

人工智能程序设计

python



```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```



# 人工智能程序设计

## 15.3 语音合成

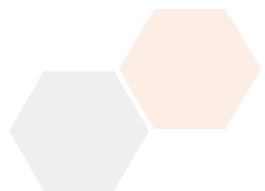
北京石油化工学院 人工智能研究院

刘 强

---

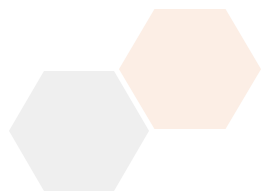
# 学习内容

- 语音合成系统架构
- Python语音合成实现
- 高级语音合成库



# 语音合成概述

语音合成技术将文本转换为自然流畅的语音，是构建智能语音交互系统的重要组成部分。  
掌握语音合成的基本原理和Python实现方法，能够为应用添加语音输出功能，提升用户体验。



# 15.3.1 文本转语音基础

## 学习内容:

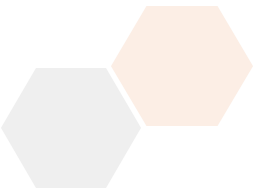
- 传统与端到端系统架构
- 语音合成处理流程



# 传统语音合成系统架构

传统语音合成系统包括四个主要模块：

模块	功能
文本分析	分词、词性标注、韵律预测
语言学处理	音素转换、重音标注
声学参数预测	预测基频、时长、频谱参数
语音波形生成	生成最终语音波形

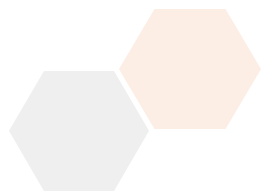


# 端到端语音合成系统

**端到端语音合成系统**使用深度神经网络直接建立文本到语音的映射关系。

**代表模型：**

- **Tacotron：** 序列到序列模型，生成梅尔频谱
- **WaveNet：** 自回归模型，生成高质量波形
- **VITS：** 端到端变分推理模型



## 15.3.2 Python语音合成实现

### 学习内容:

- pyttsx3离线合成
- gTTS在线合成
- 语音参数调整





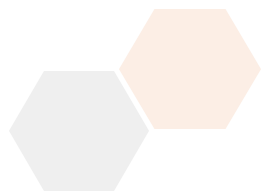
# pyttsx3库简介

**pyttsx3**是跨平台的离线语音合成库，无需网络连接。

**安装方式：**

**支持平台：** Windows、macOS、Linux

```
pip install pyttsx3
```



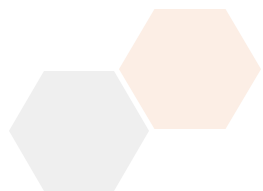
# 基础语音合成功能

使用pyttsx3库实现文本转语音的基本功能。

```
import pyttsx3

# 初始化语音引擎
engine = pyttsx3.init()

# 合成语音
text = "欢迎使用Python语音合成技术"
engine.say(text)
engine.runAndWait()
```



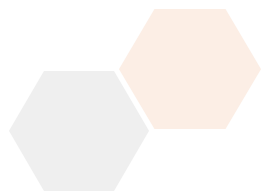
# 语音参数调整

调整语音合成的语速、音量和声音类型。

```
# 初始化引擎
engine = pyttsx3.init()

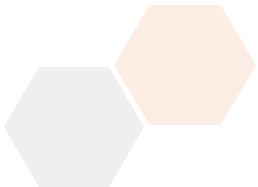
# 获取可用语音
voices = engine.getProperty('voices')

# 设置语音参数
engine.setProperty('rate', 150)      # 语速
engine.setProperty('volume', 0.8)    # 音量
engine.setProperty('voice', voices[0].id) # 语音类型
```



# 常用参数说明

参数	说明	取值范围
rate	语速（每分钟词数）	100-300
volume	音量	0.0-1.0
voice	语音类型	系统已安装的语音



# 文件输出功能

将合成的语音保存为音频文件。

```
# 初始化引擎
engine = pyttsx3.init()

# 保存语音到文件
text = "这是一段测试语音"
filename = "output.wav"
engine.save_to_file(text, filename)
engine.runAndWait()

print("语音已保存到 {}".format(filename))
```

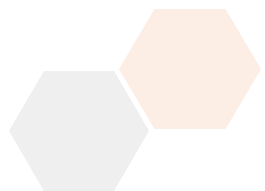


# gTTS在线语音合成

**gTTS** (Google Text-to-Speech) 使用Google服务进行高质量语音合成。

**安装方式:**

```
pip install gTTS  
pip install pygame # 播放支持
```



# gTTS使用示例

使用Google Text-to-Speech服务进行高质量语音合成。

```
from gtts import gTTS
import io
import pygame

# 创建gTTS对象
text = "欢迎使用Google语音合成"
tts = gTTS(text=text, lang='zh', slow=False)

# 保存到文件
tts.save("output.mp3")
```



# gTTS播放语音

将合成的语音保存到内存并播放。

```
# 保存到内存
fp = io.BytesIO()
tts.write_to_fp(fp)
fp.seek(0)

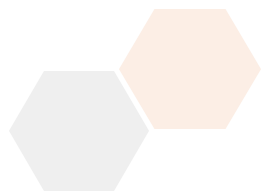
# 播放语音
pygame.mixer.init()
pygame.mixer.music.load(fp)
pygame.mixer.music.play()
```





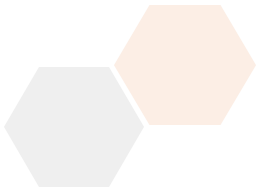
# 高级语音合成服务

服务	特点
Azure Cognitive Services	企业级服务，支持SSML标记语言
百度语音合成	中文优化，多种中文语音选择
阿里云语音合成	高质量中文语音，多种音色



# 离线与在线合成对比

特性	pyttsx3（离线）	gTTS（在线）
网络依赖	无需网络	需要网络
语音质量	一般	较高
响应速度	快	受网络影响
语言支持	系统语音	多语言支持



# 实践练习

## 练习 15.3.1：基础语音合成

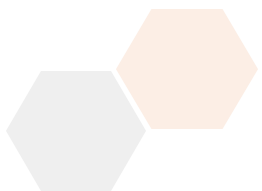
使用pyttsx3库实现基本的文本转语音功能，包括参数调整和文件保存。



# 实践练习

## 练习 15.3.2：多语言合成对比

比较不同语音合成引擎的效果，分析各自的优缺点和适用场景。



# 实践练习

## 练习 15.3.3: 情感语音实现

开发情感语音合成功能，能够根据文本内容或用户指定表达不同情感。



# 实践练习

## 练习 15.3.4: 智能播报系统

构建一个智能播报系统，能够处理新闻、天气、时间等不同类型的文本内容。

