



O prowadzącym

Kamil Musiał

Tester w Tieto (wcześniej 7 lat tester / integrator /
verification specialist w Nokia)



certyfikowany tester ISTQB
(full advanced)



od 5 lat wykładowca WSB Wrocław
(testowanie, telekomunikacja, sieci, IoT, Python)



doktorant
Politechnika Wrocławska



fan morsowania
zanim to było modne



kamil.musial@akademia.altkom.pl
kamil.musial@wroclaw.merito.pl

O prowadzącej

Natalia Kniat

CI Lead / testerka / integratorka
w Nokia



certyfikowana testerka
ISTQB



trenerka Nokia Academy
(testowanie, telekomunikacja, Python)



zorganizowana mama
z zamiłowaniem do wyrobów własnych



natalia.kniat@akademia.altkom.pl
natalia.kniat@wroclaw.merito.pl

PYTH01



01

Biblioteka
os oraz time

02

Biblioteka
datetime

03

Obsługa plików
csv bez bibliotek

04

Zadanie: znaleźć
nazwę hosta i IP

05

Powtórka:
operacje
na stringach

06

Obsługa plików
txt bez bibliotek

Tester jako developer narzędzi z pomocą Python-a - warsztaty (8h)



07

Zadanie:
Tracert: jak
daleko do googla?

08

Zadanie:
Search and destroy

09


Zadanie:
Podmiana danych

10

Biblioteka Pandas:
obsługa plików csv

Biblioteka `os` oraz `time`

Przykłady użycia biblioteki OS



```
import os
import time
```

`print(os.getcwd())` sprawdzenie aktualnej lokalizacji

`os.chdir('C:\\Users\\kamil\\Desktop')`
zmiana lokalizacji

`os.mkdir('new_folder')` utworzenie nowego folderu

`os.rename('new_folder', 'new_folder2')`
zmiana nazwy folderu

`os.rmdir('new_folder2')` usunięcie folderu

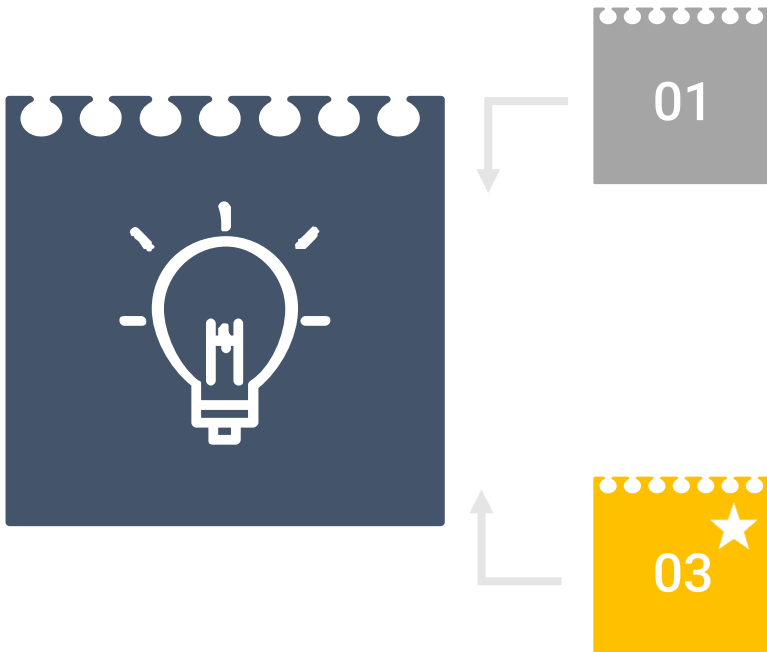
`os.system('cmd /c "cd C://Users//kamil//Desktop"')`
użycie konsoli systemowej

`time.sleep(5)` odczekanie 5 sekund

`&&` łączenie dwóch komend w terminalu

Biblioteka os oraz time

Zadania Python *



01

- przejdź na pulpit
- utwórz folder i poczekaj kilka sekund
- zmień nazwę folderu i poczekaj kilka sekund
- usuń folder

Rezultat:

Po uruchomieniu programu na pulpicie pojawia się folder, za kilka sekund folder zmienia nazwę i za kolejne kilka sekund folder ten znika.

03

Wyszukaj na pulpicie pliki `new.txt` i zwróć rezultat wyszukiwań w pliku `result.txt`

Na pulpicie pojawia się plik `results.txt` z wynikami wyszukiwania pliku `new.txt`

* PyCharm - pliki z kodem dostępne osobno

Operacje na stringach

Powtórka



Łączenie
stringów

```
my_string1 = 'mom'
```

```
my_string2 = 'dad'
```

```
sum1 = my_string1 + my_string2
```

```
print(sum1)      -> momdad
```

```
sum2 = 'my' + my_string1 + ' and my ' + my_string2
```


```
print(sum2)      -> my mom and my dad
```

Operacje na poszczególnych elementach:

```
print(sum2[3:6]) -> mom
```

```
print(sum2[-4:-1]) -> dad
```


Biblioteka datetime



Jaki dzisiaj
dzień?

```
import datetime
```

wyciągnięcie daty:

```
today = datetime.date.today()
```

%d dzień

%m miesiąc

%Y / **%y** rok (pełny, skrócony)


%B / **%b** miesiąc słownie (pełny, skrócony)

Różne sposoby przedstawienia daty:

```
d1 = today.strftime('%d/%m/%Y')    dd/mm/YY
```

```
d2 = today.strftime('%B %d, %Y')    tekstowo miesiąc, potem dzień i rok
```

Biblioteka datetime



Jaki dzisiaj
dzień?

wyciągnięcie godziny i daty:

```
now = datetime.datetime.now()
```

%d dzień

%m miesiąc

%Y / %y rok (pełny, skrócony)

%B / %b miesiąc słownie (pełny, skrócony)

%H godzina

%M minuta

%S sekunda

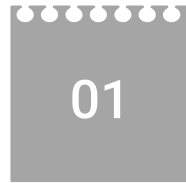
Tworzenie stringa z datą bądź godziną:

```
timer = now.strftime('%H:%M:%S')
```

```
date = now.strftime('year = %Y, month = %m, day = %d')
```

Biblioteka datetime

Zadania Python *



Wypisać 10 nazw (plików)
w odstępach 2 sekundowych o nazwach:
report<godzina>.txt

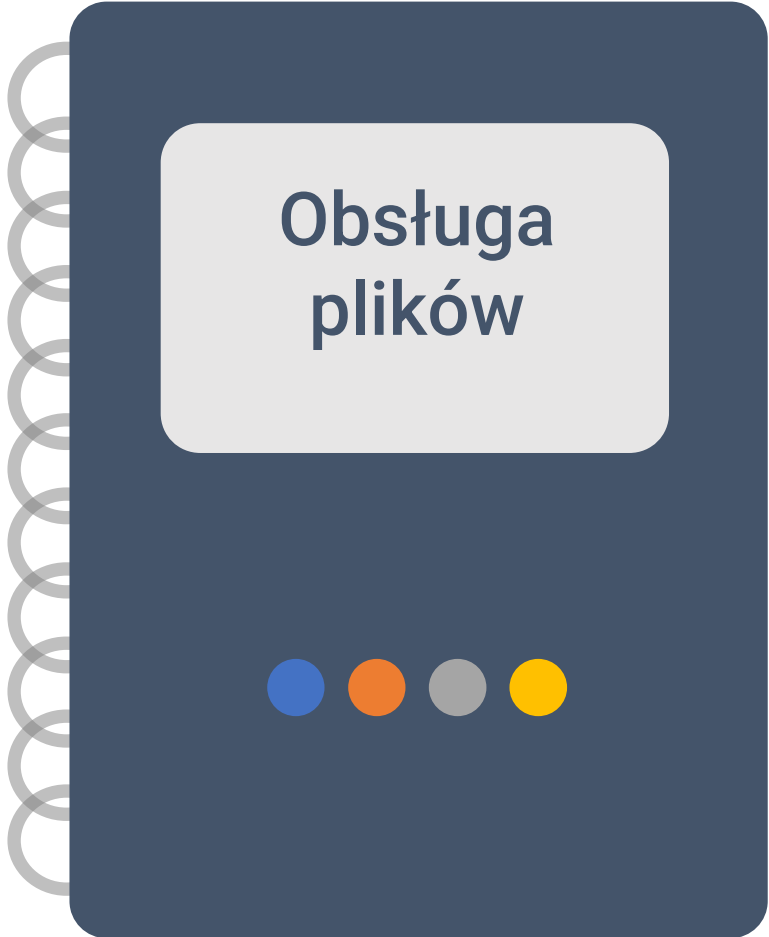
Wynik:

```
report012512.txt  
report012514.txt  
report012516.txt  
report012518.txt
```

Każdy plik stworzony o godzinie 01:25 i co 2 sekundy.

* PyCharm - pliki z kodem dostępne osobno

Obsługa plików txt bez bibliotek



Obsługa plików

Metody:

read() odczyt całego pliku jako string

read(5) odczyt 5 znaków (pierwszych bądź kolejnych)

readline() odczyt linii (pierwszej bądź kolejnej)

readlines() odczyt całego pliku jako tablicy (jedna linia, jeden element)

Przykład:

```
with open ("C://Users//kamil/Desktop//my_file.txt",'r') as plik1:  
    content = plik1.readlines()
```

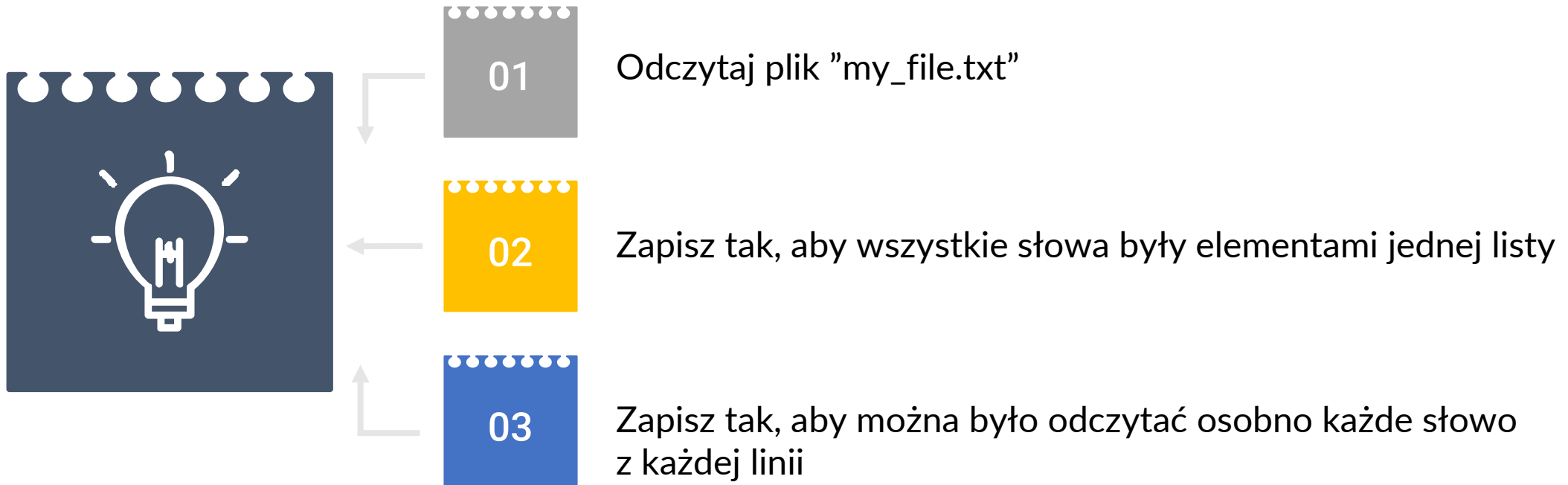
gdzie:

r - ,read'

tryb do odczytu (inne tryby: w - write, a - append)

Obsługa plików txt bez bibliotek

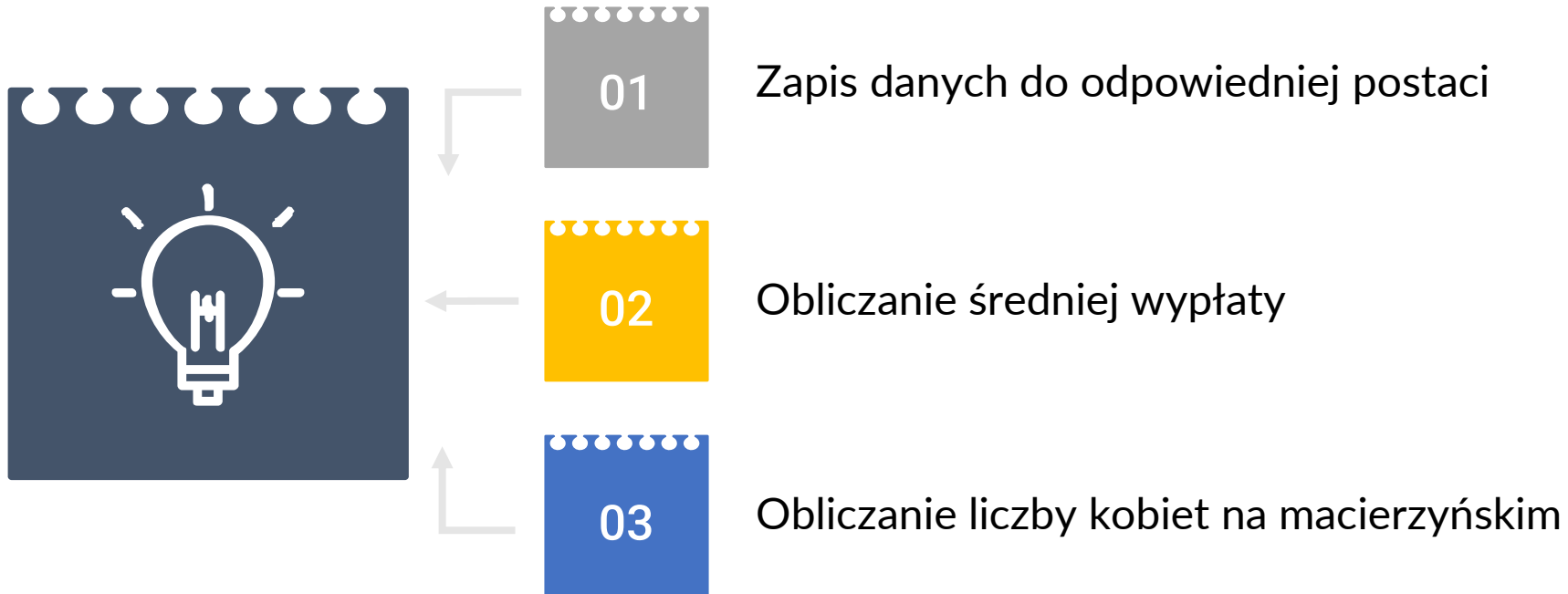
Zadania Python *



* PyCharm - pliki z kodem dostępne osobno

Obsługa plików csv bez bibliotek

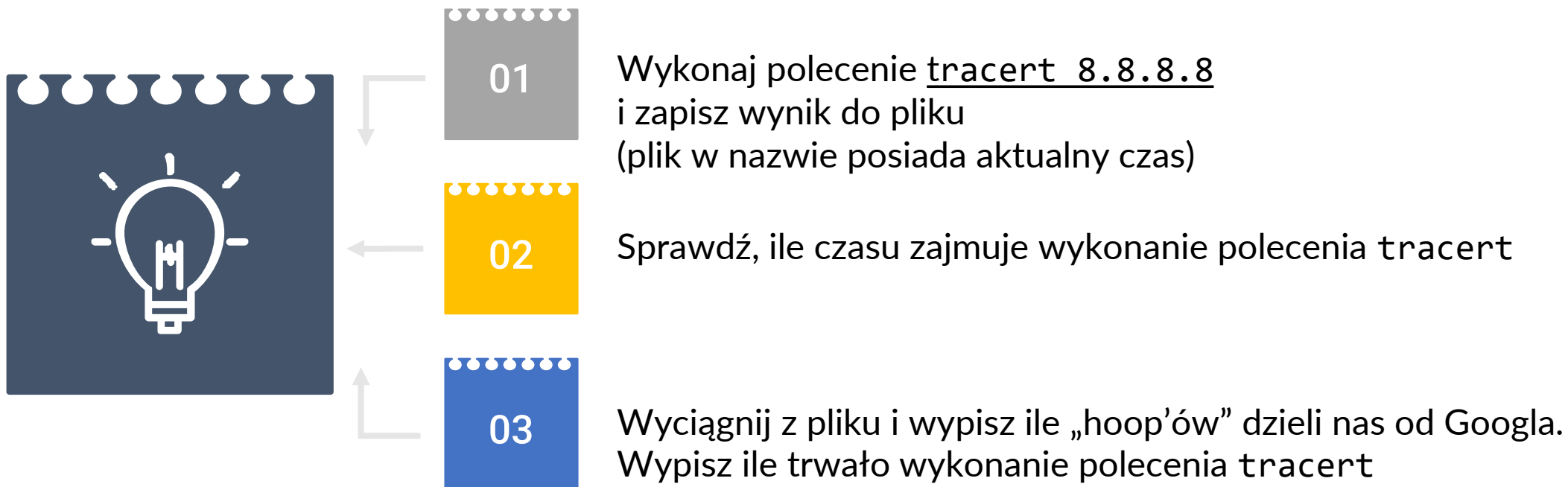
Zadania Python *



* PyCharm - pliki z kodem dostępne osobno

Zadanie: Tracert – jak daleko od Googla?

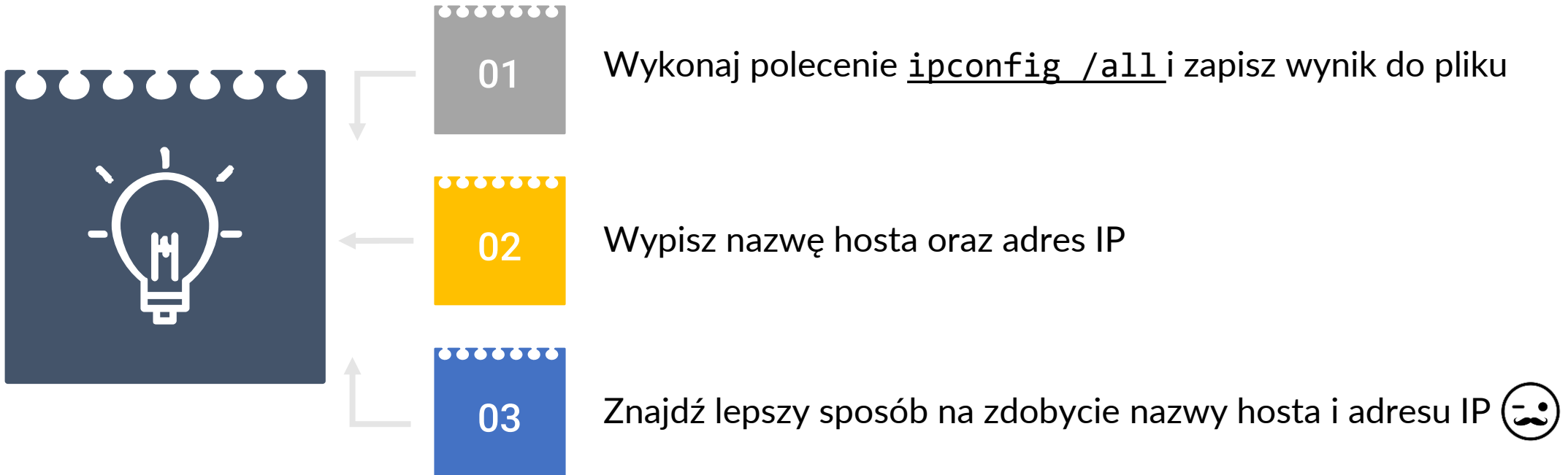
Zadania Python *



* PyCharm - pliki z kodem dostępne osobno

Zadanie: znaleźć nazwę hosta i IP

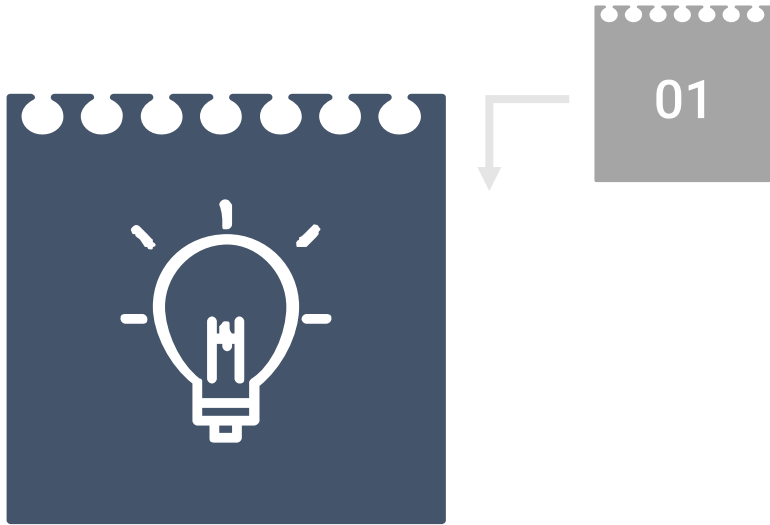
Zadania Python *



* PyCharm - pliki z kodem dostępne osobno

Zadanie: Search and destroy

Zadania Python *

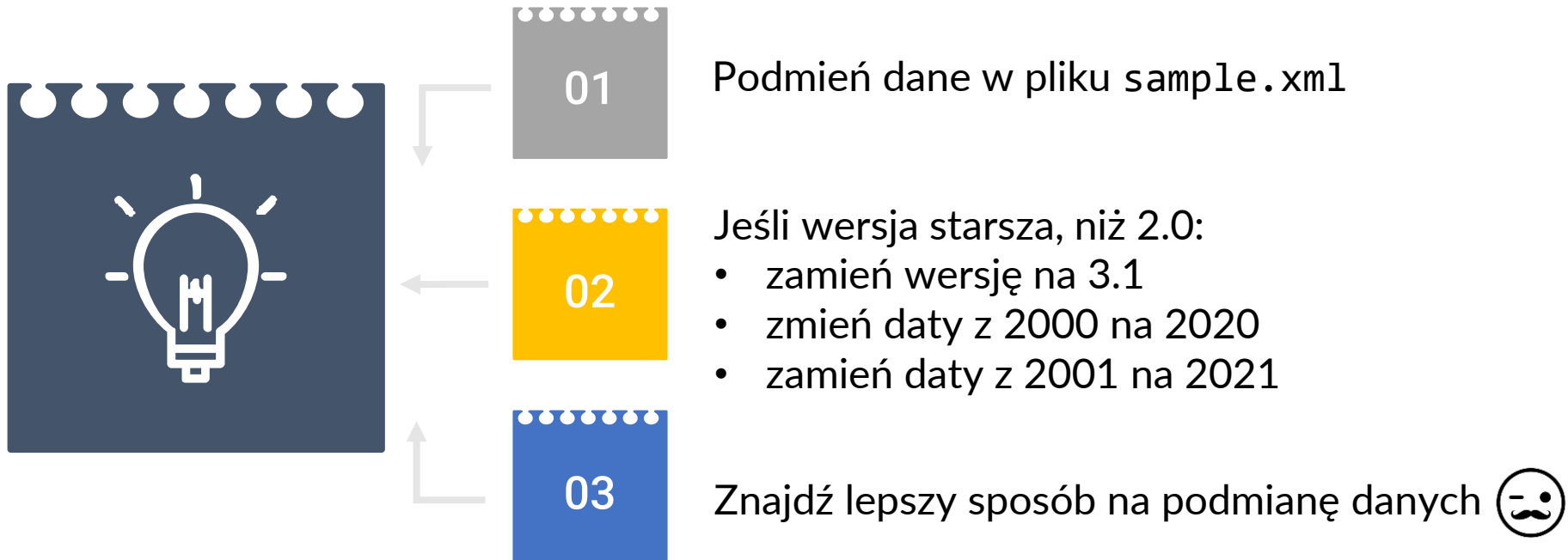


Usuń z pulpitu wszystkie pliki z rozszerzeniem "pcap"

* PyCharm - pliki z kodem dostępne osobno

Zadanie: Podmiana danych

Zadania Python *



* PyCharm - pliki z kodem dostępne osobno

Obsługa plików biblioteka Pandas

Środowisko – doinstalowanie biblioteki Pandas oraz ffspec



Biblioteka
Pandas

Import biblioteki i nadanie jej nazwy roboczej pd:

```
import pandas as pd
```

Przykład:

Zapis pliku csv z przecinkiem jako separatorem pod nazwą data:

```
data = pd.read_csv(path, delimiter=',')
```

Plik

`pokemons.csv`

Obsługa plików biblioteka Pandas

Przykłady użycia biblioteki Pandas



Biblioteka
Pandas

`print(data.head(5))` wyświetlenie 5 pierwszych wierszy

`print(data.columns)` wyświetlenie kolumn

`print(data[['Name', 'Type 1', 'HP']][0:5])`
wyświetlenie pięciu pierwszych
wierszy i kolumn Name, Type 1 i HP

`print(data.iloc[1:4])` wyświetlenie wierszy 2, 3, 4

`print (data.iloc[2,1])` wyświetlenie 2go pola w 3cim wierszu

`print (data.loc[data['Type 1'] == 'Fire'])`
wyświetlenie pozycji, posiadających Fire w Type 1

`print(data.sort_values('Name'))`
sortowanie rosnąco po Name

`print(data.sort_values('Name', ascending=False))`
posortowanie malejąco po Name

Obsługa plików biblioteka Pandas

Przykłady użycia biblioteki Pandas



Biblioteka
Pandas

Zapis danych do csv:

```
data.to_csv('C:\\Users\\kamusia1\\Desktop\\modified.csv', sep=',';')
```

lub

```
data.to_csv('C:\\Users\\kamusia1\\Desktop\\modified.csv',  
index=False, sep=';')
```

Zapis danych do txt:

```
data.to_csv('C:\\Users\\kamusia1\\Desktop\\modified.txt', sep='\\t')
```

lub

```
data.to_csv('C:\\Users\\kamusia1\\Desktop\\modified.txt',  
index=False, sep='\\t')
```

index=False

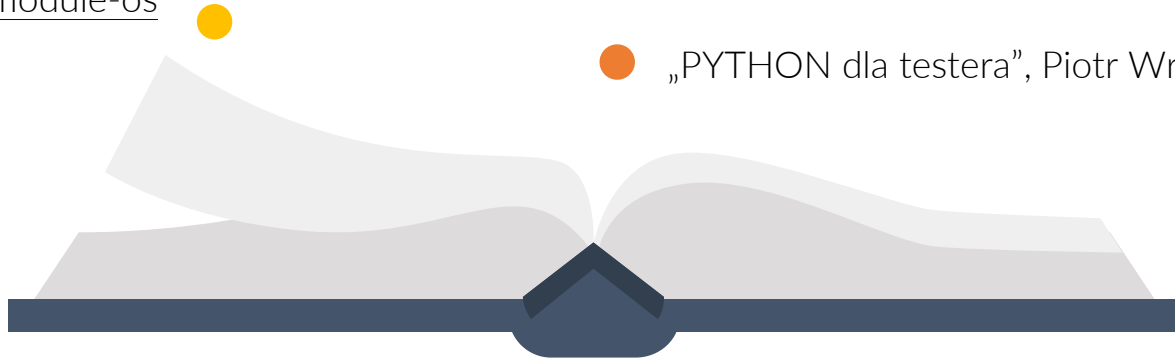
usuwa pierwszą kolumnę z indeksami

Materiały

Biblioteka os:
<https://docs.python.org/3/library/os.html?highlight=os#module-os>

● „PYTHON dla testera”, Piotr Wróblewski

● „Python 3. Projekty dla początkujących i pasjonatów”, Adam Jurkiewicz



Pandas – csv

https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html



Pandas kurs

<https://www.youtube.com/watch?v=vmEHCJofslg>



OS kurs

<https://www.youtube.com/watch?v=tJxcKyFMTGo>

Koniec

**Dziękuję za
uwagę**

