No	No
Dictionary	{"_:_"}	No	Yes	Yes	No

player_score[0,1,3,3]   tweet = {'#cat', '#cute', '#lol'}

               player_score(0,1,3,3)   eng_ger = {'I':'Ich', 'learn':'lernen'}

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Python List Example

```
veg = [['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'cilantro', 'peppers', 'zucchini']]
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Python List Example (cont.)

```
veg = [['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'cilantro', 'peppers', 'zucchini']]
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Python List Example (cont.)

```python
veg = [['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'lettuce', 'peppers', 'zucchini'],
       ['lettuce', 'cilantro', 'peppers', 'zucchini']]
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

List [ ] https://docs.python.org/3.5/tutorial/datastructures.html#more-on-lists

A list is the Python equivalent of an array, but is resizable and can contain elements of different types:

```python
x_list = ["Zuckerberg", "Musk", 42]     # Create a list
print(x_list, x_list[2])  # Prints "['Zuckerberg', 'Musk', 42] 42"
print(x_list[-1])         # Negative indices count from the end of the list;
                          # prints "42"
x_list[2] = 'Gates'       # Lists can contain elements of different types
print(x_list)             # Prints "['Zuckerberg', 'Musk', 'Gates']"
x_list.append('Bezos')    # Add a new element to the end of the list
print(x_list)             # Prints "['Zuckerberg', 'Musk', 'Gates', 'Bezos']"
x = x_list.pop()          # Remove and return the last element of the list
print(x, x_list)          # Prints "Bezos ['Zuckerberg', 'Musk', 'Gates']"
```

More list methods append(), clear(), copy(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort() to find in documentation.

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

List [ ] Slicing

Python provides concise syntax to access sublists (instead one element at the time):

```python
nums = list(range(5))      # range is a function creating a list of integers
print(nums)                # Prints "[0, 1, 2, 3, 4]"
print(nums[2:4])           # Get a slice from index 2 to 4 (exclusive);
                           # prints "[2, 3]"
print(nums[2:])            # Get a slice from index 2 to the end;
                           # prints "[2, 3, 4]"
print(nums[:2])            # Get a slice from the start to index 2 (exclusive);
                           #  prints "[0, 1]"
print(nums[:])             # Get a slice of the whole list;
                           # prints "[0, 1, 2, 3, 4]"
print(nums[:-1])           # Slice indices can be negative;
                           # prints "[0, 1, 2, 3]"
nums[2:4] = [8, 9]         # Assign a new sublist to a slice
print(nums)                # Prints "[0, 1, 8, 9, 4]
```

Technology
Arts Sciences
TH Köln

List [ ] Loop

You can loop over the elements of a list like this:

```python
animals = ['cat', 'dog', 'monkey']
for animal in animals:
    print(animal)
# Prints "cat", "dog", "monkey", each on its own line.
```

If you want access to the **index** of **each element** within the body of a loop, use the built-in **enumerate** function:

```python
animals = ['cat', 'dog', 'monkey']
for idx, animal in enumerate(animals):
    print('#%d: %s' % (idx + 1, animal))
# Prints "#1: cat", "#2: dog", "#3: monkey", each on its own line
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

List Comprehensions

- Frequently, we want to transform one type of data into another.

Example: compute price per square apartment (12€ / square)

```python
square_meters = [30, 40, 50, 60, 70]
square_prices = []
for square_meter in square_meters:
    square_prices.append(square_meter * 12)
print(square_prices)   # Prints [360, 480, 600, 720, 840]
```

- Make this code simpler using a list comprehension:

```python
square_meters = [30, 40, 50, 60, 70]
square_prices = [x * 12 for x in square_meters]
print(square_prices)   # Prints [360, 480, 600, 720, 840]
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

List Comprehensions (cont.)

- Can also contain conditions.

Example: compute price per square apartment for **small** apartments (15€ / square)

```python
square_meters = [30, 40, 50, 60, 70]
square_prices = [x * 15 for x in square_meters if x < 51]
print(square_prices)  # Prints "[450, 600, 750]"
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Tuples ( )

- Consists of **heterogenous sequences** and values separated by commas.

- **Unchangeable** ordered <u>list</u> of values.

```python
t = ('Zuckerberg', 42, 'Gates', 'Bezos') # Create a tuple
print(type(t))    # Prints "<class 'tuple'>"
print(t.count('Zuckerberg')) # Prints "1"
print(t.index('Zuckerberg')) # Prints "0"
print(t[0])                  # Prints "Zuckerberg"
# Tuples may be nested with heterogeneous sequences:
t2 = t, ('Musk','Cooper')
print(t2)         # Prints "(('Zuckerberg', 42, 'Gates', 'Bezos'),
                  #          ('Musk', 'Cooper'))"
t[1] = 'Musk'     #  TypeError: 'tuple' object does not support item assignment
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Set { } https://docs.python.org/3.5/library/stdtypes.html#set

- Unordered collection of distinct elements.

```python
s = {'Zuckerberg', 42, 'Gates', 'Bezos'}
print('Zuckerberg' in s)  # Check if an element is in a set; prints "True"
print('Musk' in s)        # prints "False"
s.add('Musk')             # Add an element to a set
print('Musk' in s)        # Prints "True"
print(len(s))             # Number of elements in a set; prints "5"
s.add('Musk')             # Adding an existing element does nothing
print(len(s))             # Prints "5"
s.remove('Musk')          # Remove an element from a set
print(len(s))             # Prints "4"
```

- More set methods add(), clear(), copy(), difference_update(), discard(), issubset(), issuperset(), remove(), union(), update() to find in documentation.

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Set { } Loop

- Iterating over a set has the same syntax as iterating over a list (loop, comprehensions).

- Since sets are unordered, you **cannot make assumptions** about the **order** in which you visit the elements of the set.

```python
s = {'Zuckerberg', 42, 'Gates', 'Bezos', 'Musk'}
for idx, answer in enumerate(s):
    print('#%d: %s' % (idx + 1, answer))
# Prints
#1: Bezos
#2: Gates
#3: Zuckerberg
#4: Musk
#5: 42
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences
TH Köln

Dictionary {"_":"_"}

- A dictionary stores (key, value) pairs.

- Duplicate keys will overwrite existing values!

```python
d = {'Tsd.': '000', 'Mil.': '000000'}  # Create a new dictionary with some data
print(d['Tsd.'])        # Get an entry from a dictionary; prints "000"
print('Tsd.' in d)      # Check if a dictionary has a given key; prints "True"
d['Bil.'] = '000000000'         # Set an entry in a dictionary
print(d['Bil.'])                # Prints "000000000"
#print(d['Tril.'])              # KeyError: 'Tril.' not a key of d
print(d.get('Tril.', 'N/A')) # Get an element with a default; prints "N/A"
print(d.get('Bil.', 'N/A'))  # Get an element with a default; prints "000000000"
del d['Bil.']                   # Remove an element from a dictionary
print(d.get('Bil.', 'N/A'))  # "Bil." is no longer a key; prints "N/A"
```

- More dictionary methods clear(), copy(), fromkeys(), get(), items(), keys(), pop(), popitem(), setdefault(), update(), values() to find in documentation.

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Dictionary {"_":"_"} Loop

- You can loop over the keys in a dictionary.

```python
d = {'person': 2, 'cat': 4, 'spider': 8}
for animal in d:
    legs = d[animal]
    print('A %s has %d legs' % (animal, legs))
# Prints "A person has 2 legs", "A cat has 4 legs", "A spider has 8 legs"
```

- If you want access to **keys and** their corresponding **values**, use the **items method**:

```python
d = {'person': 2, 'cat': 4, 'spider': 8}
for animal, legs in d.items():
    print('A %s has %d legs' % (animal, legs))
# Prints "A person has 2 legs", "A cat has 4 legs", "A spider has 8 legs"
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Think-Pair-Share #2

|		Ordered	Changeable	Indexed	Duplicates
List	[ ]	Yes	Yes	Yes	Yes
Tuple	( )	Yes	No	Yes	Yes
Set	{ }	No	Yes	No	No
Dictionary	{"_:_"}	No	Yes	Yes	No
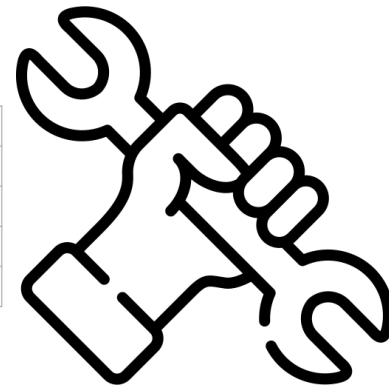
Discuss with your neighbouring peers (groups of two to three):

What container / collection type would suite

- 100 famous dinosaur species?

- All best movie grammy award winners until today?

- Shirt numbers and corresponding player names in a football league?

- Abreviations of all airports and their full name?

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Think-Pair-Share #2

		Ordered	Changeable	Indexed	Duplicates
List	[ ]	Yes	Yes	Yes	Yes
Tuple	( )	Yes	No	Yes	Yes
Set	{ }	No	Yes	No	No
Dictionary	{"_:_"}	No	Yes	Yes	No
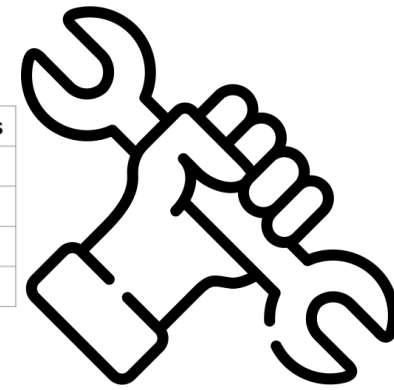
- 100 famous dinosaur species?
  - List, set

- All best movie grammy award winners?
  - Nested list or dictionary by year, set if number of grammies unrelated

- Shirt numbers and corresponding player names in a large tennis league?
  - Nested dictionary with team name: number, nested dictionary or list by player number

- Abreviations of all airports and their full name?
  - Dictionary

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Functions

- One of the most useful programming constructions.

- Recognize function calls by **( )** after the function name.

- Functions take **0** or **more arguments**.

- Some arguments are required, others have default values and can be omitted.

- Use the `def` keyword to **def**ine a **new function**.

```python
def isClassPassed(numberOfPoints):
    if numberOfPoints >= 50:
        return 'passed'
    else:
        return 'failed'
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Functions (cont.)

- Allow you to name and reuse blocks of code.

- Help you to **break complex problems** into **simpler parts**.

- Make your code more readable.

- Rule of thumb: if you copy-and-paste the same code more than once, it's probably better to encapsulate that code into a function.

```python
def isClassPassed(numberOfPoints):
    if numberOfPoints >= 50:
        return 'passed'
    else:
        return 'failed'

for pointsScored in [55, 12, 66, 1, 99]:
    print(isClassPassed(pointsScored))
#Prints "passed, failed, passed, failed, passed"
```

Programmierkurs 2 Data Science: Python II

**Technology
Arts Sciences
TH Köln**

Functions (cont.)

- Functions can become very complex and nested.

- Documentation essential.

```python
#Here's a template from Spyder - use this template every time.
def func(x, y = 0, *z, **j):
    """
    Describe your function's purpose concisely.
    Parameters
    ----------
    x : TYPE  DESCRIPTION.
    y : TYPE, optional DESCRIPTION. The default is 0.
    z : TYPE, arbitrary tuple of arguments.
    (If unkown/changing number of arguments)
    j : TYPE, arbitrary keyword arguments.


    Returns
    ----------


    """
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Functions & List Comprehensions

```python
def isClassPassed(numberOfPoints):
    if numberOfPoints >= 50:
        return 'passed'
    else:
        return 'failed'
```

- Both NumPy and Pandas become powerful with list comprehensions and functions.

```python
scoredPoints = [55, 12, 66, 1, 99];
classPassed = [isClassPassed(x) for x in scoredPoints]
print(classPassed)
#Prints "['passed', 'failed', 'passed', 'failed', 'passed']"
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Functions & List Comprehensions & Filtering

- Most "pythonic" element of Python syntax after *Lex Fridman.*

- Syntax:`[f(x) for x in container if g(x)]`

```python
def isClassPassed(numberOfPoints):
    if numberOfPoints >= 50:
        return 'passed'
    else:
        return 'failed'
```

```python
scoredPoints = [55, 12, 66, 1, 99, 300, -1];
classPassed = [isClassPassed(x) for x in scoredPoints if x in range(100)]
print(classPassed)
#Prints "['passed', 'failed', 'passed', 'failed', 'passed']"
```

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln
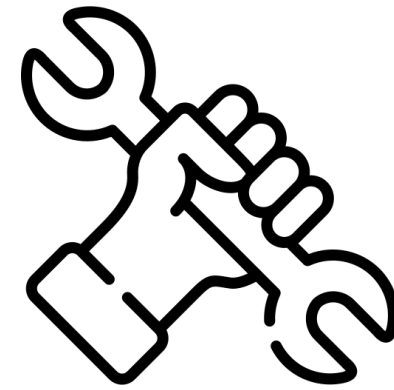
Training #2

1. Generate an array with 100 random numbers representing the number of points students scored between zero and 116 and store these in a suitable container.
   You can import the random int generator with
   ```
   from random import randint as ri
   ```
   and generate random numbers with
   ```
   ri(start_range, end_range).
   ```

2. Write a function that outputs a final grade given scored points as input. Distinct only between round grades (1,2,3,4, and failed) in this exercise.

3. Apply this function to the container storing all points (the 100 students have scored) and store these in a new container.

Punkte	Note	
116 - 94,9 %	1,0	
<94,9 - 89,5 %	1,3	1
<89,5 - 84,3 %	1,7	
<84,3 - 79,0 %	2,0	
<79,0 - 73,7 %	2,3	2
<73,7 - 68,2 %	2,7	
<68,2 - 63,1 %	3,0	
<63,1 - 57,9 %	3,3	3
<57,9 - 52,6 %	3,7	
<52,6 - 50,0 %	4,0	4
< 50,0 %	n.b.	F

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Map, Filter, and Reduce

- List comprehensions can fulfill almost all data transformation tasks with functions.

- But, in high degrees of
  - Data volume
  - Distributed storage
  - Parallel computation
  - Fault tolerance

- ...a different parallel computing topology has been invented.

> Choose a preference between comprehension list and map+filter+reduce style but understand both!

MapReduce: Simplified Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat, 2004, Google Inc.
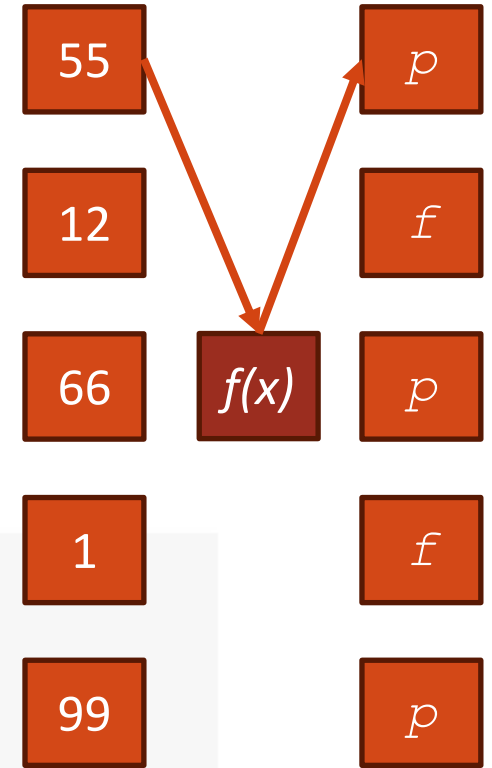
Technology
Arts Sciences
TH Köln

Map `map(func, *iterables)`

- `func` is the function on which each element in `iterables` is applied on.

- The output is a `map object` containing a list.

- To get the result as a list, use the built-in `list( )` function.

```python
def isClassPassed(numberOfPoints):
    if numberOfPoints >= 50:
        return 'passed'
    else:
        return 'failed'
```

```python
scoredPoints = [55, 12, 66, 1, 99];
classPassed = list(map(isClassPassed, scoredPoints))
print(classPassed)
#Prints "['passed', 'failed', 'passed', 'failed', 'passed']"
```

55		p
12		f
66	f(x)	p
1		f
99		p

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Map `map(func, *iterables)` (Cont.)

- What did we win? Flexibility and fault-tolerance.

- As the output of the map function is a `map object`.

```python
scoredPoints = [55, 12, 66, 1, 99];
classPassed = list(map(isClassPassed, scoredPoints))
print(classPassed)
#Prints "['passed', 'failed', 'passed', 'failed', 'passed']"
```

- We can reuse it cascadingly.

```python
scoredPoints = [55, 12, 66, 1, 99];
classPassed2 = list(map(str.upper, map(isClassPassed, scoredPoints)))
print(classPassed2)
#Prints "['PASSED', 'FAILED', 'PASSED', 'FAILED', 'PASSED']"
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Filter `filter(func, iterable)`

- `func` is the function on which each element in `iterable` is filtered through.

- The output is a `map object` containing a list with only `boolean = true` filtered elements.

- To get the result as a list, use the built-in `list( )` function.

```python
def onlyClassPassed(scoredPoints):
    return scoredPoints > 50
```

```python
scoredPoints = [55, 12, 66, 1, 99];
over50scoredPoints = list(filter(onlyClassPassed, scoredPoints))
print(over50scoredPoints)
#Prints "[55, 66, 99]"
```
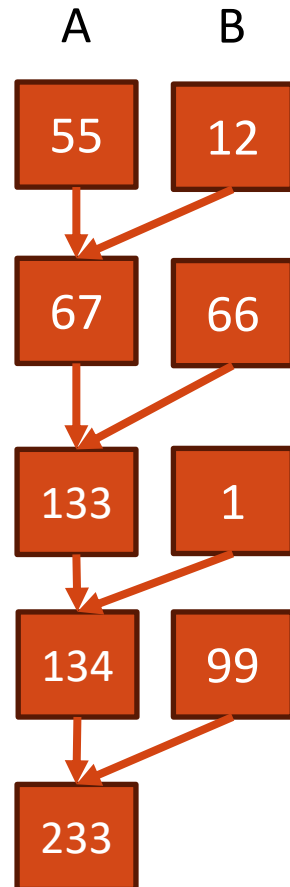
55

12

66

1

99

Programmierkurs 2 Data Science: Python II

**Technology
Arts Sciences
TH Köln**

Reduce `reduce(func, iterable[,initial])`

A   B

| 55 | 12 |

| 67 | 66 |

- `func` is the function on which each element in the iterable gets **cumulatively applied.**

- `initial` is the optional value that gets placed before the elements of the iterable in the calculation and serves as a default when the iterable is empty.

| 133 | 1 |

- The output is a single value (name source: many values "reduced" to single).

| 134 | 99 |

```python
def sumScoredPoints(scoredPointsA, scoredPointsB):
    return scoredPointsA + scoredPointsB
```

| 233 |

```python
scoredPoints = [55, 12, 66, 1, 99];
print(reduce(sumScoredPoints, scoredPoints))
#Prints "233"
```

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences TH Köln

Takeaways

- Adapting your project to an existing data science framework leads to a comprehensible and reproducible workflow.

- Follow ethical principles ☺

- Python's readability, easily defined functions, diverse containers, and container loops including list comprehensions make Python as a programming language loved by developers… and hopefully you too.

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Project Milestone #1

RACI-Matrix
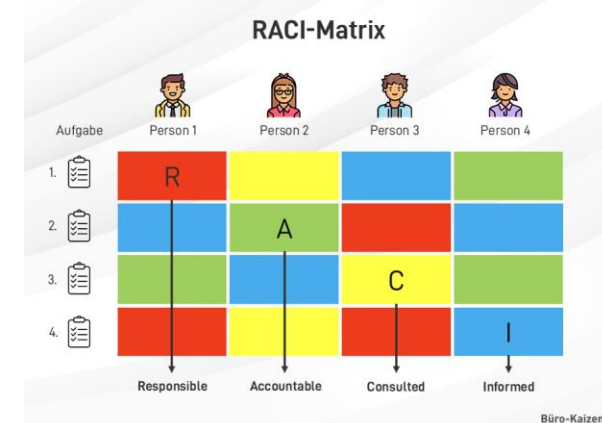
- Formed teams and a teamname (e.g., HealthyFoodies, Sportineers, ...be creative).

- Declared a domain problem everyone finds interesting in your team and found an official source to justify and motivate others.

- Chosen and summarized a Data Science Life Cycle Framework which you believe would fit your data science project. You roughly estimated in distribution of total 100% which part you want to emphasize on.

- Chosen and summarized data literacy competencies after Ridsdale (2015) you believe will become part in your project. You have planned as a team who is going to be responsible, accountable, consulted, and informed for which skill set.

- Defined collaborative tools and a rough meeting structure so everyone within the team is able to code Python, share, and document files.

- **Planning usually goes wrong but it is important to start somewhere!**

Programmierkurs 2 Data Science: Python II

Technology
Arts Sciences
TH Köln

Outlook

- In next week 4 we will dive deep into object-oriented programming with Python.

- Even though we can do a lot with functions and containers, repetitive tasks can be easily outsourced to constructors, destructors, decorator annotated and regular class methods with inheritance.

```python
## let's define player1, player2, player3
player1 = Player(player_number = 1, player_name = 'Gregor Kobel',
                 player_value = "35,00 Mio. €")

player2 = Player(player_number = 35, player_name = 'Silas Ostrzinski',
                 player_value = "150 Tsd. €")

player3 = Player(player_number = 1, player_name = 'Marc-André ter Stegen',
                 player_value = "35,00 Mio. €", club_name="FC Barcelona")
```


- We will also peak into common data formats CSV, JSON, and XML and how some of them can help us to generate datasets by ourselves.

Technology
Arts Sciences
TH Köln

See you again next week **online**!

Questions?

Programmierkurs 2 Data Science: Python II

Technology Arts Sciences
TH Köln