Software Development Practical

Computer Vision & Deep Learning

Self-Introduction

Any programming experience?

What is your Python experience?

Any Machine Learning experience?

Any Deep Learning experience?

Overview

Introduction

- Math Basics
- Human Perception & Computer Vision

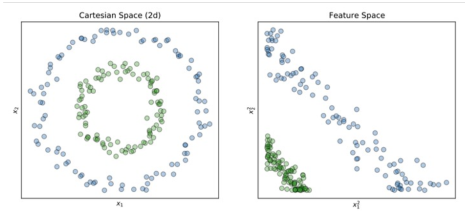
Learning from Data

- Unsupervised
- Supervised

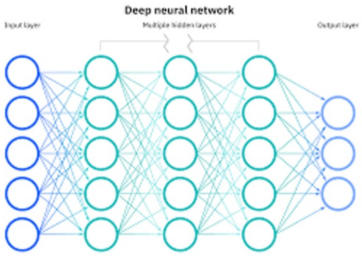
Python Fundamentals

Deep Learning

- Building Blocks
- (Convolutional) Neural Networks



https://sthalles.github.io/a-few-words-on-representation-learning/



https://www.ibm.com/topics/neural-networks

General comments

- Sometimes the math might be overwhelming on a first glance, so if you need help with understanding the concepts please don't hesitate to contact us. We will then do our best to help you.

- Contact details:

Ming Gui
ming.gui@lmu.de

Johannes Schusterbauer
joh.fischer@lmu.de

Organization

Teaching:

- Graded homework starting from next week

Final project:

- Group in teams of 4
- Focus on a given computer vision task using deep learning methods
- Submit a codebase and a group report before semester ends
- More details will follow…

Math Basics

Math notation

A symbolic representation of mathematical ideas and concepts using a set of symbols, characters, and mathematical operators

Why is it important?

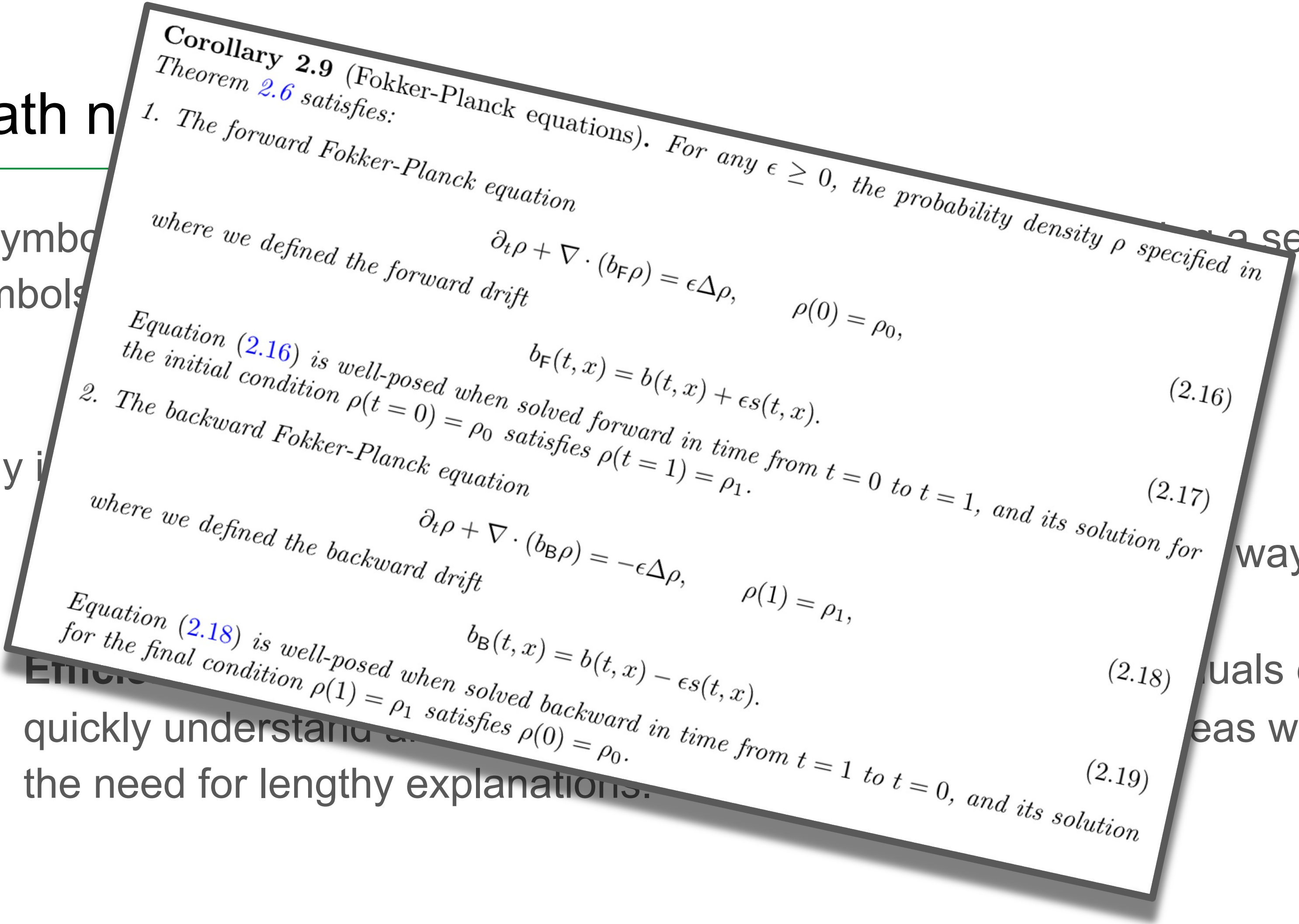
- **Clarity and Consistency**: Notation provides a clear and concise way to express mathematical concepts and ideas.
- **Efficient communication**: With a standardized notation, individuals can quickly understand and communicate complex mathematical ideas without the need for lengthy explanations.

Math n

A symbo … a set of symbols

Why i

- … way to
- **Effici** … uals can quickly understand a … eas without the need for lengthy explanations.



Corollary 2.9 (Fokker-Planck equations). *For any $\epsilon \geq 0$, the probability density ρ specified in Theorem 2.6 satisfies:*

1. *The forward Fokker-Planck equation*

$$\partial_t \rho + \nabla \cdot (b_{\mathsf{F}} \rho) = \epsilon \Delta \rho, \qquad \rho(0) = \rho_0, \tag{2.16}$$

where we defined the forward drift

$$b_{\mathsf{F}}(t,x) = b(t,x) + \epsilon s(t,x). \tag{2.17}$$

Equation (2.16) is well-posed when solved forward in time from $t = 0$ to $t = 1$, and its solution for the initial condition $\rho(t = 0) = \rho_0$ satisfies $\rho(t = 1) = \rho_1$.

2. *The backward Fokker-Planck equation*

$$\partial_t \rho + \nabla \cdot (b_{\mathsf{B}} \rho) = -\epsilon \Delta \rho, \qquad \rho(1) = \rho_1, \tag{2.18}$$

where we defined the backward drift

$$b_{\mathsf{B}}(t,x) = b(t,x) - \epsilon s(t,x). \tag{2.19}$$

Equation (2.18) is well-posed when solved backward in time from $t = 1$ to $t = 0$, and its solution for the final condition $\rho(1) = \rho_1$ satisfies $\rho(0) = \rho_0$.

Math notation: Sum and product

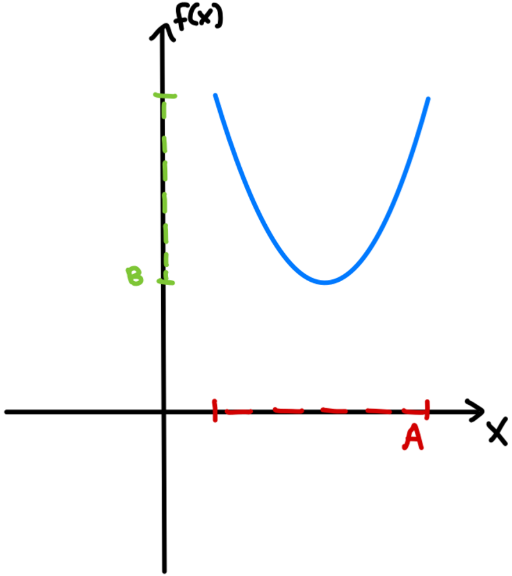
$$a_1 + a_2 + \ldots + a_n = \sum_{i=1}^{n} a_i = \sum_{i} a_i$$

$$a_1 \times a_2 \times \ldots \times a_n = \prod_{i=1}^{n} a_i = \prod_{i} a_i$$

Functions

A function ***f*** assigns to each element of its definition set ***A*** exactly one element of its target set ***B***, this is written as:

$$\begin{aligned} f\colon A &\to B\,, \\ a &\mapsto f(a)\,. \end{aligned}$$



Run an ice-cream shop!

You want to predict how many ice-cream you sell based on the temperature and whether it rains.

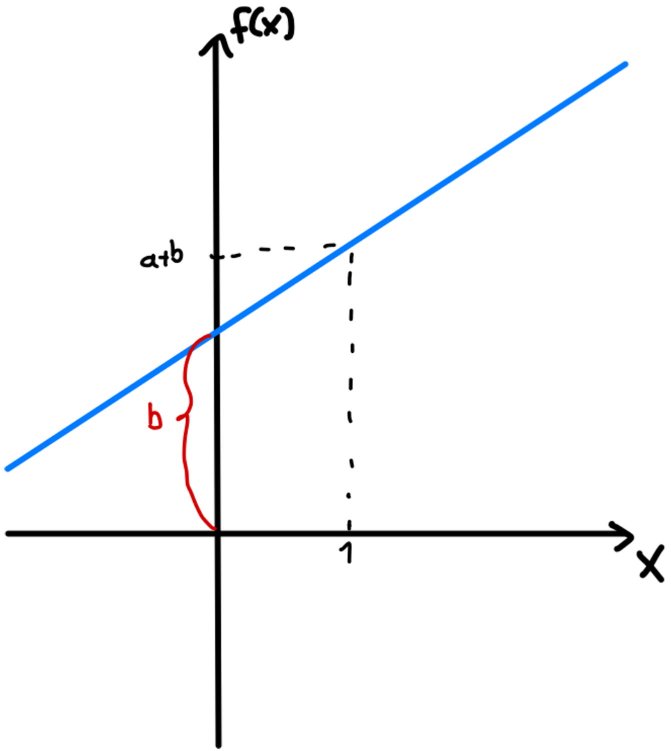
We can construct the following model using a function:

f(temperature, rain) = number of ice-cream

Functions in one-dimension

Linear:

$$\begin{aligned} f_{a,b}\colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = a \cdot x + b\,. \end{aligned}$$



Functions in one-dimension

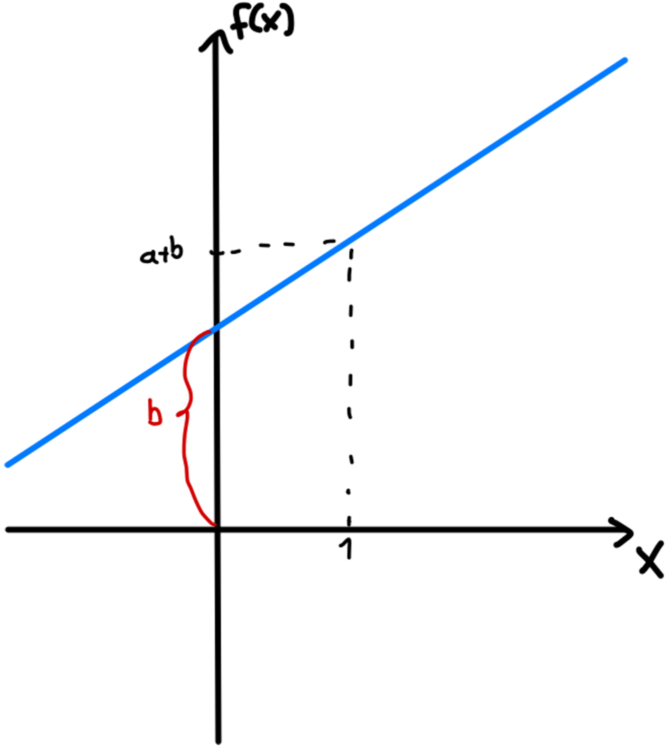
Linear:

$$\begin{aligned} f_{a,b} \colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = a \cdot x + b\,. \end{aligned}$$

Polynomial:

$$\begin{aligned} f_{a,b} \colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = \sum_i a_i x^i + b\,. \end{aligned}$$

Exponential, sinusodal, sigmoid …



1st-order polynomial

Functions in one-dimension

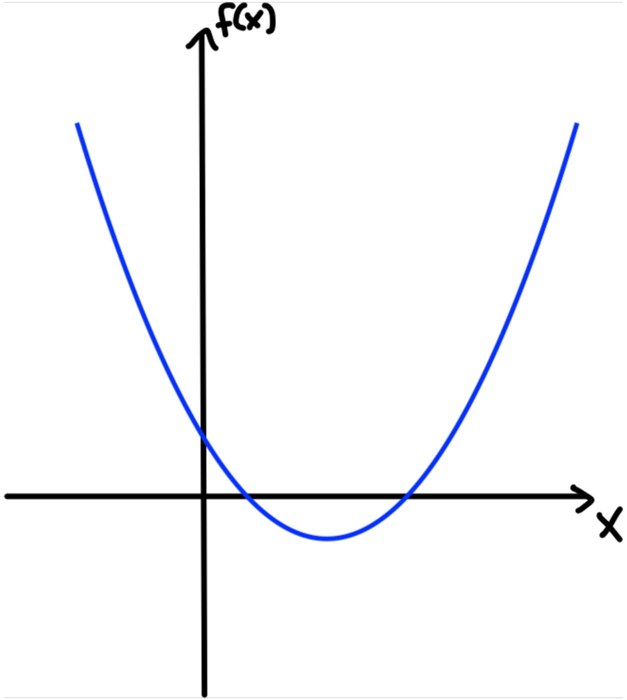
Linear:

$$\begin{aligned} f_{a,b} \colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = a \cdot x + b\,. \end{aligned}$$

Polynomial:

$$\begin{aligned} f_{a,b} \colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = \sum_i a_i x^i + b\,. \end{aligned}$$

Exponential, sinusodal, sigmoid …



2nd-order polynomial

Functions in one-dimension

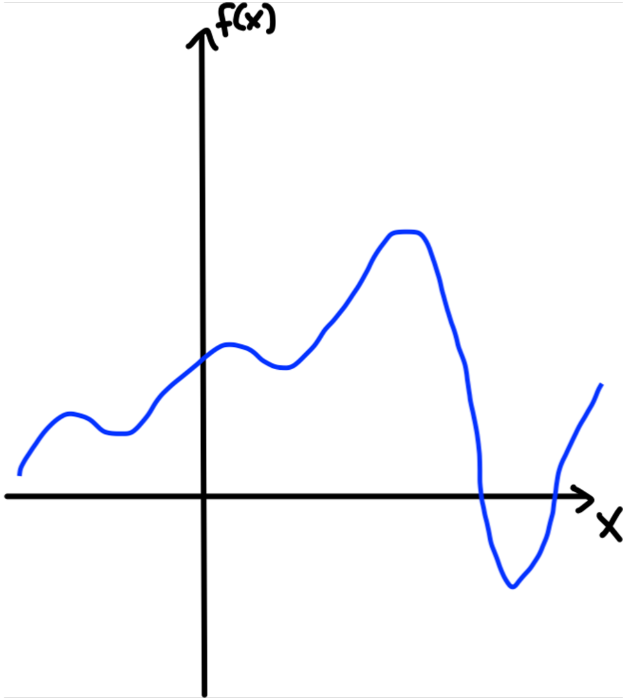
Linear:

$$\begin{aligned} f_{a,b}\colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = a \cdot x + b\,. \end{aligned}$$

Polynomial:

$$\begin{aligned} f_{a,b}\colon \mathbb{R} &\to \mathbb{R}\,, \\ x &\mapsto f(x) = \sum_i a_i x^i + b\,. \end{aligned}$$

Exponential, sinusodal, sigmoid …

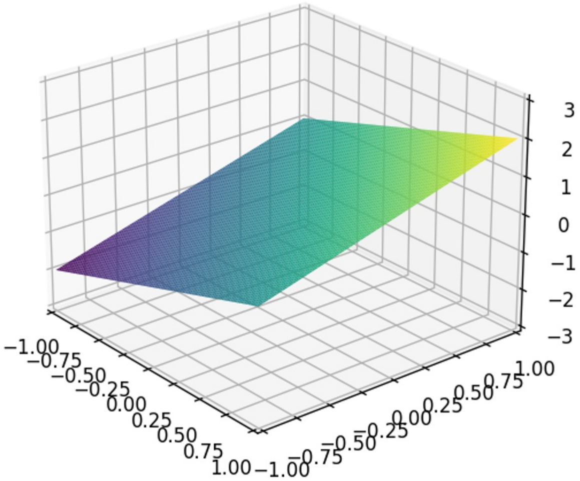


multi-order polynomial

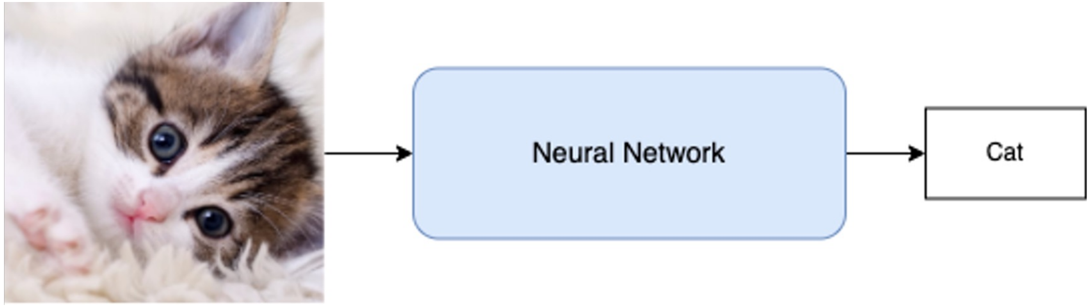
Multi-dimensional functions

2D Linear:

$$f(x_1, x_2) = a_1 x_1 + a_2 x_2 + b$$



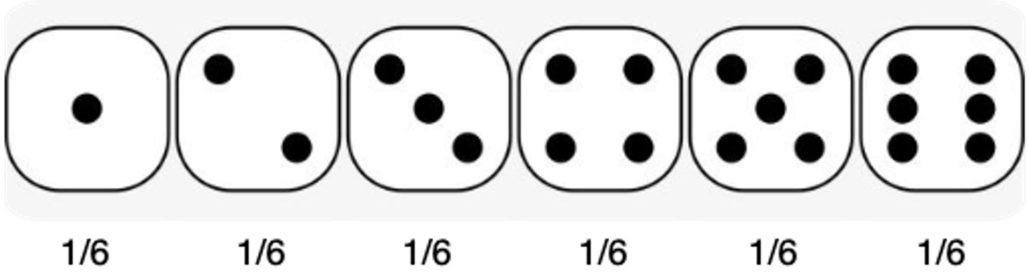
Neural networks are also functions



Neural Network

Cat

Probability



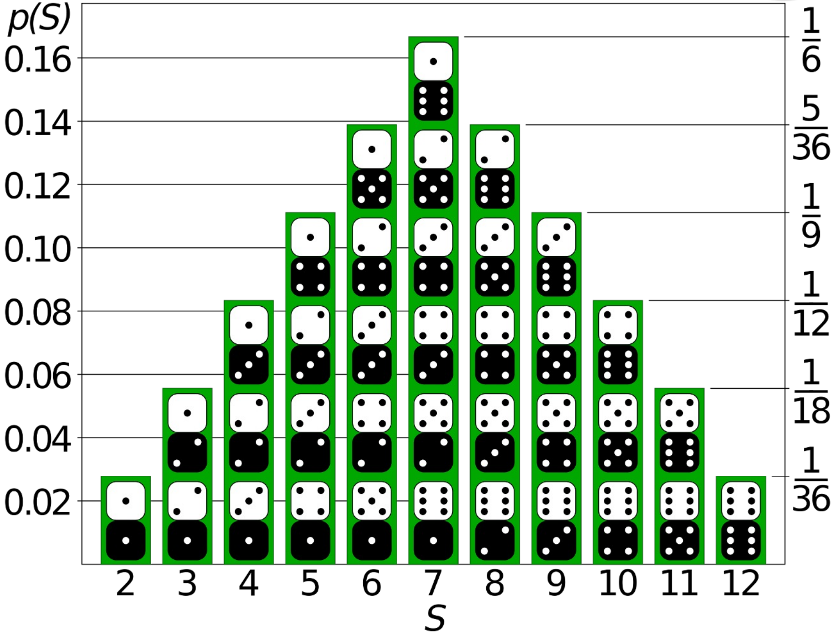
Assuming x is a random variable: $\mathbf{P}(x)$

In case of a die throw: $\mathbf{P}(x = k) = \frac{1}{6}, k = 1, \cdots, 6$

Conditional probability

Consider throwing two dice

$$\mathbf{P}(x_1 + x_2 = 5) = \frac{4}{36}$$



https://math.stackexchange.com/questions/1204396/why-is-the-sum-of-the-rolls-of-two-dices-a-binomial-distribution-what-is-define

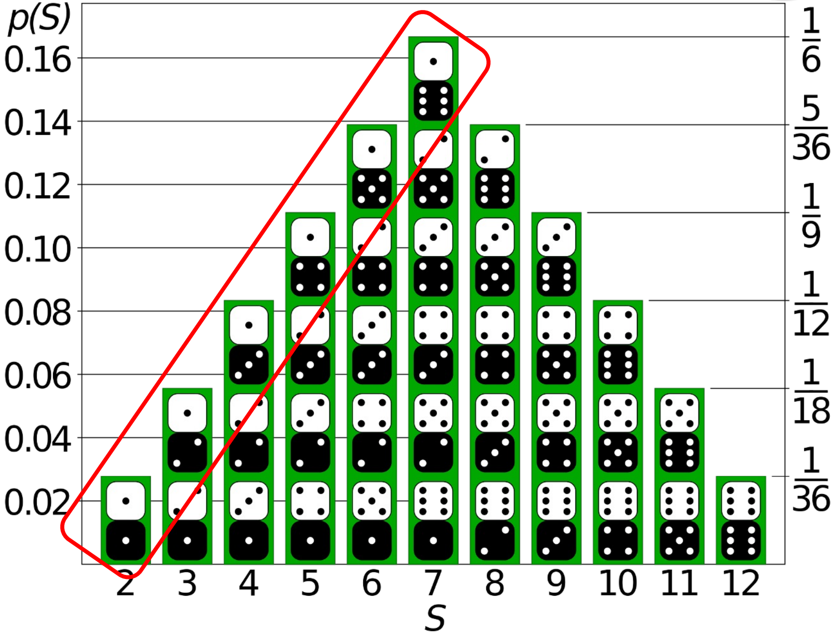
Conditional probability

Consider throwing two dice

$$\mathbf{P}(x_1 + x_2 = 5) = \frac{4}{36}$$

The conditioning changes the probability

$$\mathbf{P}(x_1 + x_2 = 5 \mid \color{red}{x_1 = 1}) = \frac{1}{6}$$



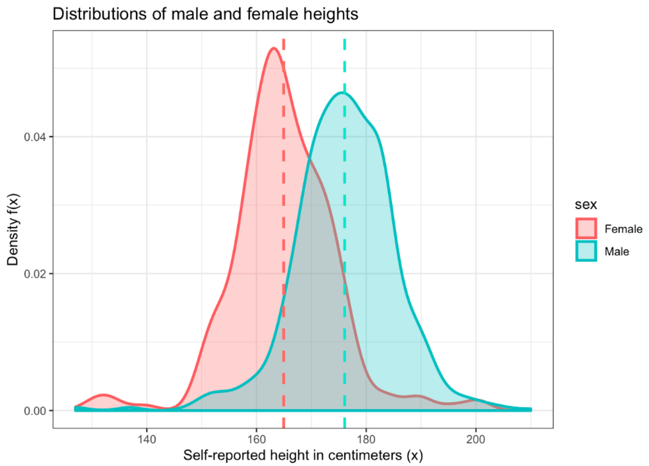
https://math.stackexchange.com/questions/1204396/why-is-the-sum-of-the-rolls-of-two-dices-a-binomial-distribution-what-is-define

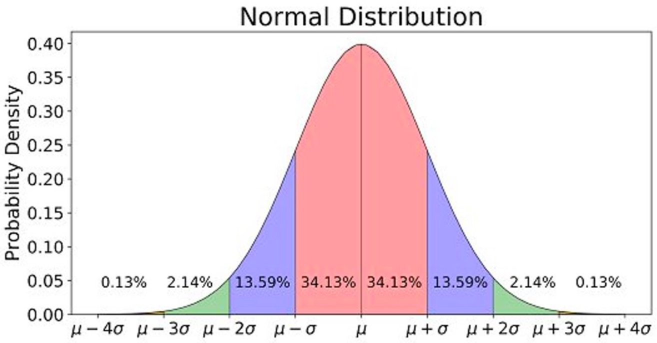
Continuous probability

Not all values are discrete
(height, rainfall…)

The likelihood is defined by the probability density function

Gaussian distribution / Normal distribution
$\mathcal{N}(\mu, \sigma^2)$





https://bookdown.org/gabriel_butler/ECON41Labs/tutorial-6-the-normal-distribution.html
https://www.thinkingondata.com/normal-distribution-what-means-in-numbers/

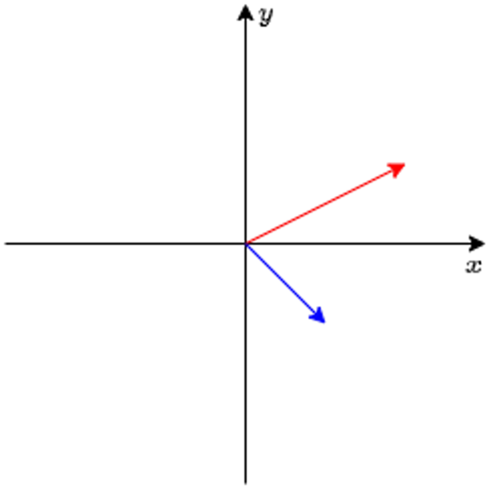
Vectors, matrices and tensors

We might need more than one number to describe the circumstance

A vector is represented as a list of numbers, where each number represents the magnitude of the vector in a particular direction.

$$a = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

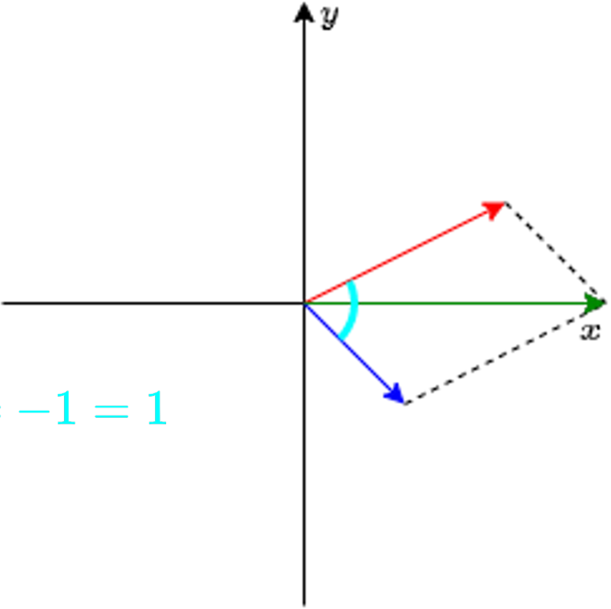
$$\mathrm{Norm}(a) = \sqrt{\sum_i a_i^2} = \sqrt{2^2 + 1^2} = \sqrt{5}$$



Vectors calculation

$$a = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Add $$a + b = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$



Inner product $$a \cdot b = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 2 * 1 + 1 * -1 = 1$$

Cosine similarity $$\frac{a \cdot b}{\|a\| \|b\|} = \frac{1}{\sqrt{5}\sqrt{2}} = \frac{1}{\sqrt{10}}$$

$$\cos^{-1}(1/\sqrt{10}) = 71.57°$$

Matrix

A matrix is just a table of scalars:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

And its transpose:

$$A^\top = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Matrix

A matrix is just a table of scalars:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

And its transpose:

$$A^\top = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

$$A^\top = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

Matrix multiplication

For $\boldsymbol{A} \in \mathbb{R}^{k \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times m}$:

$$\boldsymbol{A} \cdot \boldsymbol{B} = \begin{pmatrix} — \; \boldsymbol{a}_{1\bullet} \; — \\ — \; \boldsymbol{a}_{2\bullet} \; — \\ \vdots \\ — \; \boldsymbol{a}_{n\bullet} \; — \end{pmatrix} \cdot \begin{pmatrix} | & | & & | \\ \boldsymbol{b}_{\bullet 1} & \boldsymbol{b}_{\bullet 2} & \dots & \boldsymbol{b}_{\bullet m} \\ | & | & & | \end{pmatrix}$$

$$= \begin{pmatrix} \langle a_{1\bullet}, b_{\bullet 1} \rangle & \dots & \langle a_{1\bullet}, b_{\bullet m} \rangle \\ \vdots & \ddots & \vdots \\ \langle a_{n\bullet}, b_{\bullet 1} \rangle & \dots & \langle a_{n\bullet}, b_{\bullet m} \rangle \end{pmatrix} \in \mathbb{R}^{k \times m}$$

Matrix multiplication

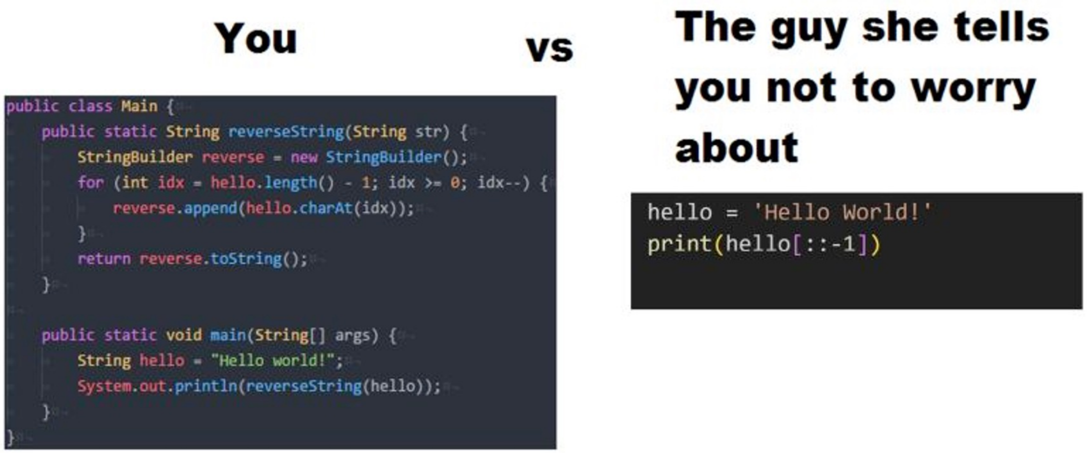
$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \qquad A \cdot B = \begin{bmatrix} \varphi & \cdot \\ \cdot & \cdot \end{bmatrix}$$

$$\varphi = \begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 7 \end{bmatrix} = \left\langle \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 5 \\ 7 \end{pmatrix} \right\rangle = 1 * 5 + 2 * 7 = 19$$

Python fundamentals

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics



```java
public class Main {
    public static String reverseString(String str) {
        StringBuilder reverse = new StringBuilder();
        for (int idx = hello.length() - 1; idx >= 0; idx--) {
            reverse.append(hello.charAt(idx));
        }
        return reverse.toString();
    }

    public static void main(String[] args) {
        String hello = "Hello world!";
        System.out.println(reverseString(hello));
    }
}
```

```python
hello = 'Hello World!'
print(hello[::-1])
```

https://media.licdn.com/dms/image/D4D22AQE9J-x5iMQztw/feedshare-shrink_800/0/1686569314381?e=1701302400&v=beta&t=HeoxoAv_KKuXM67nslW7hanSv8iKrR4Y720I6QBE_HI

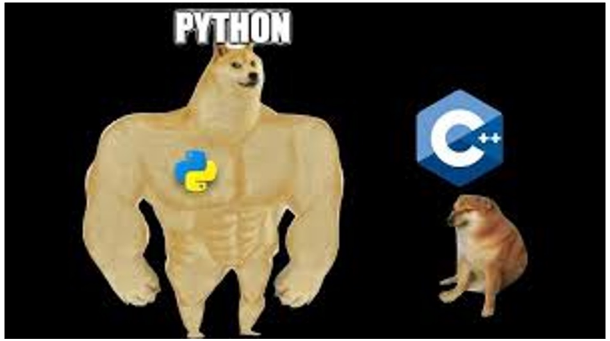
What is Python?

- Data analysis, machine learning, web developments…

- Concise and readable syntax, no compilation

- Great ecosystem with a wide range of libraries
  NumPy, SciPy, PyTorch and sooo many more …

- Easy to pick up!



Getting started with Python

Working with different environments:

- Avoid package dependency conflicts

Find how to download **miniconda** here

Create your environment using the following command:

```
conda create -n "myenv" python=3.12.0
```

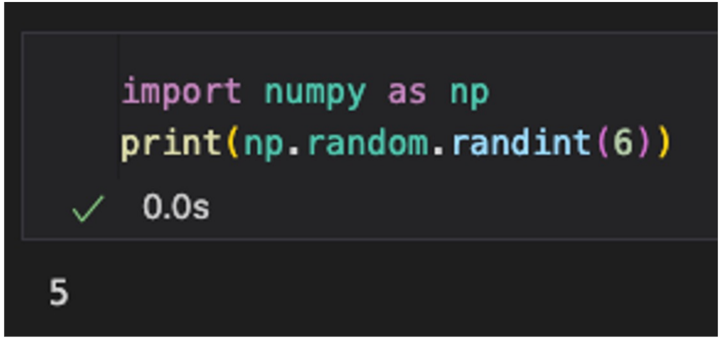
And activate the corresponding environment:

```
conda activate myenv
```

Installing packages

Use conda or pip

- Activate your environment first!
- `conda install numpy` / `pip install numpy`



```
import numpy as np
print(np.random.randint(6))
```

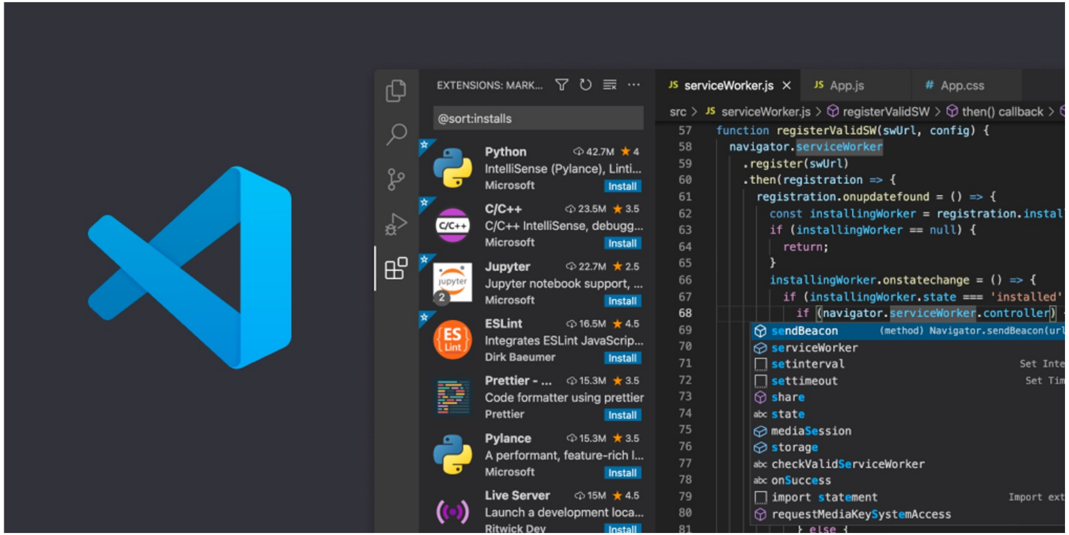
✓ 0.0s

```
5
```

Select your IDE

Integrated Development Environment

- VSCode / PyCharm / Vim



Python 42.7M ★ 4
IntelliSense (Pylance), Linti...
Microsoft Install

C/C++ 23.5M ★ 3.5
C/C++ IntelliSense, debugg...
Microsoft Install

Jupyter 22.7M ★ 2.5
Jupyter notebook support, ...
Microsoft Install

ESLint 16.5M ★ 4.5
Integrates ESLint JavaScrip...
Dirk Baeumer Install

Prettier - ... 15.3M ★ 3.5
Code formatter using prettier
Prettier Install

Pylance 15.3M ★ 3.5
A performant, feature-rich l...
Microsoft Install

Live Server 15M ★ 4.5
Launch a development loca...
Ritwick Dey Install

```
function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
      registration.onupdatefound = () => {
        const installingWorker = registration.instal
        if (installingWorker == null) {
          return;
        }
        installingWorker.onstatechange = () => {
          if (installingWorker.state === 'installed'
            if (navigator.serviceWorker.controller)
```

```
sendBeacon                    (method) Navigator.sendBeacon(url
serviceWorker
setinterval                                          Set Inte
settimeout                                           Set Tim
share
state
mediaSession
storage
checkValidServiceWorker
onSuccess
import statement                                     Import ext
requestMediaKeySystemAccess
```

```
            } else {
```

https://code.visualstudio.com

Intro into Python

Hello World

```
print("hello world")
```

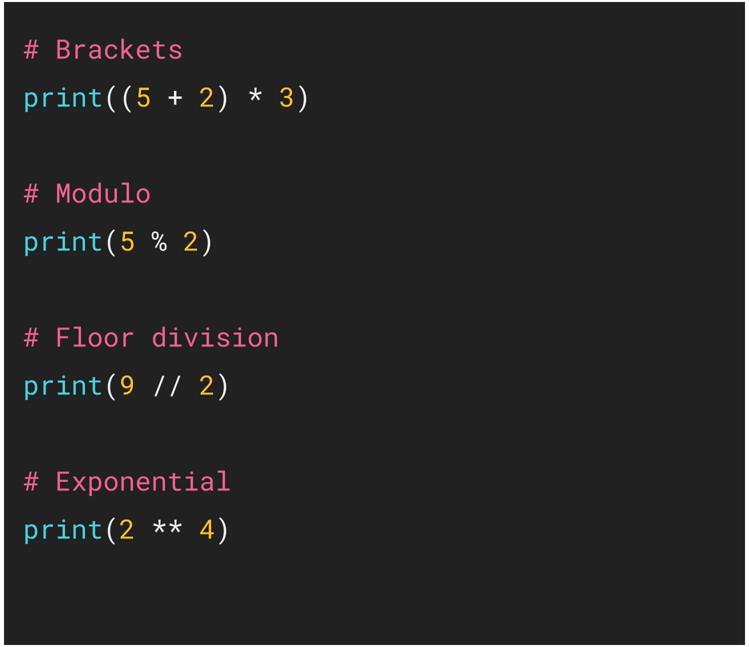
Starting point for every programming language…

Arithmetic Operators

```python
# Addition
print(1 + 3)

# Subtraction
print(3 - 1)

# Multiplication
print(5 * 3)

# Division
print(5 / 2)
```

Addition, subtraction, multiplication, division…

Arithmetic Operators

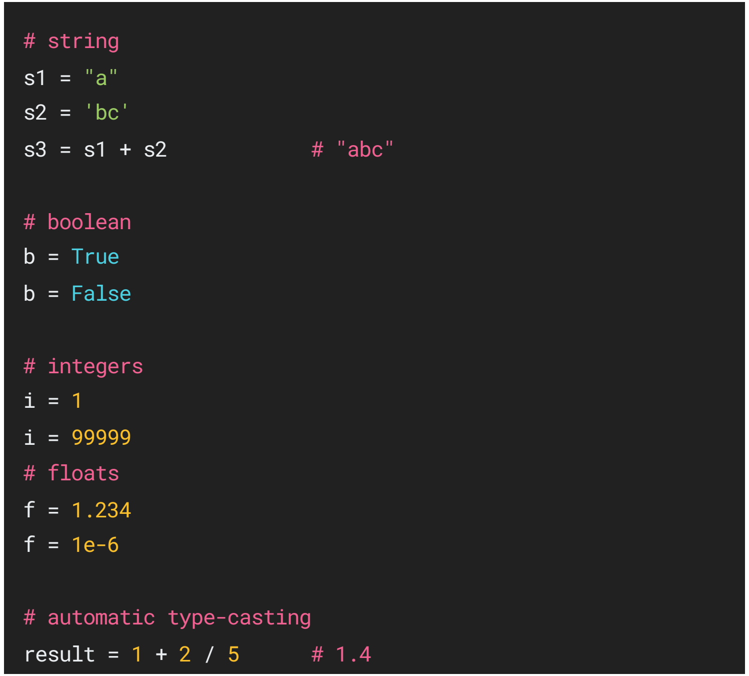


```
# Brackets
print((5 + 2) * 3)

# Modulo
print(5 % 2)

# Floor division
print(9 // 2)

# Exponential
print(2 ** 4)
```

Like in math we usually work from left to right. If in doubt, we can always use parentheses.

Modulo, floor division, exponential…

Variables



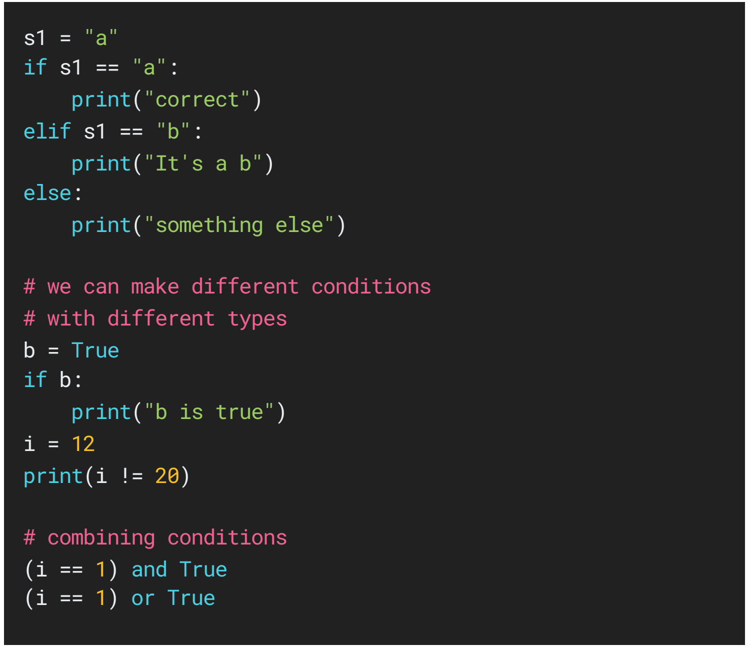
```
# string
s1 = "a"
s2 = 'bc'
s3 = s1 + s2            # "abc"

# boolean
b = True
b = False

# integers
i = 1
i = 99999
# floats
f = 1.234
f = 1e-6

# automatic type-casting
result = 1 + 2 / 5      # 1.4
```

Can not start with a number.

Must start with a lowercase letter, uppercase latter, or an underscore.

Names are case sensitive.

Conditions



```python
s1 = "a"
if s1 == "a":
    print("correct")
elif s1 == "b":
    print("It's a b")
else:
    print("something else")

# we can make different conditions
# with different types
b = True
if b:
    print("b is true")
i = 12
print(i != 20)

# combining conditions
(i == 1) and True
(i == 1) or True
```

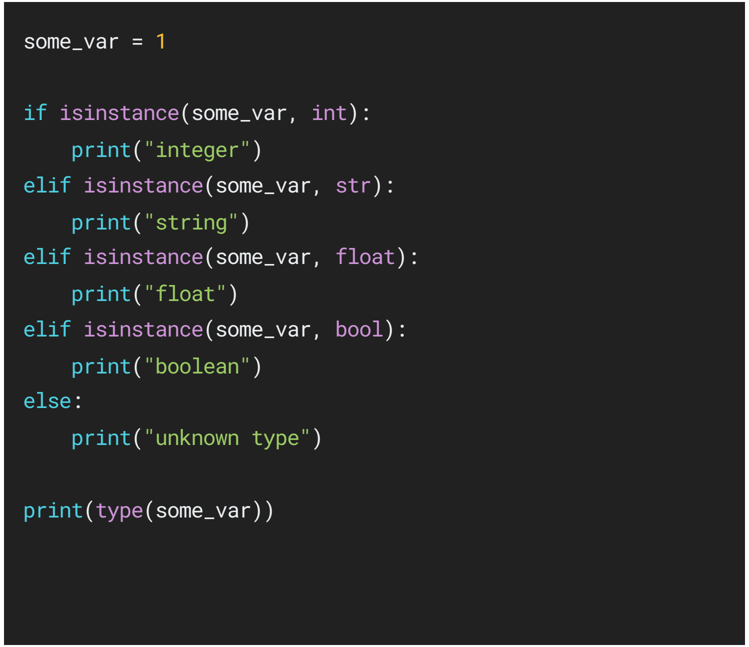
We have classic if-else clauses in python as in any other programming language.

Conditions

```
some_condition = True

if some_condition is True:
    ...

if some_condition is not True:
    ...
```

You can even write it with words…

Variable types



```
some_var = 1

if isinstance(some_var, int):
    print("integer")
elif isinstance(some_var, str):
    print("string")
elif isinstance(some_var, float):
    print("float")
elif isinstance(some_var, bool):
    print("boolean")
else:
    print("unknown type")

print(type(some_var))
```

In python everything is an object.

You can return the type of a variable with the function type(...)

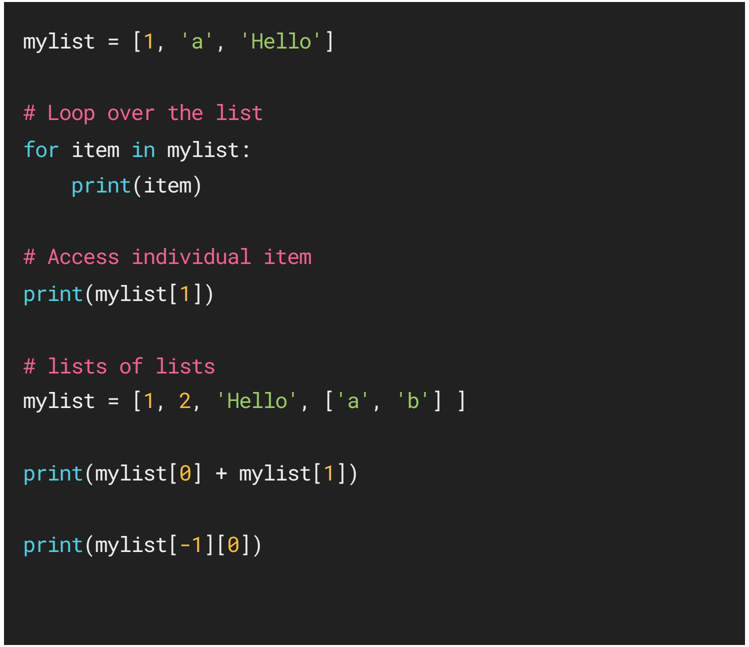
isinstance(<var>, <type>) allows you to check a variable type.

String operations and print

```python
a = 123

# combining strings and numbers
print("This is a number:", a)

new_string = f"{a} is a number"
new_string = "this is a number " + str(a)

# only print 2 decimals
float = 1.234567
print(f"{float:.2f}")
```

We can combine different types in different ways…

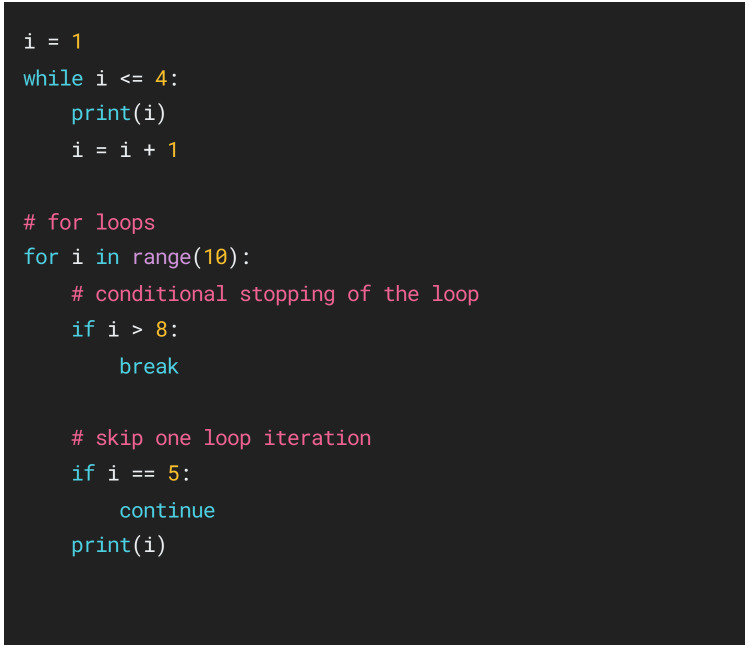
Lists



```python
mylist = [1, 'a', 'Hello']

# Loop over the list
for item in mylist:
    print(item)

# Access individual item
print(mylist[1])

# lists of lists
mylist = [1, 2, 'Hello', ['a', 'b'] ]

print(mylist[0] + mylist[1])

print(mylist[-1][0])
```

Lists don’t have to be of the same type, since everything in python is an object !

Loops



```python
i = 1
while i <= 4:
    print(i)
    i = i + 1

# for loops
for i in range(10):
    # conditional stopping of the loop
    if i > 8:
        break

    # skip one loop iteration
    if i == 5:
        continue
    print(i)
```

We have for and while loops in python.

We can skip iterations or stop the iteration if some condition is met.

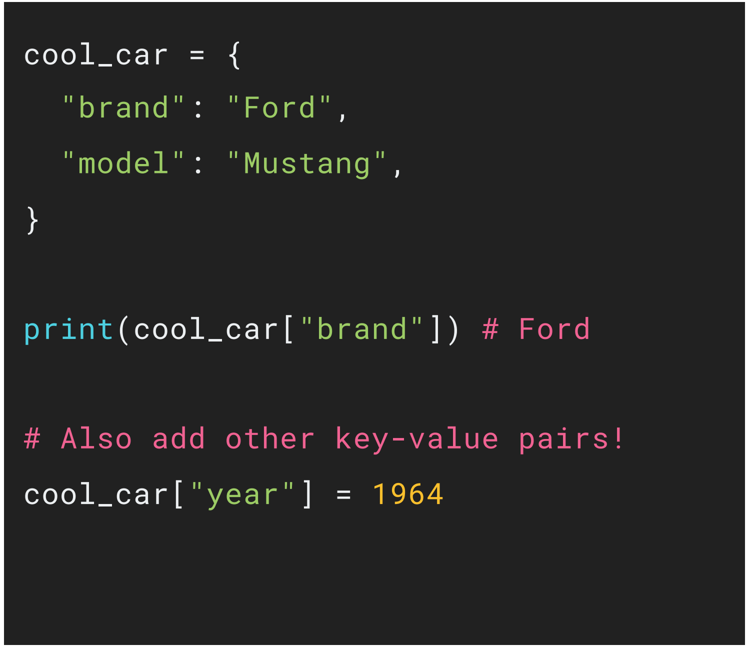
Filling a list within a loop

```
mylist = []

# adding elements
for i in range(5):
    mylist.append(i)

print(mylist)

# pop elements
last_element = mylist.pop()
first_element = mylist.pop(0)
```

We can iteratively populate a list by using append.

We can pop elements.

Dictionaries



```
cool_car = {
  "brand": "Ford",
  "model": "Mustang",
}

print(cool_car["brand"]) # Ford

# Also add other key-value pairs!
cool_car["year"] = 1964
```

Dictionaries store data values in key:value pairs.

The corresponding value can be referred to by using the key

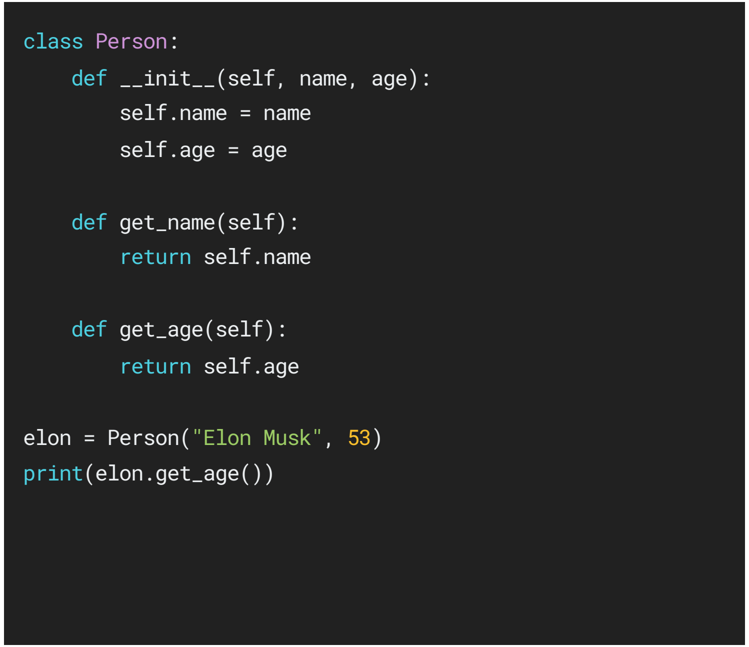
Functions

```
def add(x,y):
    return x + y

add(2,3) # should return 5
```

A function is a block of code which only runs when it is called.

You can pass parameters into a function, and it can (potentially) return data as a result.

Classes and subclasses

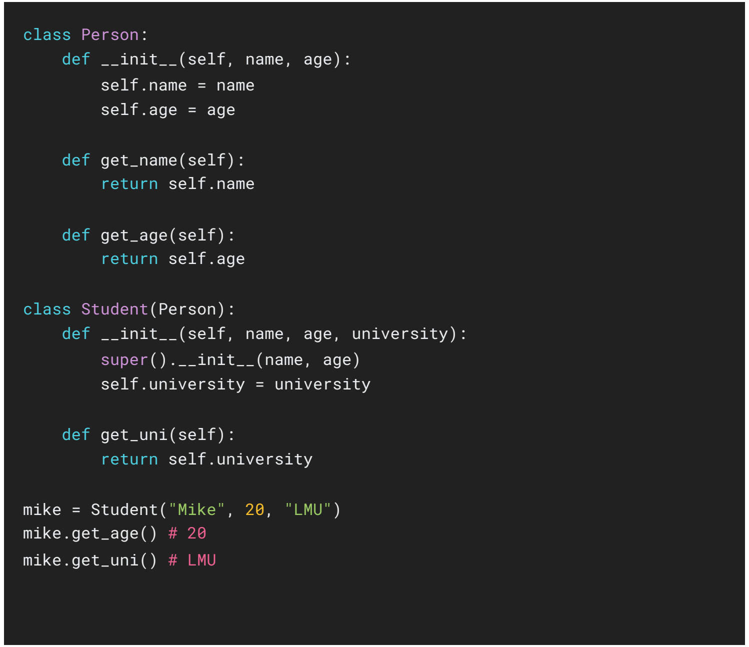


```python
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def get_name(self):
        return self.name

    def get_age(self):
        return self.age

elon = Person("Elon Musk", 53)
print(elon.get_age())
```

A Class is like an object constructor, or a "blueprint" for creating objects.

All classes have a function called __init__(), which is always executed when the class is being initiated.

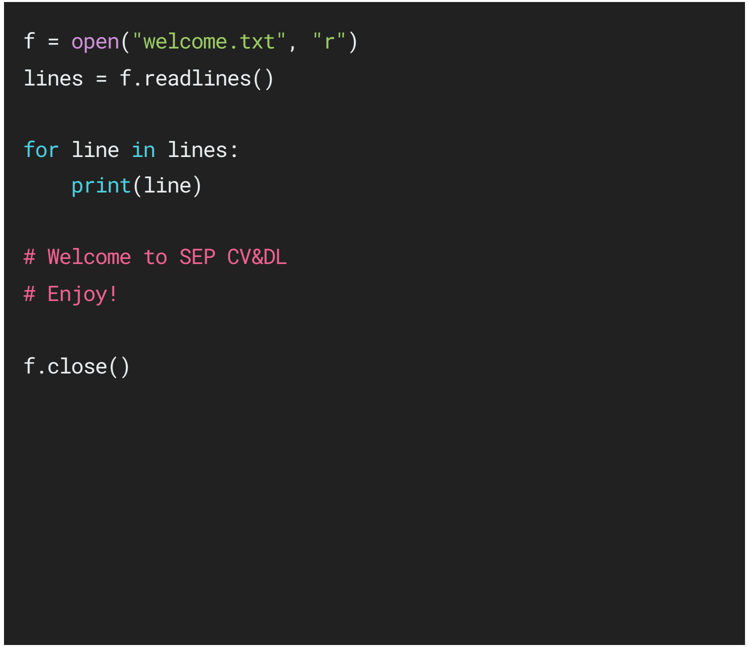
Classes and inheritance



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

    def get_name(self):
        return self.name

    def get_age(self):
        return self.age

class Student(Person):
    def __init__(self, name, age, university):
        super().__init__(name, age)
        self.university = university

    def get_uni(self):
        return self.university

mike = Student("Mike", 20, "LMU")
mike.get_age() # 20
mike.get_uni() # LMU
```

super() function inherits all the methods and properties from its parent

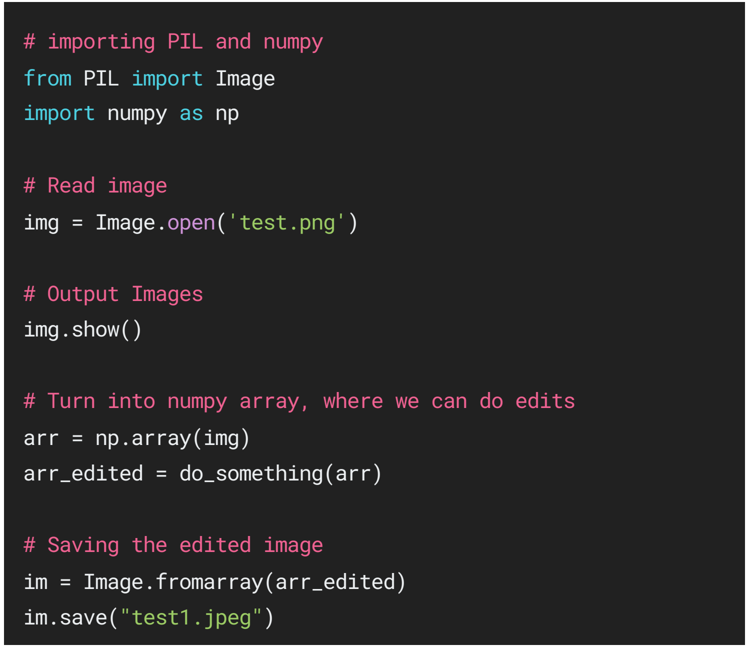
Opening files …



```python
f = open("welcome.txt", "r")
lines = f.readlines()

for line in lines:
    print(line)

# Welcome to SEP CV&DL
# Enjoy!

f.close()
```

The open() function returns a file object, which has read() and readlines() for reading the content of the file

And opening images …



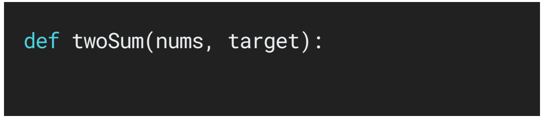
```
# importing PIL and numpy
from PIL import Image
import numpy as np

# Read image
img = Image.open('test.png')

# Output Images
img.show()

# Turn into numpy array, where we can do edits
arr = np.array(img)
arr_edited = do_something(arr)

# Saving the edited image
im = Image.fromarray(arr_edited)
im.save("test1.jpeg")
```

Pillow library (PIL) is great for reading and saving images

Turn into numpy arrays for data manipulation!

Programming exercise

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.



```python
def twoSum(nums, target):
```

Example:

Input: `nums = [2,7,11,15], target = 9`

Output: `[0,1]`

Explanation: Because `nums[0] + nums[1] == 9`, we return `[0,1]`.

https://leetcode.com/problems/two-sum/description/

Find how to download **miniconda** here

Create your environment using the following command:

```
conda create -n "myenv" python=3.12.0
```

And activate the corresponding environment:

```
conda activate myenv
```

For the coming weeks…

Bring your own laptop with you!

We will also start with homework next week. Stay tuned!

Thanks for your Attention

Next Week: Human Perception and Computer Vision