

⚡-Talks

2. Dezember 2010

im KunterBuntenSeminar

Spielregeln

- ▶ 9 Vorträge
- ▶ 5 Minuten pro Vortrag
- ▶ Gelegenheit für Fragen jeweils im Anschluss

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Tunneln mit sshuttle

Julian Fietkau

am 2. Dezember 2010
im KunterBuntenSeminar

Versuche, von zuhause aus eine beliebige Webseite aufzurufen, die nur aus dem Informatik-Netz erreichbar ist:

Zugriff verweigert!

Der Zugriff auf das angeforderte Verzeichnis ist nicht möglich. Entweder ist kein Index-Dokument vorhanden oder das Verzeichnis ist zugriffsgeschützt.

Sofern Sie dies für eine Fehlfunktion des Servers halten, informieren Sie bitte den Webmaster hierüber.

Möglichkeit 1: VPN

`fbivpn.informatik.uni-hamburg.de`

- ▶ Zugriff auf fachbereichsinterne Seiten möglich
- ▶ Taugt im UHH-WLAN gleichzeitig als Authentisierung für Internet
- ▶ Aller IP-Verkehr systemweit geht durch das VPN
- ▶ schwerfällig, Verbindungsaufbau dauert
- ▶ nicht immer stabil (Verbindungsabbrüche)

Möglichkeit 2: SSH FBI

```
home$ ssh 0nachnam@rzssh1...  
ikum$ wget ...
```

- ▶ schlank
- ▶ schnell
- ▶ kaum Konfiguration und keine zusätzliche Software nötig
- ▶ Nur die SSH-Verbindung wird getunnelt
- ▶ auf Dauer umständlich und lästig

Möglichkeit 3: SSH + DynamicForward + tsocks

```
home$ ssh fbi & tsocks  
home$ firefox ...
```

- ▶ immer noch schlanker als VPN
- ▶ immer noch schneller als VPN
- ▶ leicht zu handhaben
- ▶ Nur Traffic der tsocks-Shell wird getunnelt
- ▶ umständlich einzurichten

**There's a new tunneling
solution in town...**

sshuttle: where transparent proxy meets VPN meets ssh

„As far as I know, sshuttle is the only program that solves the following common case:

- ▶ Your client machine (or router) is Linux, FreeBSD, or MacOS.
- ▶ You have access to a remote network via ssh.
- ▶ You don't necessarily have admin access on the remote network.
- ▶ The remote network has no VPN, or only stupid/complex VPN protocols (IPsec, PPTP, etc). Or maybe you are the admin and you just got frustrated with the awful state of VPN tools

sshuttle: where transparent proxy meets VPN meets ssh

- ▶ You don't want to create an ssh port forward for every single host/port on the remote network.
- ▶ You hate openssh's port forwarding because it's randomly slow and/or stupid.
- ▶ You can't use openssh's PermitTunnel feature because it's disabled by default on openssh servers; plus it does TCP-over-TCP, which has terrible performance (see below).“

Möglichkeit 4: sshuttle

```
home$ git clone git://github.com/apenwarr/sshuttle
home$ ./sshuttle -r fbi 134.100.0.0/16
```

- ▶ schnell, im Sinne von **schnell** - subjektiv quasi nicht spürbar
- ▶ sehr leichtgewichtig
- ▶ Traffic wird systemweit getunnelt
- ▶ braucht root-Rechte auf dem Client
- ▶ Verbindung bricht manchmal ab, lässt sich jedoch automatisiert neu starten

Funktionsweise von sshuttle

- ▶ sshuttle lädt sich selbst auf den SSH-Server
- ▶ Auf dem Client werden lokale Routing-Regeln aktiviert, so dass sshuttle den Traffic abgreifen kann
- ▶ Der IP-Traffic geht über die SSH-Verbindung ans serverseitige sshuttle und von dort ans Ziel
- ▶ übrigens: komplett in Python geschrieben

Verwendung

- ▶ `sshuttle -r fbi 134.100.0.0/16` in einem lokalen Terminal
- ▶ surfen, chatten, etc. wie gewohnt - jeglicher Traffic zu Uni-Servern wird dabei transparent über `rzssh1` getunnelt
- ▶ zum Beenden: `Strg+C` in dem Terminal in dem `sshuttle` läuft
- ▶ bei Verbindungsabbruch: neu starten
 - ▶ primitive Selbstreparatur:

```
while true; do sshuttle -r ikum 134.100.0.0/16; sleep 2; done
```

Danke für die Aufmerksamkeit!



http://www.julian-fietkau.de/tunneln_mit_sshuttle



Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Awesome

Senad Licina

Kunterbuntes Seminar
Fachbereich Informatik
Uni Hamburg

02.12.2010



Was ist Awesome?

Awesome ist...

- ... ein konfigurierbarer Windowmanager für X
- ... in Lua Programmiert
 - Lua ist eine freie skriptsprache (MIT-Lizenz)
 - simpel, klein & schnell
- ... Open Source (GNU GPL)

Warum Awesome?

- Der Name ist einfach mal Awesome! =)
- Keine Fensterrahmen!
- Platz kann optimal ausgenutzt werden!
- Er läuft schnell und stabil!
- Es existiert eine gute Dokumentation mit vielen Anleitungen
- Man benutzt ein “Tag“-System
- Auf der offiziellen Seite gibt es ein HIMYM-Zitat!
 - “This gonna be LEGEN... wait for it... DARY!”
 - <http://awesome.naquadah.org/>
- Hab ich schon erwähnt, dass es sich um freie Software handelt?

Live-Demo



Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web: Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Ausspionieren der User

Nils

Inhalt

- History stealing
- Wiedererkennen eines Users
- Verteidigung

•History stealing

- Normal: 1 Monat gespeichert
- Linkdarstellung: „besucht“ durch Farbe
- Linkfarbe auslesbar

Wiedererkennung

- Plugins
- Betriebssystem
- IP
- HTML 5 Speicheroptionen
- History
- Cache

Verteidigung

- NoScript
- History löschen

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Ein C-Puzzler

Julian Fietkau

am 2. Dezember 2010
im KunterBuntenSeminar

```
1 #include <stdio.h>
2
3 void main(void)
4 {
5     double value = 10.0;
6     printf("%g", 1/value);
7 }
```

→ **0.1**

#define MACRO(x,y,z) 2*(x+y*z)

Funktionsartige Makros:

Können im Quelltext ähnlich wie Funktionen verwendet werden und bekommen Argumente übergeben.

Achtung:

Makros werden bereits zur Übersetzungszeit ausgewertet. Der Code des Makros wird an allen Stellen, an denen das Makro verwendet wird, vor dem eigentlichen Kompilieren einfach eingesetzt.

```
1 #include <stdio.h>
2
3 #define INVERSE(x) 1/x
4
5 void main(void)
6 {
7     double value = 10.0;
8     printf("%g", INVERSE(value));
9 }
```

→ **0.1**



Referenzierungs-Operator:

Liefert einen Zeiger auf eine Variable.



Dereferenzierungs-Operator:

Liefert den Inhalt der angegebenen Adresse.

```
1 #include <stdio.h>
2
3 void main(void)
4 {
5     double value = 10.0;
6     double* pValue = &value;
7     printf("%g", 1 / *pValue);
8 }
```

→ **0.1**

```
1  #include <stdio.h>
2
3  #define INVERSE(x) 1/x
4
5  void main(void)
6  {
7      double value = 10.0;
8      double* pValue = &value;
9      printf("%g", INVERSE(*pValue));
10 }
```

Ⓐ

0.1

Ⓑ

Übersetzungsfehler

Ⓒ

Laufzeitfehler

Ⓓ

noch etwas Anderes

```
1  #include <stdio.h>
2
3  #define INVERSE(x) 1/x
4
5  void main(void)
6  {
7      double value = 10.0;
8      double* pValue = &value;
9      printf("%g", INVERSE(*pValue));
10 }
```

Ⓐ

0.1

Ⓑ

Übersetzungsfehler

Ⓒ

Laufzeitfehler

Ⓓ

noch etwas Anderes

```
1  #include <stdio.h>
2
3
4
5  void main(void)
6  {
7      double value = 10.0;
8      double* pValue = &value;
9      printf("%g", 1/*pValue);
10 }
```

Ⓐ

0.1

Ⓑ

Übersetzungsfehler

Ⓒ

Laufzeitfehler

Ⓓ

noch etwas Anderes

```
1  #include <stdio.h>
2
3
4
5  void main(void)
6  {
7      double value = 10.0;
8      double* pValue = &value;
9      printf("%g", 1/*pValue);
10 }
```

(A)

0.1

(B)

Übersetzungsfehler

(C)

Laufzeitfehler

(D)

noch etwas Anderes

```
1  #include <stdio.h>
2
3  #define INVERSE(x) 1/x
4
5  void main(void)
6  {
7      double value = 10.0;
8      double* pValue = &value;
9      printf("%g", INVERSE(*pValue));
10 }
```

Ⓐ

0.1

Ⓑ

Übersetzungsfehler

Ⓒ

Laufzeitfehler

Ⓓ

noch etwas Anderes

gcc -std=c99 -E macro.c

```
687 (...)
688
689 void main(void)
690 {
691     double value = 10.0;
692     double* pValue = &value;
693     printf("%g", 1/-*pValue);
694 }
```

```
1 #define PREPEND_ASTERISK(x) *x
2 #define PREPEND_SLASH(x) /x
3
4 PREPEND_ASTERISK(literal)
5 PREPEND_SLASH(literal)
6 PREPEND_ASTERISK(*pointer)
7 PREPEND_SLASH(*pointer)
```

gcc -std=c99 -E test.c

```
7 (...)
8
9 *literal
10 /literal
11 **pointer
12 / *pointer
```

ISO/IEC 9899:TC3 (C99-Standard)

5.1.1.2 Translation phases:

3. The source file is decomposed into **preprocessing tokens⁶⁾** and sequences of white-space characters (including comments). (...) **Each comment is replaced by one space character.** (...)
4. Preprocessing directives are executed, **macro invocations are expanded**, and `_Pragma` unary operator expressions are executed. (...)

(Hervorhebungen von mir.)

```
printf("%g", INVERSE(*pValue));
```

1

2

3

printf → (→ "%g" → , → [] → INVERSE → (→
* → pValue →) →) → ;

4

printf → (→ "%g" → , → [] → 1 → / → * →
pValue →) → ;

5

6

7

8

```
printf("%g", INVERSE(*pValue));
```

1

2

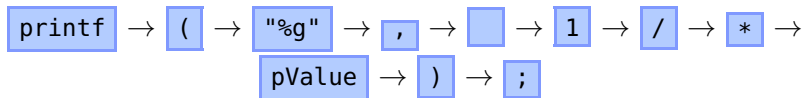
3

printf → (→ "%g" → , → [] → INVERSE → (→
* → pValue →) →) → ;

4

printf → (→ "%g" → , → [] → 1 → / → * →
pValue →) → ;

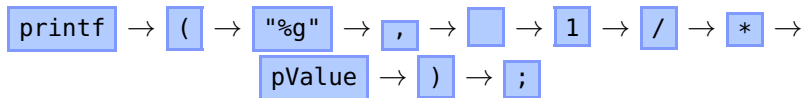
gcc -E



gcc -E

```
printf("%g", 1/*pValue);
```





gcc -E

```
printf("%g", 1/ *pValue);
```



Danke für die Aufmerksamkeit!



http://www.julian-fietkau.de/c_puzzler



Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

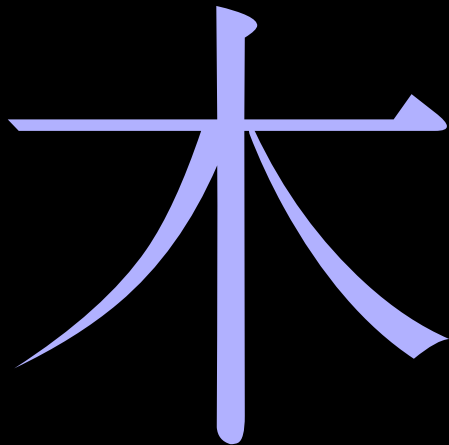
Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

漢字の起原

The Origin of Kanji

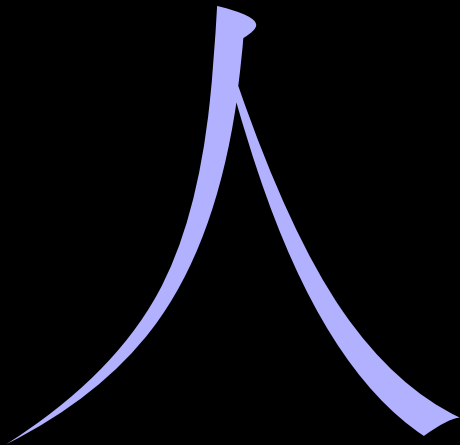
ヨツヘン サイモン
Jochen Simon

12月2日2010



林

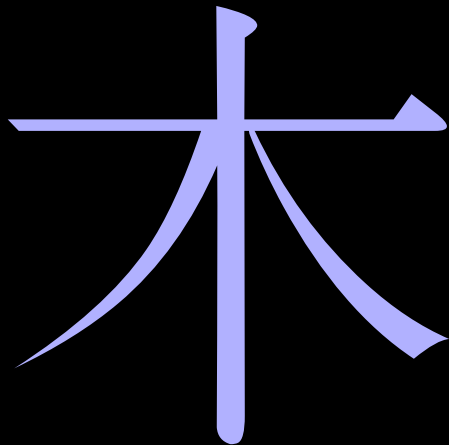
森



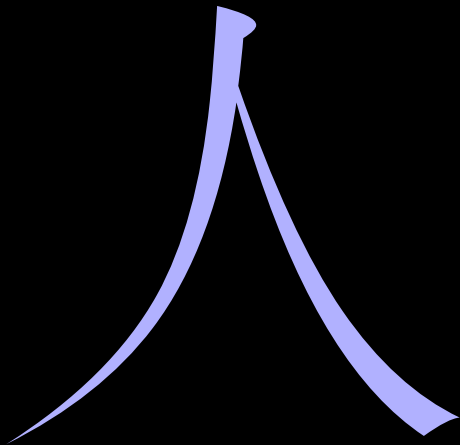
休

休

人木



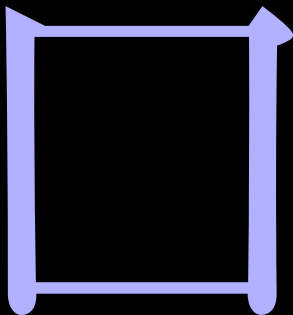
本

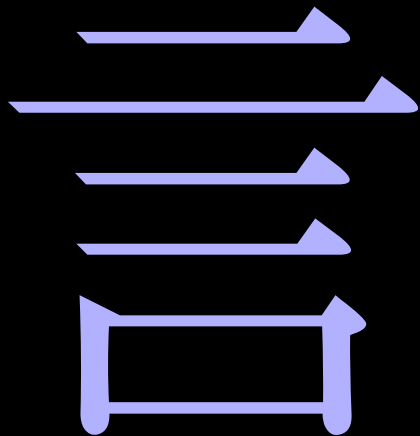


体

体

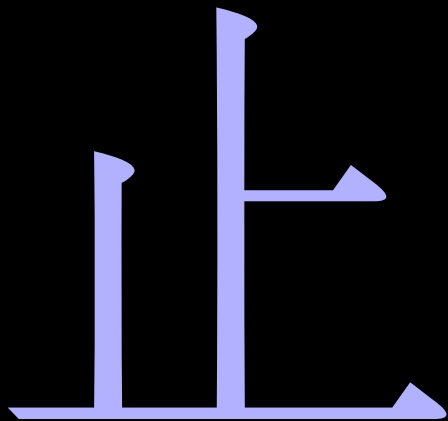
人本





信

信人



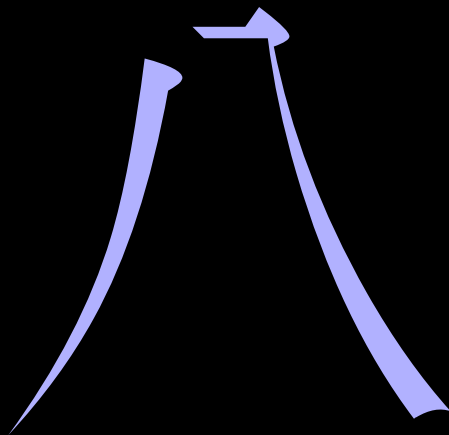
正

証言

証

言

刀



分

分

刀八

車

庫

軍

肉

豚

豚

肉

家

決定的文脈自由文法

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

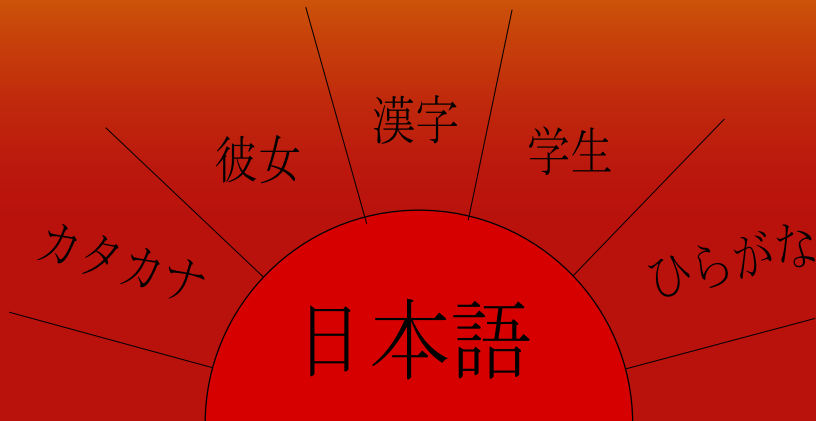
8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Kanji für Informatiker

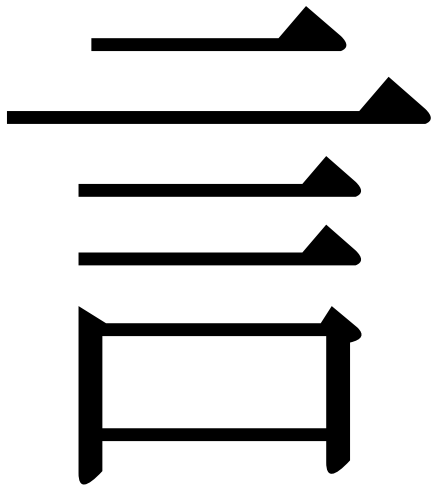
Unverkennbar bescheuerte Eselsbrücken

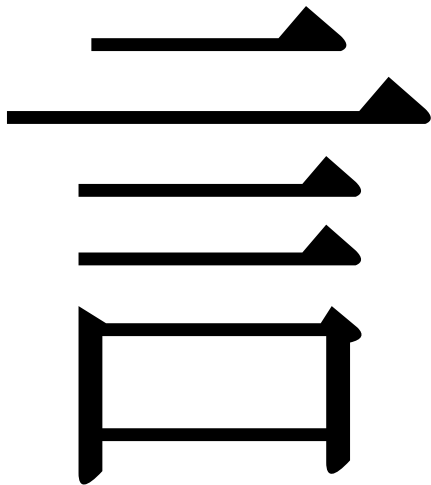
by ディミトリ 6ポポフ



Moderne Informatik

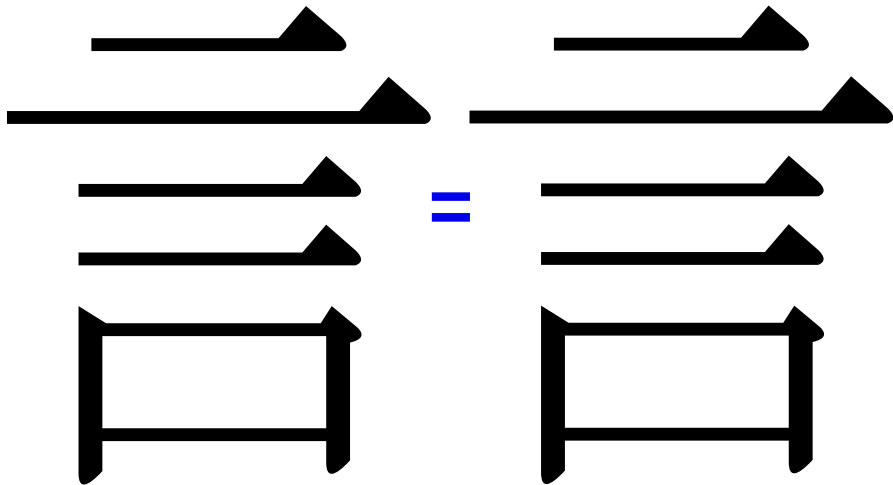
Moderne Informatik



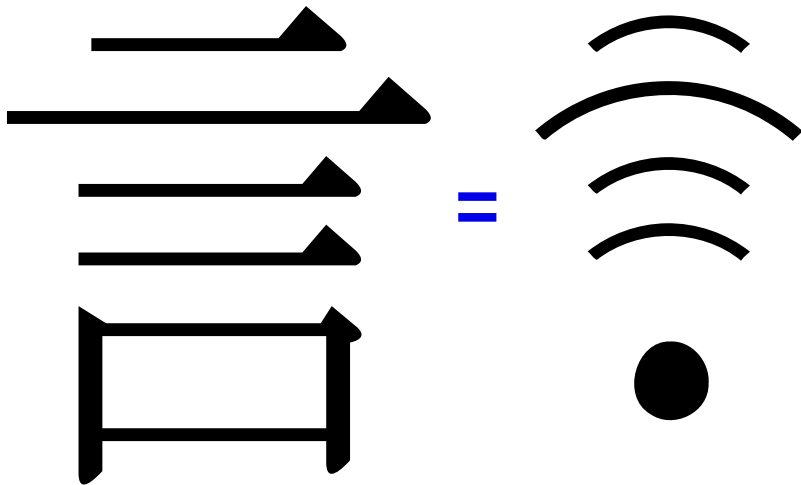


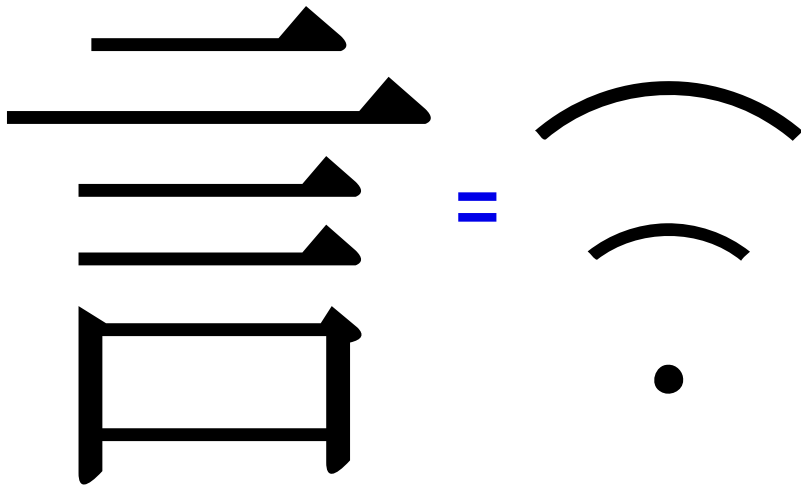
Methode:
Äquivalenzumformung

Moderne Informatik

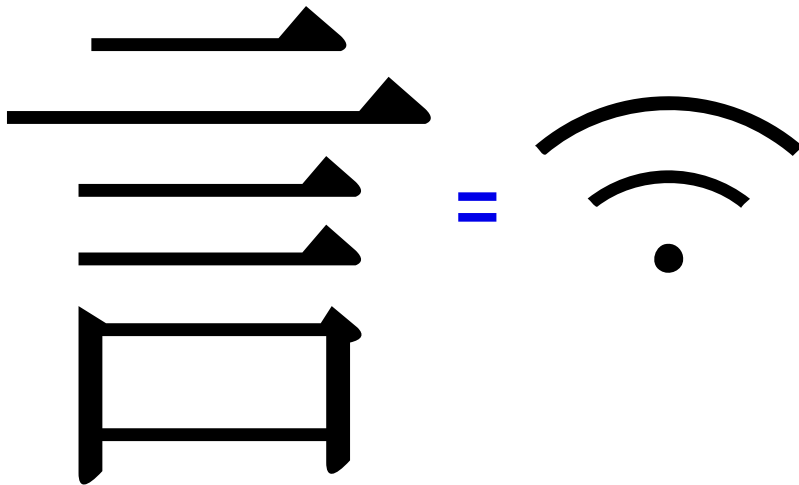


Moderne Informatik

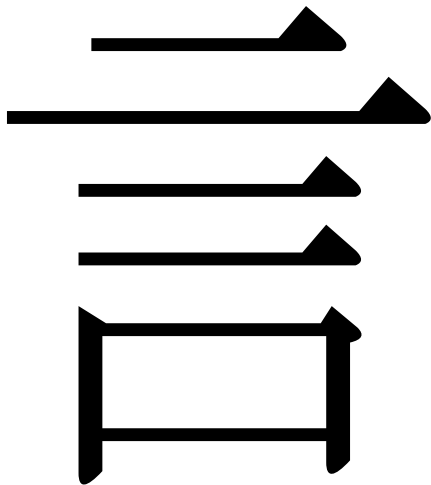




Moderne Informatik



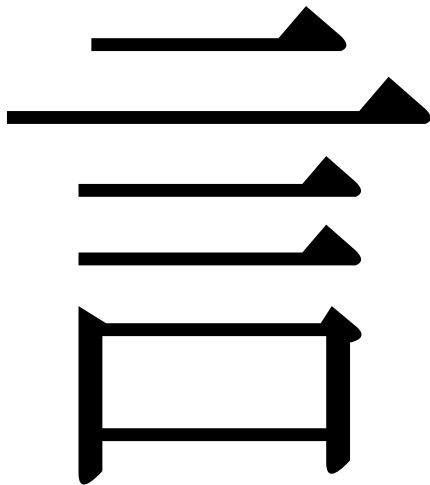
Moderne Informatik



=



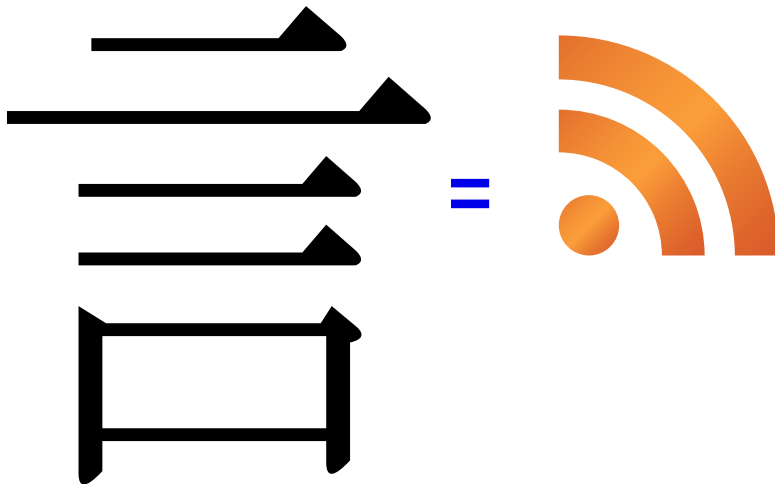
Moderne Informatik

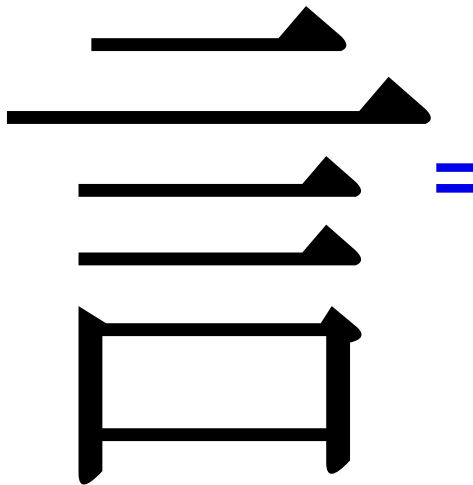


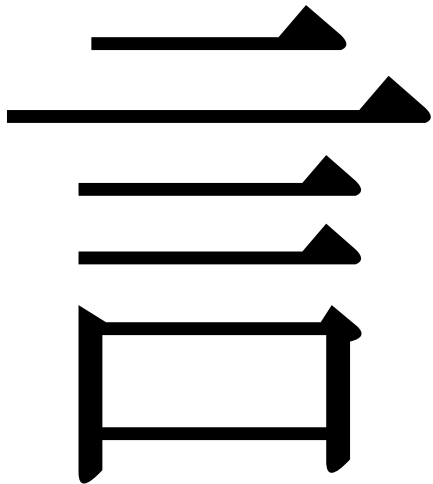
=



Moderne Informatik







etw. sagen

Begriffe aus dem Alltag

Begriffe aus dem Alltag eines Informatikers

酒

酒

Begriffe aus dem Alltag eines Informatikers

酒

Methode:
Texterkennung

酒

Begriffe aus dem Alltag eines Informatikers

酒

酒

Begriffe aus dem Alltag eines Informatikers

酒

酒

Begriffe aus dem Alltag eines Informatikers

酒

酒

Begriffe aus dem Alltag eines Informatikers

酒

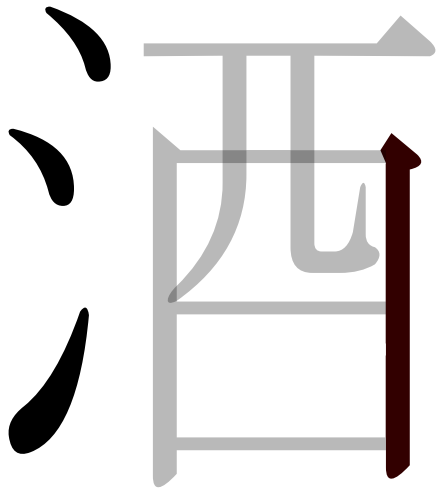
酒

Begriffe aus dem Alltag eines Informatikers

酒

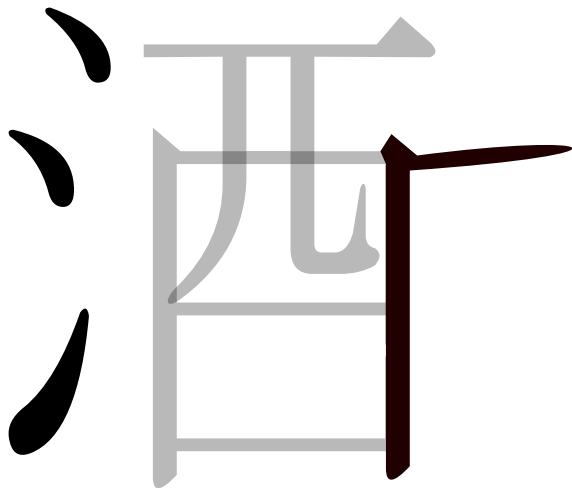
酒

Begriffe aus dem Alltag eines Informatikers



酒

Begriffe aus dem Alltag eines Informatikers



酒

Begriffe aus dem Alltag eines Informatikers

酒

酒

Begriffe aus dem Alltag eines Informatikers

兀

酒

Begriffe aus dem Alltag eines Informatikers

πE

酒

Begriffe aus dem Alltag eines Informatikers

π 日 一

酒

Begriffe aus dem Alltag eines Informatikers

沈黙

酒

Begriffe aus dem Alltag eines Informatikers

酒

“兀日”

酒

Begriffe aus dem Alltag eines Informatikers

酒

„サケ“

(jap.: „Sake“)

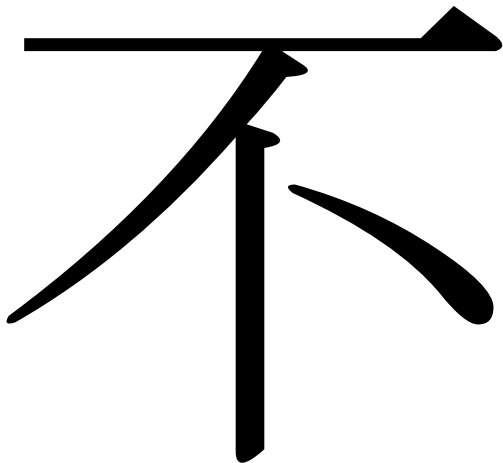
酒

Elementare Logik

不

不

Elementare Logik



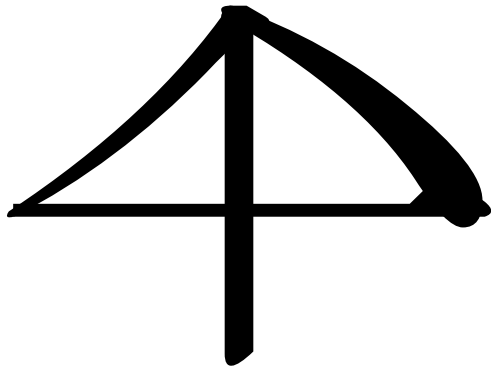
Methode:

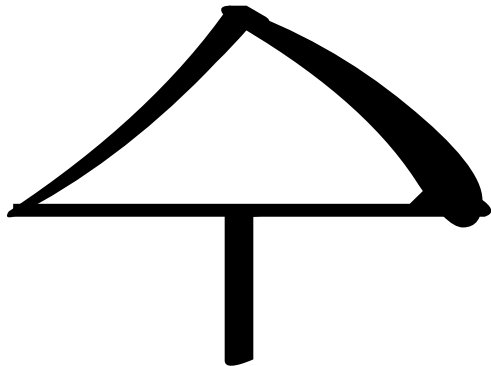
An Möller denken!

不

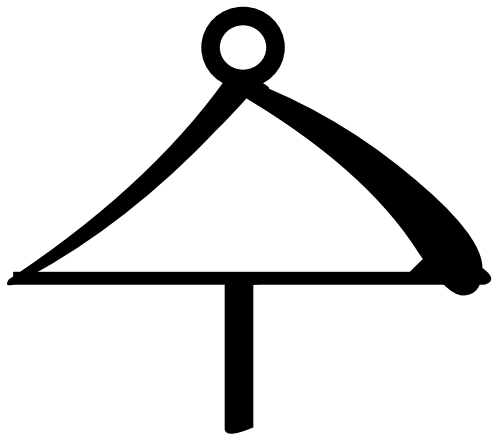
不

不



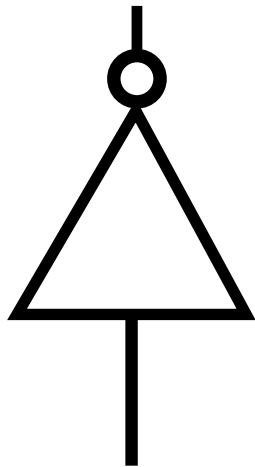
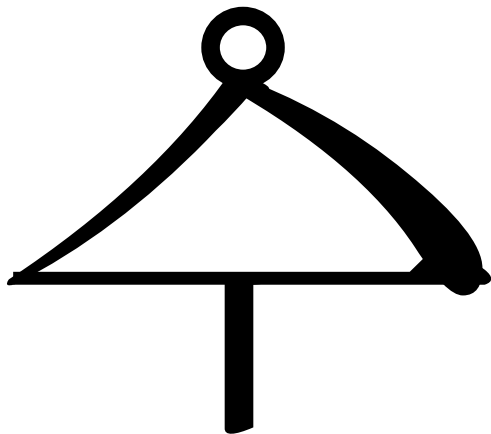


Elementare Logik



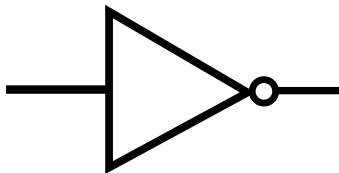
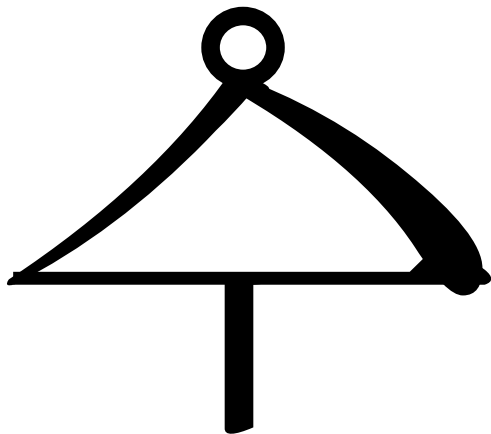
不

Elementare Logik



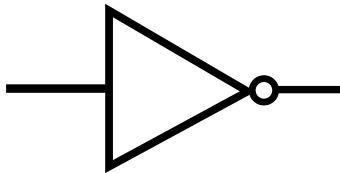
不

Elementare Logik



不

不

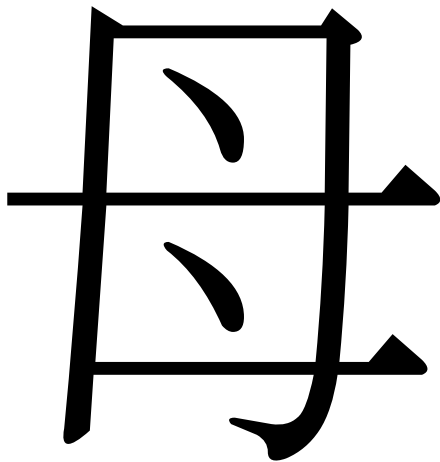


„nicht-“

不

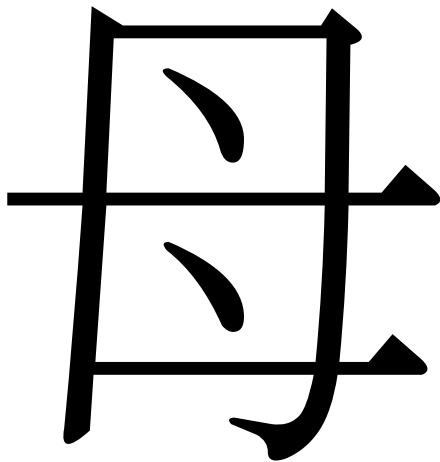
Näheres Umfeld

Näheres Umfeld



母

Näheres Umfeld



Methode:

*Verlustbehaftete
Bildkompression*

母

Näheres Umfeld

母

母

母

母

母

母

Näheres Umfeld



母

Näheres Umfeld



母

Näheres Umfeld



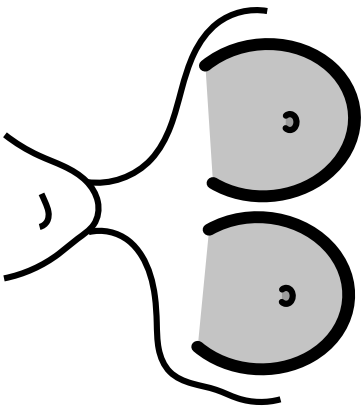
母

Näheres Umfeld



母

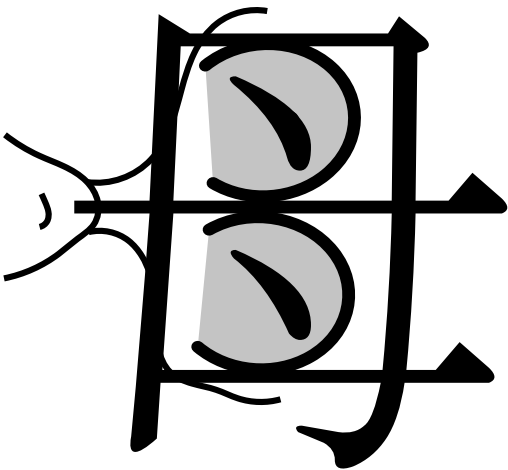
Näheres Umfeld



...

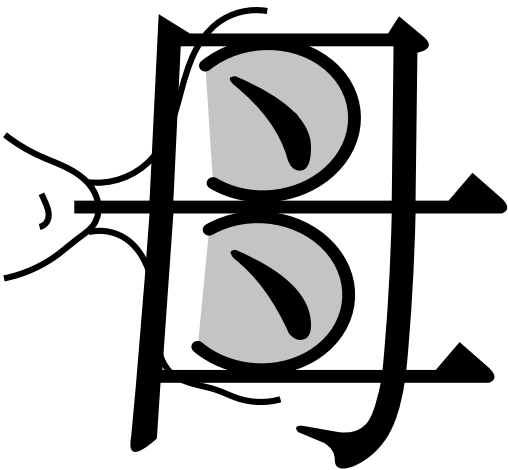
母

Näheres Umfeld



母

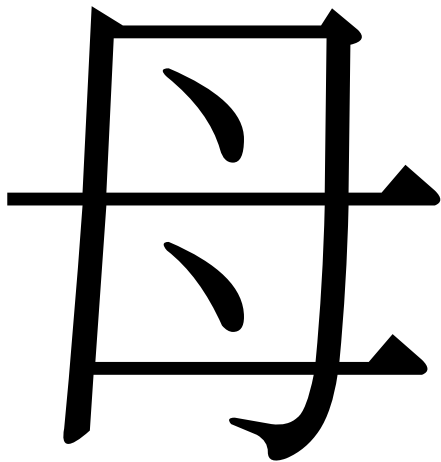
Näheres Umfeld



Mama

母

Näheres Umfeld



Mama

母

Der soziale Umgang miteinander

Der soziale Umgang miteinander

Der soziale Umgang miteinander



Der soziale Umgang miteinander

Dieses Kanji...



Der soziale Umgang miteinander

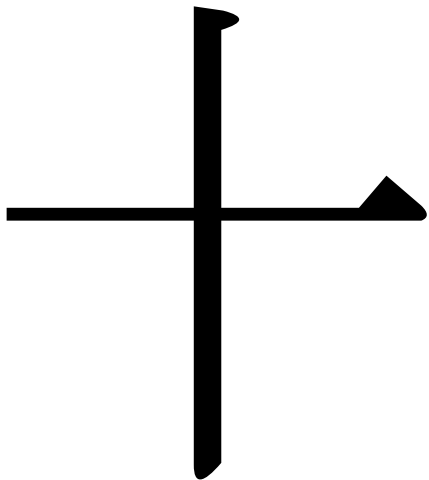
*Dieses Kanji...
gibt es nicht.*



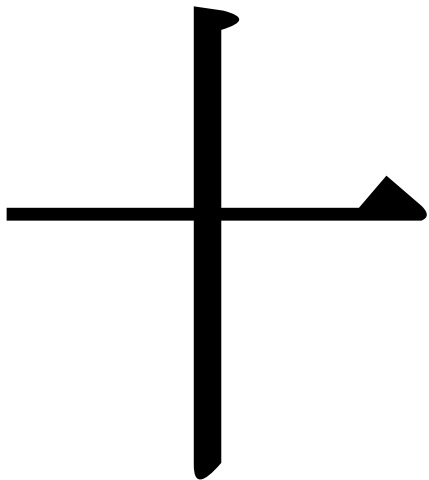
Natürliche Zahlen für Anfänger

Natürliche Zahlen für Anfänger

Natürliche Zahlen für Anfänger



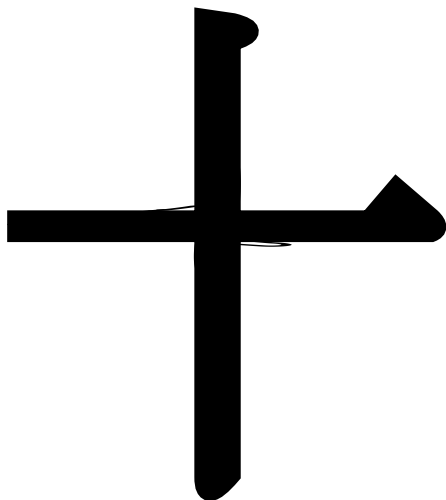
Natürliche Zahlen für Anfänger



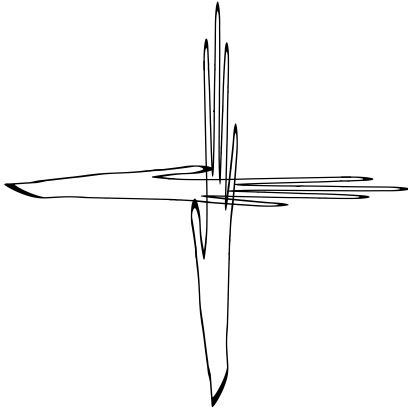
Methode:
*Elementare
Bilderkennung*



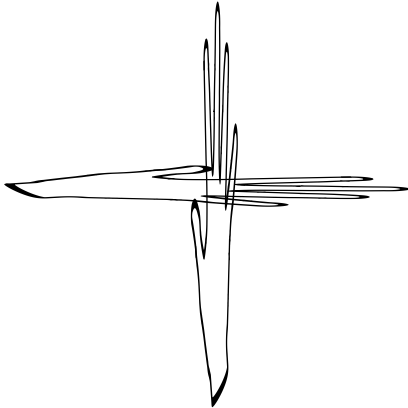
Natürliche Zahlen für Anfänger



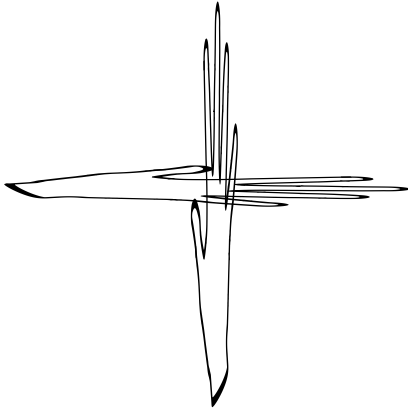
Natürliche Zahlen für Anfänger



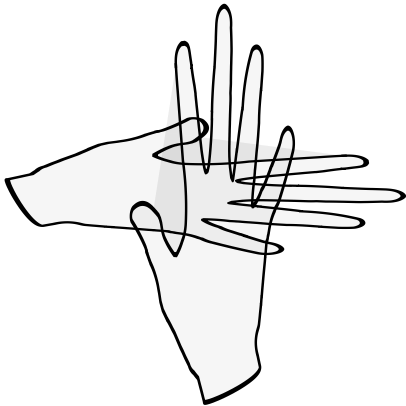
Natürliche Zahlen für Anfänger



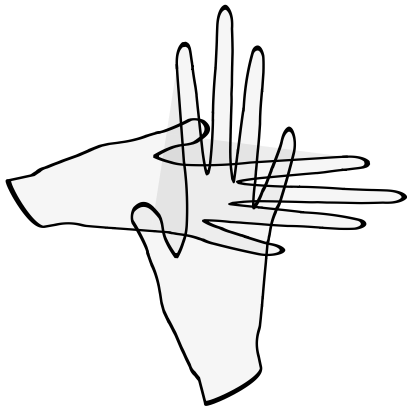
Natürliche Zahlen für Anfänger



Natürliche Zahlen für Anfänger



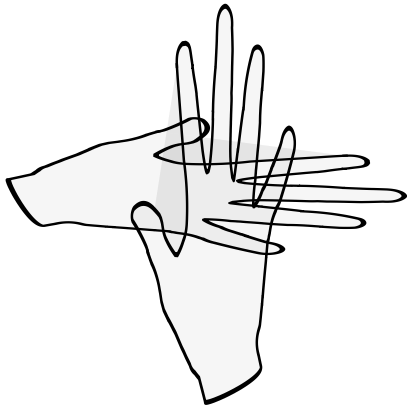
Natürliche Zahlen für Anfänger



10



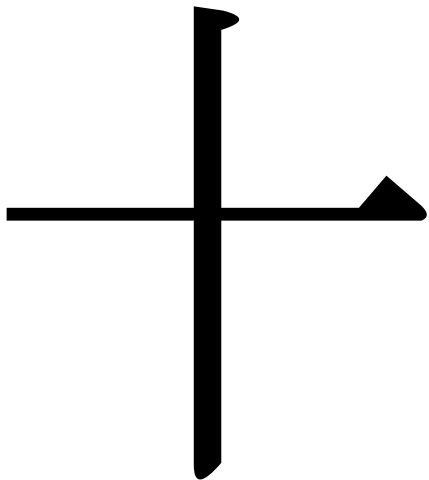
Natürliche Zahlen für Anfänger



10



Natürliche Zahlen für Anfänger



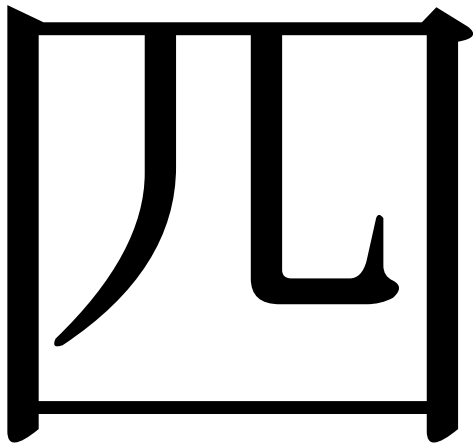
10



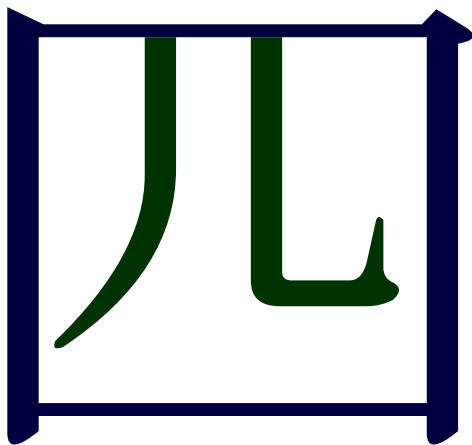
Natürliche Zahlen für Fortgeschrittene

Natürliche Zahlen für Fortgeschrittenen

Natürliche Zahlen für Fortgeschrittene

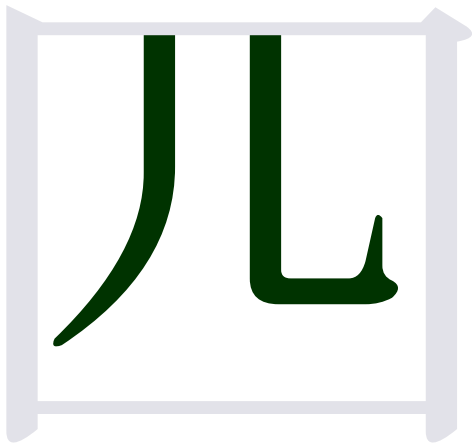


Natürliche Zahlen für Fortgeschrittene

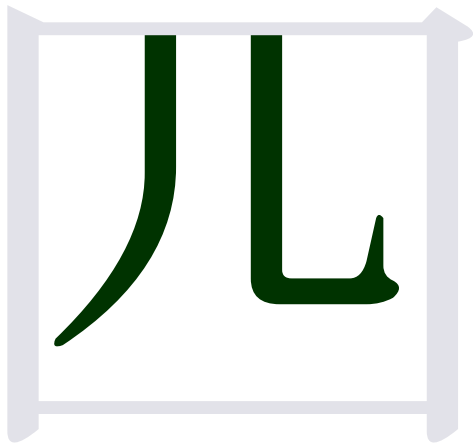


Methode:
Rechnen 7. Klasse

Natürliche Zahlen für Fortgeschrittene



Natürliche Zahlen für Fortgeschrittene



Zwei (Stäbchen?)



Natürliche Zahlen für Fortgeschrittene



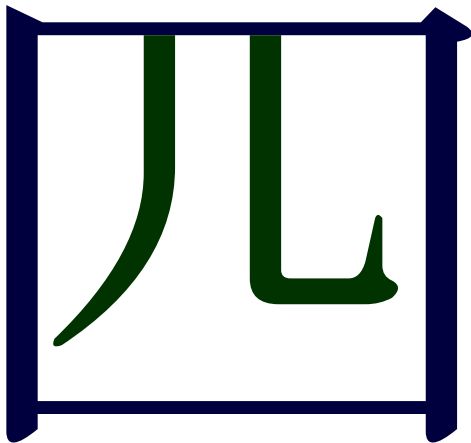
?

Natürliche Zahlen für Fortgeschrittene



Quadrat

Natürliche Zahlen für Fortgeschrittene

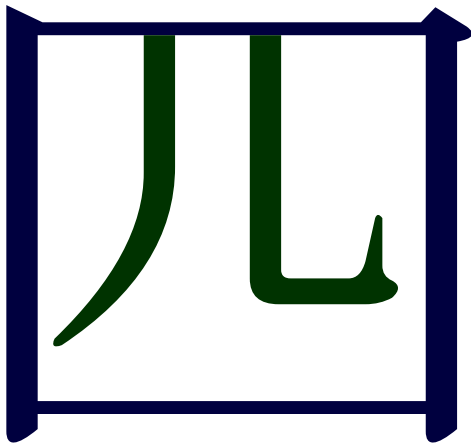


Zwei (Stäbchen?)

Quadrat



Natürliche Zahlen für Fortgeschrittene

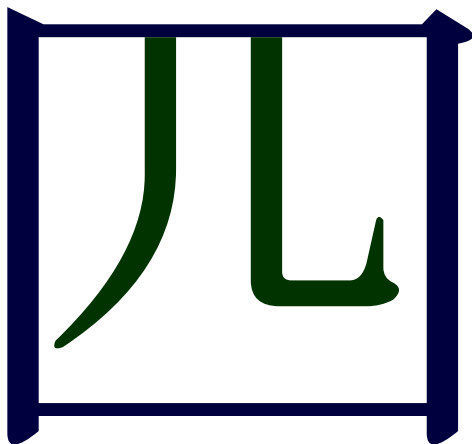


Zwei (Stäbchen?)

?

Quadrat

Natürliche Zahlen für Fortgeschrittene



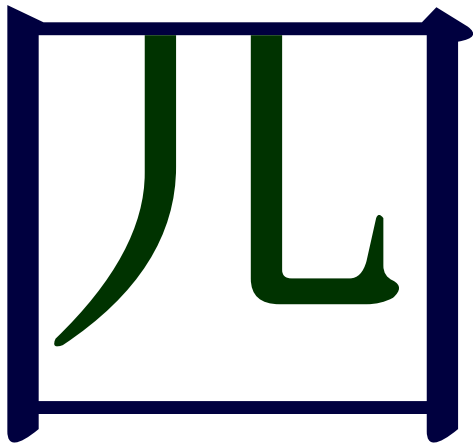
Zwei (Stäbchen?)

im

Quadrat

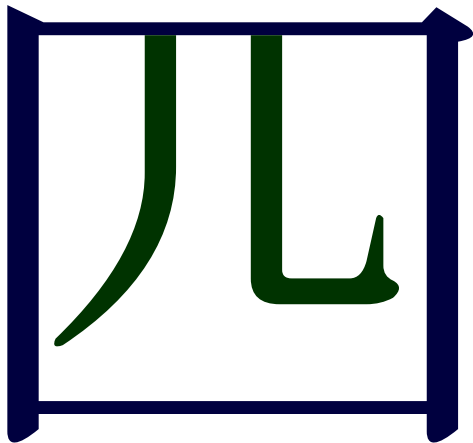


Natürliche Zahlen für Fortgeschrittene



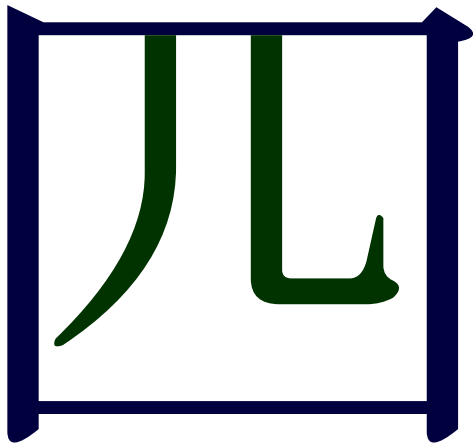
$$2^2$$

Natürliche Zahlen für Fortgeschrittene



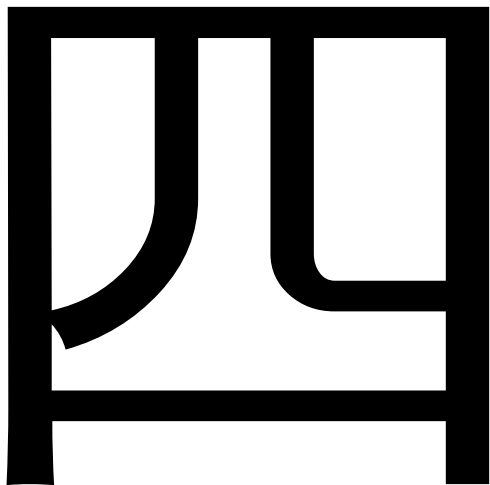
$$2^2 =$$

Natürliche Zahlen für Fortgeschrittene



$$2^2 = 4$$

Natürliche Zahlen für Fortgeschrittene



4

四

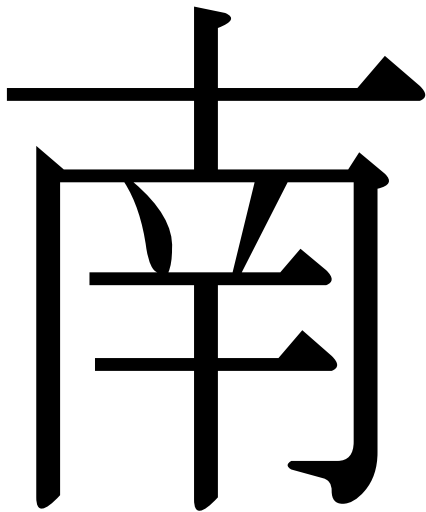
Orientierung im Freien

Orientierung im Freien

南

南

Orientierung im Freien



Methode:
Blöde Reime

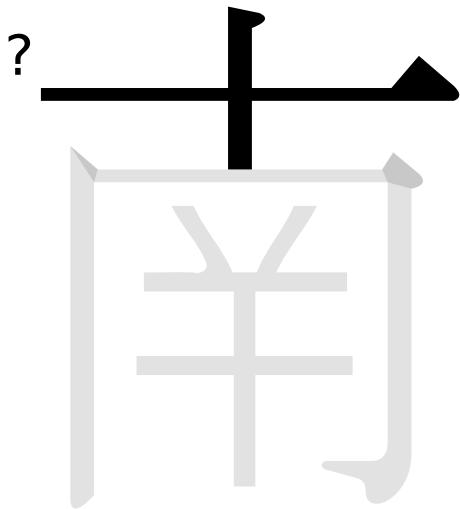
南

Orientierung im Freien

南

南

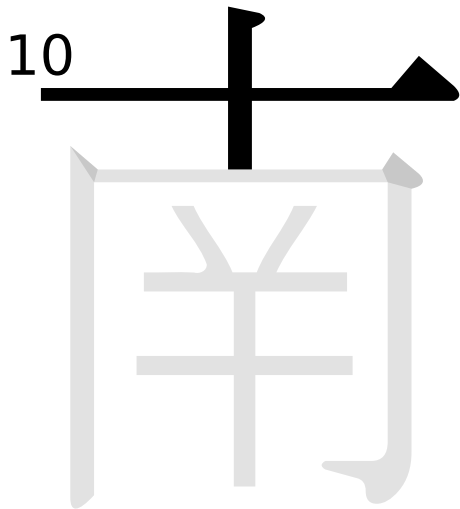
Orientierung im Freien



With

南

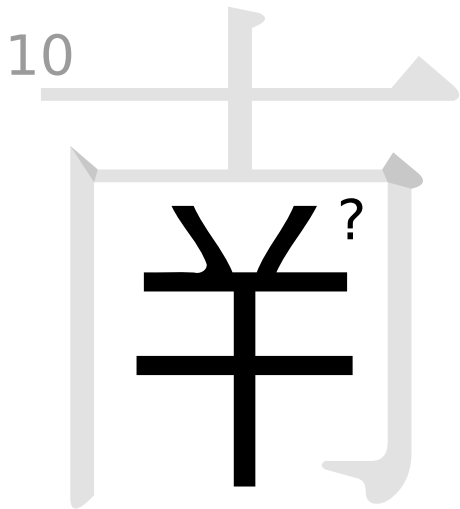
Orientierung im Freien



With *10*

南

Orientierung im Freien

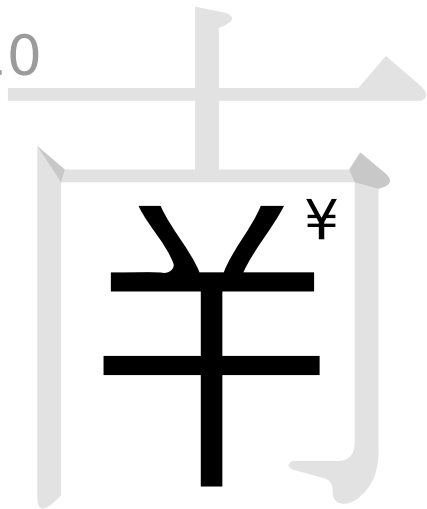


With *10*

南

Orientierung im Freien

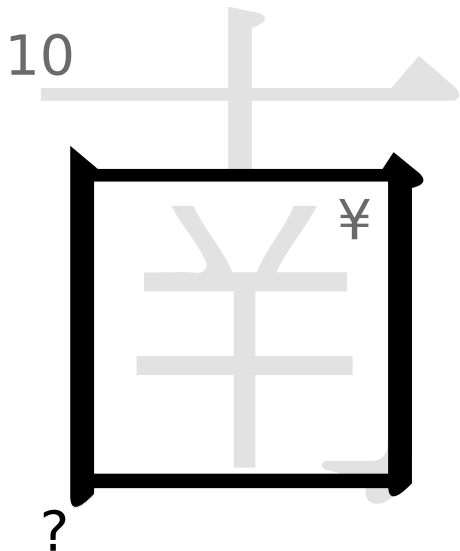
10



With *10 Yen*

南

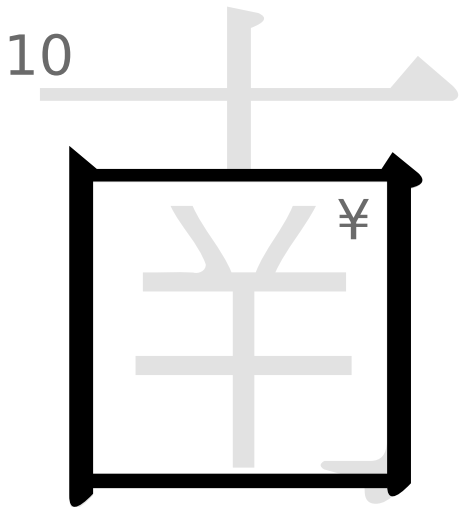
Orientierung im Freien



With *10 Yen*
inside ...

南

Orientierung im Freien

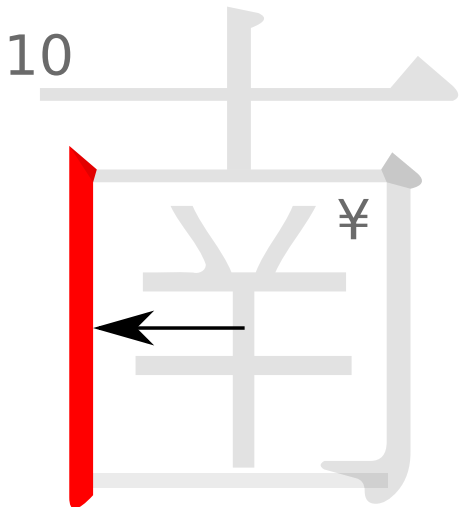


With **10 ¥en**
inside your *mouth*

Mouth

南

Orientierung im Freien



Mouth

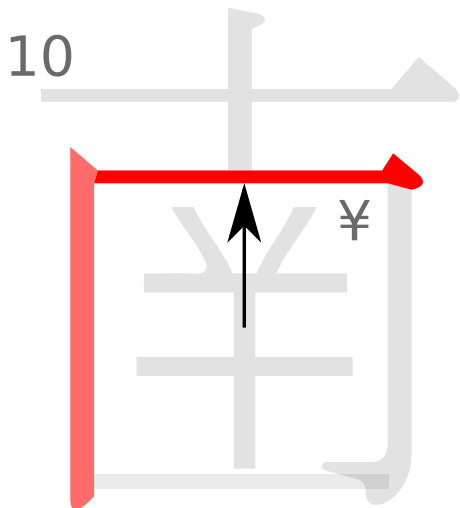
With *10 Yen*

inside your *mouth*

You can't go west

南

Orientierung im Freien



With *10 ¥en*

inside your *mouth*

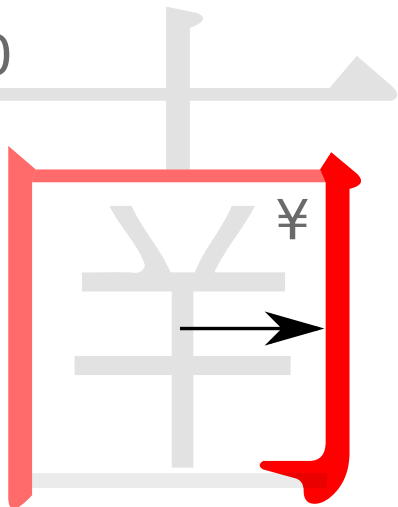
You can't go west,
You can't go north

Mouth

南

Orientierung im Freien

10



Mouth

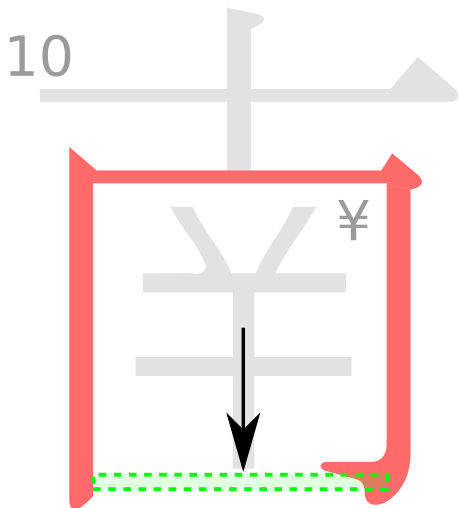
With *10 ¥en*

inside your *mouth*

You can't go west,
You can't go north,
You can't go east.

南

Orientierung im Freien



With **10 Yen**

inside your *mouth*

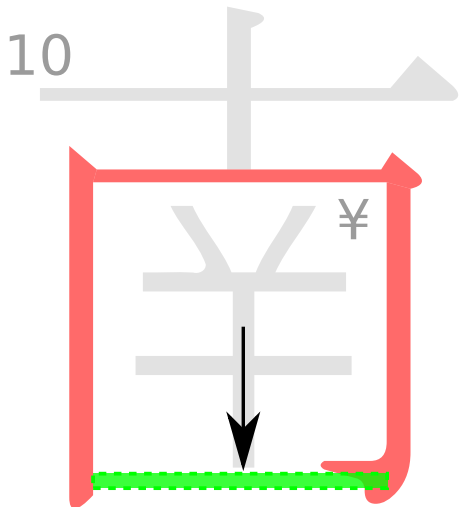
You can't go west,
You can't go north,
You can't go east.

You should head

Mouth

南

Orientierung im Freien



Mouth

With *10 Yen*

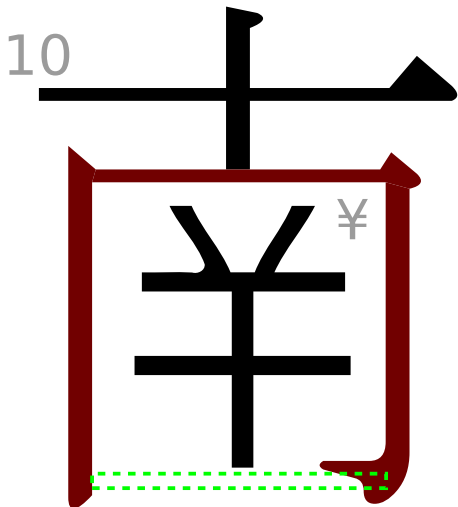
inside your *mouth*

You can't go west,
You can't go north,
You can't go east.

You should head

南

Orientierung im Freien



Mouth

With *10 ¥en*

inside your *mouth*

You can't go west,
You can't go north,
You can't go east.

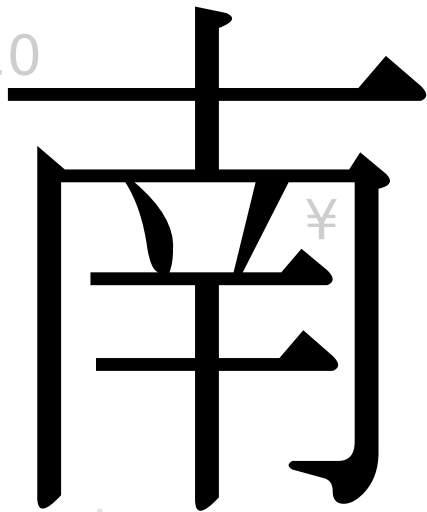
You should head

south!

南

Orientierung im Freien

10



Mouth

With *10 ¥en*

inside your *mouth*

You can't go west,
You can't go north,
You can't go east.

You should head

south!

南

Farbenlehre

赤

赤

赤

Methode:
*Rechnen für
Fortgeschrittene*

赤

赤

赤

十 丁

赤

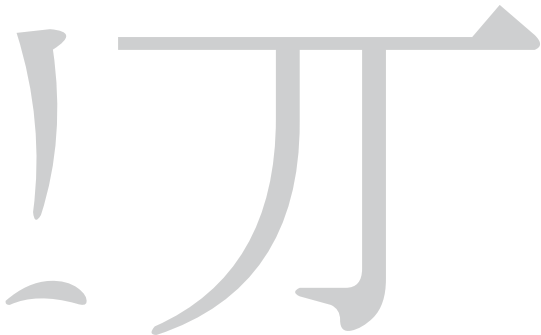
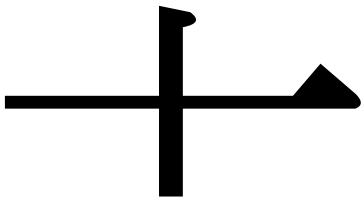
十 丁

赤

十 一 丁

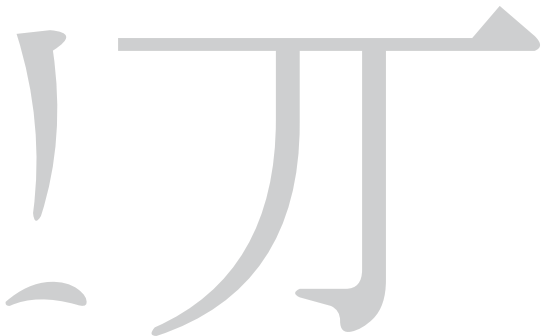
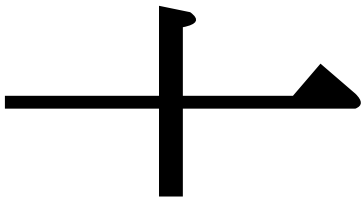
赤

?



赤

10

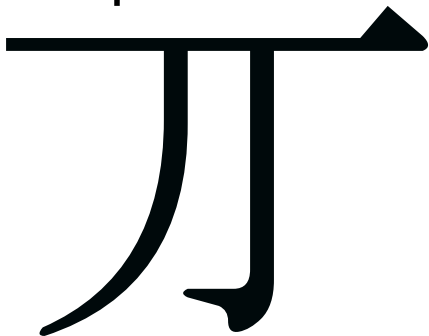


赤

10



?



赤

10



π



赤

10



?



π



赤

10



Faktorial

π



赤

¹⁰
十
!
Faktorial
^π
π

$$= 10! \cdot \pi$$

赤

¹⁰
十
!
Faktorial
^π
丁

$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \end{aligned}$$

赤

¹⁰
十
!
Faktorial
^π
π

$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \end{aligned}$$

赤

¹⁰
十
!
Faktorial
^π
π

$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \\ &= 10886400 \end{aligned}$$

赤

¹⁰
十
!
Faktorial
^π
π

$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \\ &= 10886400 \\ &= 0xa61d00 \end{aligned}$$

赤

¹⁰
十
!
Faktorial
^π
π



$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \\ &= 10886400 \\ &= \text{0xa61d00} \end{aligned}$$

赤

$10!$
Faktorial π



$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \\ &= 10886400 \\ &= \text{0xa61d00} \\ &\approx \text{0xff2800} \\ &\quad \text{(Luminanzausgleich)} \end{aligned}$$

赤

¹⁰
十
!
Faktorial
^π
π



$$\begin{aligned} &= 10! \cdot \pi \\ &= 3628800 \cdot \pi \\ &\approx 3628800 \cdot 3 \\ &= 10886400 \\ &= \text{0xa61d00} \\ &\approx \text{0xff2800} \\ &= \text{Rot} \end{aligned}$$

赤

赤

¹⁰十[!]万^π
Faktorial

Rot!

赤

Credits

Thanks for your attention

Thanks for your attention

Kanji you have learned in 5 minutes:

言、酒、不、母、十、四、南、赤

Brought to you by

Dimitri Popov

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

TCP-Tunnel durch Proxies

Wie man nur über Port 80 durch böse Proxies
durchkommt

Hintergrund

Ich schreibe meine Diplomarbeit bei einem Unternehmen, das aus Angst vor bösen rechtlichen Folgen oä. das Internet total blockiert.

- Proxy
 - Mit Anmeldung
 - Nur Port 80
 - (nicht mal https funktioniert richtig!)

Aber ich will VPN (Paper über Uni recherchieren) und SVN machen. Jabber ist auch ganz nett.

1. Schritt

Klassischer SSH-Tunnel

Hat aber 2,5 Probleme:

- Braucht nen Zielserver (hab ich)
 - Muss auch auf offenem Port (80) lauschen
- Geht nicht durch Proxy

2. Schritt

Corkscrew

„A Tool for Tunnelling SSH over HTTP Proxies“

http://wiki.kartbuilding.net/index.php/Corkscrew_-_ssh_over_https

Dann hat man einen lokalen Port der wie ein SOCKS Proxy arbeitet und über den man alles rausleiten kann.

Problem:

- Viel Software spricht nicht SOCKS
- tsocks ist manchmal echt unbequem

3. Schritt

Ein Tool um TCP-Traffic transparent über einen SOCKS Proxy zu leiten:

lcat

<https://github.com/tdrhq/tunnel-scripts/>

Problem:

- Man muss immer noch allen Programmen beibiegen über welchen Port sie gehen sollen

4. Schritt

Iptables

Eine NAT-Regel und jeder TCP-Traffic geht transparent nach draußen.

Zusammenfassend

- TCP über
- iptables NAT auf lokalen Port mit
- lcat auf lokalen Port mit
- SSH mit key-sign-in über
- Corkscrew auf Port 80 eines
- Proxy mit Anmeldung

Dauert zwar ne Minute oder zwei bis die Verbindung steht, ist aber erstaunlich schnell!

Probleme, nächste Schritte

- Iptables anpassen, dass lokaler Netztraffic nicht geNATed wird (Drucker, Lotus Notes, SMB)
- Starten des Skriptes braucht root-Rechte, hübscher wäre ohne (geht aber nicht wegen iptables)
- UDP auch machen (DNS)

Ein fertiges Skript

SSH-Schlüssel hinterlegen spart Tipparbeit und Zeit→ssh-copy-id

```
#/bin/sh
gksudo "iptables -t nat -F"
gksudo "iptables -t nat -A OUTPUT -p tcp ! -d 10.1.87.2 -o eth0 -j DNAT --to-destination 127.0.0.1:9000"
ssh -ND 1080 -p 80 vollkorn@ssh80.cryptobitch.de &
sleep 60
tsocks /home/jan/Dokumente/tunnel-scripts/lcat -p 9000 -t
killall tsocks
killall ssh
gksudo "iptables -t nat -F"
```

~/.ssh/config

```
Host *
ProxyCommand corkscrew 10.1.87.2 80 %h %p ~/.ssh/proxyauth
```

~/.ssh/proxyauth

ProxyUser:ProxyPW

Danke!

„Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen Deutschland“ in Version 3.0 (abgekürzt „CC-by-sa 3.0/de“)

<http://creativecommons.org/licenses/by-sa/3.0/de/>



Jan Girlich <3girlich@informatik.uni-hamburg.de>

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

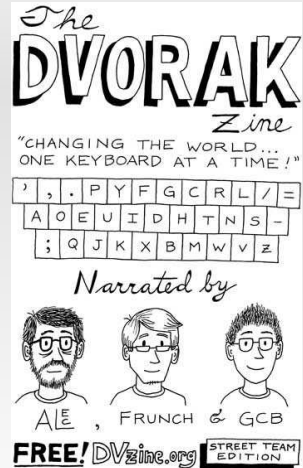
Ein für die deutsche Sprache optimiertes
Keyboard Layout

Neo2: Geschichte

Im Grunde ist Neo2 Dvorak für Deutsch, daher empfiehlt es sich die Story einfach mal zu lesen.

Hier ist sie ganz nett erzählt:

- <http://www.dvzine.org/zine/>



Neo2: Geschichte

Eckdaten und Lustiges:

- Untersuchung von Bigrammen (Buchstabenpaaren) um sie zu vermeiden
- Keine "1" geht ja auch ein "i"
- "typewriter" in der ersten Zeile hilft Verkauf
- Dvoraks Layout ist überlegen, hat sich aber nie gegen Unternehmen durchsetzen können und er starb verbittert

Neo2: Wozu überhaupt?

- Leichter zu lernen (insbesondere blind)
- Schnelleres Tippen (abwechseln der Finger)
- Einfacheres Tippen (linke Hand entlasten)
- Ergonomischer → ¬ Sehnscheidenentzündung
- Mehr Symbole direkt erreichbar
- (man lernt es zusätzlich und kann natürlich noch das alte QWERTZ "tippen")

Neo2: So sieht es aus



Neo2: und so (Ebene 2)



Neo2: und so (Ebene 3)



Neo2: und so (Ebene 4)



Neo2: und so (Ebene 5)



Neo2: und so (Ebene 6)



Neo2: Wie anfangen?

- Ubuntu: System→Einstellungen→Tastatur
- `#sudo apt-get install ktouch`
- Zum nachsehen:
 - Layout am Bildschirm OSD Neo2:
<http://www.mzuther.de/de/contents/osd-neo2>
 - Ausdruckbarer Aufsteller:
<http://neo-layout.org/grafik/aufsteller/neo20-aufsteller.png>
- Hilfe, Community, mehr:
<http://www.neo-layout.org/>

Neo2: Probleme

- Tippfehler beim Hinschauen
 - Es gibt ein paar wenige Neo2 Keyboards
 - Besser gleich blank machen:



Neo2: Zukunft

- Eigene Hardware:
<http://wiki.neo-layout.org/wiki/Hardwareentwicklung>
- Evolutionäre Algorithmen zur weiteren Optimierung der "Tippkosten":
<http://wiki.neo-layout.org/wiki/EvolvedLayouts>

Danke!

„Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen Deutschland“ in Version 3.0 (abgekürzt „CC-by-sa 3.0/de“)

<http://creativecommons.org/licenses/by-sa/3.0/de/>



Jan Girlich <3girlich@informatik.uni-hamburg.de>

Ablauf

7fietkau - Tunneln mit sshuttle:

Ein VPN für Arme über SSH und seine Möglichkeiten für Hamburger Informatiker.

7licina - Elitäre Windowmanager:

awesome, xmonad, und mehr tolle Möglichkeiten, Fenster hin- und herzuschieben.

0rokita - Spionage im Web:

Klauen von History.

7fietkau - Ein C-Puzzler:

Eine Quizfrage zu einem Stückchen C-Code in der Tradition der bekannten Java-Quizzes.

6simon - The Origin of Kanji:

The ideas behind some Japanese glyphs.

6popov - Japanisch für Informatiker:

Unverkennbar bescheuerte Eselsbrücken.

3girlich - transparente TCP-Tunnel durch Proxies:

Wie man nur über Port 80 durch böse Proxies durchkommt.

3girlich - Neo2:

Kurze Geschichte und aktueller Entwicklungsstand von deutschen Keyboard-Designs.

8pridoeh - Breaking CAPTCHAs:

Wie man einfache CAPTCHAs mit einfachen Methoden automatisiert lösen kann.

Breaking CAPTCHAs

Wie man einfache CAPTCHAs mit einfachen Methoden
automatisiert lösen kann

Henning (8pridoeh)

December 2, 2010



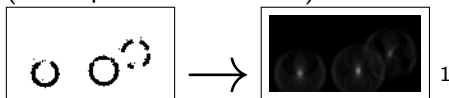
Hintergrund entfernen

- ▶ b_{orig} sei eine Funktion $\mathbb{N} \times \mathbb{N} \times \{r, g, b\} \rightarrow \{0, \dots, 255\}$
- ▶ b_{ref} sei ein Referenzbild, definiert wie b_{orig}
- ▶ s sei ein Schwellwert, klein gewählt (z. B. 20)
- ▶ $b_{clean}(x, y) = \begin{cases} 255 & \text{falls } \sum_{c \in \{r, g, b\}} |b_{orig}(x, y, c) - b_{ref}(x, y, c)| < s \\ 0 & \text{sonst} \end{cases}$
- ▶ Bei mehreren Referenzbildern: Berechnung fuer alle durchfuehren, waehle das b_{clean} mit der kleinsten Summe (Grauwerte addiert)
- ▶ **Ergebnis:** Nur noch schwarze Kreise auf dem Bild



Hough transform

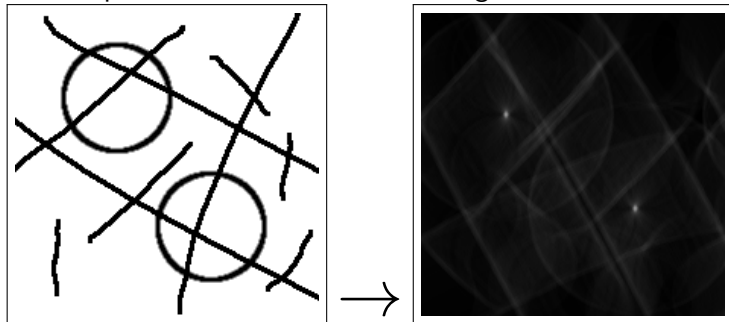
- ▶ Kreisgleichung: $r = (x - a)^2 + (y - b)^2$
- ▶ $(a|b)$ sei Kreismittelpunkt (Parameter fuer Hough-Raum), $(x|y)$ sei ein Bildpunkt, r ein *fester* Radius (hier 10 Pixel)
- ▶ Verfahren zur Erstellung des Hough-Raums:
 1. Erstelle eine Matrix M in Bildgroesse $m \times n$ und initialisiere diese mit 0.
 2. Fuer jeden Bildpunkt und jeden Parameter a, b der Gleichung: Inkrementiere $M_{a,b}$ um 1.
- ▶ **Ergebnis:** Hough-Raum in dem man die Haeufungspunkte (Mittelpunkte der Kreise) sieht



¹Werte im Hough-Raum mit 3 multipliziert zur Visualisierung

Hough transform – Groesseres Beispiel

Kein Captcha – Nur zur Verdeutlichung



Suche von Haeufungspunkten im Hough-Raum

- ▶ Verfahren zur Suche Haeufungspunkten im Hough-Raum:
 1. Suche in M einen unmarkierten Punkt p , der ueber einem Schwellwert s (hier 25%) liegt.
 2. Ermittle alle Punkte die von p eine Distanz kleiner d (hier 5 Pixel) haben und ueber s liegen.
 3. Berechne den Mittelpunkt m von p und den ermittelten Punkten und deren Durchschnitt der Werte w .
 4. Markiere p und ermittelte Punkte
 5. Falls noch unmarkiere Punkte: Gehe zu 1. Sonst: Gib Haeufungspunkte mit Gewicht zurueck
- ▶ **Ergebnis:** Haeufungspunkte mit einem Gewicht w .
Geringstes w ist (meist) der unterbrochene Kreis



Mission accomplished :-)

Danke!

Folien:

https://mafiasi.de/KunterBuntesSeminar-WS10/Lightning_Talks

Nächster Lightning-Talk-Termin: **21.01.11**

Hat's euch heute gefallen? Haltet selbst einen Talk. :)

Weiterhin gesucht:

Hosts und Referenten für die kommenden Wochen!

<https://mafiasi.de/KBS>