|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Beschreibung** | **Element** | **Beschreibung** |
| ^ | Start des Strings | \* | 0 oder mehr |
| $ | Ende des Strings | + | 1 oder mehr |
| \s | Leerschlag | ? | 0 oder 1 |
| \S | Kein Leerschlag | {4} | Genau 4 |
| \d | Ziffer | {4,} | 4 oder mehr |
| \w | Buchstabe (oder eine Ziffer!) | {4,8} | 4,5,6,7 oder 8 |
| \n | Zeilensprung | [^a] | Nicht a |
| \t | Tabulator |  |  |
| . | Beliebiges Zeichen (ausser Zeilensprung) |  |  |
| (a|b) | a oder b |  |  |
| (abc) | Gruppierung |  |  |
| [a-g] | Zeichen von a bis g |  |  |
| [0-9] | Ziffer von 0 bis 9 (analog \d) |  |  |

# Regex

# Python

**print(**"Unendlicher Fibanacci-Generator"**)**

**def** fibonacci**():**

a**,** b **=** 0**,** 1

**while** **True:**

**yield** a

a**,** b **=** b**,** a**+**b

f **=** fibonacci**()**

counter **=** 0

**for** x **in** f**:**

**print(**"fib(" **+** str**(**counter**)** **+**") = " **+** str**(**x**))**

counter**+=**1

**if** **(**counter **>** 25**):** **break**

**for i in range(5):**

**print(i)**

**import** sys

**for** arg **in** sys**.**argv**:**

**print(**arg**)**

**with** open**(**'Data1.csv'**,** newline**=**''**)** **as** f**:**

reader **=** csv**.**reader**(**f**,** delimiter**=**';'**)**

**for** row **in** reader**:**

**for** col **in** row**:**

|  |  |  |
| --- | --- | --- |
| **Konstrukt** | **Beschreibung** | **Beispiel** |
| class *Klassenname* | Klassendefinition | class Fahrzeug: |
| def *Methodenname* | Methodendefinition (mit 2 Parametern) | def FahrezuZiel(self, Zielname): |
| *Variablenname* = Klassenname() | Instanzierung einer neuen Klasse in einer Variable | Tesla = Fahrzeug() |
| *Variablen-name.Methodenname* | Aufruf einer Methode | Tesla.FahrezuZiel('Bern') |

# Bash

## Vergleich von Zeichenketten

|  |  |
| --- | --- |
| **Operator** | **Bedeutung** |
| **=** | Gleich |
| != | Ungleich |
| -z | Leere Zeichenkette |
| -n | Nicht leere Zeichenkette |

## Vergleich von numerischen Werten

|  |  |
| --- | --- |
| **Operator** | **Bedeutung** |
| -eq | Gleich |
| -ne | Ungleich |
| -lt | Kleiner als |
| -le | Kleiner oder gleich |
| -gt | Grösser als |
| -ge | Grösser oder gleich |

## Abfrage von Dateiattributen

|  |  |
| --- | --- |
| **Operator** | **Bedeutung** |
| -e | Existiert die Datei |
| -r | Leserecht gesetzt |
| -w | Schreibrecht gesetzt |
| -x | Ausführungsrecht gesetzt |
| -f | Normale Datei |
| -d | Verzeichnis |
| ! | Nicht |

## Shell-Variablen

|  |  |
| --- | --- |
| **Variable** | **Bedeutung** |
| $BASH\_VERSION | Version der Bash-Shell |
| $HOME | Heimverzeichnis |
| $HOSTNAME | Computername |
| $OLDPWD | Letztes Arbeitsverzeichnis |
| $PS1 | Aussehen der Shell |
| $PWD | Aktuelles Arbeitsverzeichnis |

## Parameter

|  |  |
| --- | --- |
| **Parameter** | **Bedeutung** |
| $0 | Name des Skripts |
| $1 -$9 | 1- 9. Parameter |
| $# | Anzahl Parameter |
| $\* | Alle Parameter als Einheit |

## Operatoren

|  |  |
| --- | --- |
| **Operator** | **Erklärung** |
| > | Umleiten |
| >> | Anhängen |
| | | Pipe |
| % | Modulo |

# Linux-Befehle

|  |  |
| --- | --- |
| **Befehl** | **Erklärung** |
| **cd** verzeichnis | Verzeichnis wechseln |
| **mv** datei1 datei2 | Datei verschieben/umbenennen |
| **cp** datei1 datei1 | Datei kopieren |
| **cat** datei | Datei ausgeben |
| **adduser** | Benutzer erstellen |
| **chmod** 777 datei | Rechte verteilen |
| **sqlite3** „SELECT \*…“ | sqlite Befehle benutzen |
| **ls** –l | Dateien & Berechtigungen auflisten |
| **which** bash | Installationspfad anzeigen |
| **deluser** Benutzer | Benutzer löschen |
| **grep** –iE „(tux|root)“ Datei | Nach tux oder root in Datei suchen |
| **apt-get** install Programm | Programm installieren |

# http://upload.wikimedia.org/wikipedia/commons/thumb/6/6b/Flowchart_de.svg/285px-Flowchart_de.svg.pngDokumentation

## Teile einer Dokumentation

* Aufgabenbeschreibung
* Programmbeschreibung
* Programmcode
* Testfälle und –protokolle
* Bedienungsanleitung (Manual)