

# 大学计算机与人工智能基础：实验与习题

## 第 7 章 人工智能实验一

### 一. 实验目的

1. 了解人工智能和 python 语言的关系
2. 理解 python 语言的基本结构和作用
3. 理解人工智能在智能机器人中的应用

### 二. 实验内容

人工智能涉及大量的数据计算，而 Python 编程语言简单易用，拥有数量众多并功能强大的工具库，对数据处理有着得天独厚的优势，所以已成为人工智能领域中广泛使用的编程语言之一。

本次实验有如下两个内容：

1. 安装 Python 解释器并熟悉 Python 集成开发环境
2. 用 Python 语言完成 4 个简单聊天机器人的对话功能

### 三. 实验步骤

1. 安装 Python 解释器

Python 解释器可在 Python 官网下载页面（图 7-1）进行下载，网址如下：

<https://www.python.org/downloads/>

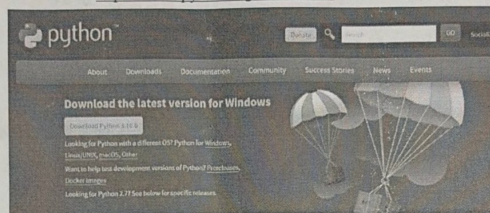


图 7-1 Python 主页面

根据操作系统版本选择相应的 Python 系列（图 7-2）进行安装，建议选择 3.5 以上版本。

Looking for a specific release?  
Python releases by version number.

Release version	Release date		Click for more
Python 3.10.6	Aug 2, 2022	Download	Release Notes
Python 3.9.15	June 6, 2022	Download	Release Notes
Python 3.9.13	May 17, 2022	Download	Release Notes
Python 3.10.4	March 24, 2022	Download	Release Notes
Python 3.9.12	March 23, 2022	Download	Release Notes
Python 3.10.3	March 16, 2022	Download	Release Notes
Python 3.9.11	March 16, 2022	Download	Release Notes

View older releases

图 7-2 Python 不同版本

安装成功后打开 Python 集成开发环境 IDLE，运行 Python 程序有两种方式：交互式（图 7-3）和文件式（图 7-4）。交互式用于少量代码的编写与调试，文件式更为常用。



在交互式界面的菜单栏点击“File”即可打开文件式编程环境，在该界面中写好程序后，点击菜单栏的“Run”——“Run Module”，会提示是否要将该程序保存，保存后方可运行并显示程序结果。程序关闭后想再打开，可以选中该文件，单击右键选择“Edit with IDLE”进行打开。

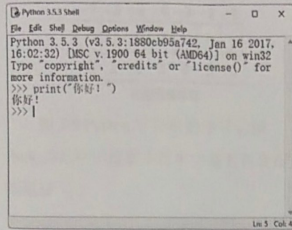


图 7-3 交互式 Python 运行环境

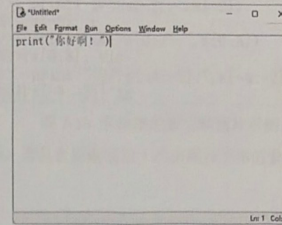


图 7-4 文件式 Python 运行环境

## 2. 简单聊天机器人 1

打开 IDLE，新建文件，完成如图 7-5 的程序输入并执行，运行结果如图 7-6 所示。

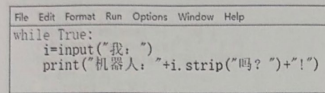


图 7-5 聊天机器人 1 程序参考

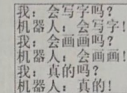


图 7-6 聊天机器人 1 运行结果参考

while 为条件控制循环，通过条件表达式控制循环结束。

strip()方法用于移除字符串头尾指定的字符或字符序列，该方法只能删除开头或结尾的字符，不能删除中间部分的字符。

## 3. 简单聊天机器人 2

打开 IDLE，新建文件，完成如图 7-7 的程序输入并执行，运行结果如图 7-8 所示。

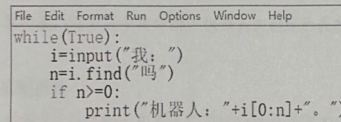


图 7-7 聊天机器人 2 程序参考

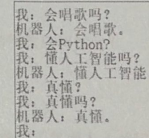


图 7-8 聊天机器人 2 运行结果参考

if 语句是通过一条或多条语句的执行结果（True 或者 False）来决定执行的代码块。

索引和切片是字符串的两个重要操作。

索引：返回字符串中单个字符，格式为：<字符串>[M]

切片：返回字符串中一段字符串，格式为：<字符串>[M: N]（不包含 N）

字符串的序号有正向和逆向两种方式（如图 7-9），其索引和切片的示例如图 7-10 所示。



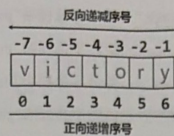


图 7-9 Python 字符串的序号示例

```
>>> s="victory"
>>> print(s)
victory
>>> print("索引s[0]:",s[0])
索引s[0]: v
>>> print("索引s[-2]:",s[-2])
索引s[-2]: r
>>> print("切片s[0:3]:",s[0:3])
切片s[0:3]: vic
>>> print("切片s[-4:-2]:",s[-4:-2])
切片s[-4:-2]: to
```

图 7-10 字符串的索引和切片示例

find()方法用于检索字符串中是否包含指定子串,若包含则返回第1次出现该子串的索引值,反之,则返回 -1。

#### 4. 简单聊天机器人 3

打开 IDLE, 新建文件, 完成如图 7-11 的程序输入并执行, 运行结果如图 7-12 所示。

```
File Edit Format Run Options Window Help
t=0
print("机器人: 你好呀!")
while(True):
    i=input("我: ")
    robot=i.replace("吗","呢")
    robot=robot.replace("?", "! ")
    robot=robot.replace("你","我")
    if i=="再见! ":
        print("再见!")
        break
    elif robot==i:
        print("真心不懂求原谅, 换个问题试试? ")
        t=t+1
    else:
        print("机器人: "+robot)
```

图 7-11 聊天机器人 3 程序参考

```
机器人: 你好呀!
我: 你会打字吗?
机器人: 我会打字呢!
我: 你会弹吉他吗?
机器人: 我会弹吉他呢!
我: 厉害啊!
真心不懂求原谅, 换个问题试试?
我: 会唱歌?
机器人: 会唱歌!
我: 行啊!
真心不懂求原谅, 换个问题试试?
我: 再见!
再见!
```

图 7-12 聊天机器人 3 运行结果参考

Replace()方法用于将字符串中的旧字符串替换成新字符串。

if-elif-else 用于判断多个条件的语句, if 为真则执行 if 后面的语句。

break 用来终止循环语句, 即循环条件没有 False 条件也会停止执行循环语句。

思考题: 以上三个简单聊天机器人的程序存在哪些问题, 如何改进和优化?

#### 四. 实验报告

参考以上三个例子和本程序的运行结果(图 7-13), 完成简单聊天机器人 4 的程序填空。

```
name=""
i=0
while ①____:
    i= ②____
    print("循环第, i,"次")
    name = input("请输入名字: ")
    if name == ' ③____':
        break
```

```

age = input("请输入年龄: ")
height = input("请输入身高: ")
print("你的名字小写: ",name.lower())
print("你的名字大写: ",name.upper())
print("你的名字长度: ",len(④))
print("你的名字的后三位字母: ", name[-3:])
print("a 在你的名字中出现次数: ", name.count("a"))
print("{}同学: 年龄为{}岁, 身高为{}米。".format(name, ⑤))
print("再过 15 年, {}岁。".format(age+15))

print("循环结束")

```

lower()方法将字符串中的大写字母转换为小写字母。

upper()方法将字符串中的小写字母转换为大写字母。

len()函数用于返回对象(字符、列表、元组等)长度或项目个数。

count()方法用于统计字符串里某个字符或子字符串出现的次数。

format()函数通过字符串中的花括号{}来识别替换字段, 完成字符串的格式化。

```

循环第 1 次
请输入名字: wangfan
请输入年龄: 16
请输入身高: 1.65
你的名字小写: wangfan
你的名字大写: WANGFAN
你的名字长度: 7
你的名字的后三位字母: fan
a 在你的名字中出现次数: 2
wangfan同学: 年龄为16岁, 身高为1.65米。
再过15年, 31岁。
循环第 2 次
请输入名字: hexing
请输入年龄: 17
请输入身高: 1.8
你的名字小写: hexing
你的名字大写: HEXING
你的名字长度: 6
你的名字的后三位字母: ing
a 在你的名字中出现次数: 0
hexing同学: 年龄为17岁, 身高为1.8米。
再过15年, 32岁。
循环第 3 次
请输入名字: end
循环结束

```

图 7-13 聊天机器人 4 运行结果参考

