

python

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned below the word "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)
```

Python语言程序设计

深入理解Python语言



嵩 天
北京理工大学





单元开篇

深入理解Python语言



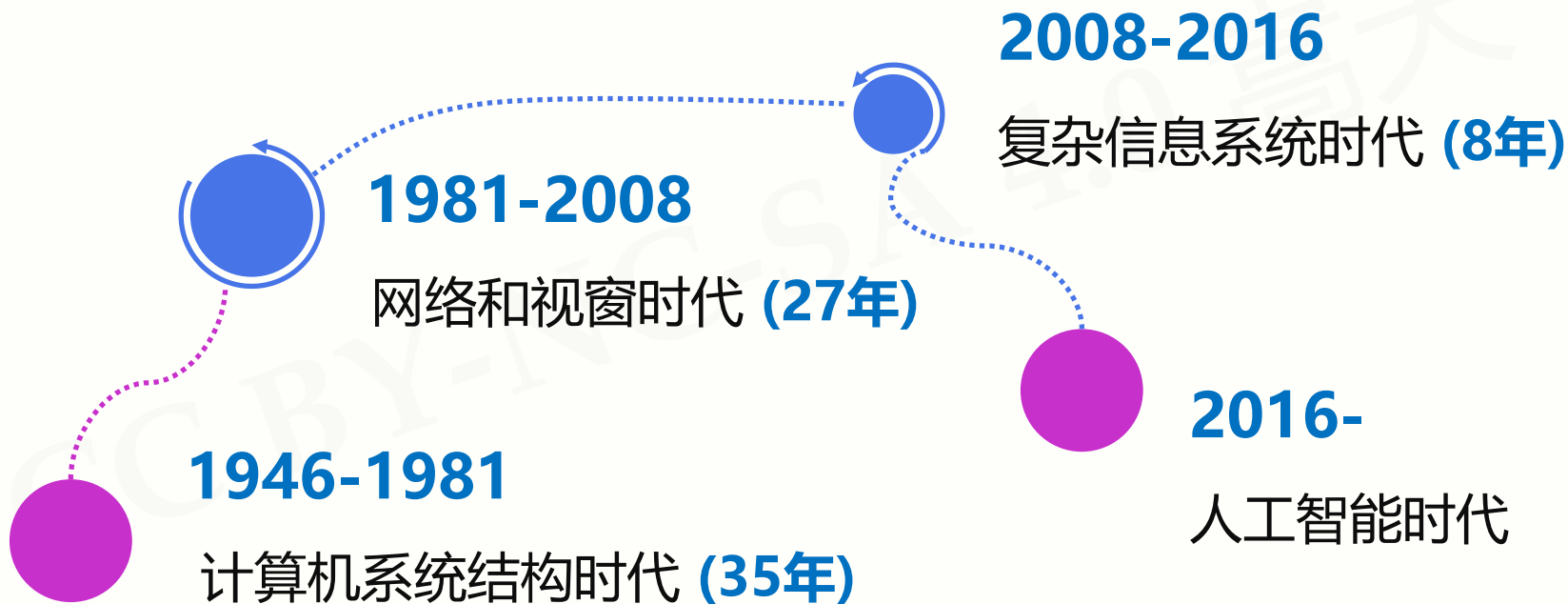
- 计算机技术的演进
- 编程语言的多样初心
- Python语言的特点
- "超级语言"的诞生





计算机技术的演进

计算机技术的演进过程



计算机技术的演进过程

2017-

人工智能时代

人类的问题

新计算时代

2008-2016

复杂信息系统时代

数据问题

1981-2008

网络和视窗时代

交互问题

1946-1981

计算机系统结构时代

计算能力问题



编程语言的多样初心

编程语言有哪些？

**Basic, C, C++, C#, CSS, Fortran, Go, HTML, Java,
JavaScript, Lisp, Lua, Matlab, Object C, Pascal, Perl, PHP,
PostScript, Python, Ruby, Scala, SQL, Swift, VBA,
VB.NET, Verilog, VHDL, Visual Basic**

编程语言，也是一个江湖！

不同编程语言的初心和适用对象

编程语言	学习内容	语言本质	解决问题	适用对象
C	指针、内存、数据类型	理解计算机系统结构	性能	计算机类专业
Java	对象、跨平台、运行时	理解主客体关系	跨平台	软件类专业
C++	对象、多态、继承	理解主客体关系	大规模程序	计算机类专业
VB	对象、按钮、文本框	理解交互逻辑	桌面应用	不确定
Python	编程逻辑、第三方库	理解问题求解	各类问题	所有专业

各编程语言所处历史时期和使命不同，Python是**计算时代演进**的选择！

2018年以后的计算环境...

计算机性能不再是解决一般问题的瓶颈

移动互联网广泛普及

大数据、云计算、物联网、信息安全、人工智能等需求爆发

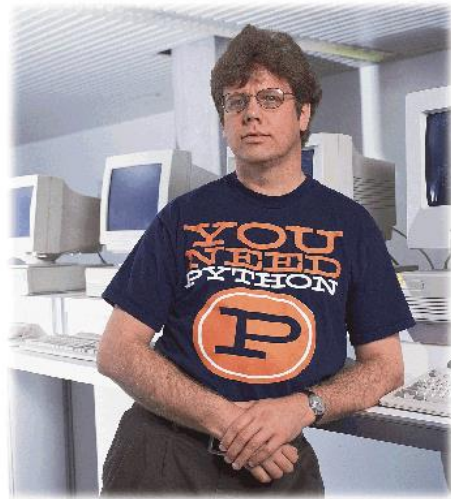
解决日益增长的计算需求，用什么语言？



Python语言的特点



- Python语言是通用语言
- Python语言是脚本语言
- Python语言是开源语言
- Python语言是跨平台语言
- Python语言是多模型语言



Guido van Rossum

Python语言创立者

2002年，Python 2.x

2008年，Python 3.x

Python特点与优势

语法简洁



10x

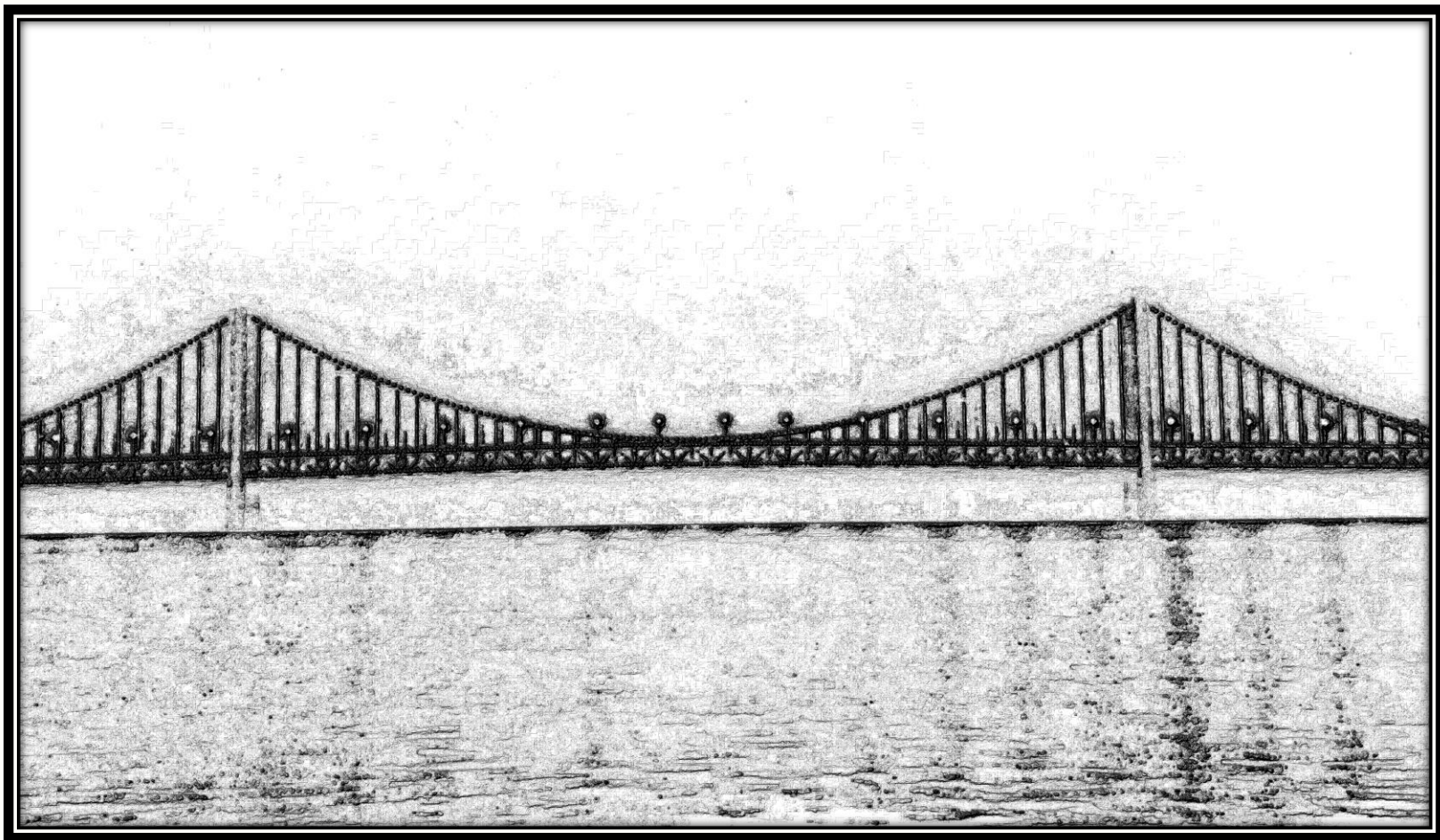
10x



生态高产

- **C代码量的10%**
- **强制可读性**
- **较少的底层语法元素**
- **多种编程方式**
- **支持中文字符**
- **>13万第三方库**
- **快速增长的计算生态**
- **避免重复造轮子**
- **开放共享**
- **跨操作系统平台**





Python 21行代码

如何看待Python语言？

人生苦短，我学Python

- C/C++：Python归Python，C归C
- Java：针对特定开发和岗位需求
- HTML/CSS/JS：不可替代的前端技术，全栈能力
- 其他语言：R/Go/Matlab等，特定领域

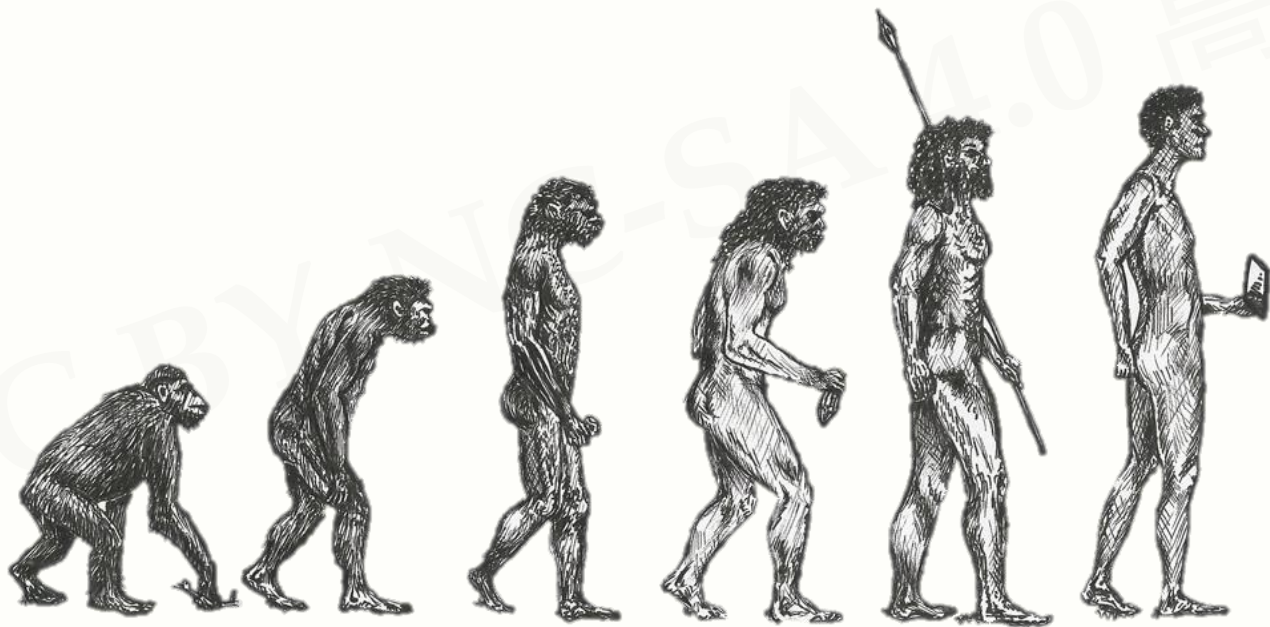
如何看待Python语言？

Python是最高产的程序设计语言及.....

- 掌握**抽象并求解**计算问题综合能力的语言
- 了解产业界解决**复杂计算问题**方法的语言
- 享受利用编程**将创新变为实现**乐趣的语言

如何看待Python语言？

工具决定思维：关注工具变革的力量！





"超级语言"的诞生

编程语言的种类

机器语言

- 一种二进制语言，直接使用二进制代码表达指令
- 计算机硬件(CPU)可以直接执行，与具体CPU型号有关
- 完成 $2+3$ 功能的机器语言

11010010 00111011

编程语言的种类

汇编语言

- 一种将二进制代码直接对应助记符的编程语言
- 汇编语言与CPU型号有关，程序不通用，需要汇编器转换
- 完成 $2+3$ 功能的汇编语言

`add 2,3,result`

编程语言的种类

高级语言

- 更接近自然语言，同时更容易描述计算问题
- 高级语言代码与具体CPU型号无关，编译后运行
- 完成 $2+3$ 功能的高级语言

`result = 2 + 3`

编程语言种类的发展

超级语言

- 粘性整合已有程序，具备庞大计算生态

高级语言

- 接近自然语言，编译器，与CPU型号无关

汇编语言

- 有助记符，汇编器，与CPU型号有关

机器语言

- 代码直接执行，与CPU型号有关

编程语言的种类


超级语言

- 具有庞大计算生态，可以很容易利用已有代码功能
- 编程思维不再是刀耕火种，而是集成开发
- 完成 $2+3$ 功能的超级语言

```
result = sum(2,3)
```

Python: 唯一的"超级语言"！

Python前进的步伐不可阻挡



单元小结

深入理解Python语言

- 计算机系统结构时代到人工智能时代的演进路线
- 五种编程语言的初心和历史使命
- Python语言的通用性、简洁性和生态性
- Python是以计算生态为标志的“超级语言”



