PYTHON PROGRAMMING FOR ABSOLUTE BEGINNERS

36C3

Ingo Kleiber

@KleiberIngo @IngoKleiber
mastodon.social

36C3 RESOURCE EXHAUSTION

Who’s That?

Ingo (Kleiber)

– (Computational) Linguist & teacher educator at Heidelberg University (HSE)

– Interested in a wide range of (often unrelated) things such as (digital) education, languages, coffee, photography, artificial intelligence, (political) philosophy, economics, …

– Not a programmer; similarly to the fact that you’re not an ‘e-mailer’

Today’s Aims

You will be able to ...

- describe what programming essentially is about
- name and describe some basic programming terminology
- model simple problems in terms of data structures and simple algorithms
- implement a simple solution to a problem in Python

Programming

“It’s difficult not to have a love/hate relationship with computer programming if you have any relationship with it at all.”

(Rosenberg 2006)

Code Along!

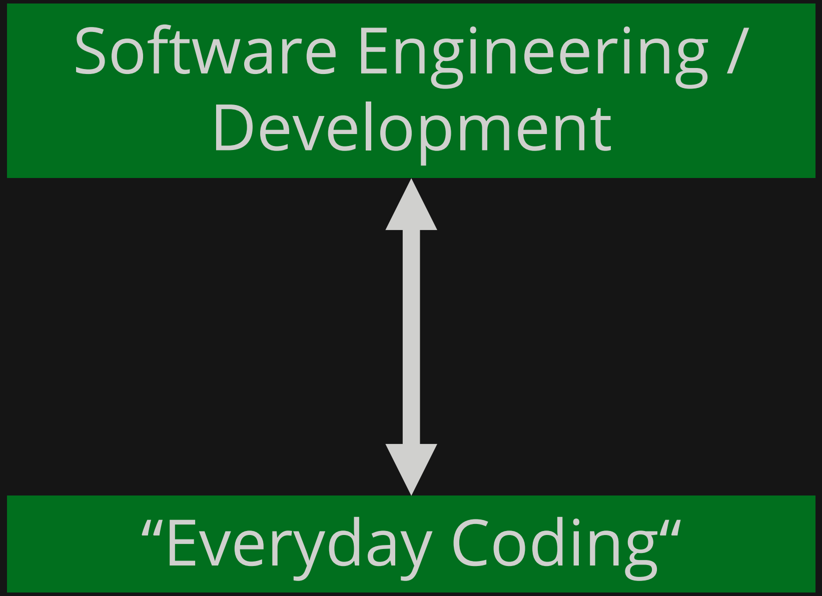
If you like, you can **code and experiment along**!

https://github.com/IngoKl/36c3-workshops

(then use *Binder*)

Programming is …

- instructing machines and computers
- problem solving
- thinking differently (computationally)
- modeling problems and other things
- an art
- fun
- …



Disclaimer

Everything that follows should be considered a (gross) oversimplification of reality!

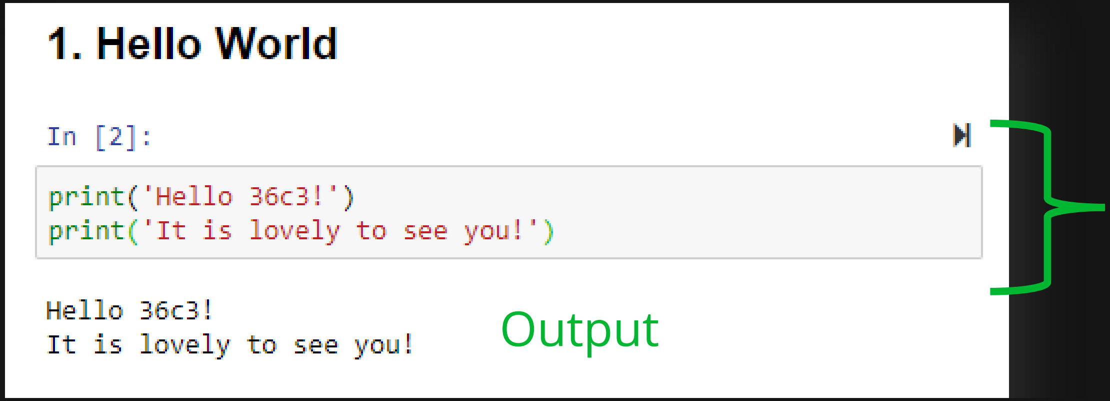
Python

Python is one of hundreds of programming languages.

- free, open, and available on almost any platform
- modern and widely used; great community
- relatively easy to learn; hard to master
- legacy Python (2.x) vs. **modern Python** (3.x)

What does Code Look Like?

Usually, something like this …


1. Hello World

In [2]:

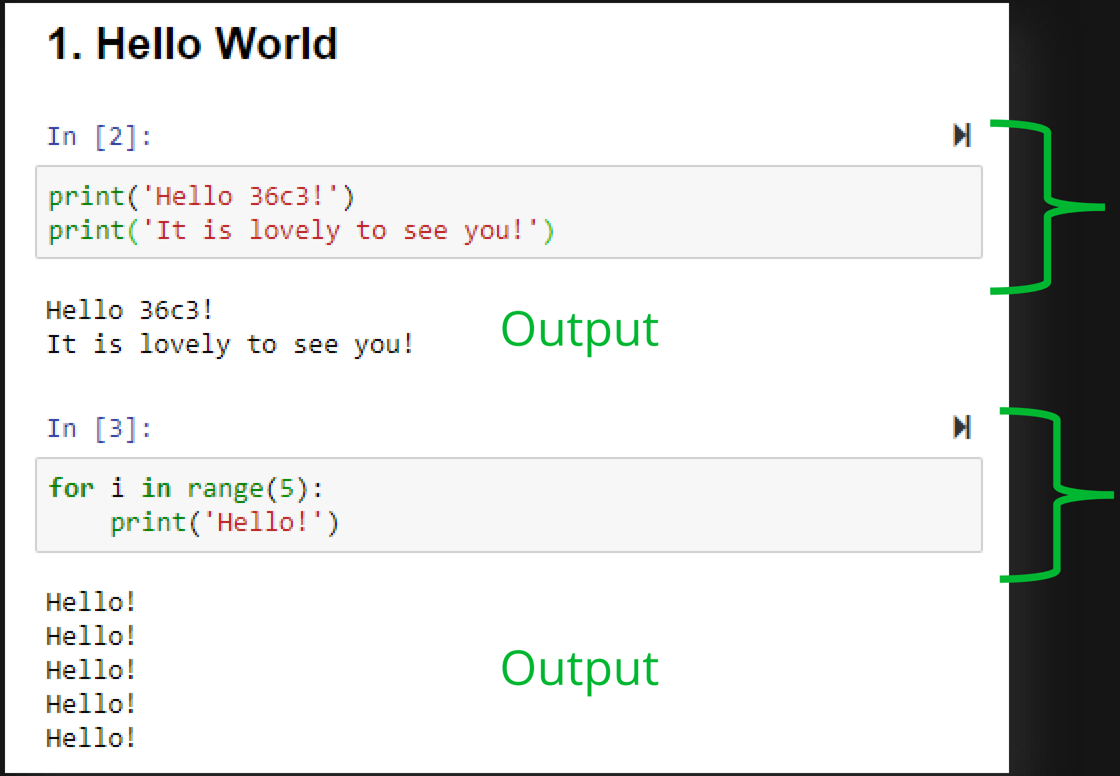
```
print('Hello 36c3!')
print('It is lovely to see you!')
```

```
Hello 36c3!
It is lovely to see you!
```

Two lines of code
Each line = one command
Executed in order

What does Code Look Like?

Usually, something like this …



1. Hello World

In [2]:

```
print('Hello 36c3!')
print('It is lovely to see you!')
```

```
Hello 36c3!
It is lovely to see you!
```

In [3]:

```
for i in range(5):
    print('Hello!')
```

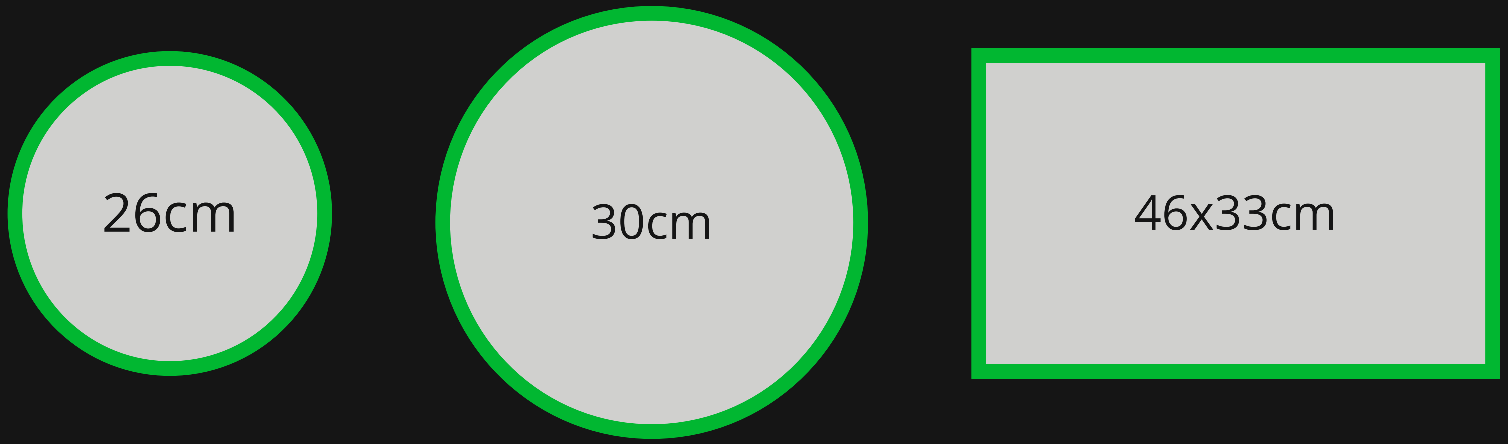
```
Hello!
Hello!
Hello!
Hello!
Hello!
```

Two lines of code
Each line = one command
Executed in order

Block of code
One 'main' line and multiple indented lines
A unit of functionality

A Real Problem: The Pizza Problem!

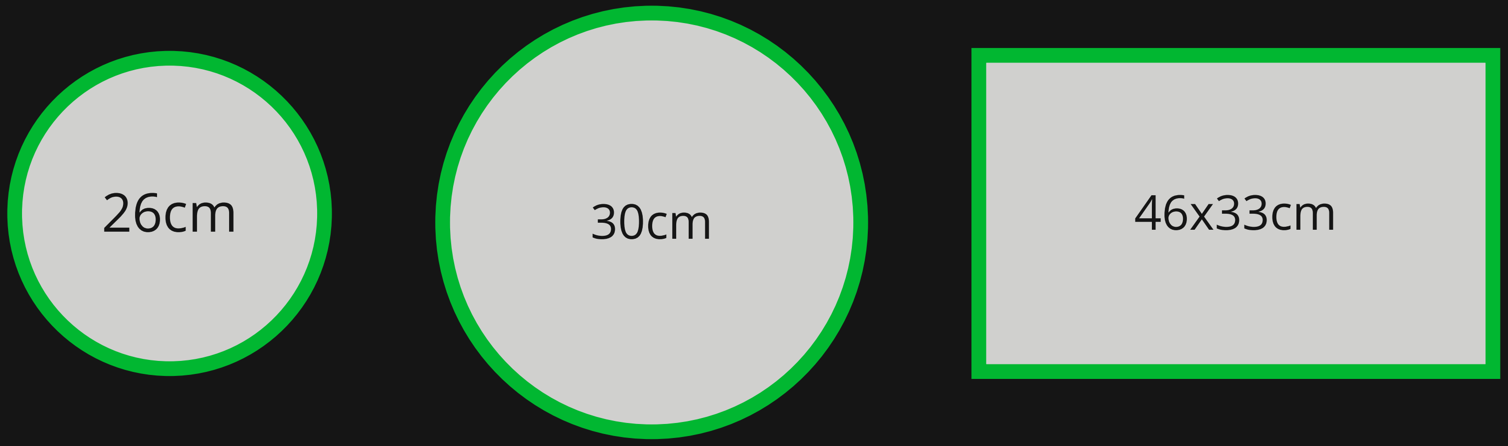
At Sue’s Pizza, you can oder **three types of pizza**:



Small for 4.80 **Large** for 5.50 **Party** for 13.00

A Real Problem: The Pizza Problem!

At Sue’s Pizza, you can oder **three types of pizza**:



Small for 4.80 **Large** for 5.50 **Party** for 13.00

$A = \pi \times r^2 \quad A = a \times b$

$A_S = 531cm^2 \longrightarrow 111/€$

$A_L = 707cm^2 \longrightarrow 128/€$ ✓

$A_P = 1518^2 \longrightarrow 116/€$

A Real Problem: The Pizza Problem!

For every (coding) **problem**, there are various solutions and approaches …

In **programming**, some common measures for **good solutions** are:

(1) simplicity *(2)* reusability *(3)* testability *(4)* understandability

(5) compliance *(6)* maintainability *(7)* efficiency *(8)* robustness

→ We’re aiming for a solution which is **just good enough**!

A Real Problem: The Pizza Problem!

Back to the pizza problem …

1. Determine sizes, prices, and shapes of n pizzas
2. For each pizza, determine its area (A)
3. For each pizza, calculate the pizza to Euro ration (PTER)
4. Determine the best PTER

Coding/Python Basics

In order to do this, we are going to need **some basics** ...

- **Variables** = a container to put data in (r = 13)
- **Lists** = a list of data-things (e.g. variables) (l = [1,2,3])
- **Loops** = repeating something until some condition is met
- **If-Constructions** = do something if some condition is met
- **Functions** = a unit of code that completes a specific task
- **Dictionaries**

1 & 2 - Variables and Lists

```
a = 13
b = ‘Hello 36c3’
c = 36.3
```

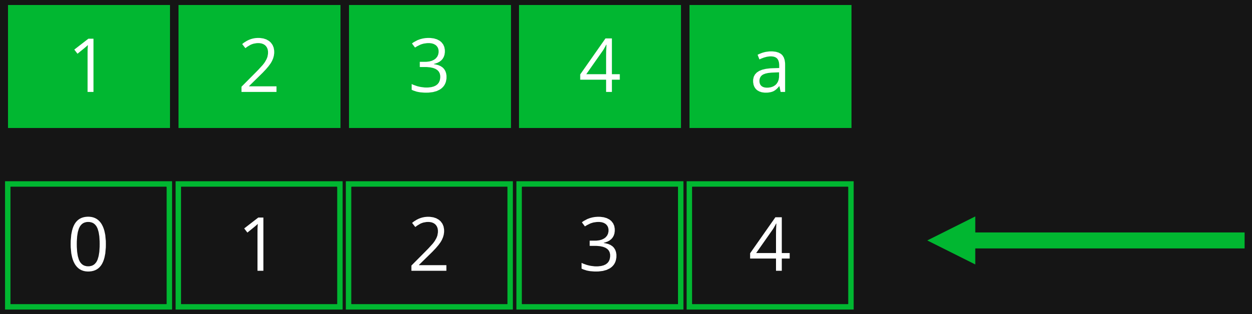
Three variables (containers) of three different types: *integer, string, and float*

```
l = [1, 2, 3, 4, a]
```

A list (named l) containing 4 integers and the variable a.

1 & 2 - Variables and Lists

l = [1, 2, 3, 4, a]



We always start counting at 0

l[0] → 1
l[3] → 4

1 & 2 - Variables and Lists

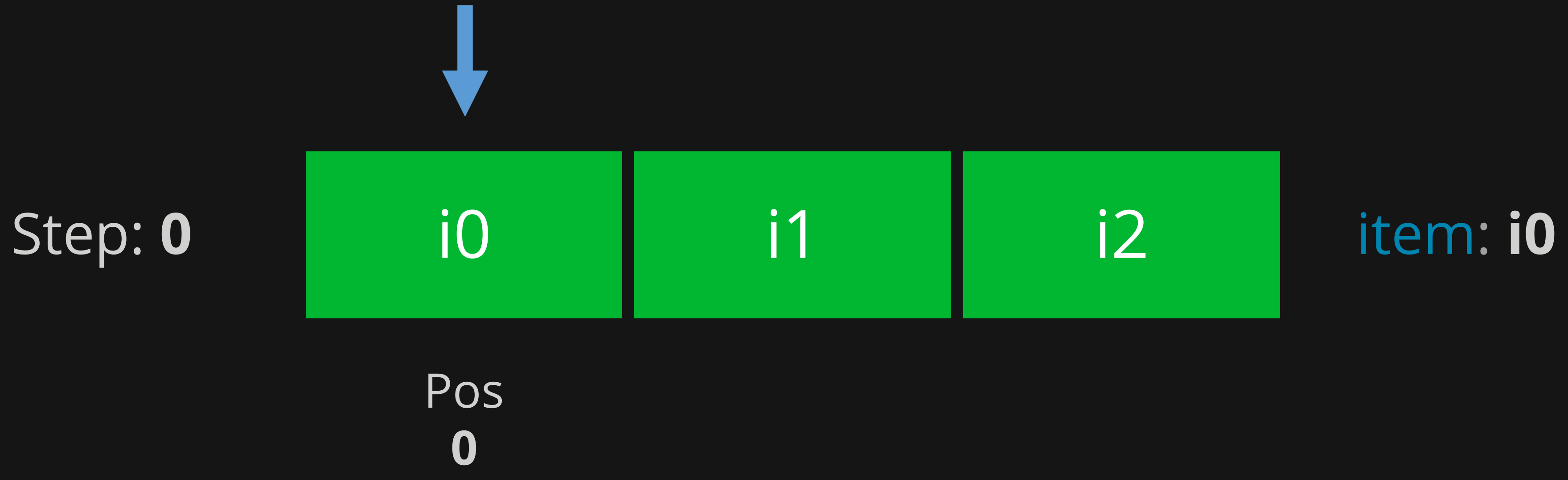
la = [1, 2, 3]
lb = [4, 5, 6]

lol = [la, lb] → [[1,2,3], [4,5,6]] } A list of lists

lol[0][1] → 2

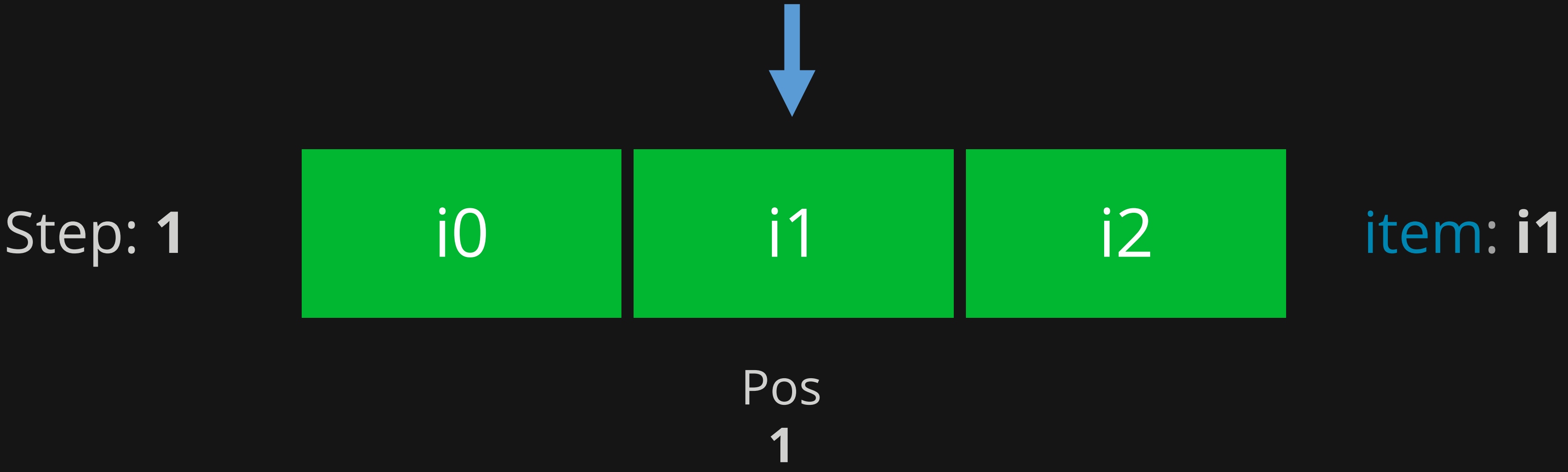
3 - Loops

```
box = ['i0', 'i1', 'i2']
for item in box:
        print(item)
```



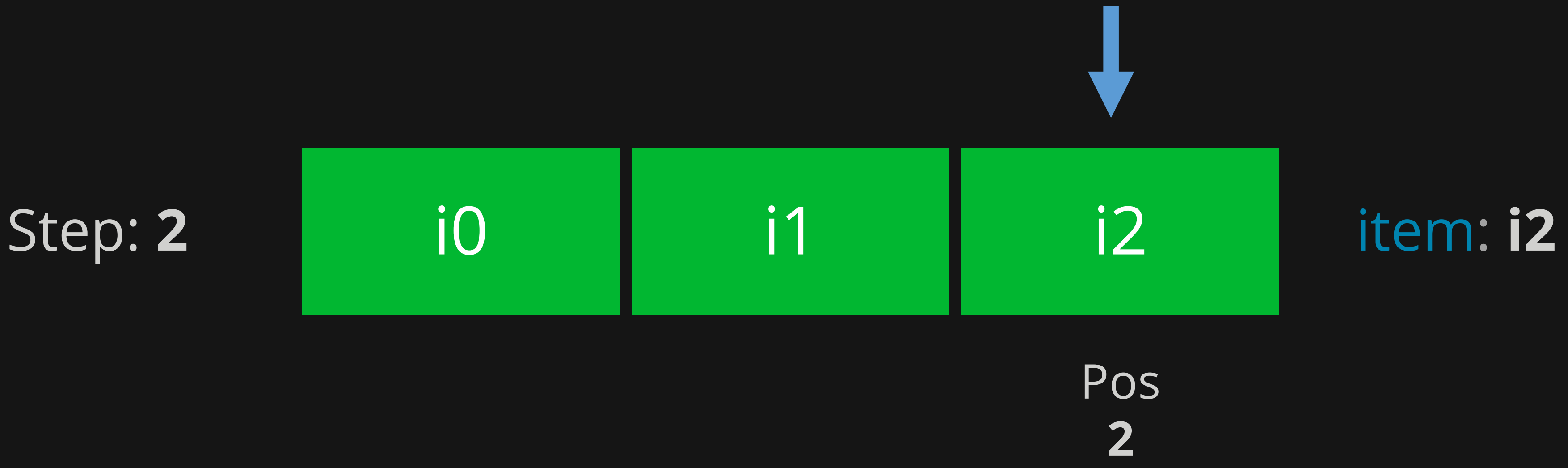
3 - Loops

```
box = ['i0', 'i1', 'i2']
for item in box:
        print(item)
```



3 - Loops

```
box = ['i0', 'i1', 'i2']
for item in box:
        print(item)
```



4 – If-Construction

```
a = 10

if a > 15:
    print (‘A is greater than 15’)
else:
    print (‘A is not greater than 15’)
```

5 – Functions

Two parameters which we pass to the function.

```
def add(a, b):
   result = a + b

   return result
```

What the function returns

```
add(5, 10) → 15
add(2, 2) → 4
```

Modeling Pizza as a List

Type Size Price Shape

```
ps = ['small', [26, 0], 4.80, 'circle']
```

↑

Shape is, implicitly,
encoded here as well!

A Very Simple Algorithm

Imagine we wanted to **find the youngest and the oldest person** in the room …

Bonus Exercises

1. How can we find the ideal (i.e. best priced) combination of pizzas for a given area that is being requested?
2. What if we were looking to optimize for as much/little crust as possible?
3. What about a second/third size dimension (i.e. height)?

What's Next?

A Small Selection of Books

Learn Python the Hard Way (Z. A. Shaw)

Python Crash Course (E. Matthes)

Python 3 for Absolute Beginners (T. Hall and J-P. Stacey)

A Small Selection of Courses

Codecademy

DataCamp

FreeCodeCamp

django girls

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

36C3 RESOURCE EXHAUSTION

Works Cited

– Rosenberg, Scott. 2006. *Dreaming in Code*. New York: Three Rivers Press.