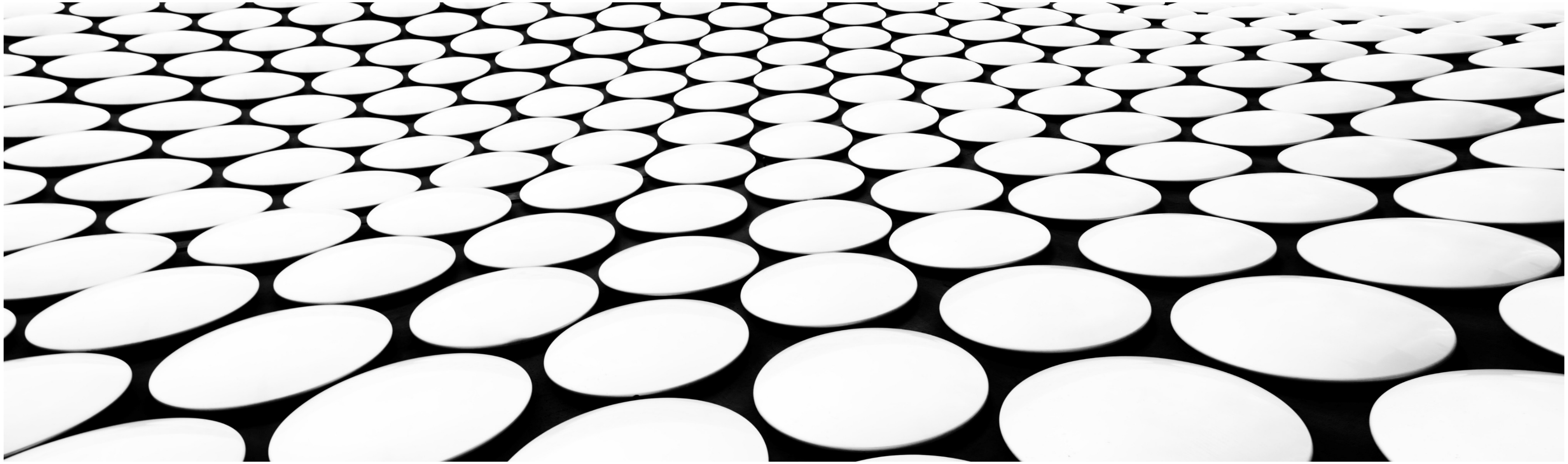
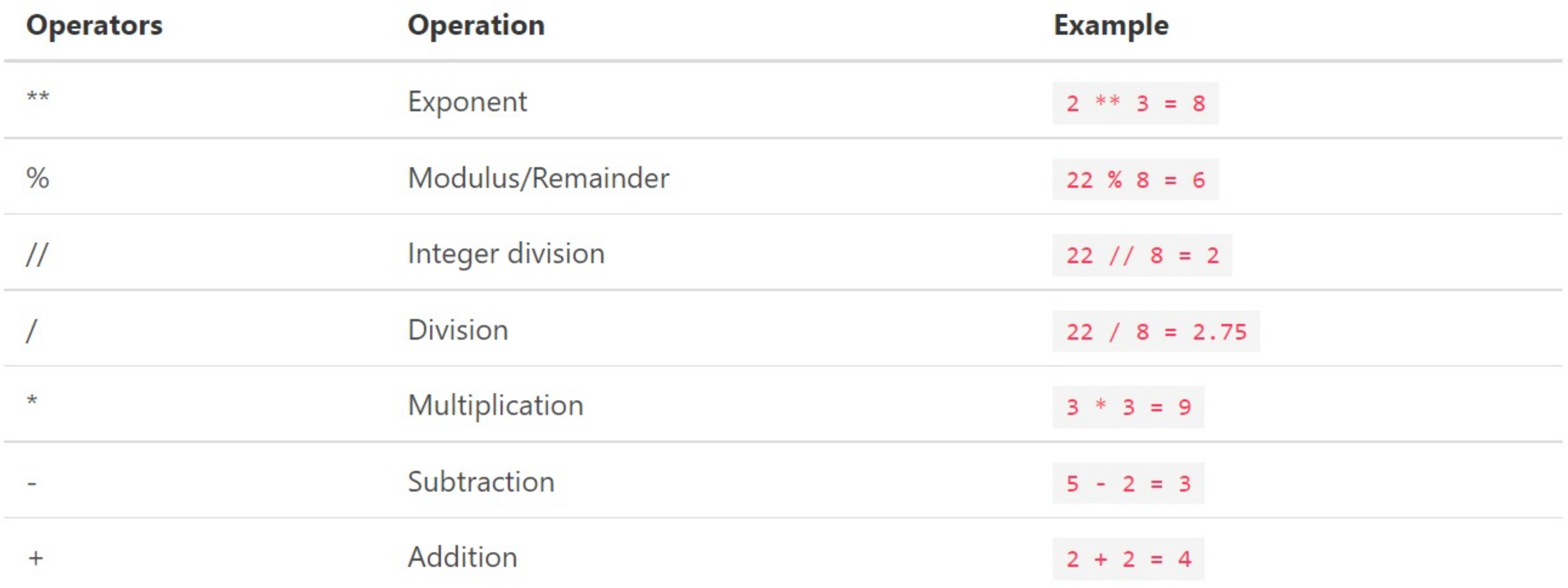
PYTHON REFRESHER



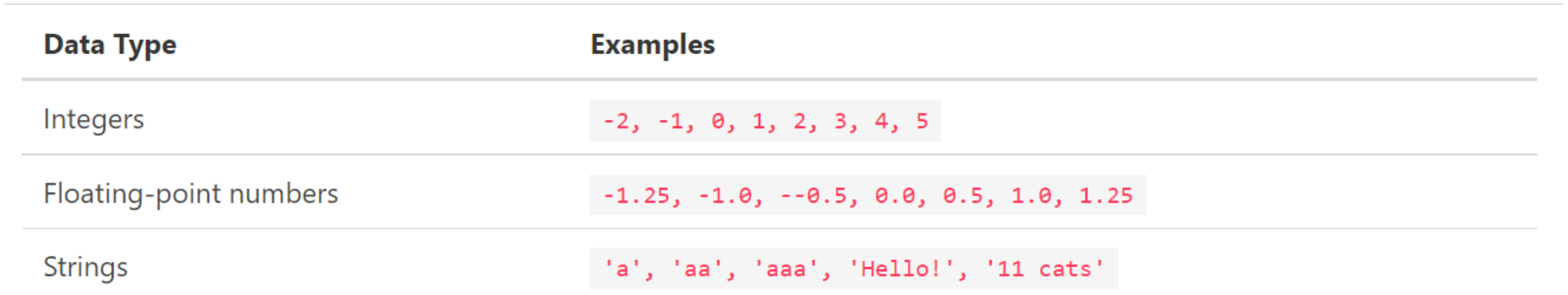
DAVID PINEZICH, DAVID.PINEZICH@GMAIL.COM

OPERATOREN



Operators	Operation	Example
**	Exponent	`2 ** 3 = 8`
%	Modulus/Remainder	`22 % 8 = 6`
//	Integer division	`22 // 8 = 2`
/	Division	`22 / 8 = 2.75`
*	Multiplication	`3 * 3 = 9`
-	Subtraction	`5 - 2 = 3`
+	Addition	`2 + 2 = 4`

DATENTYPEN



Data Type	Examples
Integers	`-2, -1, 0, 1, 2, 3, 4, 5`
Floating-point numbers	`-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25`
Strings	`'a', 'aa', 'aaa', 'Hello!', '11 cats'`

KOMMENTARE & DOCSTRINGS

```python
# This is a comment
```

```python
# This is a
# multiline comment
```

```python
a = 1  # initialization
```

```python
def foo():
    """
    This is a function docstring
    You can also use:
    ''' Function Docstring '''
    """
```

WICHTIGE FUNKTIONEN

- print()
- len(), range()
- str(), int(), float(), bool()
- input()
- abs(), round(), min(), max(), sum()
- type()
- enumerate(), iter()
- map(), filter(), zip()

Map

```
nums = [1, 2, 3, 4, 5]

def sq(n):
    return n*n

square = list(map(sq, nums))

print(square)

Output: [1, 4, 9, 16, 25]
```

Filter

```
seq = [0, 1, 2, 3, 4, 5]

# result contains odd numbers of the list
result = filter(lambda x: x % 2, seq)
print(list(result))

# result contains even numbers of the list
result = filter(lambda x: x % 2 == 0, seq)
print(list(result))

Output:
[1, 3, 5]
[0, 2, 4]
```

Zip

```
name = ["Manjeet", "Nikhil", "Shambhavi"]
roll_no = [4, 1, 3]
marks = [40, 50, 60]

mapped = zip(name, roll_no, marks)

print(list(mapped))

Output:
[('Manjeet', 4, 40), ('Nikhil', 1, 50), ('Shambhavi', 3, 60)]
```

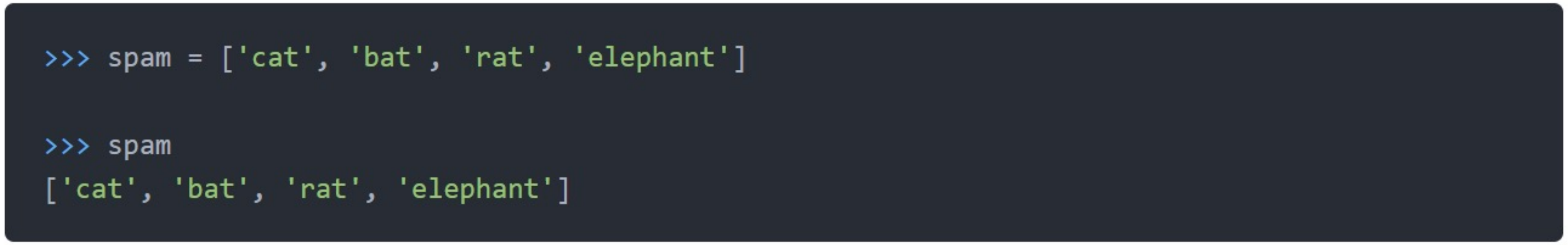
https://medium.com/@lokeshsharma596/python-lambda-map-filter-reduce-and-zip-function-c59d8946a3ce

FOR & WHILE

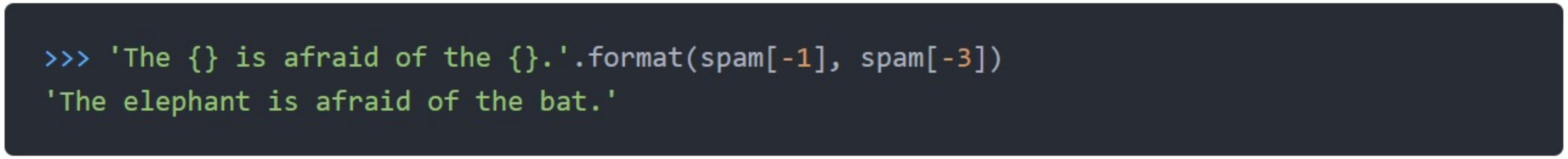
```
>>> for i in [1, 2, 3, 4, 5]:
>>>    if i == 3:
>>>        break
>>> else:
>>>    print("only executed when no item of the list is equal to 3")
```

```
while True:
    print('Who are you?')
    name = input()
    if name != 'Joe':
        continue
    print('Hello, Joe. What is the password? (It is a fish.)')
    password = input()
    if password == 'swordfish':
        break
print('Access granted.')
```

LISTS

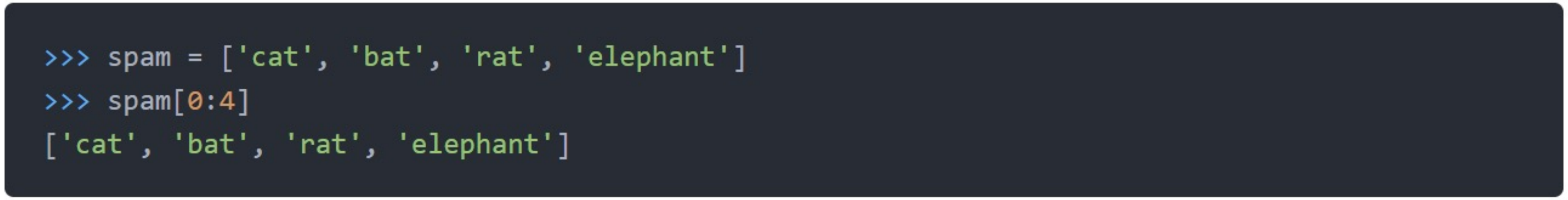


```
>>> spam = ['cat', 'bat', 'rat', 'elephant']

>>> spam
['cat', 'bat', 'rat', 'elephant']
```

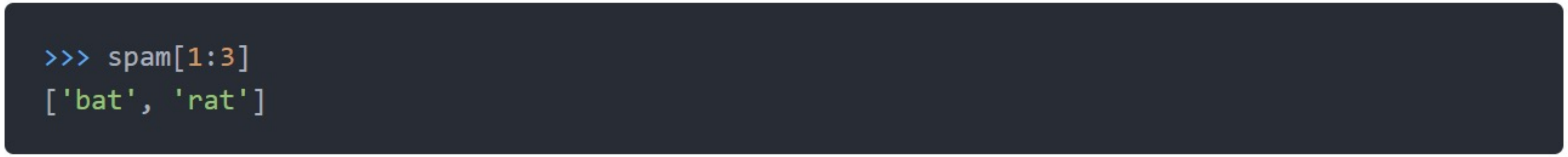


```
>>> 'The {} is afraid of the {}.'.format(spam[-1], spam[-3])
'The elephant is afraid of the bat.'
```

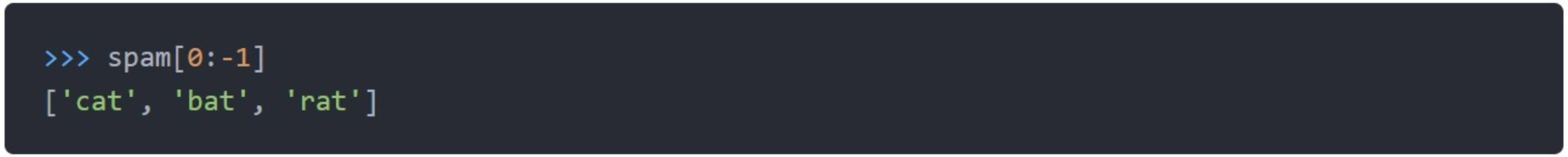
LISTS AND SLICES



```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[0:4]
['cat', 'bat', 'rat', 'elephant']
```



```
>>> spam[1:3]
['bat', 'rat']
```



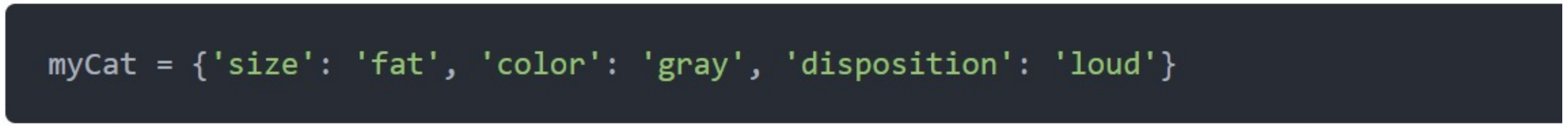
```
>>> spam[0:-1]
['cat', 'bat', 'rat']
```

TUPLES

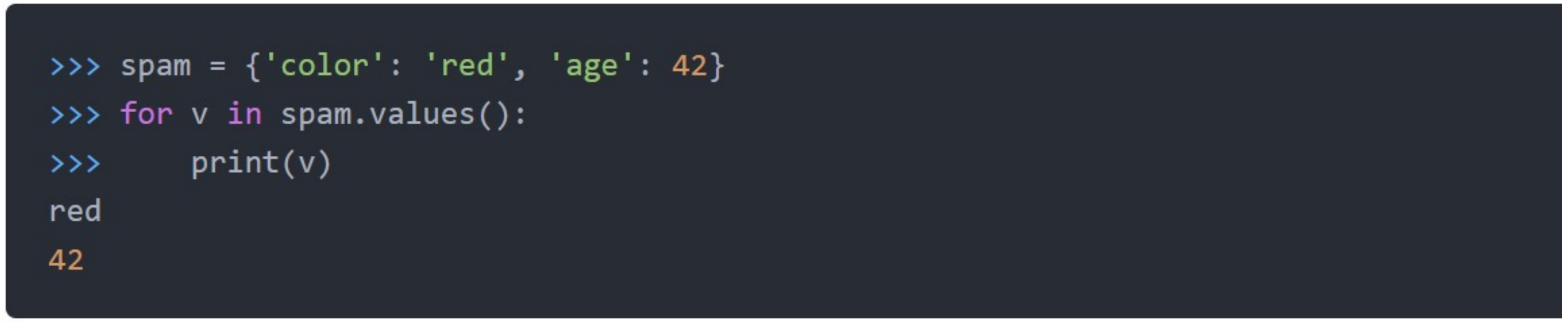
```
>>> eggs = ('hello', 42, 0.5)
>>> eggs[0]
'hello'
```

```
>>> eggs[1:3]
(42, 0.5)
```

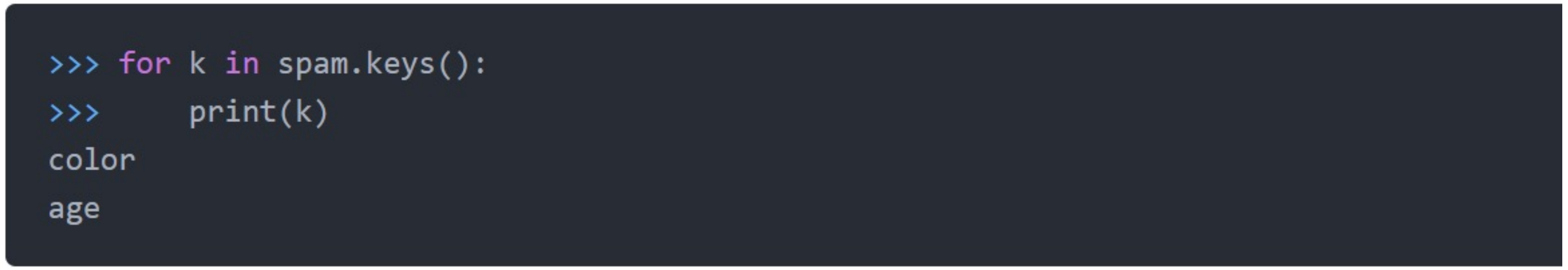
DICTIONARIES



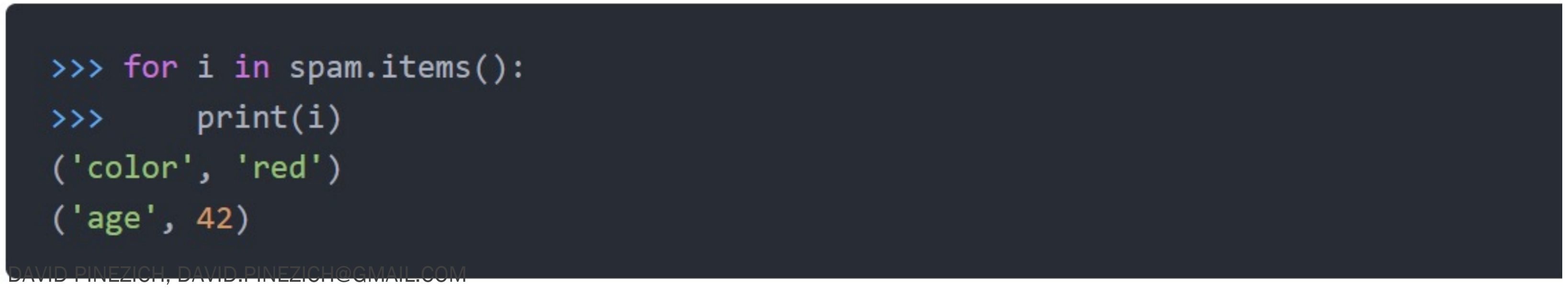
```
myCat = {'size': 'fat', 'color': 'gray', 'disposition': 'loud'}
```



```
>>> spam = {'color': 'red', 'age': 42}
>>> for v in spam.values():
>>>     print(v)
red
42
```



```
>>> for k in spam.keys():
>>>     print(k)
color
age
```



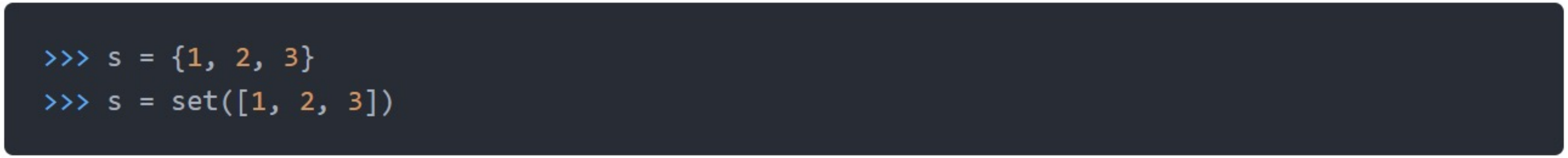
```
>>> for i in spam.items():
>>>     print(i)
('color', 'red')
('age', 42)
```

GET()

```
>>> picnic_items = {'apples': 5, 'cups': 2}

>>> 'I am bringing {} cups.'.format(str(picnic_items.get('cups', 0)))
'I am bringing 2 cups.'
```

```
>>> 'I am bringing {} eggs.'.format(str(picnic_items.get('eggs', 0)))
'I am bringing 0 eggs.'
```

SETS



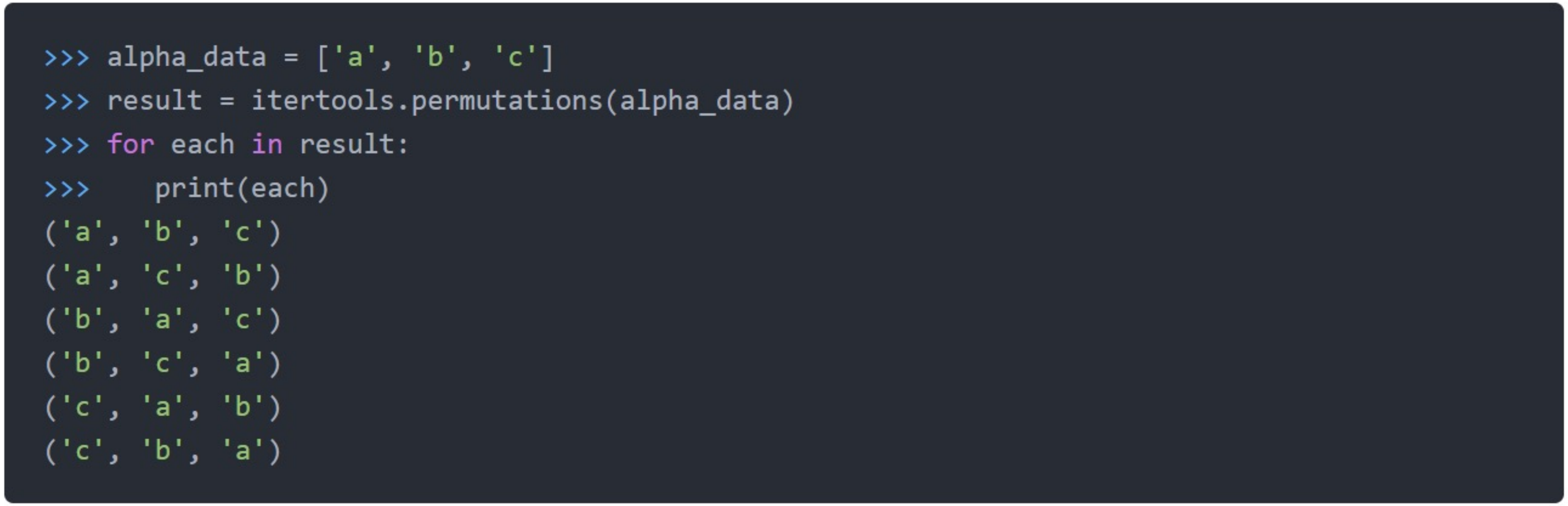
```
>>> s = {1, 2, 3}
>>> s = set([1, 2, 3])
```

SORTING

```
>>> spam = [2, 5, 3.14, 1, -7]
>>> spam.sort()
>>> spam
[-7, 1, 2, 3.14, 5]
```

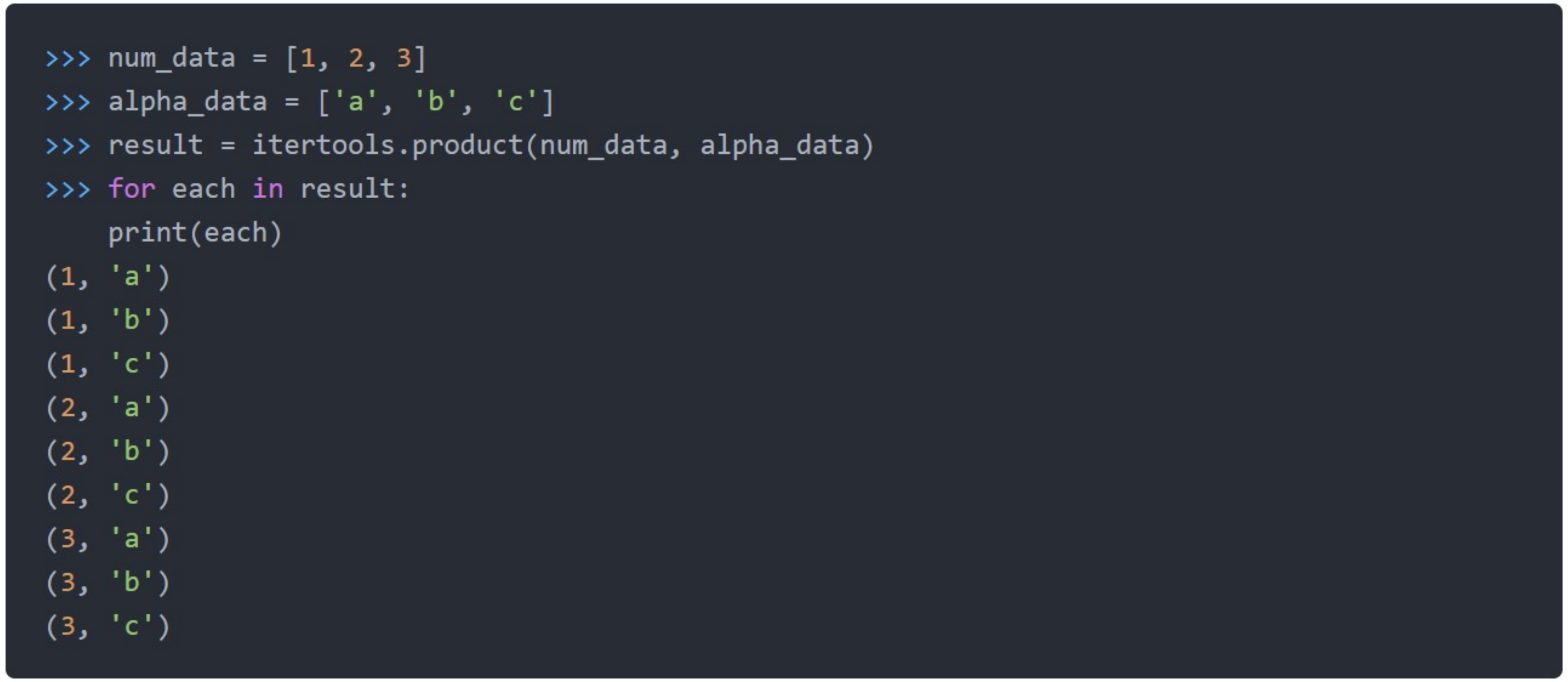
```
>>> spam = ['ants', 'cats', 'dogs', 'badgers', 'elephants']
>>> spam.sort()
>>> spam
['ants', 'badgers', 'cats', 'dogs', 'elephants']
```

PERMUTATIONS()



```
>>> alpha_data = ['a', 'b', 'c']
>>> result = itertools.permutations(alpha_data)
>>> for each in result:
>>>    print(each)
('a', 'b', 'c')
('a', 'c', 'b')
('b', 'a', 'c')
('b', 'c', 'a')
('c', 'a', 'b')
('c', 'b', 'a')
```

PRODUCT()



```
>>> num_data = [1, 2, 3]
>>> alpha_data = ['a', 'b', 'c']
>>> result = itertools.product(num_data, alpha_data)
>>> for each in result:
    print(each)
(1, 'a')
(1, 'b')
(1, 'c')
(2, 'a')
(2, 'b')
(2, 'c')
(3, 'a')
(3, 'b')
(3, 'c')
```

COMPREHENSIONS

List comprehension

```
>>> a = [1, 3, 5, 7, 9, 11]

>>> [i - 1 for i in a]
[0, 2, 4, 6, 8, 10]
```

Set comprehension

```
>>> b = {"abc", "def"}
>>> {s.upper() for s in b}
{"ABC", "DEF"}
```

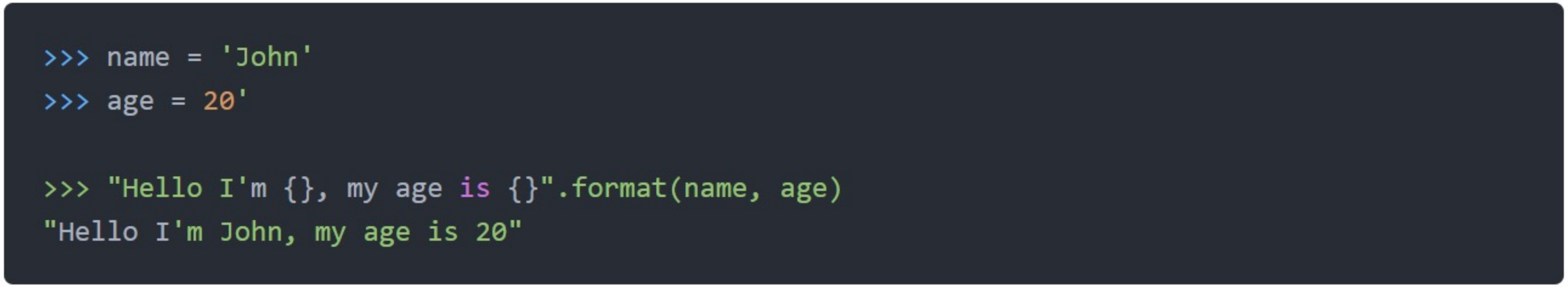
Dict comprehension

```
>>> c = {'name': 'Pooka', 'age': 5}
>>> {v: k for k, v in c.items()}
{'Pooka': 'name', 5: 'age'}
```

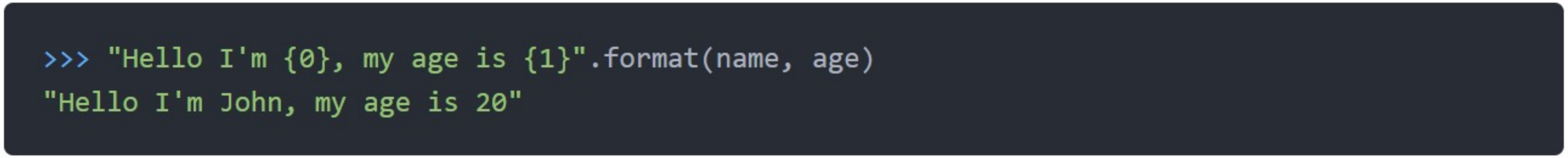
EXCEPTION HANDLING

```
>>> def spam(divideBy):
>>>     try:
>>>         return 42 / divideBy
>>>     except ZeroDivisionError as e:
>>>         print('Error: Invalid argument: {}'.format(e))
>>>     finally:
>>>         print("-- division finished --")
>>> print(spam(2))
-- division finished --
21.0
>>> print(spam(12))
-- division finished --
3.5
>>> print(spam(0))
Error: Invalid Argument division by zero
-- division finished --
None
>>> print(spam(1))
-- division finished --
42.0
```

STRING FORMATIERUNG

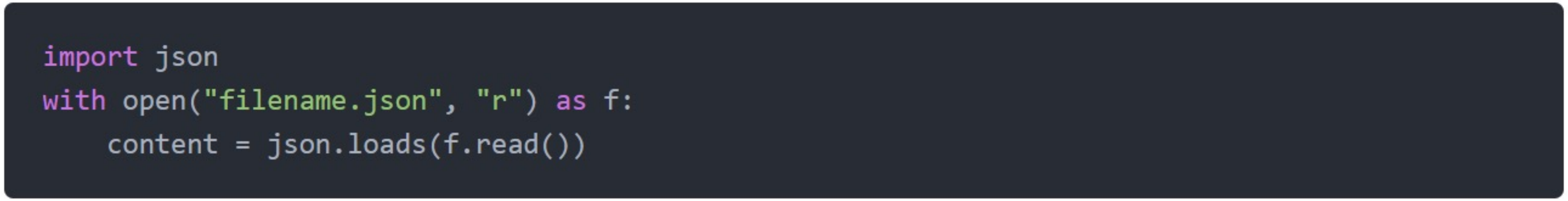


```
>>> name = 'John'
>>> age = 20'

>>> "Hello I'm {}, my age is {}".format(name, age)
"Hello I'm John, my age is 20"
```



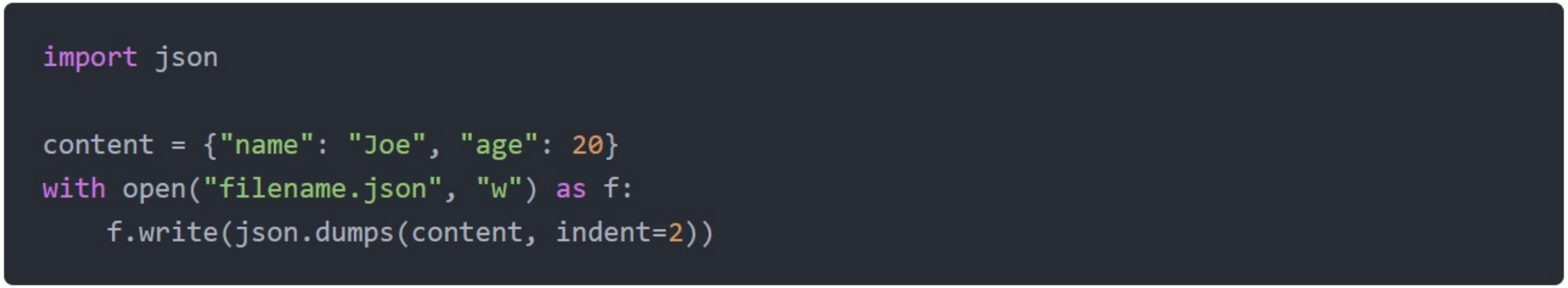
```
>>> "Hello I'm {0}, my age is {1}".format(name, age)
"Hello I'm John, my age is 20"
```

FILE HANDLING – JSON



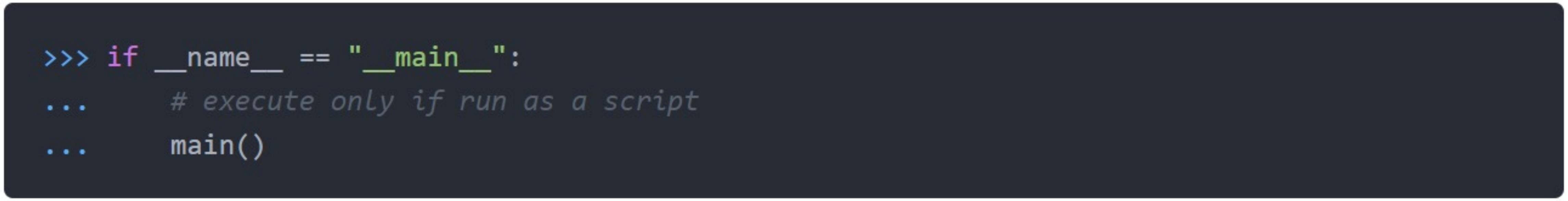
```python
import json
with open("filename.json", "r") as f:
    content = json.loads(f.read())
```

FILE HANDLING – JSON



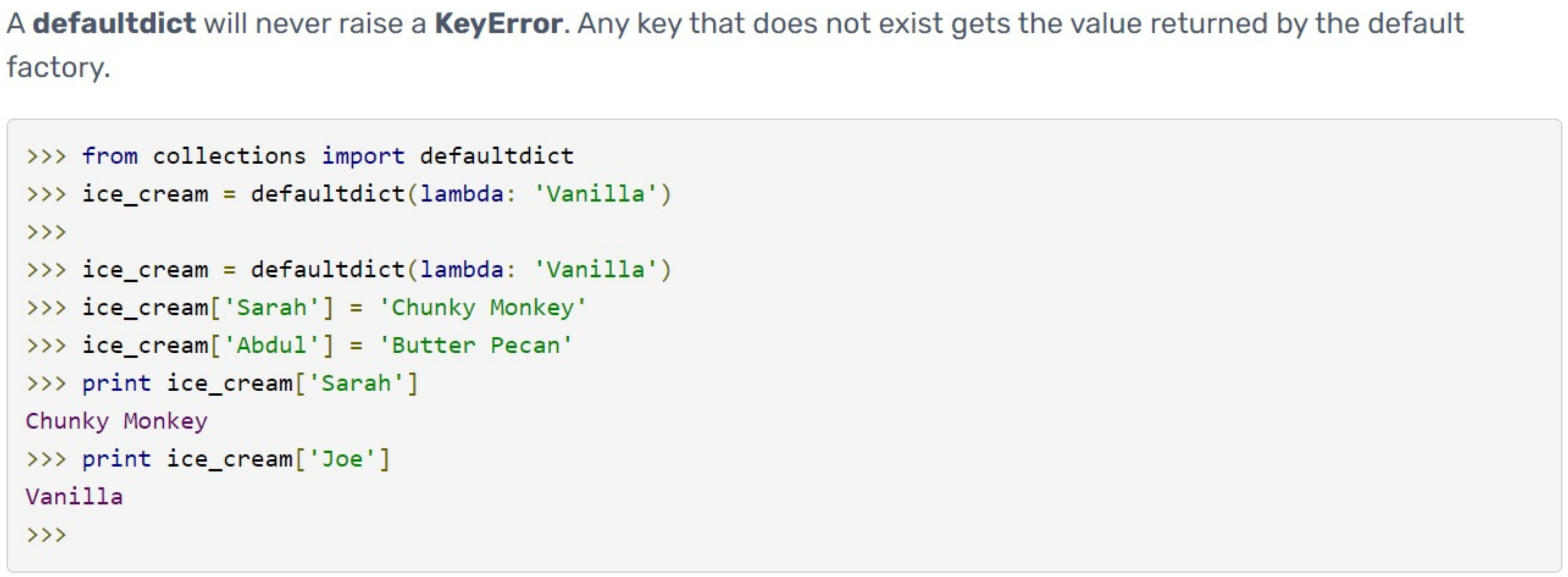
```python
import json

content = {"name": "Joe", "age": 20}
with open("filename.json", "w") as f:
    f.write(json.dumps(content, indent=2))
```

__MAIN__



```
>>> if __name__ == "__main__":
...     # execute only if run as a script
...     main()
```

BONUS: DEFAULTDICT



A **defaultdict** will never raise a **KeyError**. Any key that does not exist gets the value returned by the default factory.

```
>>> from collections import defaultdict
>>> ice_cream = defaultdict(lambda: 'Vanilla')
>>>
>>> ice_cream = defaultdict(lambda: 'Vanilla')
>>> ice_cream['Sarah'] = 'Chunky Monkey'
>>> ice_cream['Abdul'] = 'Butter Pecan'
>>> print ice_cream['Sarah']
Chunky Monkey
>>> print ice_cream['Joe']
Vanilla
>>>
```

SOURCES

- https://www.pythoncheatsheet.org/#Python-Basics
- https://www.accelebrate.com/blog/using-defaultdict-python