



编译原理与设计：编程语言

计卫星

jwx@bit.edu.cn

北京理工大学 计算机学院

<https://github.com/jiweixing/build-a-compiler-within-30-days>





内容

- 语言发展历史
- 语言分类
- 语言排名
- 语言实现
- Lab. 1





语言发展历史

- 程序设计语言的演化
 - 机器语言, 1940s

[op		rs		rt		rd		shamt		funct]	
	0		1		2		6		0		32	decimal
	000000		00001		00010		00110		00000		100000	binary

[op		rs		rt		address/immediate]	
	35		3		8		68	decimal
	100011		00011		01000		00000 00001 000100	binary

程序设计语言是用于计算方法描述中表示和标识方法





语言发展历史

- 程序设计语言的演化
 - 汇编语言, 1950s 早期

```
LDF  R2, id3  
MULF R2, R2, #60.0  
LDF  R1, id2  
ADDF R1, R1, R2  
STF  id1, R1
```





语言发展历史

- 程序设计语言的演化
 - 高级程序设计语言, 1950s后期
 - FORTRAN 科学计算
 - SQL 商业数据处理

```
write (*,*) "Enter the points to average:"
read (*,*) points

! Take the average by summing points and dividing by number_of_points
if (number_of_points > 0) average_points = sum(points) / number_of_points

! Now form average over positive and negative points only
if (count(points > 0.) > 0) then
    positive_average = sum(points, points > 0.) / count(points > 0.)
end if

if (count(points < 0.) > 0) then
    negative_average = sum(points, points < 0.) / count(points < 0.)
end if
```

```
SELECT Book.title AS Title,
       COUNT(*) AS Authors
FROM   Book
JOIN   Book_author
ON     Book.isbn = Book_author.isbn
GROUP BY Book.title;

(defun factorial (n)
  (if (= n 0) 1
      (* n (factorial (- n 1)))))
```





语言发展历史

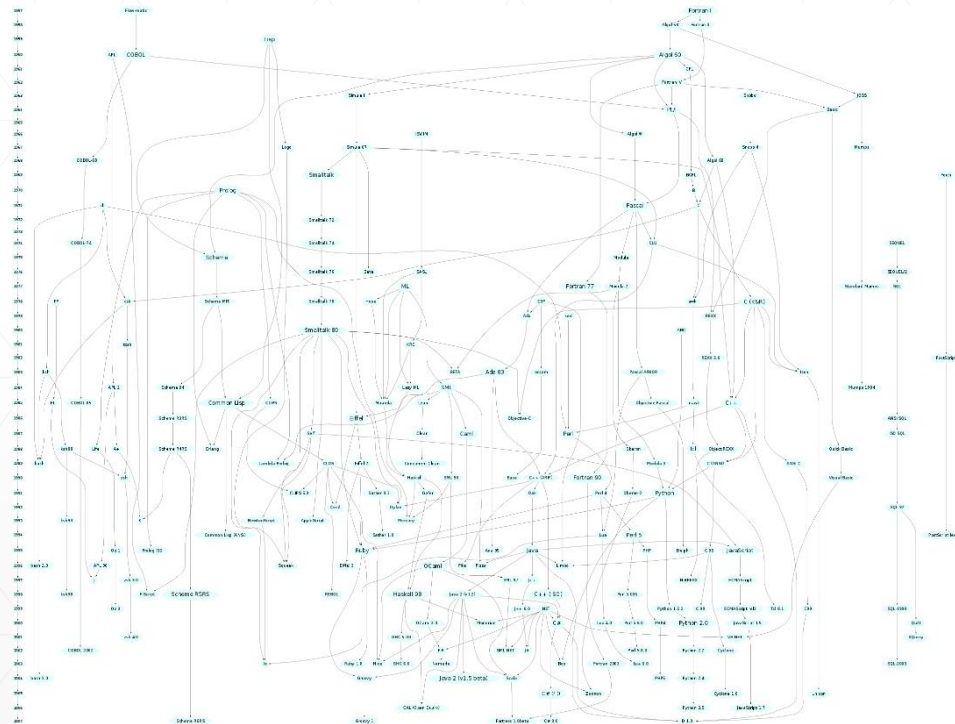
- 程序设计语言的演化
 - First-generation languages
 - Machine languages
 - Second-generation languages
 - Assembly languages
 - Third-generation languages
 - Higher-level languages: Fortran, Cobol, Lisp, C, C++, C# and Java
 - Forth-generation languages
 - Designed for specific application: NOMAD, SQL
 - Fifth-generation languages
 - Prolog and OPS5





语言发展历史

■ Family Tree



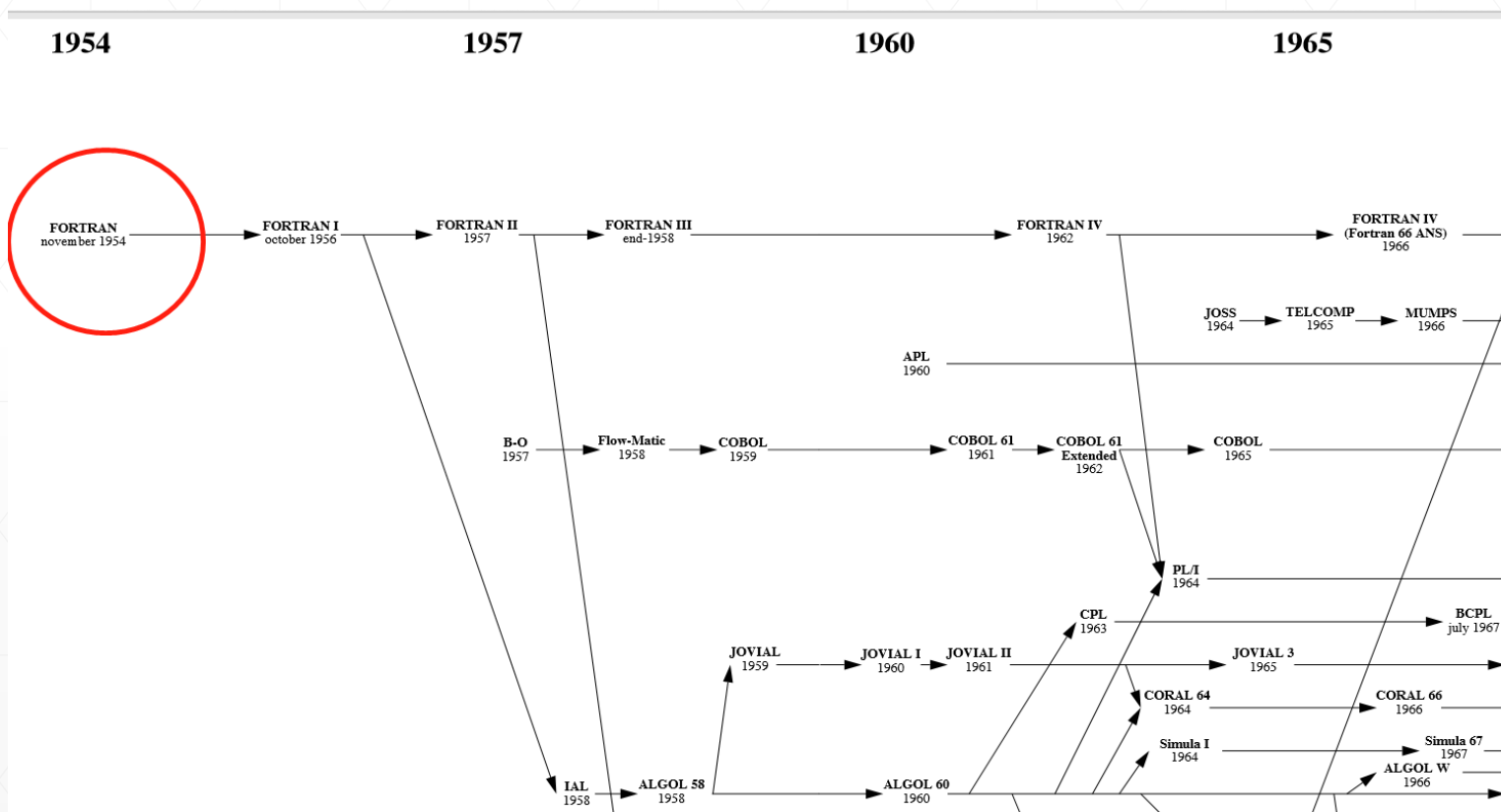
<https://github.com/stereobooster/programming-languages-genealogical-tree>





语言发展历史

■ 第一个程序设计语言



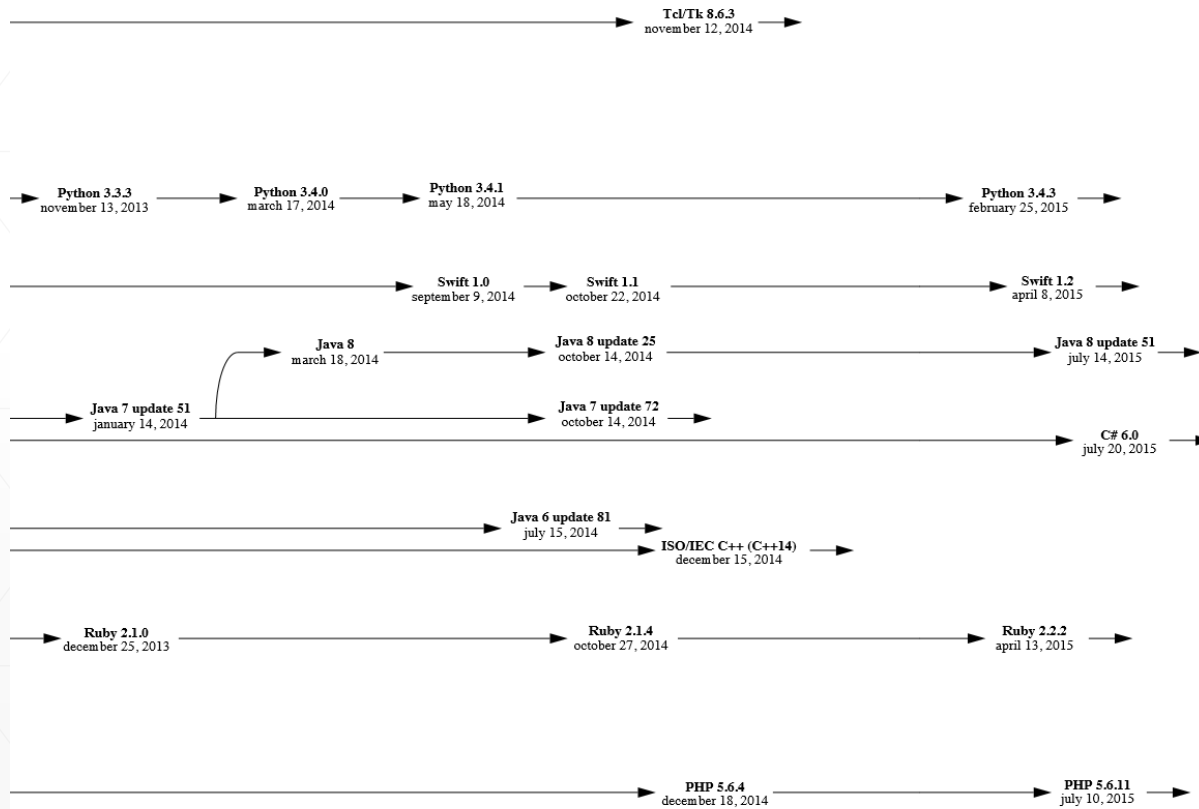
<https://github.com/stereobooster/programming-languages-genealogical-tree>





语言发展历史

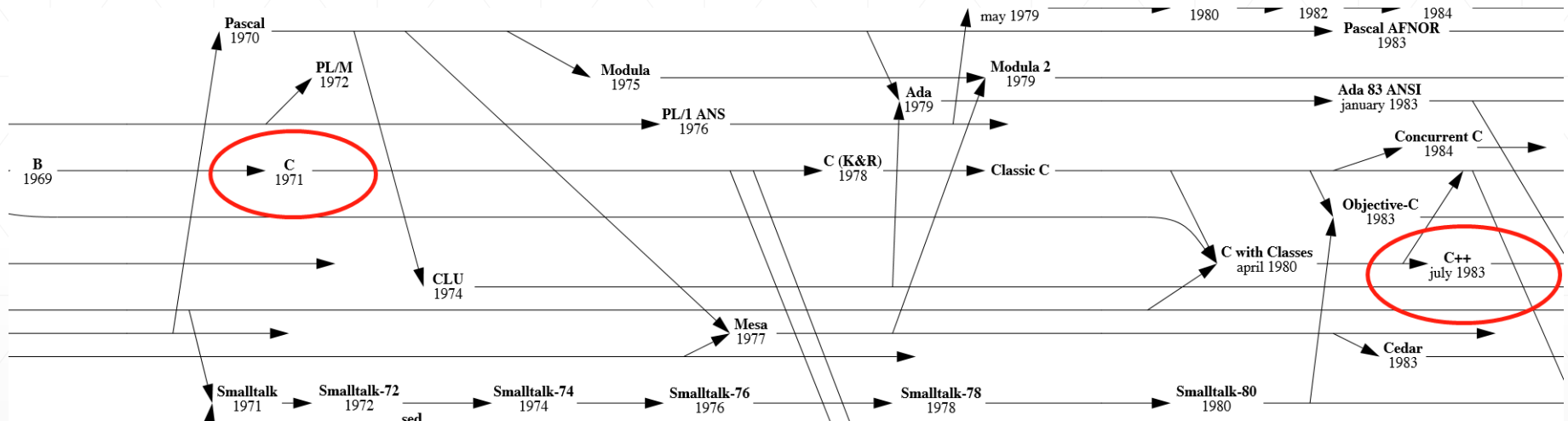
■ 新语言





语言发展历史

■ C/C++



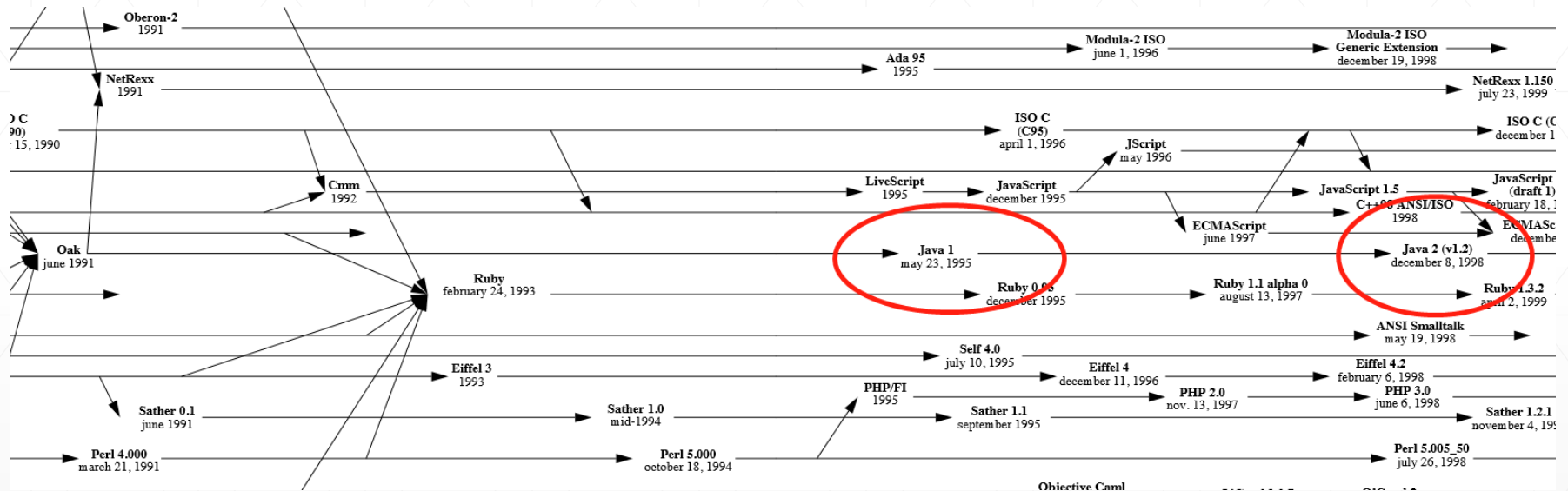
<https://github.com/stereobooster/programming-languages-genealogical-tree>





语言发展历史

■ Java



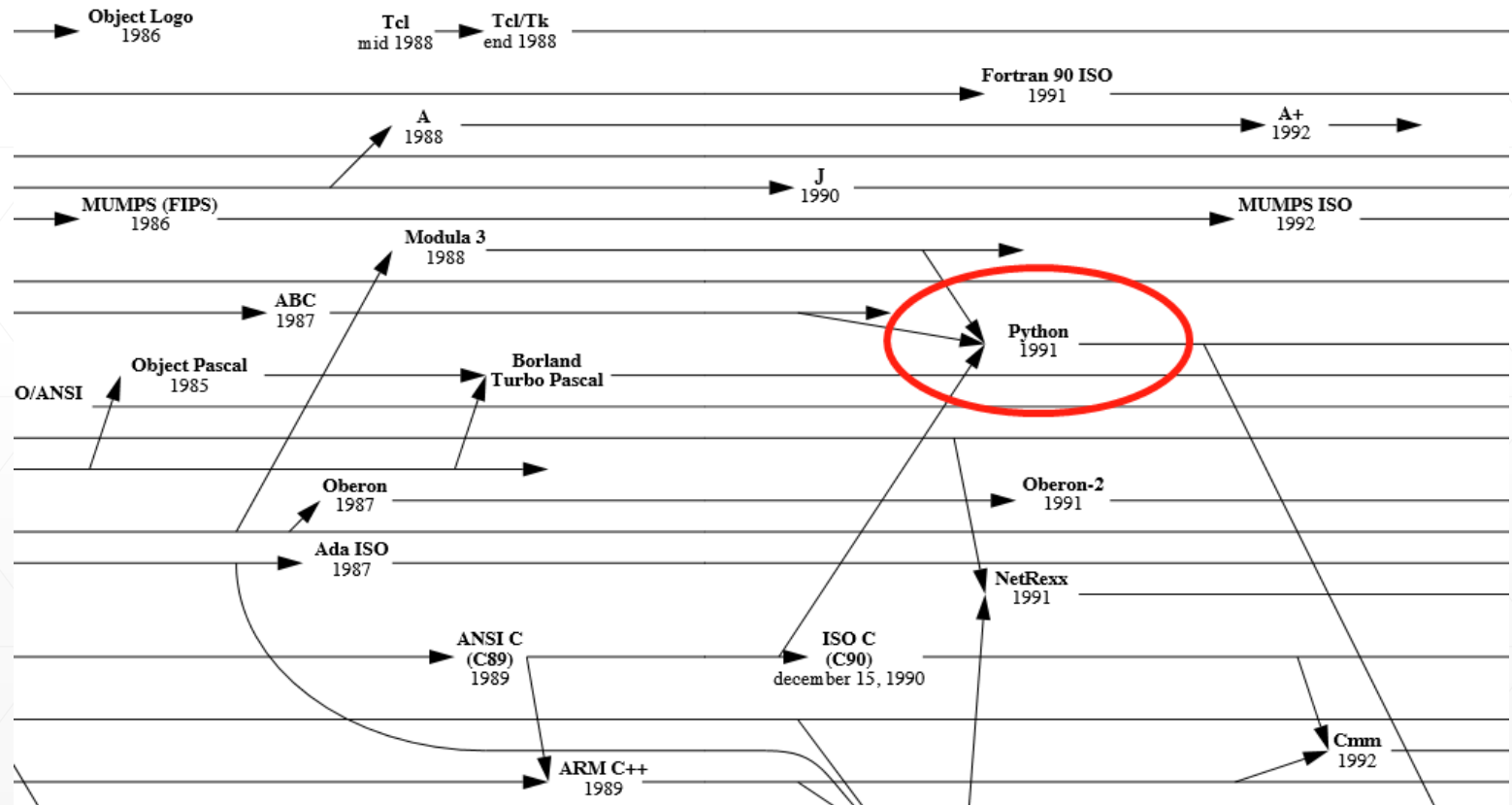
<https://github.com/stereobooster/programming-languages-genealogical-tree>





语言发展历史

■ Python





语言分类

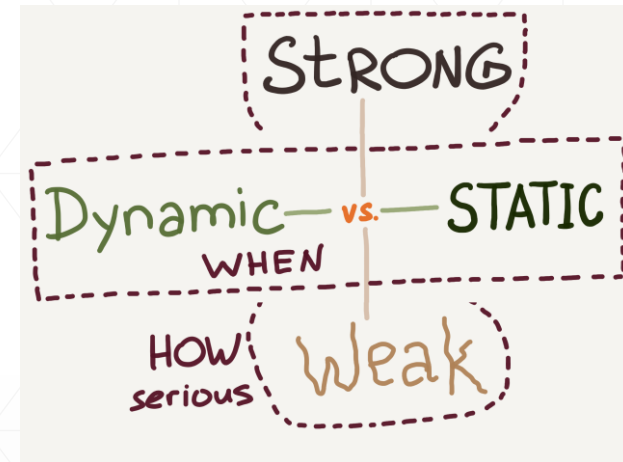
- 分类标准:描述方式
 - **命令式：编程告诉计算机如何完成工作**
 - 面向过程语言
 - 面向对象语言
 - **声明式：编程告诉计算机要完成哪些工作**
 - 函数式语言
 - 逻辑编程语言





语言分类

- 分类标准: Language by typing
 - 静态类型
 - 运行前类型检查: C, C++, and Java
 - 动态类型
 - 运行时检查类型: Python, Ruby等
 - 强类型
 - 不允许隐式数据类型转换
 - 弱类型
 - 允许隐式数据类型转换: JavaScript

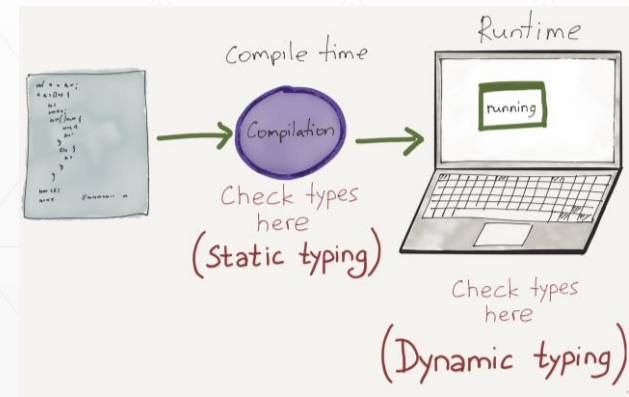


```
4 + '7';      // '47'
4 * '7';      // 28
2 + true;     // 3
false - 3;    // -3
```



语言分类

- 分类标准：实现方式
 - 编译型
 - 解释型
- 分类标准：按照应用领域
 - 科学计算(FORTRAN)
 - 商业应用(SQL)
 - 系统编程(C/C++)





语言排名

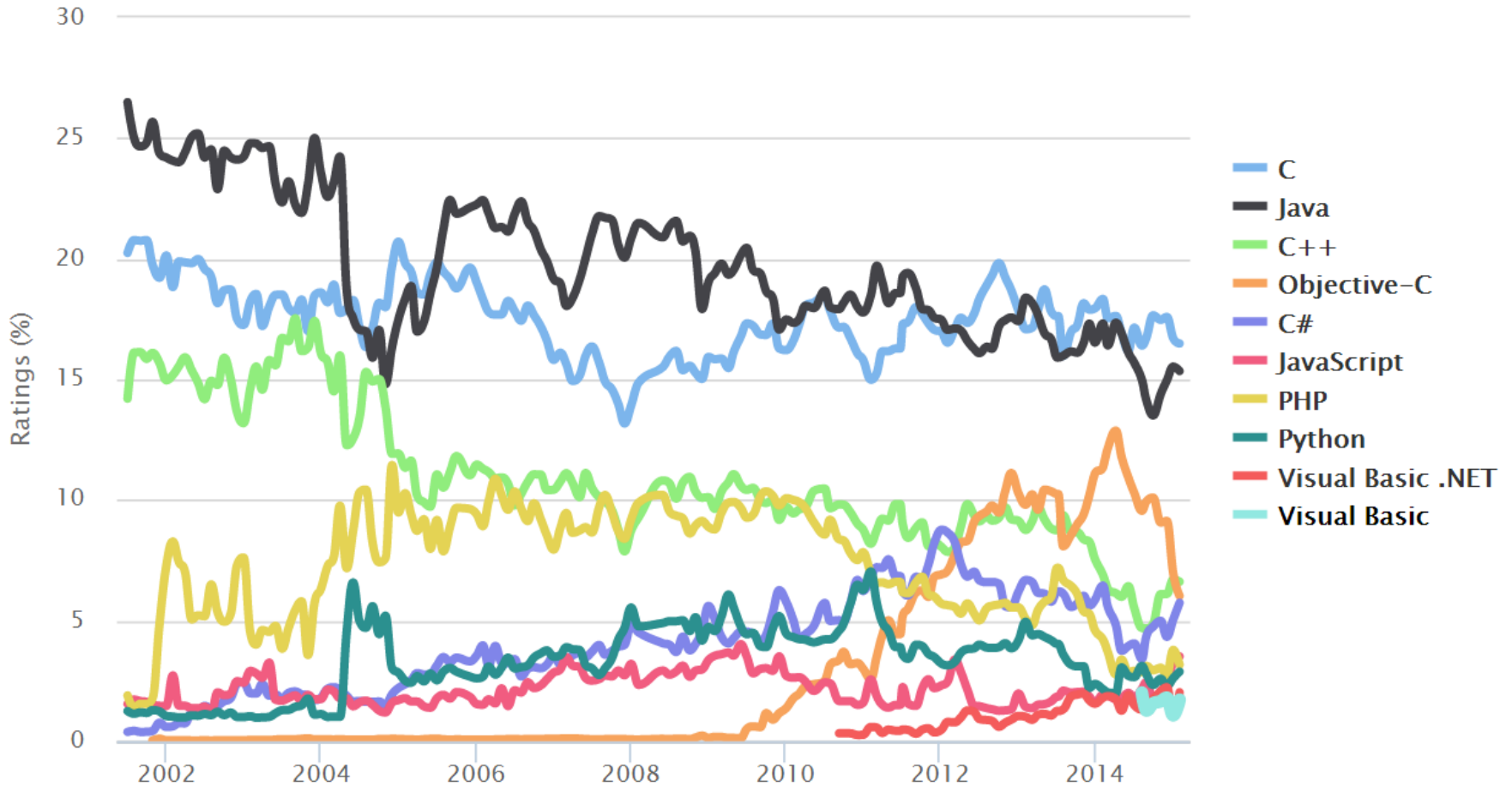
Feb 2015	Feb 2014	Change	Programming Language	Ratings	Change
1	1		C	16.488%	-1.85%
2	2		Java	15.345%	-1.97%
3	4	▲	C++	6.612%	-0.28%
4	3	▼	Objective-C	6.024%	-5.32%
5	5		C#	5.738%	-0.71%
6	9	▲	JavaScript	3.514%	+1.58%
7	6	▼	PHP	3.170%	-1.05%
8	8		Python	2.882%	+0.72%
9	10	▲	Visual Basic .NET	2.026%	+0.23%
10	-	▲▲	Visual Basic	1.718%	+1.72%
11	20	▲▲	Delphi/Object Pascal	1.574%	+1.05%
12	13	▲	Perl	1.390%	+0.50%
13	15	▲	PL/SQL	1.263%	+0.66%
14	16	▲	F#	1.179%	+0.59%
15	11	▼▼	Transact-SQL	1.124%	-0.54%
16	30	▲▲	ABAP	1.048%	+0.69%
17	14	▼	MATLAB	1.033%	+0.39%

<http://www.tiobe.com/>





语言排名



Source: <http://www.tiobe.com/>





语言排名

Programming Language	2015	2010	2005	2000	1995	1990	1985
C	1	2	1	1	2	1	1
Java	2	1	2	3	-	-	-
Objective-C	3	22	39	-	-	-	-
C++	4	4	3	2	1	2	12
C#	5	5	8	8	-	-	-
PHP	6	3	4	28	-	-	-
Python	7	6	6	24	22	-	-
JavaScript	8	8	9	6	-	-	-
Perl	9	7	5	4	9	13	-
Visual Basic .NET	10	-	-	-	-	-	-
Pascal	16	13	67	12	3	15	5
Lisp	18	16	13	15	5	3	2
Ada	30	25	15	16	6	4	3

Source: <http://www.tiobe.com/>





语言排名

- 不断有新的语言涌现
 - 为什么会有这么多种语言?
 - 什么是好的语言?



语言排名

- 为什么会有这么多语言出现?
 - 语言演化
 - 不断发现有更好的方式做事情
 - 特殊目的
 - 为了特定的应用各领域和问题设计新的语言
 - 大数据、人工智能
 - 个人喜好



语言排名

- 什么样的语言会更成功？
 - 表达方式强大：在某一方面抽象程度高
 - 新手容易学习
 - 容易实现
 - 小机器上也可以实现、容易移植
 - 开源且社区活跃
 - 强大的编译器支持
 - ...



语言实现

- Python
 - Cpython、Jython、IronPython
- Ruby
 - CRuby、JRuby





语言实现

- 用数组实现Stack
 - 数组存储压入的元素
 - 使用一个整数标识当前的栈顶
- 用链表实现Stack
 - 使用指针和动态分配
- 其它方式





语言实现

■ 基于数组的实现

```
char Store[MAX];
int top = 0;
void push(char x)
{
    if (top < MAX)
        Store[top++] = x;
    else
        printf("full\n");
}
```

```
char pop()
{
    if (top > 0)
        return Store[--top];
    else
        printf("empty\n");
}
...
```




Lab. 1 语言认知实验

- 见乐学平台上相关材料

