

Chapitre 1

Introduction C

Plan

- 1. Popularité des langages de programmation**
2. Origine du C/C++
3. Avantage / désavantages
4. Exemples de programmes
5. La compilation

1.1 Popularité des langages

TIOBE

Communauté qui élabore des méthodologies standards pour l'élaboration de codes.

PYPL











PopularitY of Programming Language index

IEEE

Institute of Electrical and Electronics Engineers

1.1 Popularité des langages

TIOBE, septembre 2025

Sep 2025	Sep 2024	Change	Programming Language		Ratings	Change
1	1			Python	25.98%	+5.81%
2	2			C++	8.80%	-1.94%
3	4	↗		C	8.65%	-0.24%
4	3	↘		Java	8.35%	-1.09%
5	5			C#	6.38%	+0.30%
6	6			JavaScript	3.22%	-0.70%
7	7			Visual Basic	2.84%	+0.14%
8	8			Go	2.32%	-0.03%
9	11	↗		Delphi/Object Pascal	2.26%	+0.49%
10	27	↗		Perl	2.03%	+1.33%

1.1 Popularité des langages

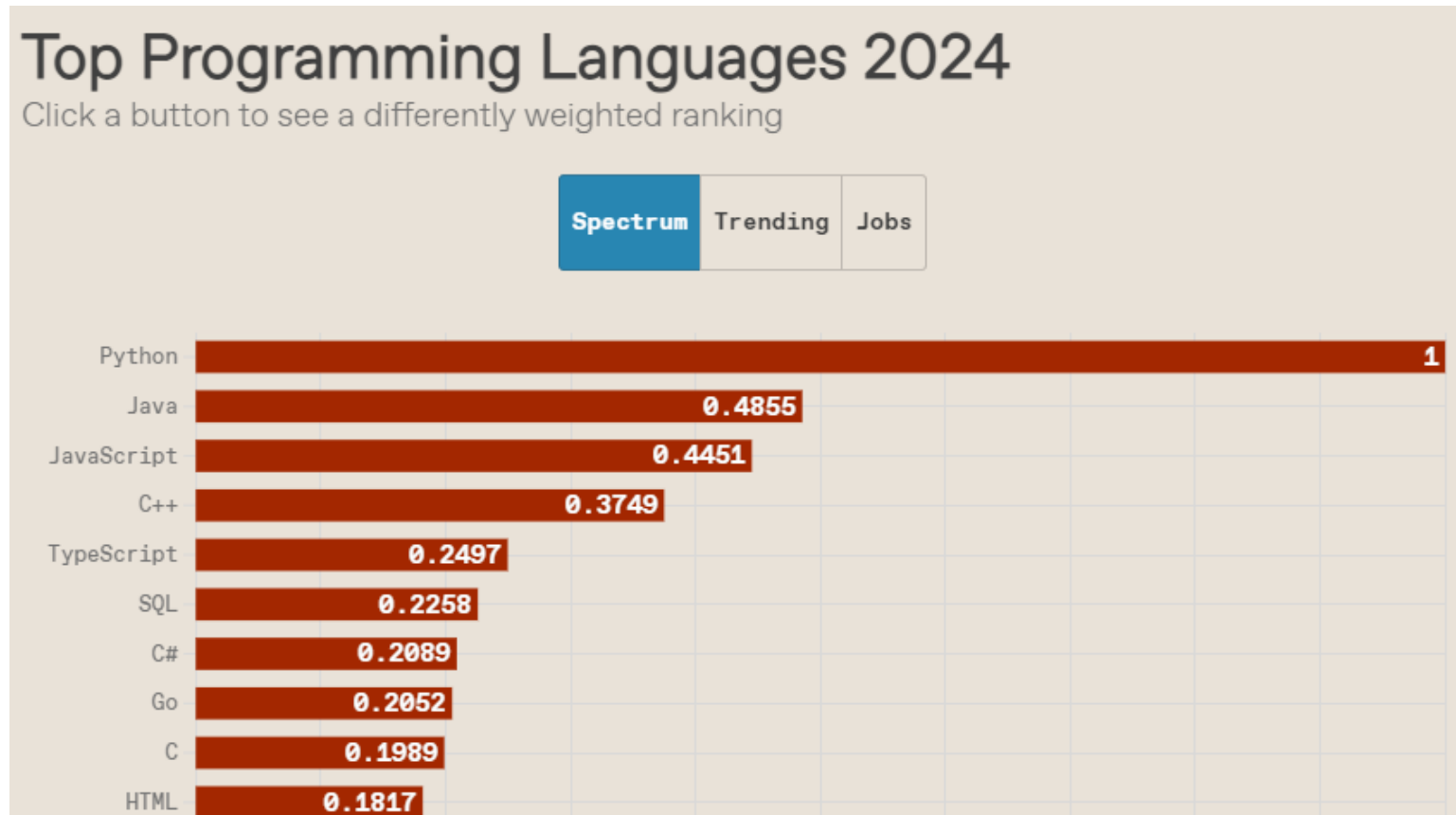
PYPL, septembre 2025

Worldwide, Sept 2025 :

Rank	Change	Language	Share	1-year trend
1		Python	29.69 %	+0.2 %
2		Java	14.72 %	-0.7 %
3	↑	C/C++	9.27 %	+2.5 %
4	↓	JavaScript	6.79 %	-1.4 %
5	↑	R	5.26 %	+0.6 %
6	↓	C#	4.81 %	-1.8 %
7	↑↑↑↑↑	Objective-C	4.17 %	+1.7 %
8	↓	PHP	3.34 %	-0.8 %
9	↑	Rust	2.73 %	+0.2 %
10	↓	Swift	2.72 %	+0.0 %

1.1 Popularité des langages

IEEE, 2024



Plan

1. Classement des langages de programmation
- 2. Origine du C/C++**
3. Avantage / désavantages
4. Exemples de programmes
5. La compilation

1.2 Origine des langages C/C++

1960 - **Le langage CPL** - *Combined Programming Language*

1967 - **Le langage BCPL** - *Basic Combined Programming Language* [Martin Richard]

1970 - **Le langage B** possédant des structures de contrôle indépendant de la machine et permettant d'écrire des compilateurs, récrit UNIX en langage B. [Ken Thompson]

1972 - Le langage C s'inspire fortement des langages B et BCPL des laboratoires Bell [Dennis Ritchie]

1.2 Histoire des langages C et C++

1973 - K&R C : UNIX est écrit à 90% en K&R C. Cette première version du C des auteurs Kernighan et Ritchie est définie dans le livre *The C Programming Language* (1978)

1983 - ANSI C : *American National Standards Institute*, résultat d'un effort de standardisation remplace le K&R C, et évolue jusqu'au standard connu comme **C ANSI**, **C89** ou **C90**

1985 - C++ : évolution du C par Bjarne Stroustrup
Le principal "+" du C++ est la possibilité d'utiliser les concepts de la programmation orientée objet avec l'efficacité du ANSI C

1.2 Histoire des langages C et C++

1999 - C99 : l'ISO standardise une extension de la norme du langage C, connue sous le nom de **C99**

2011 - C11 : nouvelle norme du langage C depuis décembre 2011, avec la définition du nouveau standard **C11**

2018 - C17

2024 - C23

Plan

1. Classement des langages de programmation
2. Origine du C/C++
- 3. Avantage / désavantages**
4. Exemples de programmes
5. La compilation

1.3 Avantages de C

Rapide

Universel

Compact, *code et empreinte mémoire*

Accès bas niveau

Indépendant de la machine

Multi-plateforme

1.3 Désavantages de C

Clarté non garantie
concours du code obscur

Portabilité et bibliothèques
de fonctions

Discipline de programmation

```
#include <math.h>
#include <sys/time.h>
#include <X11/Xlib.h>
#include <X11/Xkeysym.h>

double L, o, P,
    _dt, T, Z, D=1, d,
    s[999], E, h= 8, I,
    J, K, w[999], M, m, 0,
    n[999], j=33e-3, i=
    1E3, r, t, u, v, W, S=
    74.5, l=221, X=7.26,
    a, B, A=32.2, c, F, H;
int N, q, C, y, p, U;
Window z; char f[52];
; GC k; main(){ Display *e=
XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0),BlackPixel(e,0))
; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(0)){ struct timeval G={ 0,dt*1e6};
; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+= *_P; r=E*K; W=cos( 0); T=sin(0); O+=D*_F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d*_D- *_F*E; P=W*E*B-T*D; for (o+=(I-D)*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
]= 0|K <fabs(W-T*r-I*E +D*P) |fabs(D-t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2; K
*D; N=1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_*(X*t +P*M+m*1); T=X*X+ 1*1+M *M;
XDrawString(e,z,k ,20,300,f,17); D=v/l*15; i+= (B *1-M*r -X*Z)*_; for(; XPending(e); u *=CS!-N){
XEvent z; XNextEvent(e ,&z);
++*((N-XLookupKeysym
(&z->key,0))-IT?
N-LT? UP-N&E E:&
J:& u: &h); --*(
DN -N? N-DT ?N==
RT?&u: & W:&h:&J
); } m=15*F/l;
c+=(I-M/ 1,1*H
+I*M+a*X)*_; H
=A*r+v*X-F*1+(
E=-.1+X*4.9/l,t
=T*m/32-I*T/24
)/S; K=F*M+(
h* 1e4/l-(T+
E*5*T*E)/3e2
)/S-X*d-B*A;
a=2.63 /l*d;
X+=( d*1-T/S
*(.19*E +a
*.64+J/1e3
)-M* v +A*
Z)*_; 1 +=
K *_; W=d;
sprintf(f,
"%5d %3d"
"%d",p =1
/l.7,(C=9E3+
0*57.3)%0550,(int)i); d+=T*(.45-14/l*
X-a*130-J* .14)*_/_125e2+F*_v; P=(T*(47
*I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
179*v)/2312; select(p=0,0,0,0,&G); v--(
W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
)/107e2)*_; D=cos(o); E=sin(o); }
```

Plan

1. Classement des langages de programmation
2. Origine du C/C++
3. Avantage / désavantages
- 4. Exemples de programmes**
5. La compilation

1.4 Exemple de code

Fichier `hello.c`

```
#include <stdio.h>
```

Fonction principale

```
int main(void)
```

```
{
```

```
    printf("Hello World!\n");
```

```
    return 0;
```

```
}
```

Bloc

1.4 Exemple de code

Fichier **add.c**

```
#include <stdio.h>

int main(void)
{
    int a = 10;
    int b = 5;
    printf("%d", a+b);
    return 0;
}
```


Plan

1. Classement des langages de programmation
2. Origine du C/C++
3. Avantage / désavantages
4. Exemples de programmes
- 5. La compilation**

1.5 La compilation

Le C est un langage **compilé**

Un programme C est décrit par un fichier texte, appelé **fichier source**.

Il faut le **traduire**, le **compiler**, en langage machine car le texte n'est pas exécutable par le microprocesseur

La compilation

La compilation est effectuée par un programme appelé **le compilateur**.

Elle se réalise en **4 phases** successives :

1. Le prétraitement : *preprocessing*
2. La compilation : *compilation*
3. L'assemblage : *assembly*
4. L'édition des liens : *linking*

Exercices



Exercices du chapitre 01