



# Python和人工智能基础课程（第一课）

张威



# 课程介绍

- 计算机知识科普
- Python编程基础知识
- Python编程实例
- Python实现人工智能
- 终极目标：构建自己的编程知识体系，可以应对新的问题
- 个人主页：<https://wei2624.github.io/about>



# 课程要求

- 英语：会读，最好会听
  - 较为通顺的阅读编程相关的文章和博客
  - 一篇文章生词和语法导致的检索数量<10次
- 勤奋：
  - 完成练习
  - 主动尝试
  - 主动建立自己的编程知识体系
- 工作量：
  - 上课
  - 作业，开始简单，后面难
  - 有什么问题，可以在群里讨论



# 今日摘要

- 什么是编程？编程解决了怎么样的问题？为什么学习编程？
- 编程的基本概念
- 编程语言介绍
- Hello World
- 什么是环境？什么是Python环境？怎么设置环境？
- 安装Python环境，介绍并安装Intergated Development Environment (IDE)
- 开始Python编程
- 学习资源

# 什么是编程？ 电脑是什么？



听说

思考决策

短期记忆

长期记忆

语言交流



输入输出

计算 (CPU, GPU)

内存 (和编程**强相关**)

硬盘 (和编程**弱相关**)

程序



# 编程解决了什么样的问题？

- 初级问题：
  - 计算--每秒10亿次量级的计算
- 次初级问题：
  - 存储，短时期存储，长时间存储
- 进阶问题：
  - 控制电脑处理一系列任务
    - 计算并存储，之后读取.....
- 高级问题：
  - 处理现实任务，实现自动化
    - 网络爬虫
    - 黑客
    - 数据挖掘
- 终极问题：
  - 人工智能的应用和普及



# 编程的基本思路和概念

- 找到对应一个问题的解决方案
- 解决方案是什么：
  - 一系列简单步骤
  - 流程控制：告诉电脑**什么时候**执行**哪个步骤**
  - 决定什么时候结束整个流程
- 我们称上述的过程为**算法 (Algorithm)**
- 当我们面对一个新问题的时候，我们应该思考：
  - 这个问题**有没有**可自动化（一定程度上可重复）的流程
  - 这种流程**能不能**被程序解决
  - 这样的程序需要我的**硬件**（比如内存）有多大
  - 如果都满足，我的开发**成本**（时间成本）是多少
- 如果都可以，那么就可以开始群求解决方案了



# 编程语言介绍

- 编程语言的种类可以从三个角度来介绍：
  - 不同的**功能**对应不同的编程语言
  - 不同的编写**难度**对应不同的编程语言
  - Object Oriented Language (OOP)



# 功能分类

- 前端：网页，APP界面，
  - 功能：显示, 网页功能
  - 举例：HTML, CSS, JavaScript
- 后端：服务器，数据库
  - 功能：后台计算和存储
  - 举例：Python, PHP, ruby, SQL
- 数学和统计：
  - 功能：数学用途
  - 举例：MATLAB, R
- 命令行脚本：
  - 功能：执行多组命令行
  - 举例：Shell Script (Unix)
- 通用语言：
  - C, C++, C#, Java, Objective-C, Python



# 难度分类

- 什么是“Python比较好用”？
  - 越接近人类语言顺序的，就是越好的用。（英语）
- 容易编写程序的：
  - Python
  - MATLAB
  - R
- 有一些难度的：
  - C#
  - Java
  - JavaScript
- 非常难的：
  - C, C++
- 越不接近人类语言的，越接近计算机语言的，越难编写



# OOP

- Object-Oriented Programming: is a programming language based on the concept of 'object'. It has functions and attributes.
- 举例：汽车，前进后退（功能），价值（属性）
- OOP languages:
  - Python, C#, C++, Java
- Not OOP languages:
  - C
  - FORTRAN
  - CLEAN



# Hello World

- 实例：打印（print）“Hello World” 在屏幕上

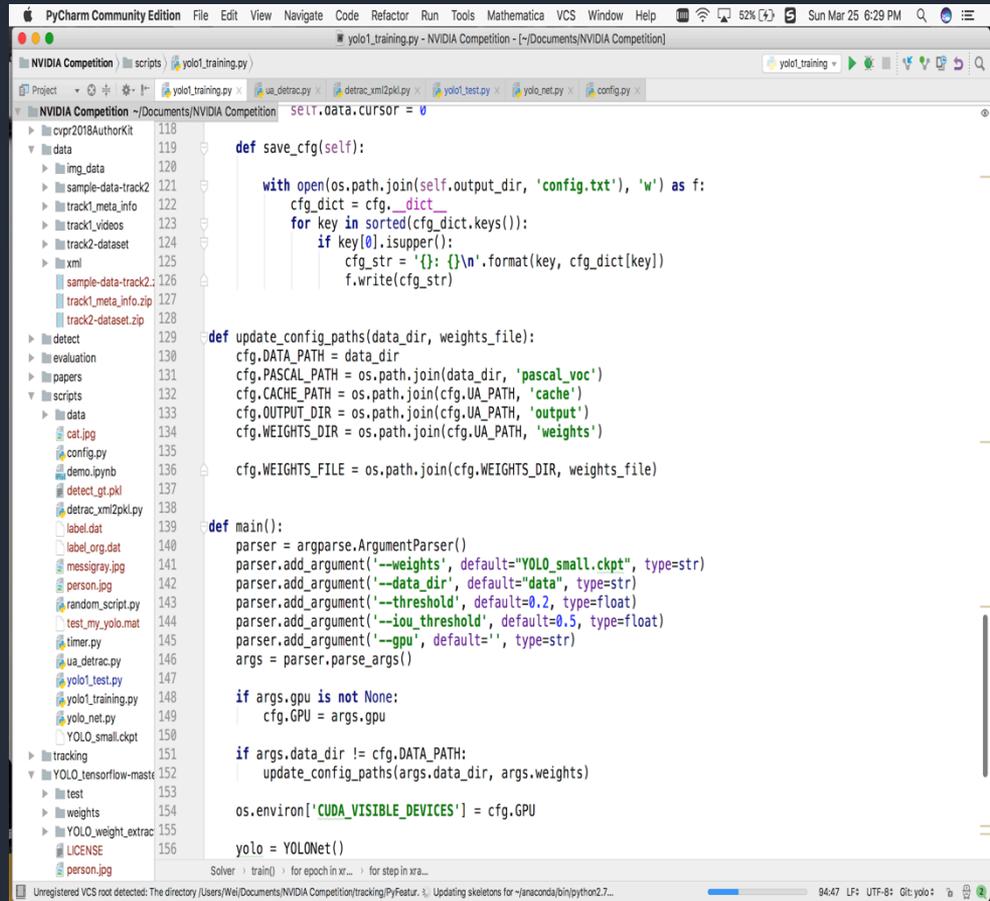


# Python 环境

- 跑一段程序，电脑到底发生了什么？
  - 加载代码
  - 寻找依赖包 (package)
  - 编译器
  - 把代码转换成010101....., 这个过程：编译
  - 让电脑去执行命令
  - 完成
- Python环境：
  - 搜集每一个组成部分-->构建环境
  - 下载依赖包：
    - 随着项目改变而改变
  - Anaconda:
    - 通过创建虚拟环境来管理依赖包
    - 针对Python

# Python 环境和集成开发环境

- 安装Python环境
  - 测试
- 集成开发环境 (IDE)
  - 帮助快速的开发
    - 没有IDE, 我们会怎么样?
      - 实例
  - 帮助更好的检测bug
  - 帮助更好的管理项目
- 常用Python IDE:
  - Sublime Text
  - PyCharm (有免费版本)



```
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

def save_cfg(self):
    with open(os.path.join(self.output_dir, 'config.txt'), 'w') as f:
        cfg_dict = cfg._dict_
        for key in sorted(cfg_dict.keys()):
            if key[0].isupper():
                cfg_str = '{}: {}'.format(key, cfg_dict[key])
                f.write(cfg_str)

def update_config_paths(data_dir, weights_file):
    cfg.DATA_PATH = data_dir
    cfg.PASCAL_PATH = os.path.join(data_dir, 'pascal_voc')
    cfg.CACHE_PATH = os.path.join(cfg.UA_PATH, 'cache')
    cfg.OUTPUT_DIR = os.path.join(cfg.UA_PATH, 'output')
    cfg.WEIGHTS_DIR = os.path.join(cfg.UA_PATH, 'weights')

    cfg.WEIGHTS_FILE = os.path.join(cfg.WEIGHTS_DIR, weights_file)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', default="YOLO_small.ckpt", type=str)
    parser.add_argument('--data_dir', default="data", type=str)
    parser.add_argument('--threshold', default=0.2, type=float)
    parser.add_argument('--iou_threshold', default=0.5, type=float)
    parser.add_argument('--gpu', default='', type=str)
    args = parser.parse_args()

    if args.gpu is not None:
        cfg.GPU = args.gpu

    if args.data_dir != cfg.DATA_PATH:
        update_config_paths(args.data_dir, args.weights)

    os.environ['CUDA_VISIBLE_DEVICES'] = cfg.GPU

    yolo = YOLONet()
```



# 开始编程

- 和练习相关
- 目标：获取user input, 然后print在后台终端
- 实例



# 学习资源

- Google is ALWAYS better!
- VPN:
  - Express VPN
- Github
  - 程序员社区
  - 有很多现成的代码
  - Python的好处
- 把自己的问题放到社区里，让专业的人回答
  - Stack Overflow