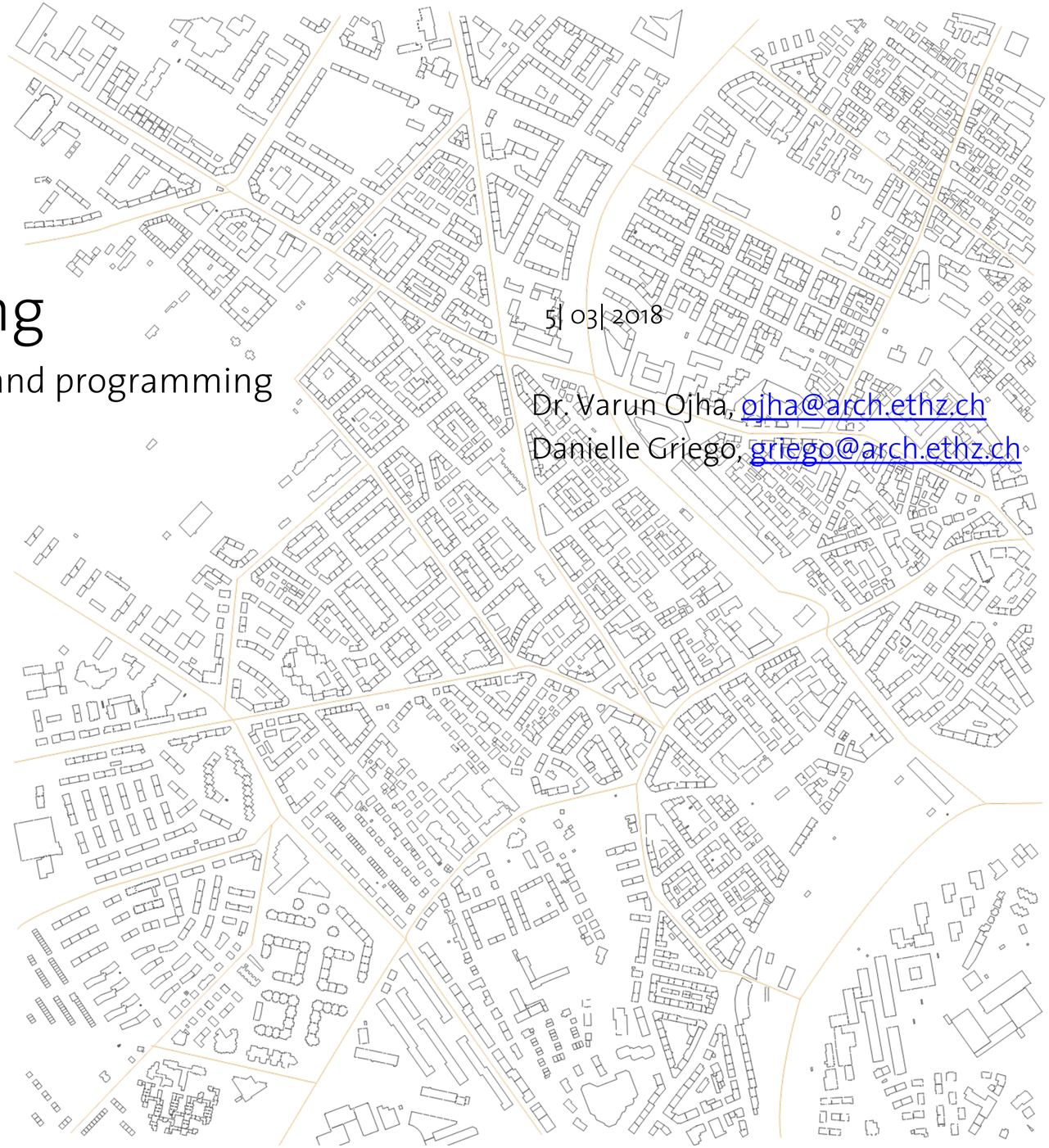
Creative Data Mining

Lecture 03 : Concepts of python and programming

5| 03| 2018

Dr. Varun Ojha, ojha@arch.ethz.ch
Danielle Griego, griego@arch.ethz.ch



ETH zürich DARCH iA Chair of Information Architecture

What we'll cover today

- Introduction to programming and python
- Hands-on python tutorial
  - Get familiar with the environment
  - Data types, Operators, Data structures, conditional statements, functions, packages...

First, did everyone get python and spyder installed?

1. Install Python and Spyder through Anaconda https://www.anaconda.com/download/
2. Can also install Python and Spyder from https://www.python.org/downloads/ and https://pythonhosted.org/spyder/

What is programming?

- Programming is about solving problems and puzzles. You describe a precise set of instructions which the computer follows exactly.

For example if we want to sort a list:

```
list=[23, 44, 5, 17, 8, 90, 102]
list.sort()
print(list)

output: [5, 8, 17, 23, 44, 90, 102]
```

Types and Values

- *Values* are for example: 5, 12.6, True or “Hello world”
- Each value is of a certain *type*
  - Numbers are: →numeric (integer or float)
  - True is: →boolean (only two values or states)
  - “Hello world” is: →a string (list of characters)

Operators

- Operators represent a value manipulation (+, -, /, %, ….)
- It is important to check that the data type is correct, otherwise you might get an unexpected result.

```
                                Output:
p=int(17/9)          1
print(p)

p=round(17/9)        2
print(p)

p=float(17/9)        1.8888888888
print(p)
```

Data Structures

There are three basic data structures in Python:

1. Lists- A sequence of values of any type

2. Lists provide several functionalities:

Example:

```
cdm=['There are', [20, 17, 11, 7],'students registered and', 2, 'instructors']
```

```
cdm[2]          #Accesses the third value
cdm.remove(2)   #Removes the first occurrence of 2
cdm.append(2)   #Adds 2 to the end of the list
```

Data Structures

There are three basic data structures in Python:

2. Tuples- a sequence of values of any type, however cannot be altered after it is initiated.

Example:

```
#tuple
cdm_t=('There are', [20, 17, 11, 7],'students registered and', 2, 'instructors')

Vs.

#list
cdm_l=['There are', [20, 17, 11, 7],'students registered and', 2, 'instructors‘]
```

Data Structures

There are three basic data structures in Python:

3. Dictionaries- store data as key-value pairs. Each key has an associated value.

Example

```
CDM={'Number of students so far': [12, 15, 17, 8]}

CDM['Number of students so far']
# Returns [12,15,17,8]

CDM['final number']='usually 3/4 of first day'
#Adds a new element with key 'final number' and value 'usually 3/4 of
the first day'
```

Logical Operators

Logical Operators are expressions that evaluate to either *True* or *False*.
They are normally used in conditional statements.
Relational operators: ==, !=, <, >, <=, >=, and, or, not
They stand for: equal, not equal, smaller, greater, smaller equal, greater equal, and, or, not

Examples:
x < 2 or x >= 0
x == 0 or x != 0
not (x == 0 or x > 100)

Functions

If you do not want to retype the same code over and over again, you can define a function that you can call over and over again

Structure:

def funtionName(parameters):

    instructions

    return variable

Example:

```
def square(a):
    a = a * a
    return a

# To get the square of 6, now just write:

square(6)
```

Built-in python Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

https://docs.python.org/3/library/functions.html

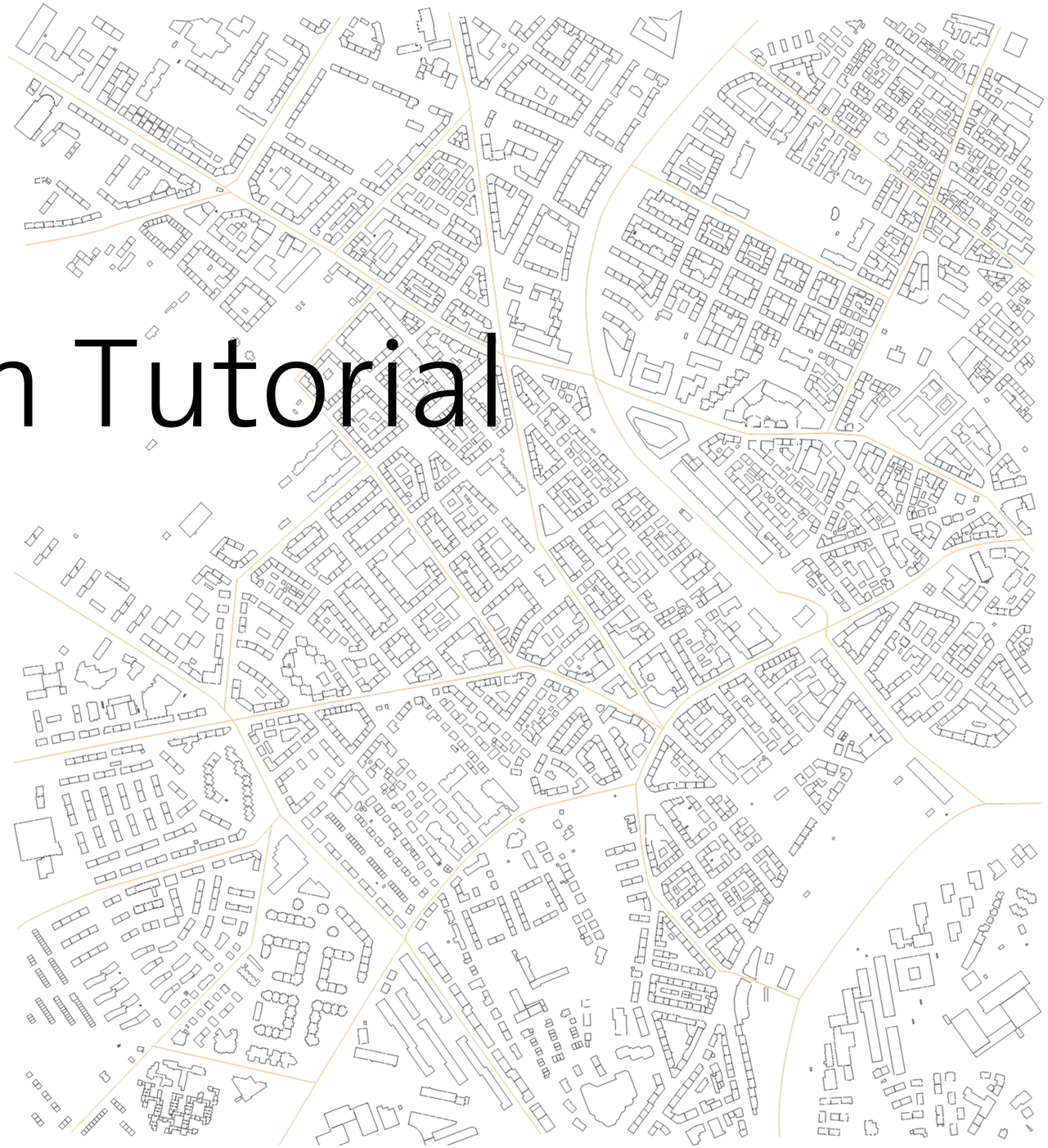
Packages

- Often, more complicated code is already implemented and can be used as Packages. They help to be more efficient, because the programmer does not need to implement everything from scratch.
- When installed, packages can be integrated in the code using the keyword *import*.

Example:

```
import pandas as pd
# imports the Pandas package, and now you can access the built in functions
```

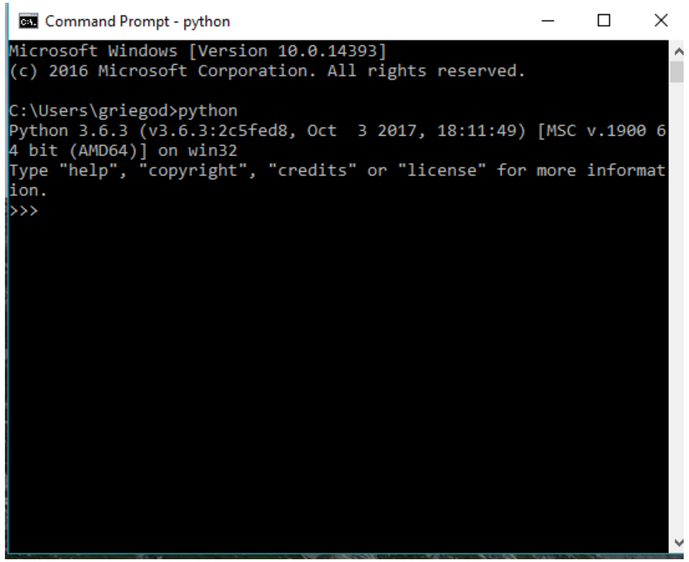
Hands-on Tutorial



ETH zürich DARCH iA Chair of Information Architecture

Python is the programming language

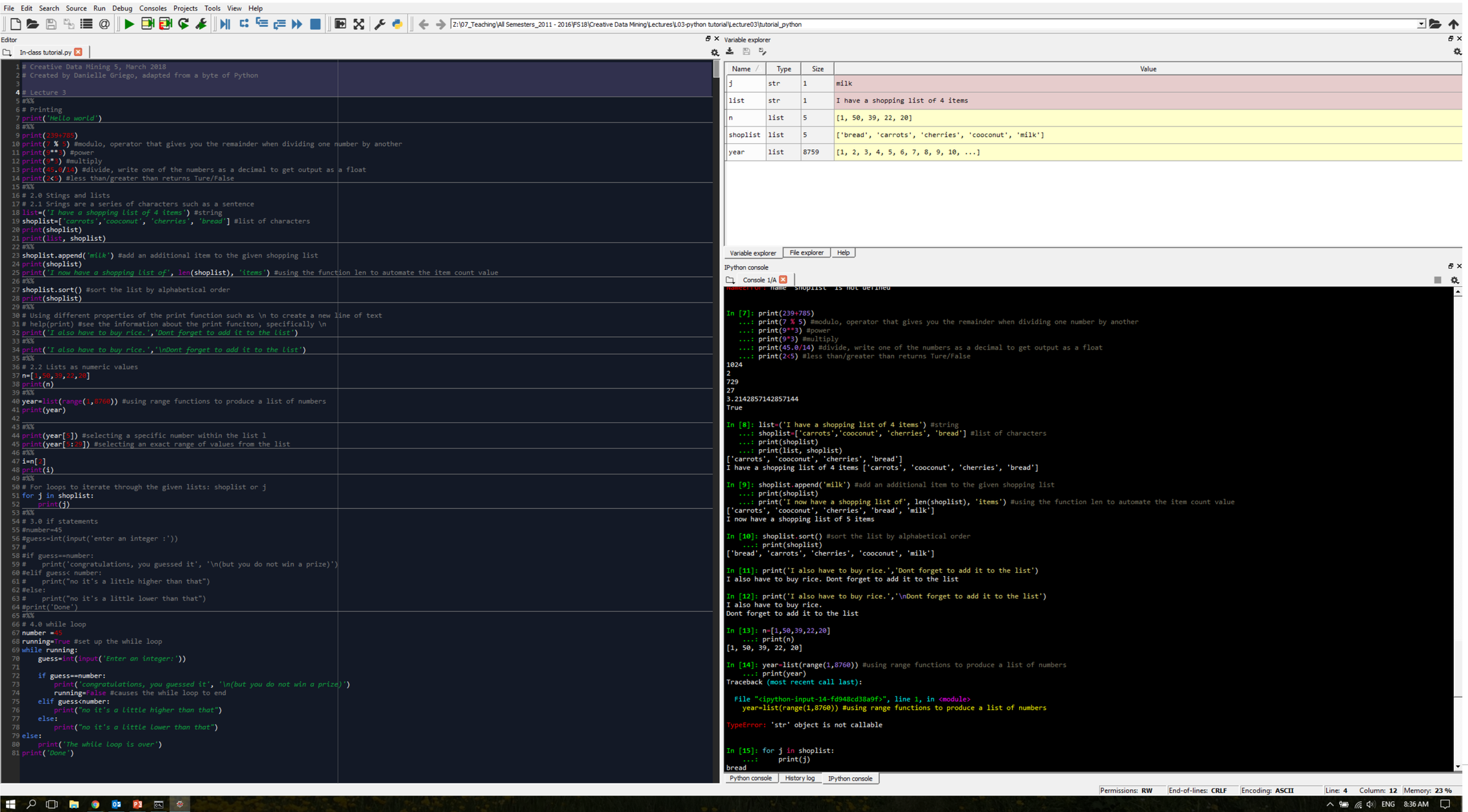
We can simply access it from the command prompt/terminal



Python Interpreter version 2.x or 3.x

Spyder is the IDE (interactive development environment)

There are many IDE's to choose from, we will show all tutorials in Spyder in the course



Questions?

Lecture 3: Python I

5 | 03 | 2018

Dr. Varun Ojha, ojha@arch.ethz.ch
Danielle Griego, griego@arch.ethz.ch

ETH zürich DARCH iA Chair of Information Architecture