Programming languages

- Two types of programming languages
- Compiler (C, Fortran, Java)
  - Source file (main.c), compile
  - Object file (main.obj), link
  - Execution file (main.exe)
- Interpreter (R, Python, VBA)
- Artificial Intelligence, Machine Learning
  - keras, tensorflow
- Data analysis
  - pandas
- Install Anaconda
  - www.anaconda.com
  - Python + some basic packages + Jupyter notebook

Textbook

“Python Crash Course” 2nd, by Eric Matthes, 2019

Ch 1 : Getting started

Ch 2 : Variables and simple data types

Ch 3 : Introducing lists

Ch 4 : Working with lists

Ch 5 : If statements

Ch 6 : Dictionaries

Ch 7 : User input and while loops

Ch 8 : Functions

Ch 9 : Classes

Ch 10 : Files and exceptions

Ch 1: Getting started

- Checking the list of installed packages
- Checking the version of installed packages
- Installing a package
- Uninstalling a package
- Running python in a terminal session
- Running a python program in a python session
- Editing a python file
- Running a python file

pip (package installer for python)

```
C:\users\user>pip list
    (Package)    (Version)
    gym
    Keras
    matplotlib
    numpy
    pandas
    scikit-learn
    scipy
    tensorflow
```

```
pip --version
pip install --upgrade pip
pip --help
```

```
C:\users\user>pip uninstall gym
C:\users\user>pip install gym
```

Ch 2: Variables and simple data types

- Function
- Argument
- String
- Variable
- Value
- Assignment operator
- Assignment statement
- Date type
- Method
- String concatenation
- f-string

- Arithmetic operator
- Comparison operator
- Logical operator
- Expression
- Boolean expression

Data types

int

float

str

bool (ch 5)

list (ch 3, ch 4)

tuple (ch 4)

set

dict (ch 6)

range (ch 4)

date

array (numpy)

series (pandas)

dataframe (pandas)

tensor (tensorflow)

....

Arithmetic/Comparison/Logical operator

```
5 + 2
5 - 2
5 * 2
5 / 2
5 ** 2
5 // 2
5 % 2
```

```
5 > 2
5 >= 2
5 < 2
5 <= 2
5 == 2
5 != 2
```

```
1 == 1 and 5 > 2
1 == 0 and 5 > 2
1 == 0 or 5 > 2
not 5 > 2
```

Expression

An entity in a programming language that may be evaluated to determine its value

Boolean Expression

Ch 2: review

Python Functions

print()
type()
id()
len()
sum()
max()
min()

String Methods

title()
upper()
lower()
rstrip()
lstrip()
strip()

Data types (int, float, str, bool, list, tuple, set, dict)

(range, date, timedelta, array, series, dataframe, tensor, ….)

function, argument

variable, value, assignment operator

string, character, string concatenation, f-string

arithmetic/comparison/logical operators

Ch 3: Introducing Lists

- Accessing elements in a List
- Changing, adding, and removing elements
- del statement
- Sorting a List permanently with the sort() method
- Sorting a List temporarily with the sorted() function
- Reversing the order of a List permanently
- Return value of a function
- Return value of a method

Ch 3: review

Python Functions

print()

type()

id()

len()

sum()

max()

min()

sorted()

list()

int()

float()

str()

set()

List Methods

index()

append()

insert()

pop()

remove()

sort()

reverse()

string + string

list + list

list * int

del statement

return value of a function

return value of a method

None

Ch 4: Working with Lists

- Looping through an entire List
  - for loop
  - indentation
  - block
- range() function
- List comprehension
- Working with part of a list
  - Slicing
- Copying a list
- Tuple
- Mutable and immutable

Ch 4: review

Python Functions

print()

type()

id()

len()

sum()

max()

min()

sorted()

int()

float()

str()

set()

range()

list() str → list, range → list

for loop

indentation

block

range() function

data type (range)

list comprehension

slicing

copying a list (b = a[:], b = a)

tuple

mutable, immutable

character ‘\n’

Ch 5: If statements

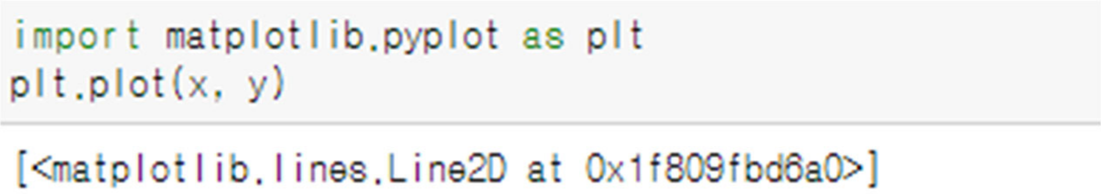
- If statement
- Conditional test
- Checking whether a value is in a list
- Checking whether a value is not in a list
- If-else statement
- If-elif-else statement
- Multiple elif blocks
- Checking that a list is not empty

Ch 6: Dictionaries

- Accessing values in a dictionary
- Adding new key-value pairs
- Modifying values in a dictionary
- Removing key-value pairs
- Using get() method to access values
- Looping through a dictionary
- Nesting

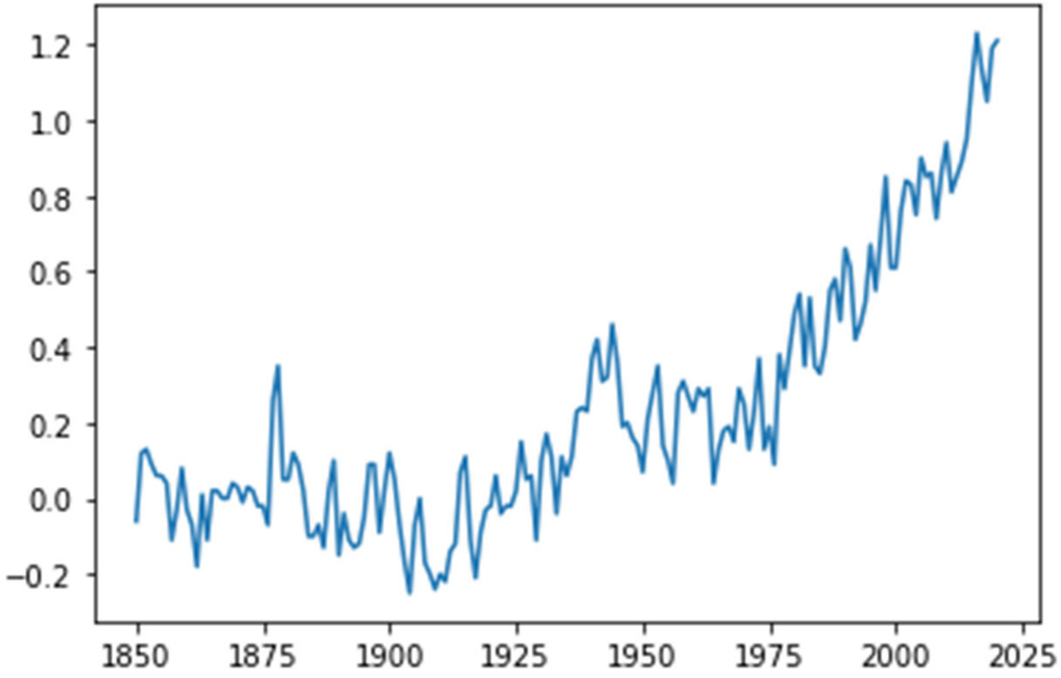
List of dictionaries

https://public.wmo.int/en/our-mandate/climate/wmo-statement-state-of-global-climate



```
import matplotlib.pyplot as plt
plt.plot(x, y)
```

```
[<matplotlib.lines.Line2D at 0x1f809fbd6a0>]
```



Ch 6: review

Dictionary Methods

get()

items()

keys()

values()

Looping through a dictionary

```
for key, value in dict.items():
for key in dict.keys():
for value in dict.values():
```

Nesting

- List of dictionaries
- A list in a dictionary
- A dictionary in a dictionary

Ch 7: User input and while loops

- How the input() function works?
- Introducing while loops
- Using a flag
- Using break to exit a loop
- Using continue in a loop
- Using while loops with a list
- Removing all instances of specific values from a list

Ch 7: review

Python Functions

print()
type()
id()
len()
sum()
max()
min()
sorted()
int()
float()
str()
set()
range()
list()
input()

while loop
flag
break
continue

Ch 8: Functions

- Built-in function
- User-defined function
- Positional arguments
- Keyword arguments
- Making an argument optional
- Global variable, local variable
- Passing a list
- Passing an arbitrary number of arguments
- Module
- Built-in module

Ch 8: review

Built-in Functions

print()
type()
id()
len()
sum()
max()
min()
sorted()
int()
float()
str()
set()
range()
list()
input()
dir()

```
import pizza
import pizza as p
from pizza import make_pizza
from pizza import make_pizza as mp
```

Built-in Modules

```
import math
math.exp()
math.log()
```

Ch 9: Classes

- Creating a class
- __init__() method
- Making an instance from a class
- Object oriented programming
- Methods and attributes
- Parent class and child class
- Super class and sub class
- Inheritance
- Override
- Importing classes

class Dog

```
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age

my_dog = Dog('Willie', 6)
your_dog = Dog('Lucy', 3)
```

sit() method

```
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def sit(self):
        print(f'{self.name} is now sitting’)

my_dog.sit()
```

class Car

```
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year
    def get_descriptive_name(self):
        long_name = f'{self.year} {self.make} {self.model}'
        return long_name.title()

my_new_car = Car('audi', 'a4', 2021)
```

class ElectricCar

```
class ElectricCar(Car):
    def __init__(self, make, model, year):
        super().__init__(make, model, year)

my_tesla = ElectricCar('tesla', 'model s', 2021)
```

override

```
class Car:
    …
    def fill_gas_tank(self):
        print('The gas tank is full.’)

class ElectricCar(Car):
    …
    def fill_gas_tank(self):
        print('This car does not need a gas tank.’)

my_tesla.fill_gas_tank()
```

class Battery

```
class Battery:
    def __init__(self, battery_size = 100):
        self.battery_size = battery_size
    def describe_battery(self):
        print(f'This car has a {self.battery_size}-kWh battery.')

class ElectricCar(Car):
    def __init__(self, make, model, year):
        super().__init__(make, model, year)
        self.battery = Battery()

my_tesla.battery.battery_size
my_tesla.battery.describe_battery()
```

Ch 9: review

```
class Dog:
    def  __init__(self, name, age):
        self.name = name
        self.age = age

my_dog = Dog('Willie', 6)
your_dog = Dog('Lucy', 3)
```

class

instance

object

parent class and child class

super class and sub class

inheritance

overriding

```
import car
from car import Car, ElectricCar
```

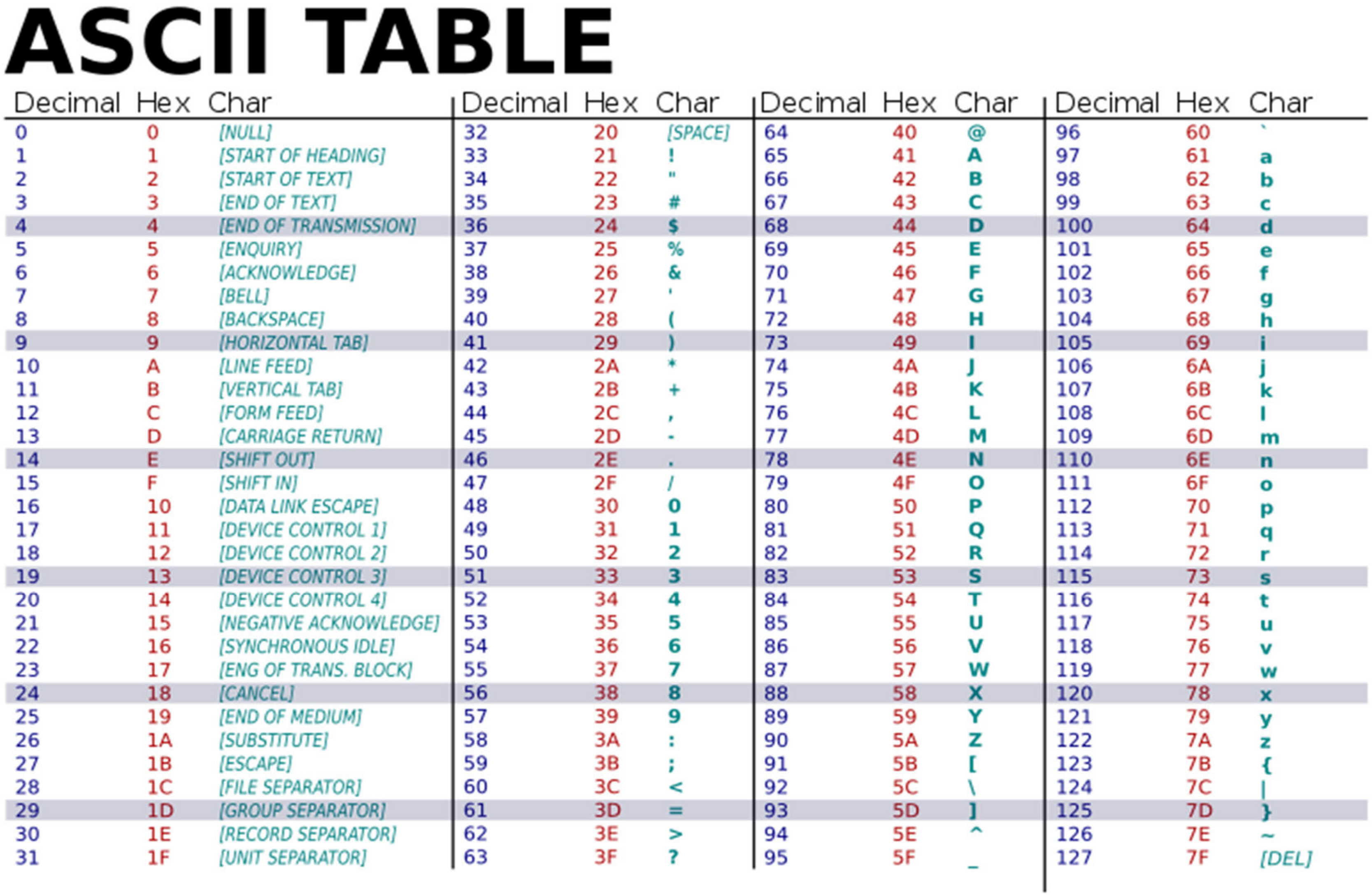
Ch 10: Files and exceptions

- Reading a file
- Reading line by line
- Making a list of lines from a file
- Writing to a file
- Appending to a file
- Try-except-else block
- Handling FileNotFoundError
- Counting the number of words in a file
- JSON (JavaScript Object Notation) data format
- Using json.dump() and json.load()

Encoding and decoding

- Encoding and decoding
  - character (‘a’, ‘1’, ‘#’) → code (01001, 10010)
  - code → character
- ASCII code (7-bit)
- Extended ASCII (8-bit)
  - Include European (France, Germany, …) characters
- Unicode
  - International standard code
  - Include all characters in the world
- UTF (Unicode Transformation Format)
  - utf-8, utf-16, utf-32, …
- UTF-8
  - English + numbers + symbols (1 byte)
  - Korean characters, Chinese characters (3 bytes)
- UTF-8-sig (signature)

ASCII code (7-bit)



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Encoding Korean characters

- 조합형(초성, 중성, 종성)
  - ㄱㄴㄷ, ㄱㄴㄷ, ㅎㅏㅂ
- 완성형
  - 가나다라
- EUC-KR (extended unix code)
  - English + numbers + symbols (1 byte)
  - Korean characters (2 bytes)
  - 2,350 Korean characters
- cp949
  - English + numbers + symbols (1 byte)
  - Korean characters (2 bytes)
  - 11,172 Korean characters (똠,뜽, 홥)

Ch 10: review

Built-in Functions

print()
type()
id()
len()
sum()
max()
min()
sorted()

int()
float()
str()
set()
range()
list()
input()
dir()
open()

try-except-else block

with open(filename) as file_object:

pass

utf-8-sig, cp949

File object Methods

read()

readlines()

write()

Built-in Modules

import math
math.exp()
math.log()
import json
json.dump()
json.load()

Ch1 - Ch10: review(1)

function, argument, variable, value

data types (int, float, str, bool, list, tuple, set, dict, range)

(date, timedelta, array, series, dataframe, tensor, ….)

Assignment operator
Arithmetic operator
Comparison operator
Logical operator
For loop
While loop
Continue
Break
Pass

If-else statement
If-elif-else statement
Boolean expression
Class, child class, inheritance
Method, attribute
Try-except-else
Encoding, decoding
With as
JSON data format

Ch1 - Ch10: review(2)

Built-in Functions

print()
type()
id()
len()
sum()
max()
min()
sorted()

int()
float()
str()
set()
range()
list()
input()
dir()
open()

String Methods

title()
upper()
lower()
split()

rstrip()
lstrip()
strip()

Built-in Modules

import math
math.exp()
math.log()

import json
json.dump()
json.load()

List Methods

index()
append()
insert()
pop()
remove()
sort()
reverse()

Dictionary Methods

get()
items()
keys()
values()