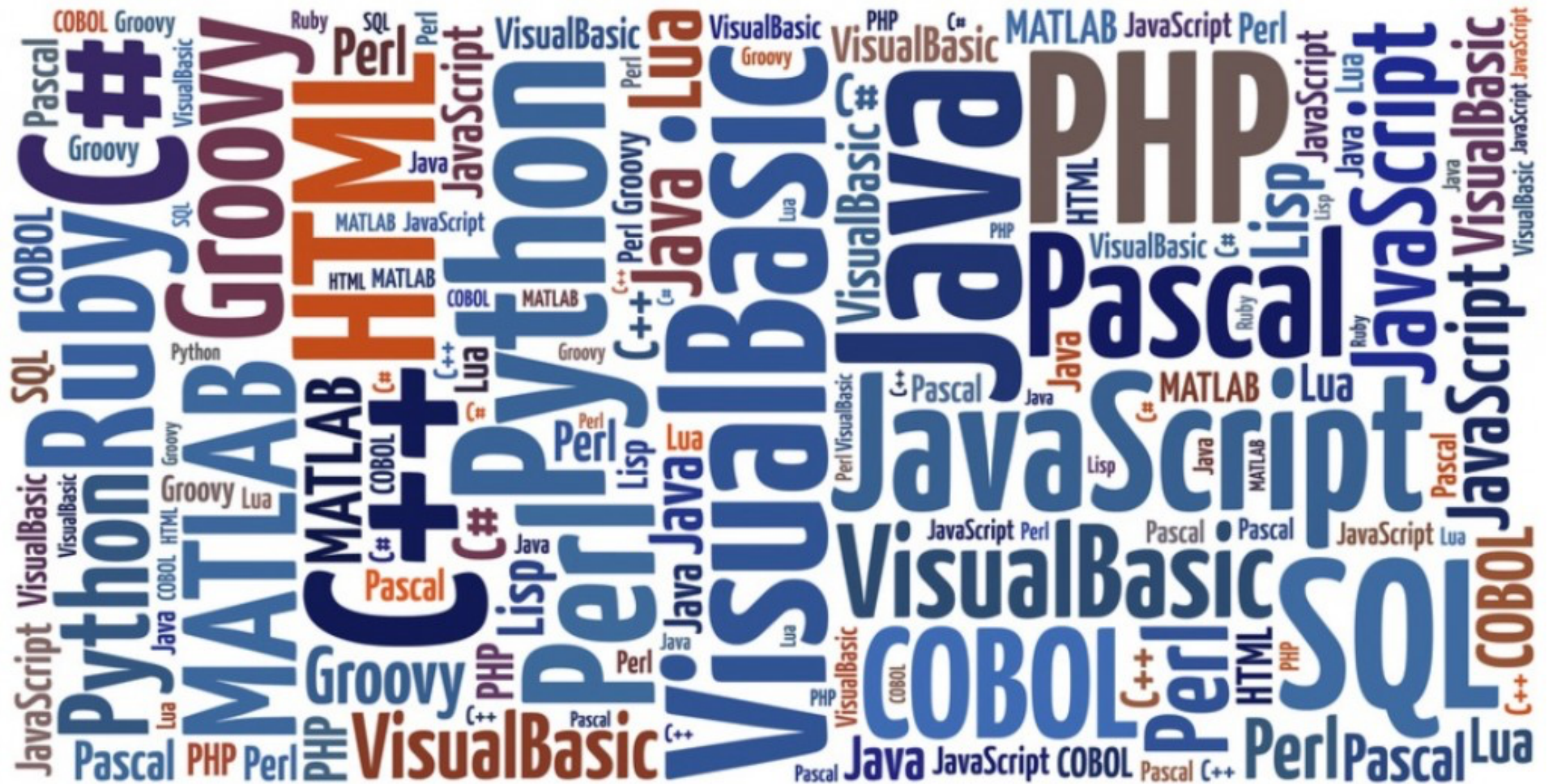


Programmieren in Python

Einführung



Was sind Programmiersprachen?

Was sind Programmiersprachen?

Die Popularität der Programmiersprachen

Apr 2019	Apr 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.035%	-0.74%
2	2		C	14.076%	+0.49%
3	3		C++	8.838%	+1.62%
4	4		Python	8.166%	+2.36%
5	6	⬆	Visual Basic .NET	5.795%	+0.85%
6	5	⬇	C#	3.515%	-1.75%
7	8	⬆	JavaScript	2.507%	-0.99%
8	9	⬆	SQL	2.272%	-0.38%
9	7	⬇	PHP	2.239%	-1.98%
10	14	⬆	Assembly language	1.710%	+0.05%
11	18	⬆	Objective-C	1.505%	+0.25%
12	17	⬆	MATLAB	1.285%	-0.17%
13	10	⬇	Ruby	1.277%	-0.74%
14	16	⬆	Perl	1.269%	-0.26%
15	11	⬇	Delphi/Object Pascal	1.264%	-0.70%
16	12	⬇	R	1.181%	-0.63%
17	13	⬇	Visual Basic	1.060%	-0.74%
18	19	⬆	Go	1.009%	-0.17%
19	15	⬇	Swift	0.978%	-0.56%
20	68	⬆	Groovy	0.932%	+0.82%

TIOBE Index for April 2019

Mar 2020	Mar 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.78%	+2.90%
2	2		C	16.33%	+3.03%
3	3		Python	10.11%	+1.85%
4	4		C++	6.79%	-1.34%
5	6	⬆	C#	5.32%	+2.05%
6	5	⬇	Visual Basic .NET	5.26%	-1.17%
7	7		JavaScript	2.05%	-0.38%
8	8		PHP	2.02%	-0.40%
9	9		SQL	1.83%	-0.09%
10	18	⬆	Go	1.28%	+0.26%
11	14	⬆	R	1.26%	-0.02%
12	12		Assembly language	1.25%	-0.16%
13	17	⬆	Swift	1.24%	+0.08%
14	15	⬆	Ruby	1.05%	-0.15%
15	11	⬇	MATLAB	0.99%	-0.48%
16	22	⬆	PL/SQL	0.98%	+0.25%
17	13	⬇	Perl	0.91%	-0.40%
18	20	⬆	Visual Basic	0.77%	-0.19%
19	10	⬇	Objective-C	0.73%	-0.95%
20	19	⬇	Delphi/Object Pascal	0.71%	-0.30%

TIOBE Index for März 2020

TIOBE-Index

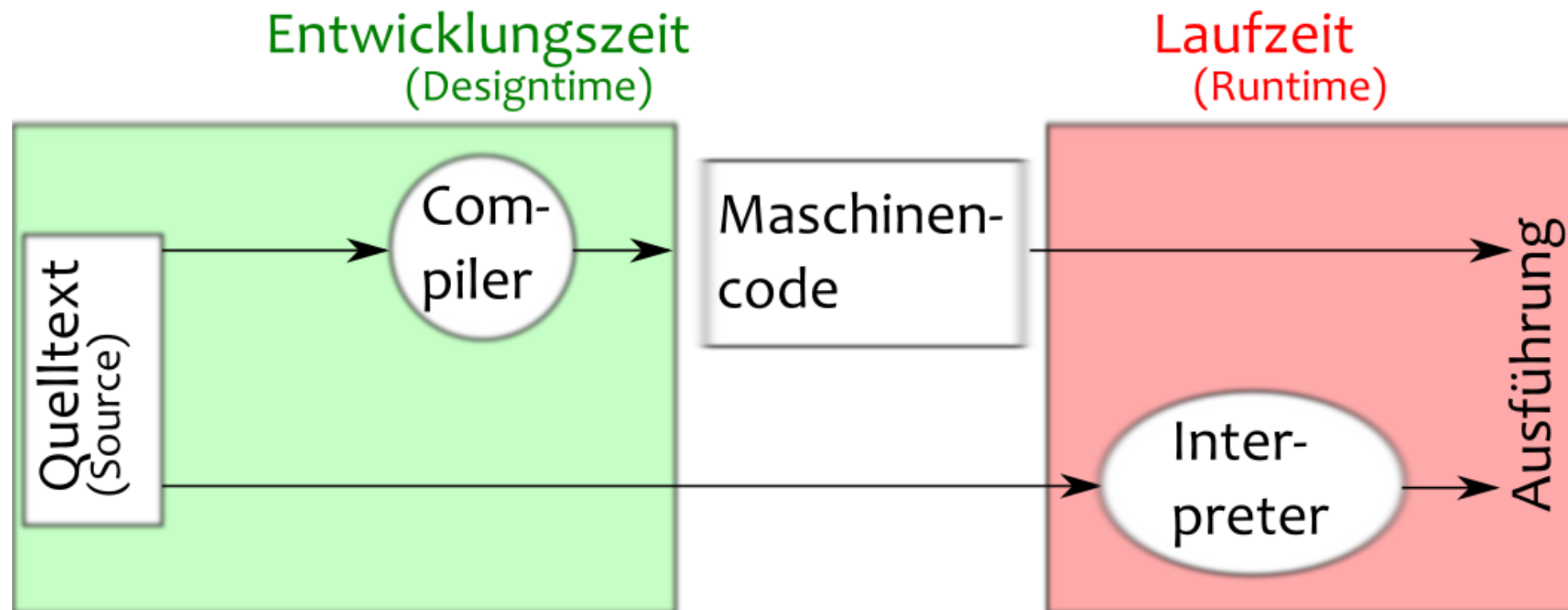
<https://www.tiobe.com/tiobe-index/>

Compiler und Interpreter?

Was ist ein Compiler?

Was ist ein Interpreter?

Compiler und Interpreter



Vorteile Compiler

- Der entstandene Code ist dem Prozessor angepasst = Maximale Ausführungsgeschwindigkeit
- Während des Übersetzungsvorganges werden die meisten Fehler bereits gefunden.
- Der Quelltext des Programmes ist geschützt (wird nicht benötigt)

Vorteile Interpreter

- Zur Ausführung ist kein Übersetzungsvorgang und -programm nötig.
- Quelltexte sind einsehbar, Programmschnipsel leicht wiederverwendbar.

Fazit: Kurze "Skripts" eignen sich für Interpreter, ganze Applikationen sind besser mit Compiler zu entwickeln.

Warum Python?

“Hello, World”

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```

- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```

- now in Python

```
print "Hello, World!"
```

Einführung in Python

Python ist eine freie, plattformunabhängige, interpretierende, Open-Source-Script-Programmiersprache, die leistungsstark und einfach zu erlernen ist.

Einige Vorteile von Python:

- Einfach zu erlernen und ausgezeichnet für Anfänger geeignet, und doch großartig für Experten
- Umfassend skalierbar, geeignet für große Projekte oder kleine einmalige Programme, die als Skripte bezeichnet werden
- Übertragbar, plattformunabhängig
- Stabil und ausgereift
- Große Benutzergemeinschaft
- Die Programmiersprache Python unterstützt die wichtigsten Programmier-Paradigm heutiger Software-Entwicklungsmethoden: Strukturierte Programmierung, Object-Orientierte-Programmierung (OOP) und Aspect-Oriented-Programming (AOP).

Einführung in Python

- In den meisten Programmiersprachen - so auch in Python - ist ein Programm eine einfache Textdatei, welche mit einem beliebigen Texteditor bearbeitet werden kann.
- Das Ausführen des Programms bedeutet einfach, dass Python die Anweisungen in der Textdatei Zeile für Zeile durcharbeitet, bis es am Ende der Datei angelangt ist.
- Das erste Beispielprogramm, welches üblicherweise in einer Programmiersprache verfasst wird, ist ein so genanntes „Hello World“-Programm, welches nichts anderes macht, als den Text „Hello World“ auf dem Bildschirm auszugeben.

```
1 print("Hello World!")
```

Was brauchen wir zum Programmieren in Python?

1. Python-Interpreter (Laufzeitumgebung)

- Python3 (Windows, Mac OS X, Ubuntu) - <https://www.python.org>

2. Editoren und IDEs

- Um Software in Python zu entwickeln, reicht in der Theorie ein einfacher Texteditor.
- Besser ist die Verwendung einer **integrierten Entwicklungsumgebung - IDE** (integrated development environment) mit Python-Support

Vorteile:

- Das Programm kann direkt im IDE ausgeführt werden und muss nicht separat aufgerufen werden.
- Die Befehle werden automatisch vervollständigt, so dass man nicht die ganzen Befehle tippen muss.
- Gewisse Fehler im Programm werden schon während dem programmieren erkannt und als solche gekennzeichnet.

Editoren und IDEs

Ein Quelltext-Editor erlaubt das Lesen und Schreiben von Programmen. Es gibt viele, und es ist eine sehr persönliche Wahl – wie ein Tennisspieler seinen Schläger oder ein Koch sein Lieblingsmesser wählt. Um anzufangen benötigst du nur einen simplen, einfach zu benutzenden Editor der dir nicht in die Quere kommt, aber immer noch effektiv beim Schreiben vom Pythonquelltext ist.

Hier sind ein paar Vorschläge:

Dedizierte IDEs für Python:

- IDLE, PyCharm, Komodo, Spyder,

Beliebte IDE's mit Python Support:

- Visual Studio, Visual Studio Code, Eclipse, Sublime Text, Vim, Emacs, Atom

Online Editor

- repl.it - a Powerful and simple online compiler, IDE, interpreter, and REPL

TIPP

Wordpad, TextEdit, Notepad und Word sind keine geeigneten Quelltexteditoren.

los geht's!