Vorkurs in Informatik

Eine Einführung ins Informatikstudium an der Universität Zürich

Emanuel Giger, Giacomo Ghezzi, Michael Würsch, and Harald Gall

University of Zurich, Switzerland

University of Zurich
Department of Informatics

s.e.a.l.
software evolution & architecture lab

Zielsetzung

Sanfter Einstieg ins Studium der Informatik.

Kleinsten gemeinsamen Nenner schaffen.

Socializing.

Ablauf

1. Tag: Grundlagen
2. Tag: Software Engineering by Example
3. Tag: Einführung in die Programmierung

Ablauf: Tag 1

09:30 bis 12:00

- Was ist ein Computer?
- Wie ist ein Computer aufgebaut?
- Das Rechnen mit Wahrheitswerten
- Zahlensysteme
- Wie bringe ich den Computer dazu, für mich Probleme zu lösen?

13:00 bis 16:00

- Eine Einführung in die Programmierung mit Scratch

Ablauf: Tag 2

09:30 bis 12:00

- Eine Einführung in das systematische Entwickeln von Software (aka. Software Engineering)
- Beginn Gruppenarbeiten: Ein kleines eigenes Projekt mit Scratch

13:00 bis 16:00

- Fortsetzung vom Morgen

Ablauf: Tag 3

09:30 bis 12:00

- Kurzpräsentationen der Gruppenarbeiten vom Vortag
- Eine Einführung in die Programmierung mit Groovy

13:00 bis 16:00

- Fortsetzung vom Morgen

Das Institut für Informatik (ifi)

s.e.a.l.
software evolution & architecture lab

CSG Communication Systems Group

VMML
Surfing and snowboarding lab
university of zürich

DIS Dynamic and Distributed Information Systems

Information Management Research Group

ai lab

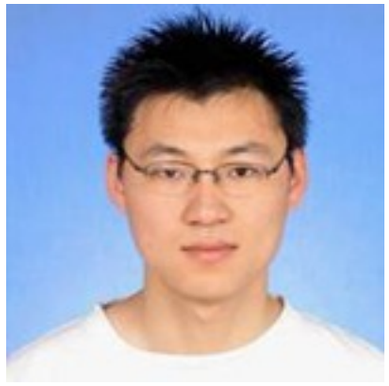
RERG IFI REQUIREMENTS ENGINEERING RESEARCH GROUP

Database Technology Research Group

IFI INFORMATICS AND SUSTAINABILITY GROUP

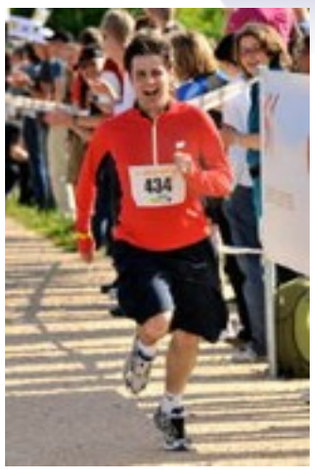
Institut für Computerlinguistik

Wer sind wir?



s.e.a.l.

software evolution & architecture lab



Was sind unsere Forschungsschwerpunkte?

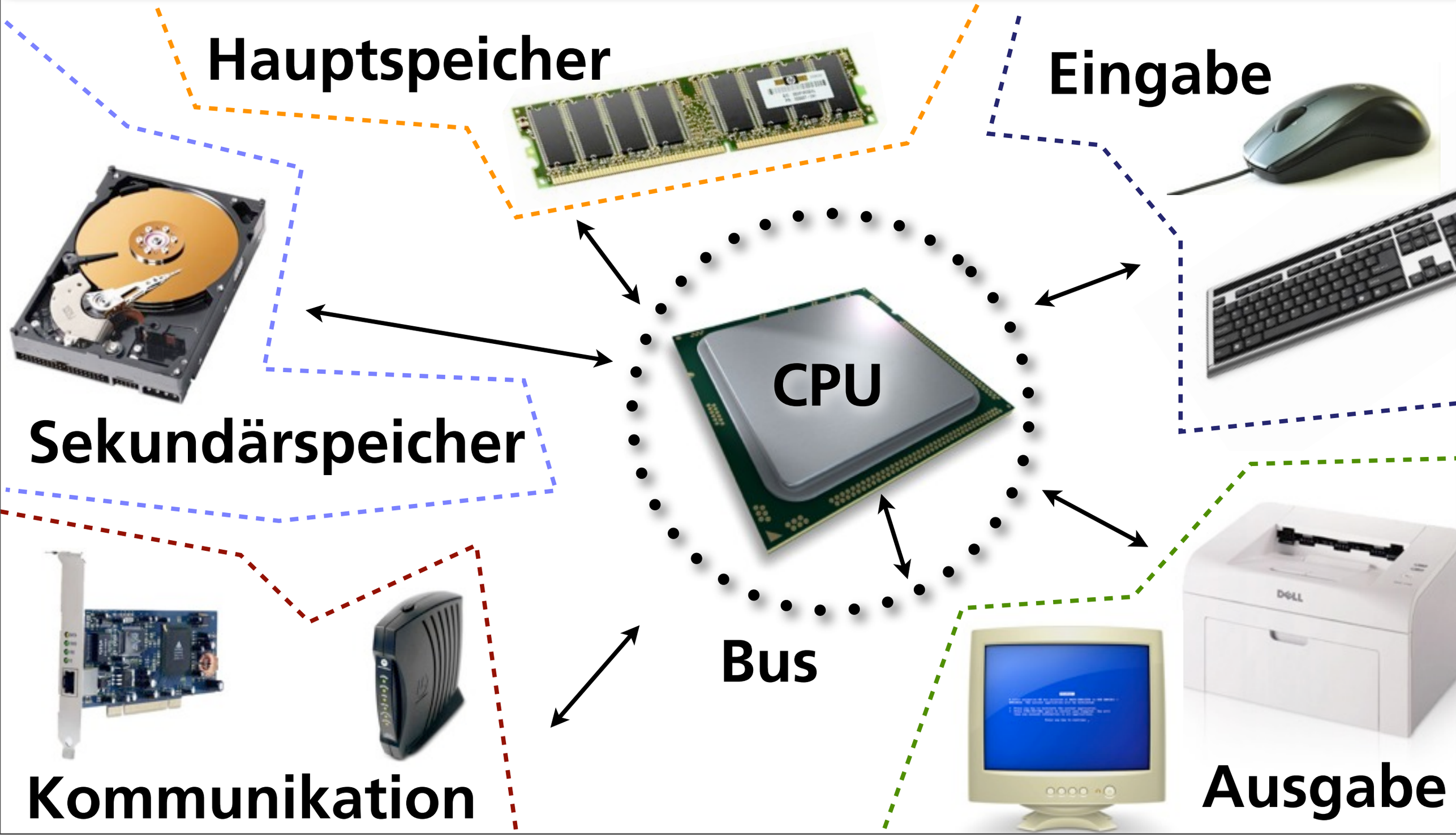
- Software Evolution
- Software Wartung und Reengineering
- Software Architekturen
- Produkt-/Programmfamilien
- Verteilte Software Engineering Prozesse
- Methodologien und Paradigmen für Software Entwicklung
- Semantic Web Engineering & Recommender Systems

Programmieren in der Assessment-/Bachelorstufe

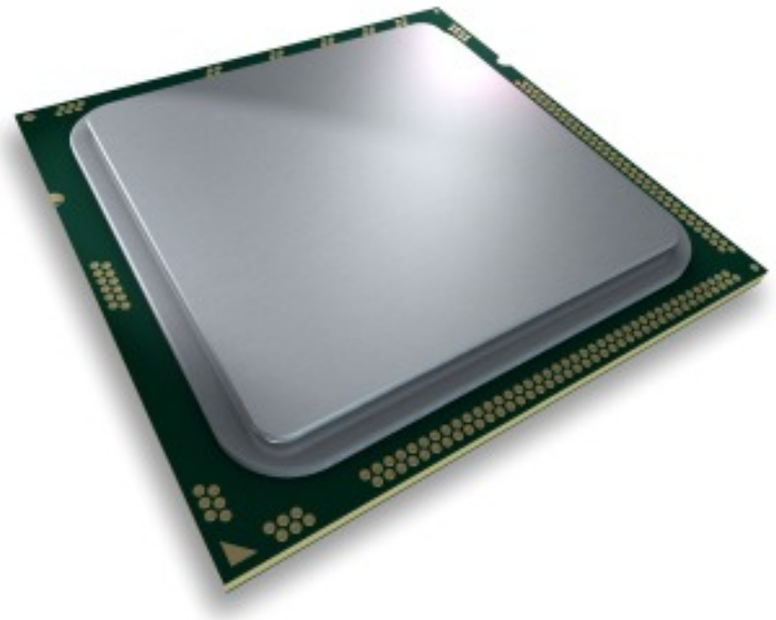
1. Semester: Einführung in die Programmierung (Informatik I, Java)
2. Semester: Algorithmen und Datenstrukturen (C++)
3. Semester: ??
4. Semester: Software Praktikum (aka. SoPra, Java)

Was ist ein Computer?

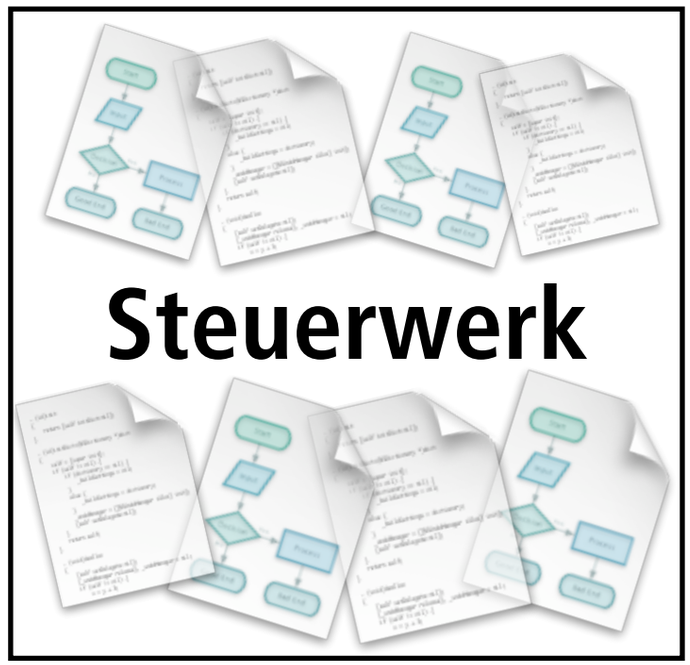
Wie ist ein Computer aufgebaut?

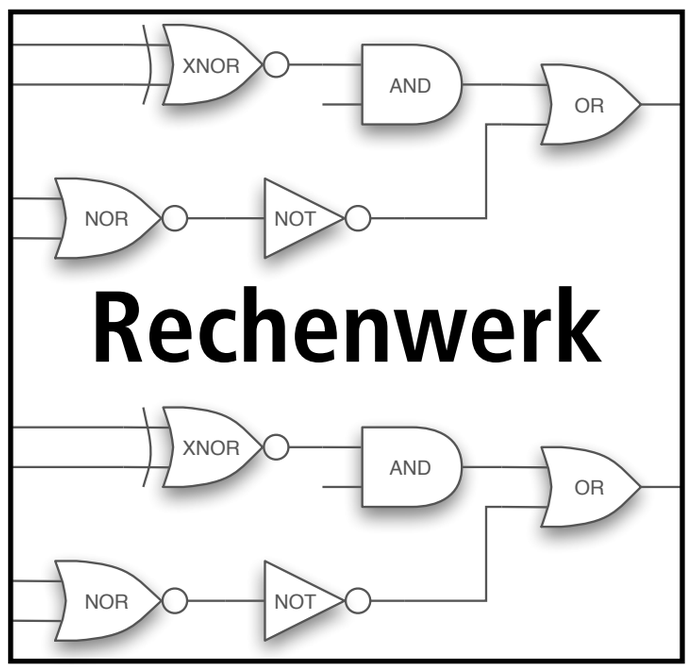


(C)entral (P)rocessing (U)nit



=





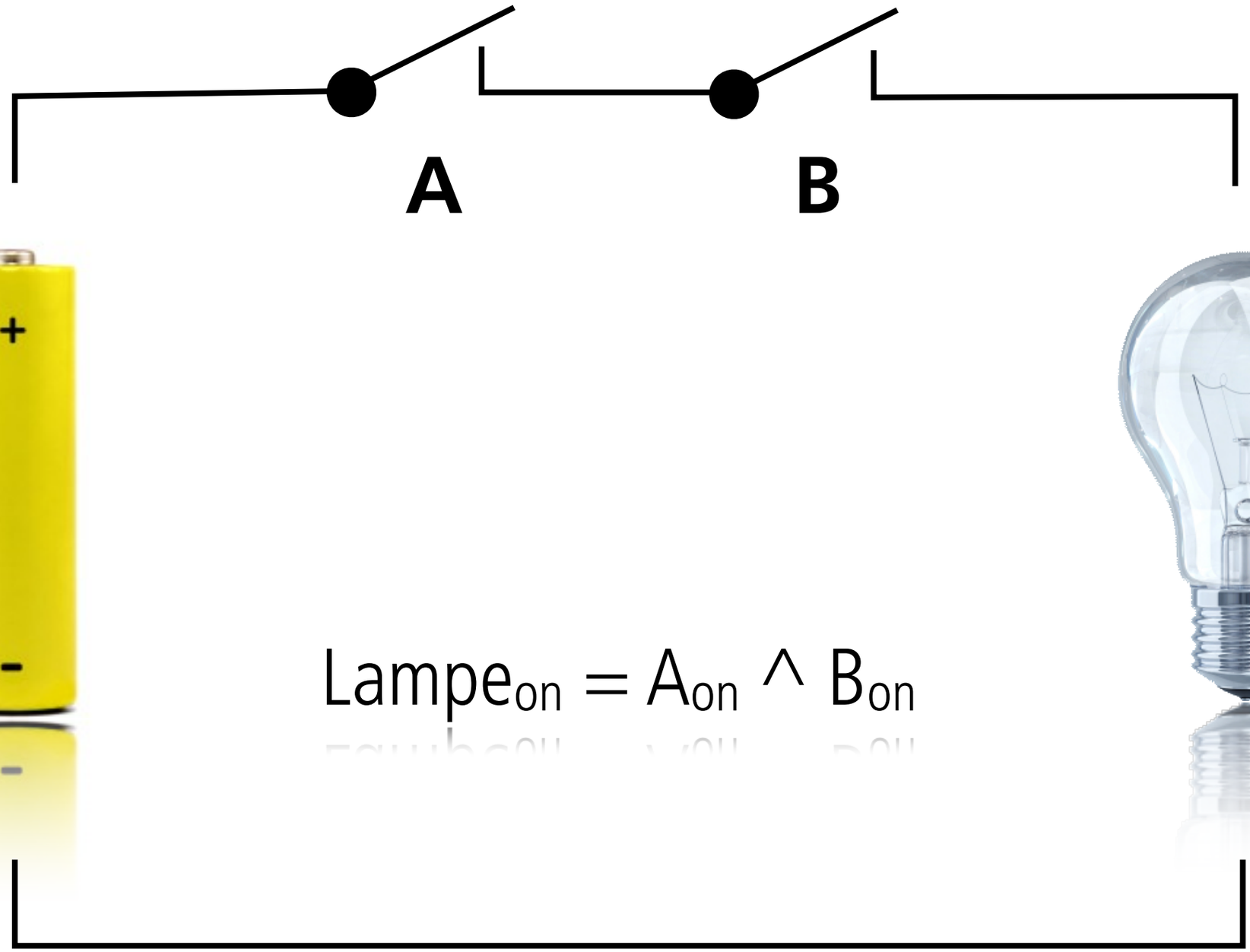
Bool'sche Algebra

wahr oder falsch?

Bsp: Eine Bestellung kann nur in Auftrag gegeben werden, wenn der Kunde bereits ein Kundenkonto besitzt und eine gültige Kreditkarte angegeben hat.

$Bestellung_{ok} = Konto_{Existiert}$ und $Kreditkarte_{gültig}$

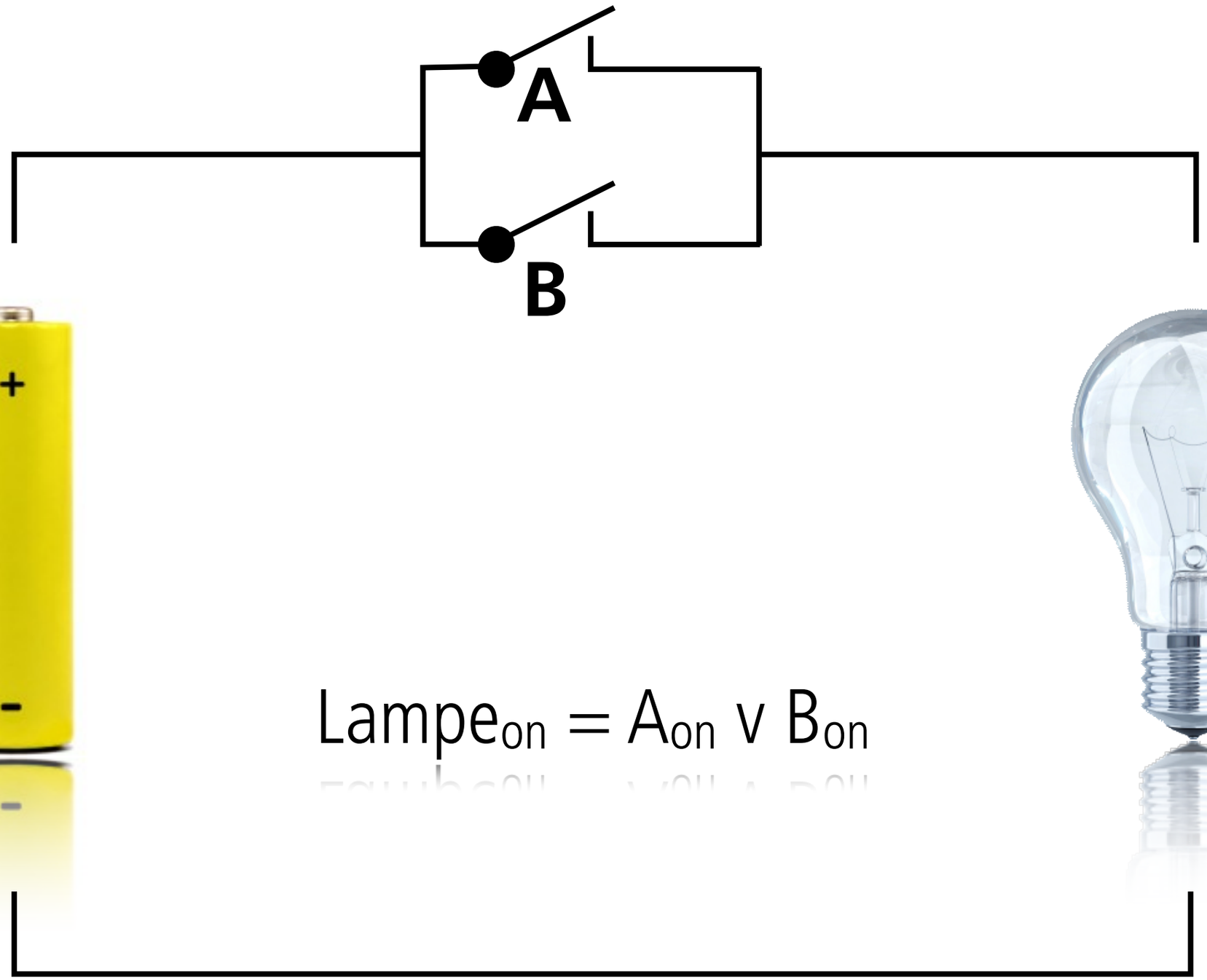
Bool'sche Algebra: AND



Wahrheitstabelle: AND

AND	T	F
T	T	F
F	F	F

Bool'sche Algebra: OR



Wahrheitstabelle: OR

OR	T	F
T	T	T
F	T	F

Aufgabe: Bool'sche Ausdrücke

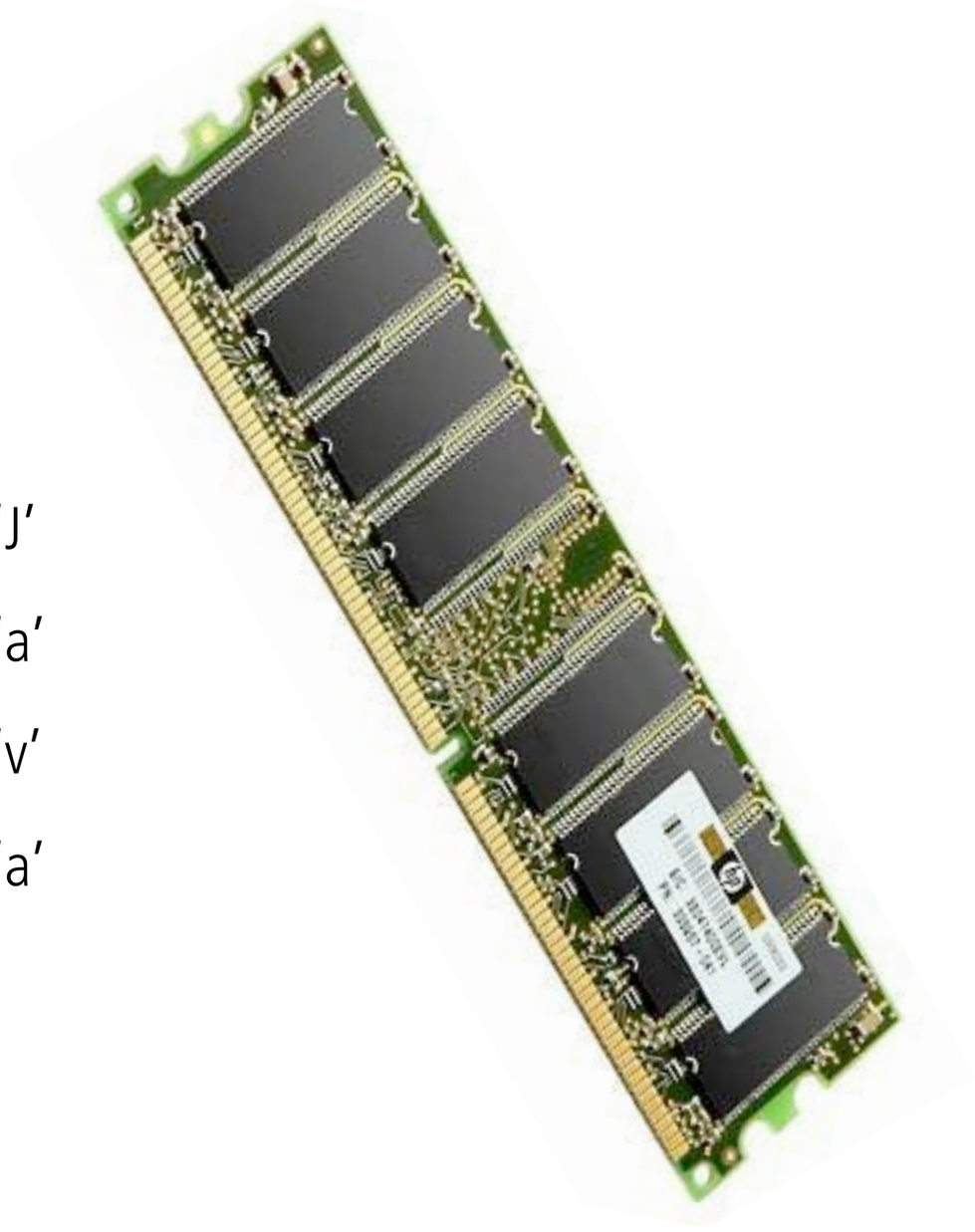
Wann sind die folgenden Ausdrücke wahr?

(A v B) ^ B

(A v B) ^ ¬B

Hauptspeicher

•	•	
•	•	
•	•	
2000	01001010	Kodierung für Buchstabe ‘J’
2001	01100001	Kodierung für Buchstabe ‘a’
2002	01110110	Kodierung für Buchstabe ‘v’
2003	01100001	Kodierung für Buchstabe ‘a’
2004	00000011	Kodierung für Zahl 3



Zahlensysteme

Dezimal

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Binär

0, 1

Hexadezimal

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A , B, C, D, E, F

Konversion: Dezimal nach Binär

Beispiel: 123_{10}

Quotient →	61	30	15	7	3	1	0
2 |	123	61	30	15	7	3	1
	122	60	30	14	6	2	0
Rest →	1	1	0	1	1	1	1

Leserichtung

Resultat: 1111011_2

Aufgabe: Dezimal nach Binär

Aufgabe: 17_{10} nach Binär

Aufgabe: Dezimal nach Binär

Aufgabe: 17_{10} nach Binär

Quotient →	8	4	2	1	0
	$2\overline{)17}$	$2\overline{)8}$	$2\overline{)4}$	$2\overline{)2}$	$2\overline{)1}$
	16	8	4	2	0
Rest →	1	0	0	0	1

Aufgabe: Dezimal nach Binär

Aufgabe: 17_{10} nach Binär

Quotient →	8	4	2	1	0
	$2 \overline{)\ 17}$	$2 \overline{)\ 8}$	$2 \overline{)\ 4}$	$2 \overline{)\ 2}$	$\overline{)\ 1}$
	16	8	4	2	0
Rest →	1	0	0	0	1

Resultat: 10001_2

Konversion: Binär nach Dezimal

Beispiel: 1111011_2

	1	1	1	1	0	1	1
Sum:	1×2^6	1×2^5	1×2^4	1×2^3	0×2^2	1×2^1	1×2^0
Sum:	64	32	16	8	0	2	1 → 123

Resultat: 123_{10}

Aufgabe: Binär nach Dezimal

Aufgabe: 10101_2 nach Dezimal

Aufgabe: Binär nach Dezimal

Aufgabe: 10101_2 nach Dezimal

	1	0	1	0	1	
Sum:	1×2^4	0×2^3	1×2^2	0×2^1	1×2^0	
Sum:	16	0	4	0	1	→ 21

Aufgabe: Binär nach Dezimal

Aufgabe: 10101_2 nach Dezimal

	1	0	1	0	1	
Sum:	1×2^4	0×2^3	1×2^2	0×2^1	1×2^0	
Sum:	16	0	4	0	1	→ 21

Resultat: 21_{10}

Addieren von Binärzahlen

	1	1	1	1	0	1	1
+	1	0	1	1	0	1	1
1	1	0	1	0	1	1	0

Addieren von Binärzahlen

	1	1	1	1	0	1	1
+	1	0	1	1	0	1	1
1	1	0	1	0	1	1	0

Kann zu einem Überlauf führen!

Aufgabe: Addition

	1	0	1	1
+	1	0	1	1

Aufgabe: Addition

Aufgabe: Addition

	I	0	I	I
+	I	0	I	I
I	I	I	I	
I	0	I	I	0

Sekundärspeicher

Software

Gängige Vorurteile/ Ausreden

“Computer sind intelligent.”

“Der Computer ist abgestürzt.”

“Der Computer erlaubt das nicht.”

“Der Computer hat die Datei verloren/ kaputt gemacht.”

Computer...

...sind universelle Maschinen. Sie führen das Programm aus, das man ihnen eingibt.

Die guten Neuigkeiten:

1. *Der Computer wird genau das tun, was das Programm ihm vorschreibt.*
2. *Er wird es sehr schnell tun.*

Die schlechten Neuigkeiten:

1. *Der Computer wird genau das tun, was das Programm ihm vorschreibt.*
2. *Er wird es sehr schnell tun.*

Software schreiben ist schwierig

Programme stürzen ab

Programme, die nicht abstürzen, funktionieren aber auch nicht unbedingt richtig

Programmierer sind verantwortlich für das korrekte Funktionieren ihrer Programme

Beispiel: Ariane 5

4. Juni 1996: Millionen Dollar Schaden wegen einfachem Programmierfehler:

64-bit float -> 32-bit int

Weitere Beispiele

28. July, 1962 - Mariner I space probe.

Eine mit Bleistift niedergeschriebene Formel wird falsch abgeschrieben. Resultat: Die Sonde muss gesprengt werden.

1982 - Soviet gas pipeline.

Die CIA lies absichtlich einen Fehler in eine Steuerungssoftware einbauen. Resultat: Die grösste nicht-nukleare Explosion in der Geschichte der Menschheit.

1985-1987 - Therac-25 medical accelerator

Gerät zur Strahlentherapie dessen Betriebssystem von einem unerfahrenen Programmierer zusammengebastelt worden war. Resultat: Mindestens 5 Patienten sterben, mehrere werden schwer verletzt.

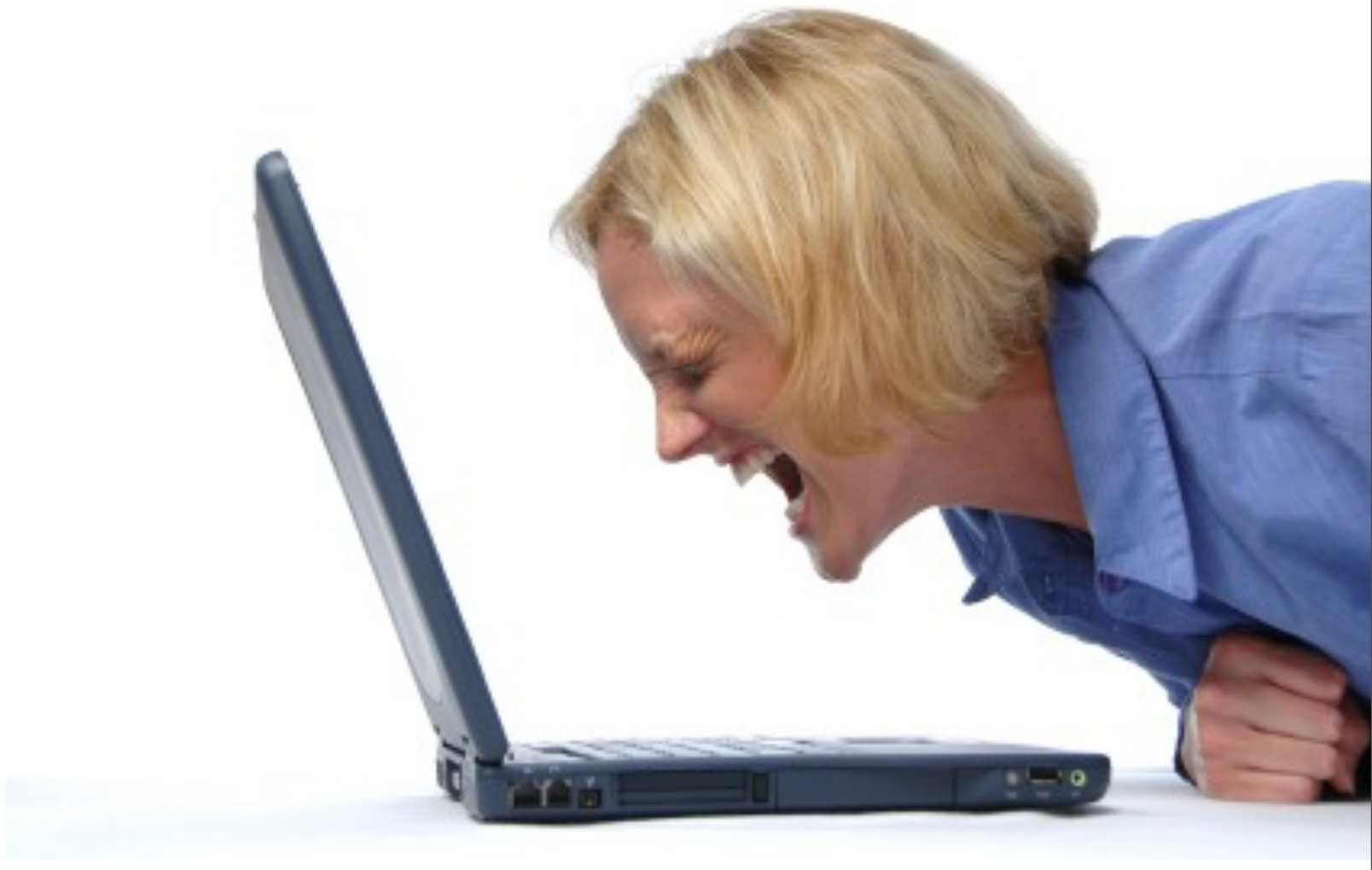
Software schreiben macht Spass

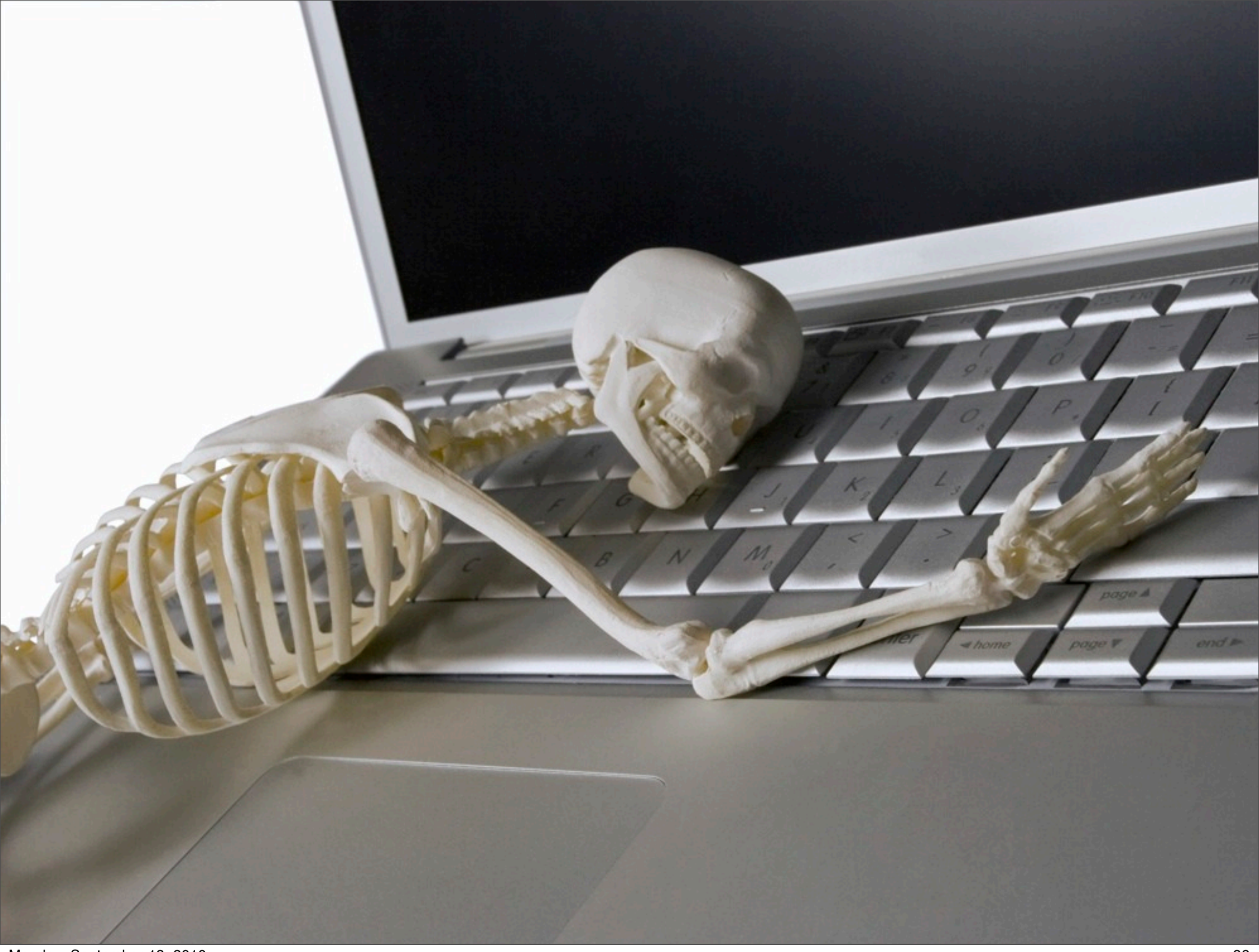
Ein Programmierer entwirft und baut eigene Maschinen

Er kann kreativ sein und seine Vorstellungskraft gebrauchen

Es ist faszinierend, wenn ein selbstgeschriebenes Programm läuft und vielleicht sogar seinen Benutzern den Alltag erleichtert

Wie bringe ich den Computer dazu,
das zu tun, was ich will?





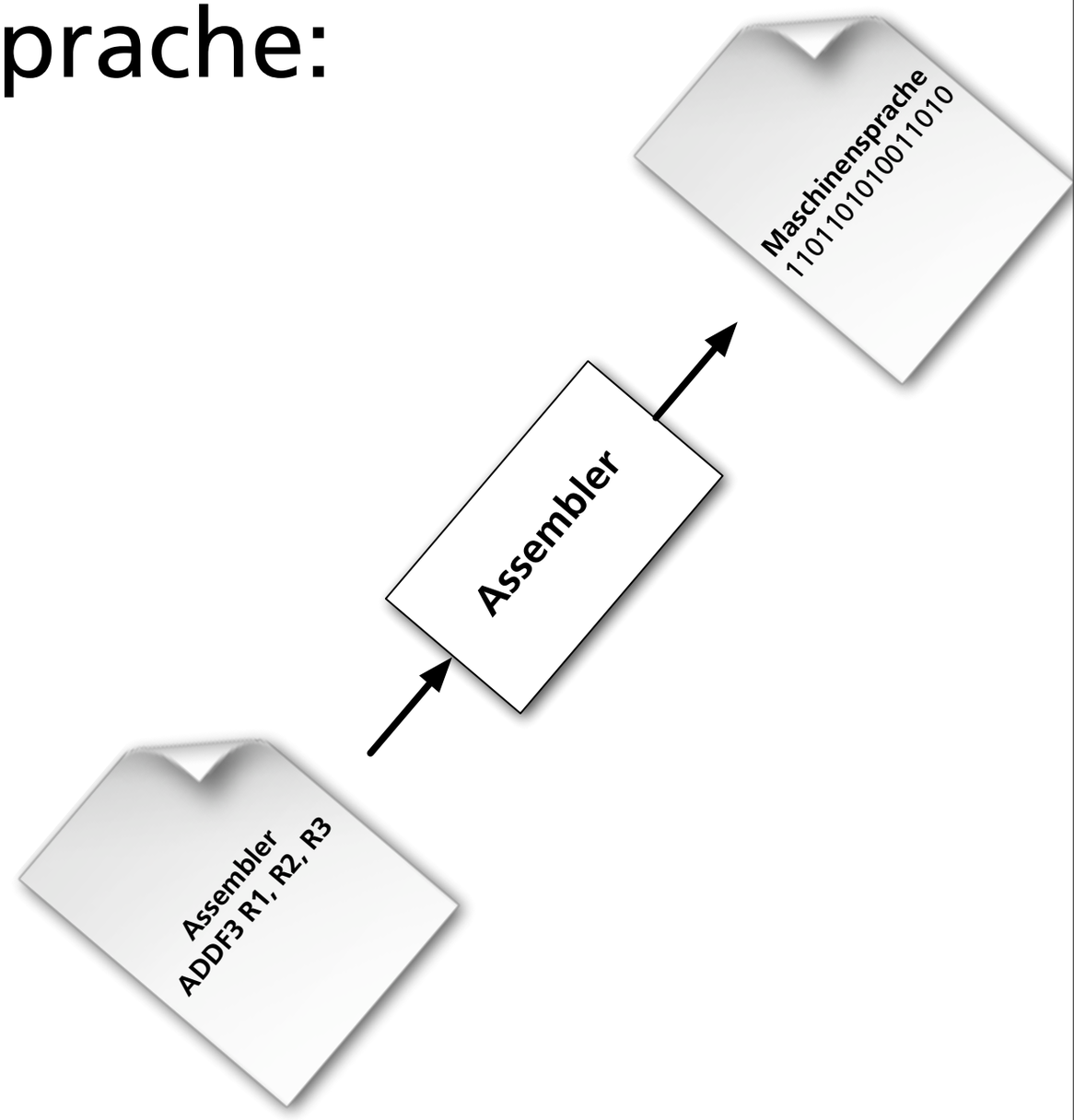
Maschinensprache vs. Assembler

Addieren in Maschinensprache:

```
1101101010011010
```

Addieren in Assembler:

```
ADDF3 R1, R2, R3
```



Höhere Programmiersprachen

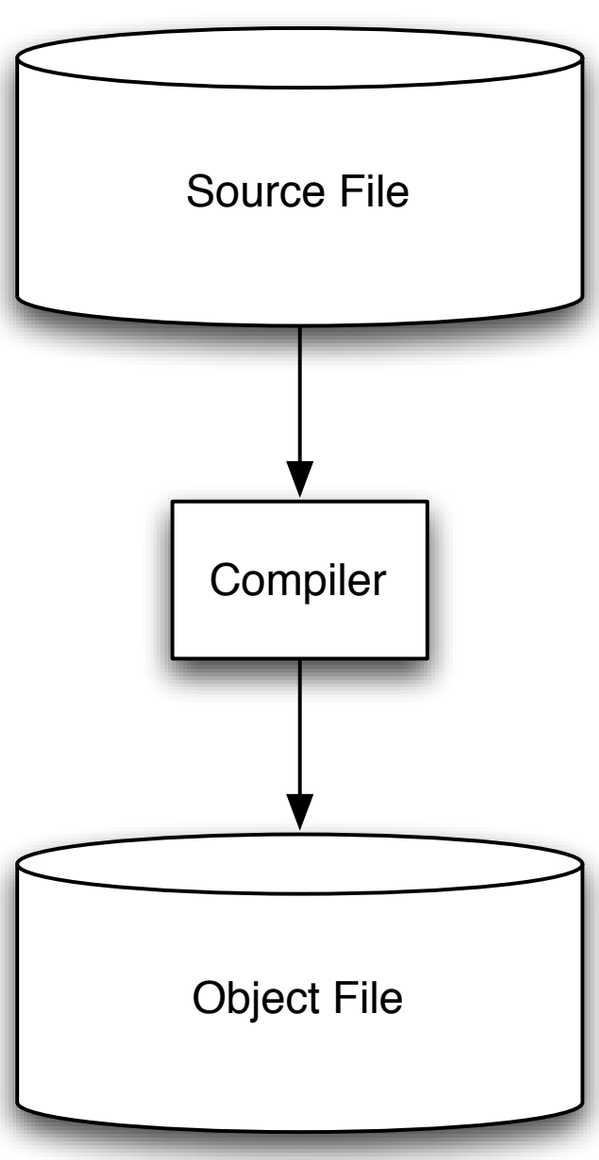
- Java
- COBOL (COmmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (benannt nach Blaise Pascal)
- Ada (benannt nach Ada Lovelace)
- C
- Visual Basic
- Delphi
- C++
- ...

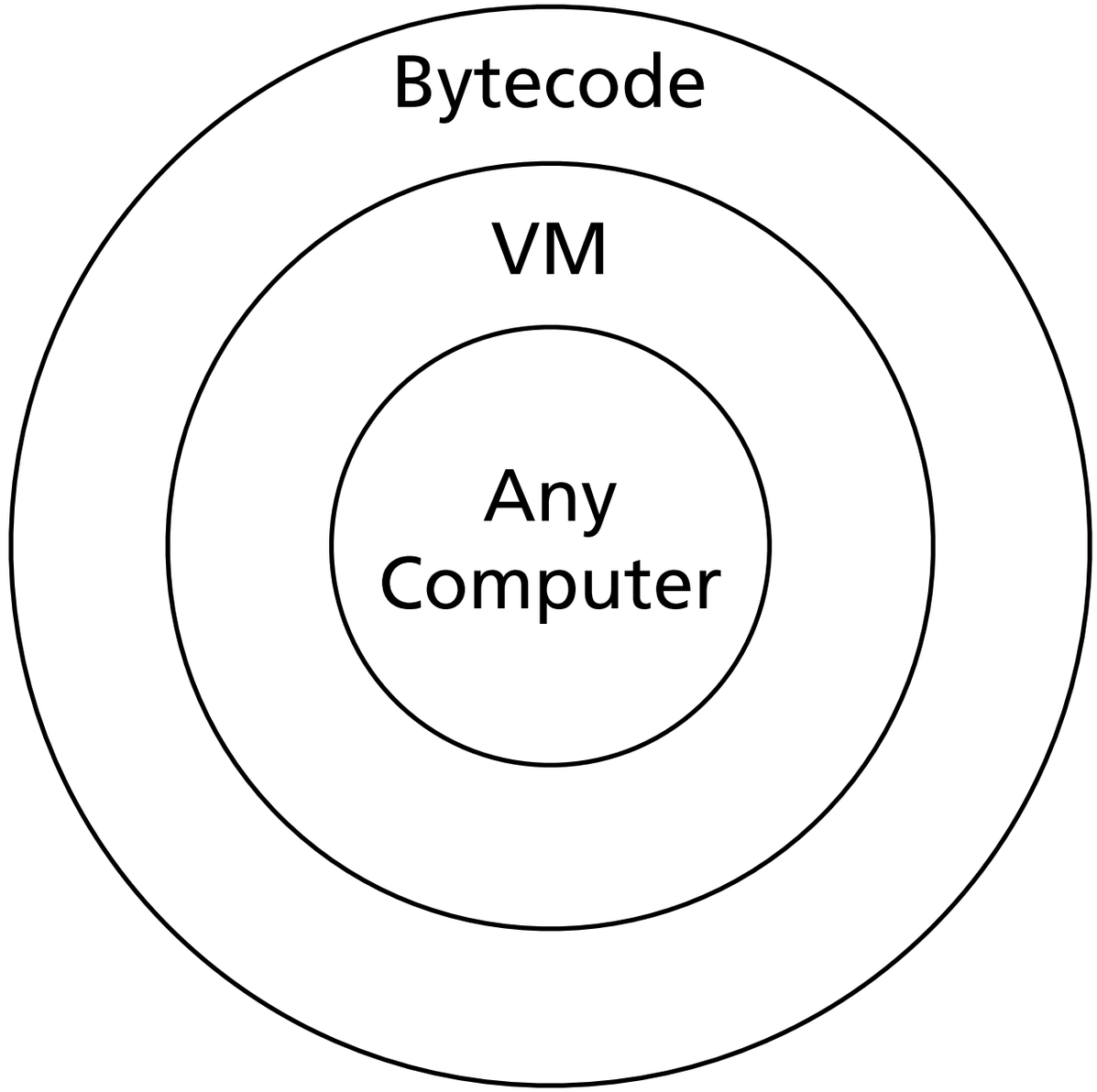
Code-Beispiele:

```
float area = 5 * 5 * 3.1415;
new Window(“This is the title”).setVisible(true);
```

Der Kompiliervorgang

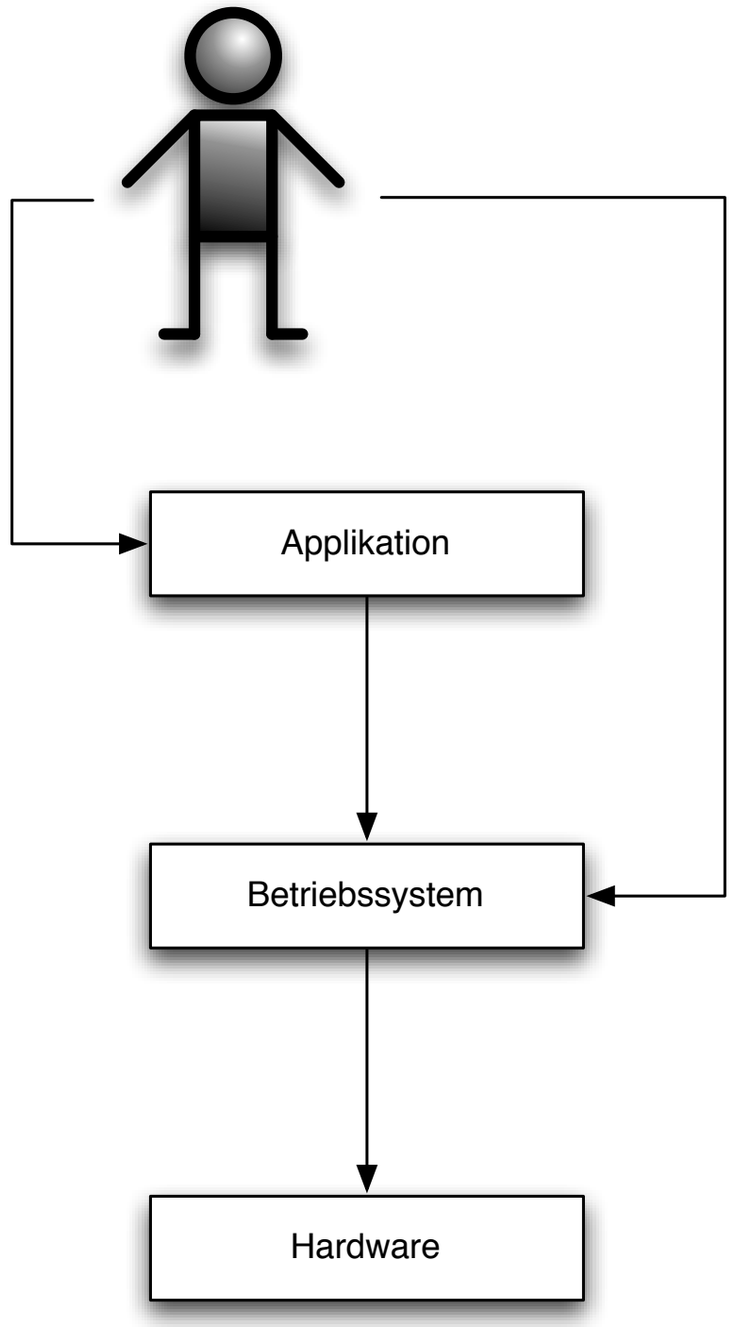
Virtuelle Maschinen





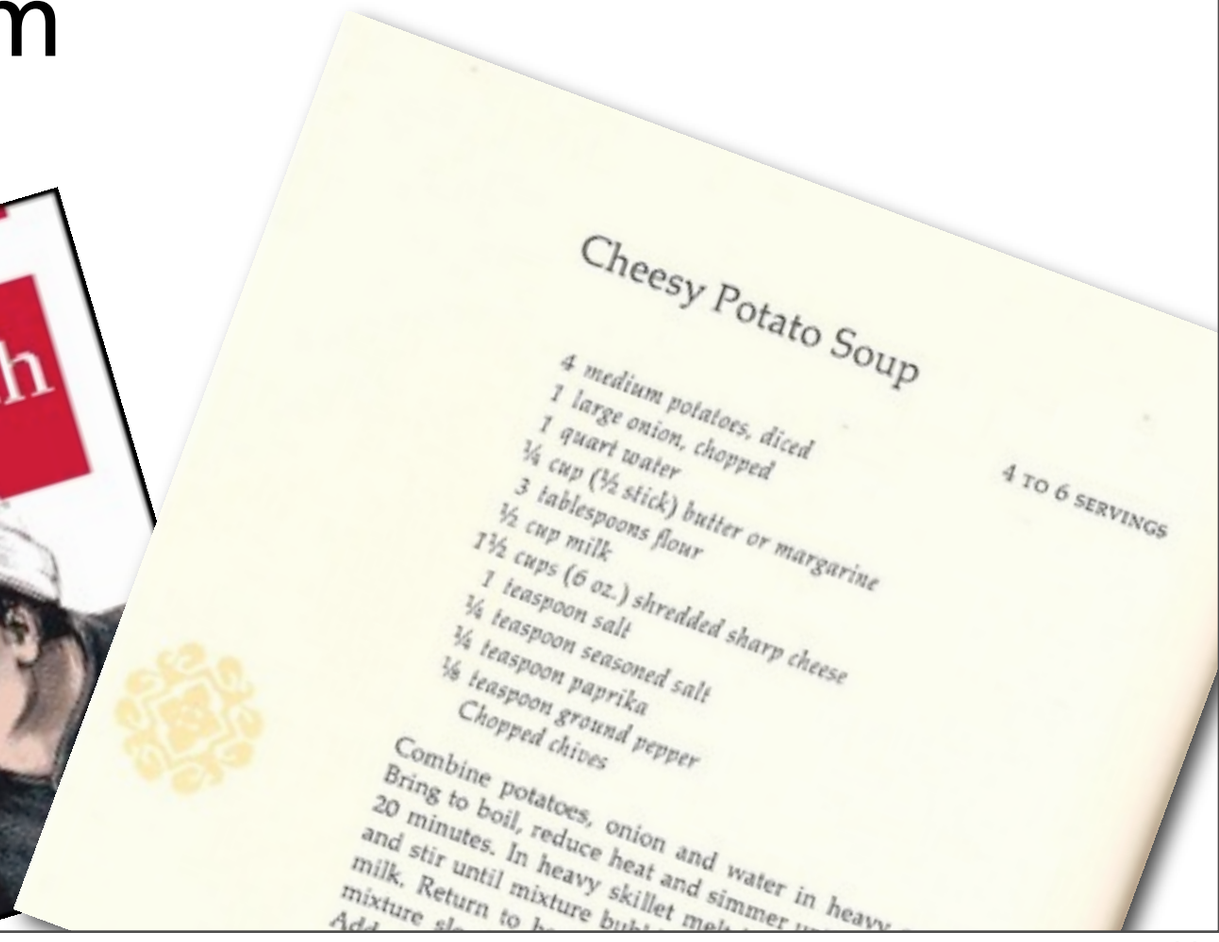
Das Betriebssystem

- Controlling und Monitoring von Systemaktivitäten
- Allokation und Zuweisung von Systemressourcen
- Scheduling von Prozessen



Algorithmisches Denken

Vom Rezept zum ausführbaren Computerprogramm



Der Risotto-Algorithmus

Zutaten für 2 Personen:

1 dl	Wein
ca. 8 dl	Rinderbouillon
1x	kleine Zwiebel
1-2	Knoblauchzehe(n)
2 Esslöffel	Olivenöl
150 g	Rundkorn Reis
50 g	Parmesan
eine Messerspitze	Safranfäden

1. Das Olivenöl erhitzen, die Zwiebel und den Knoblauch fein würfeln und darin anschwitzen.

2. Den Reis beigeben und 3-4 Minuten unter Rühren glasig werden lassen.

3. Mit dem Wein ablöschen und solange köcheln lassen, bis die Flüssigkeit verdunstet ist.

4. Den Safran beigeben.

5. Die Rinderbouillon aufkochen und eine Kelle voll zum Reis hinzugeben und verdunsten lassen.

6. Diesen Vorgang solange wiederholen, bis sämtliche Bouillon aufgebraucht ist.

7. Den Parmesan unterrühren und heiss servieren.

Weitere Algorithmen im Alltag

Prozess	Ausführender	Algorithmus	typische Anweisung
Kuchen backen	Bäcker	Rezept	nimm 1/2 kg Mehl...
Spielen einer Klaviersonate	Pianist	Partitur	♫♬♩♪
Bedienung eines Handys	Anrufer	Bedienungsanleitung	drücken sie die # Taste
Bau eines Radios	Bastler	Schaltplan und Montageanleitung	verbinde Transistor T1 mit T5

Der Euklidische Algorithmus

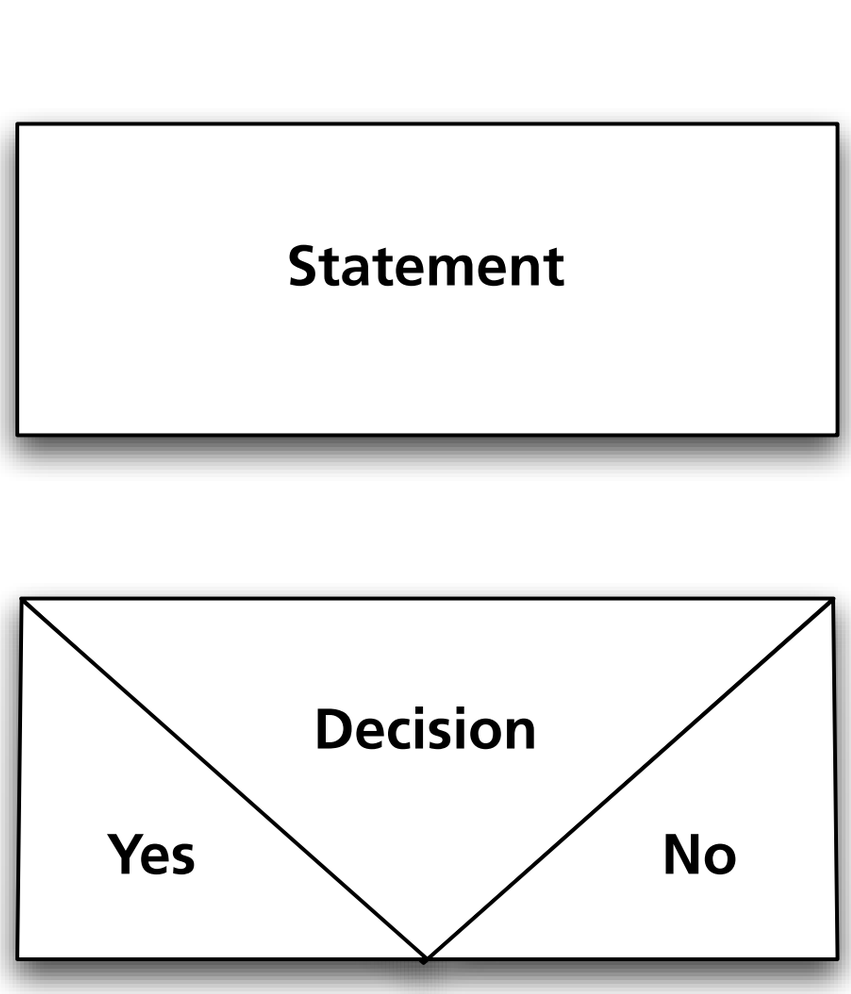
Der Euklidische Algorithmus (um 300 v. Chr. beschrieben dient zur Ermittlung des grössten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen A und B.

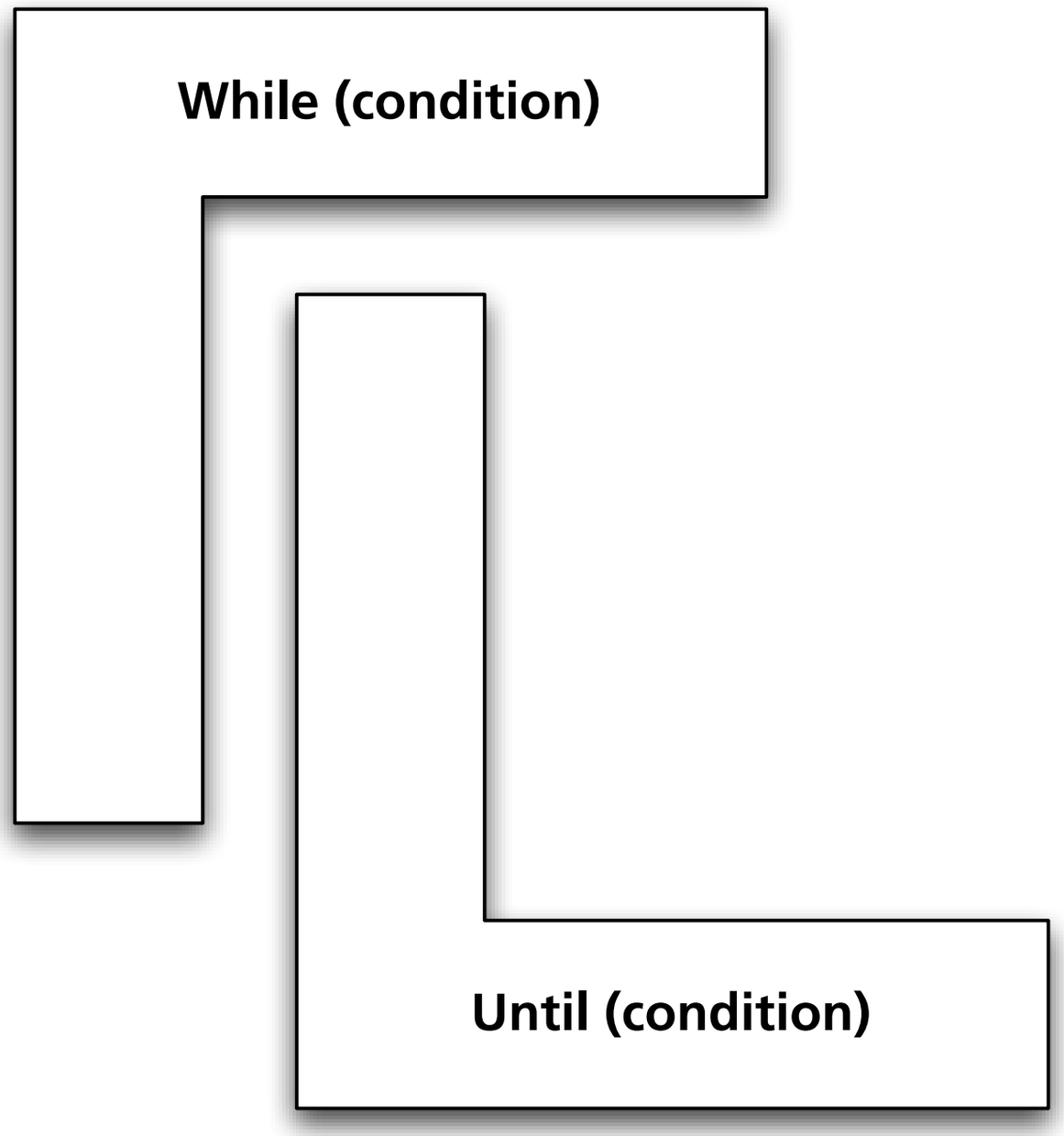
1. Sei A die grössere der beiden Zahlen A und B (entsprechend vertauschen, falls dem noch nicht so ist)
2. Setze A = A - B
3. Wenn A und B ungleich sind, dann fahre fort mit Schritt 1. Wenn sie gleich sind, dann beende den Algorithmus: Diese Zahl ist der ggT

Beispiel: ggT von 14 und 8

Schritt	A	B	A - B
1.	14	8	6
2.	8	6	2
3.	6	2	4
4.	4	2	2
5.	2	2	gleich

Nassi-Shneiderman Diagramme





Aufgabe: Der Risotto-Algorithmus als NSD

Zutaten für 2 Personen:

1 dl	Wein
ca. 8 dl	Rinderbouillon
1x	kleine Zwiebel
1-2	Knoblauchzehe(n)
2 Esslöffel	Olivenöl
150 g	Rundkorn Reis
50 g	Parmesan
eine Messerspitze	Safranfäden

1. Das Olivenöl erhitzen, die Zwiebel und den Knoblauch fein würfeln und darin anschwitzen.

2. Den Reis beigeben und 3-4 Minuten unter Rühren glasig werden lassen.

3. Mit dem Wein ablöschen und solange köcheln lassen, bis die Flüssigkeit verdunstet ist.

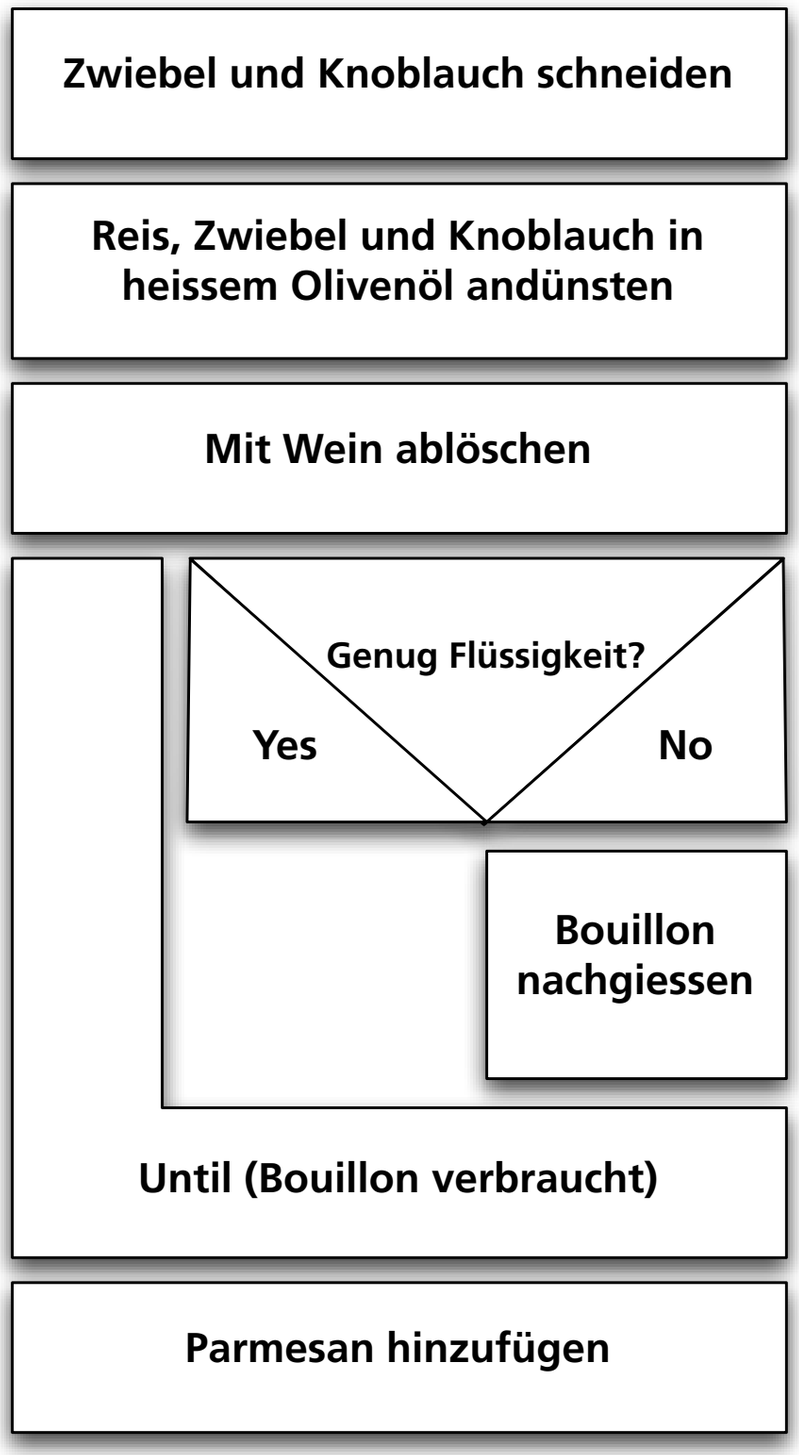
4. Den Safran beigeben.

5. Die Rinderbouillon aufkochen und eine Kelle voll zum Reis hinzugeben und verdunsten lassen.

6. Diesen Vorgang solange wiederholen, bis sämtliche Bouillon aufgebraucht ist.

7. Den Parmesan unterrühren und heiss servieren.

Lösung: Der Risotto-Algorithmus als NSD

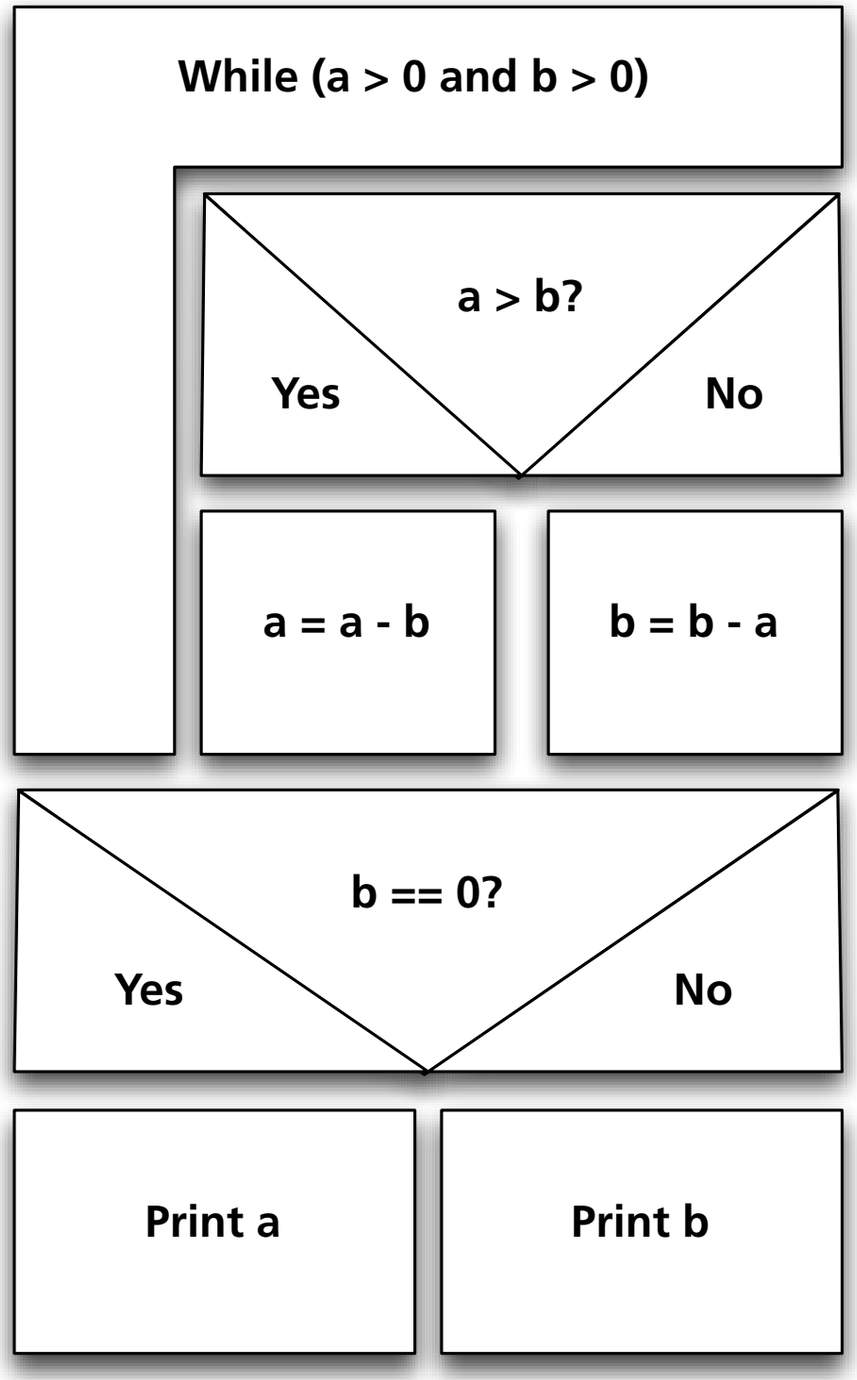


Aufgabe: Der Euklidische Algorithmus als NSD

Der Euklidische Algorithmus (um 300 v. Chr. beschrieben dient zur Ermittlung des grössten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen A und B.

1. Sei A die grössere der beiden Zahlen A und B (entsprechend vertauschen, falls dem noch nicht so ist)
2. Setze A = A - B
3. Wenn A und B ungleich sind, dann fahre fort mit Schritt 1. Wenn sie gleich sind, dann beende den Algorithmus: Diese Zahl ist der ggT

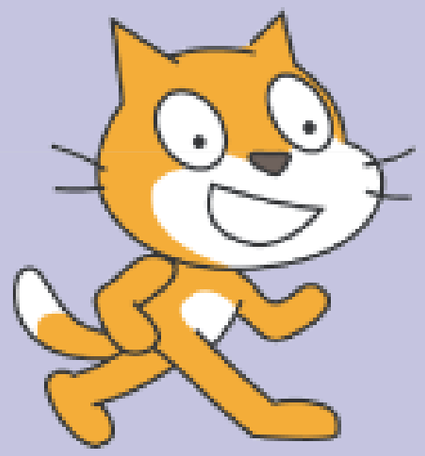
Lösung: Der Euklidische Algorithmus als NSD



Getting Started with

SCRATCH

version 1.4

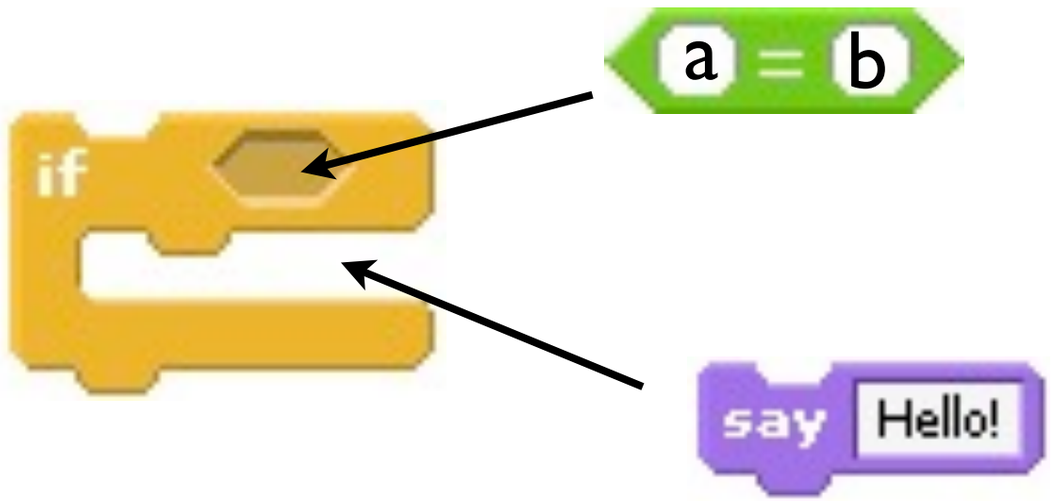


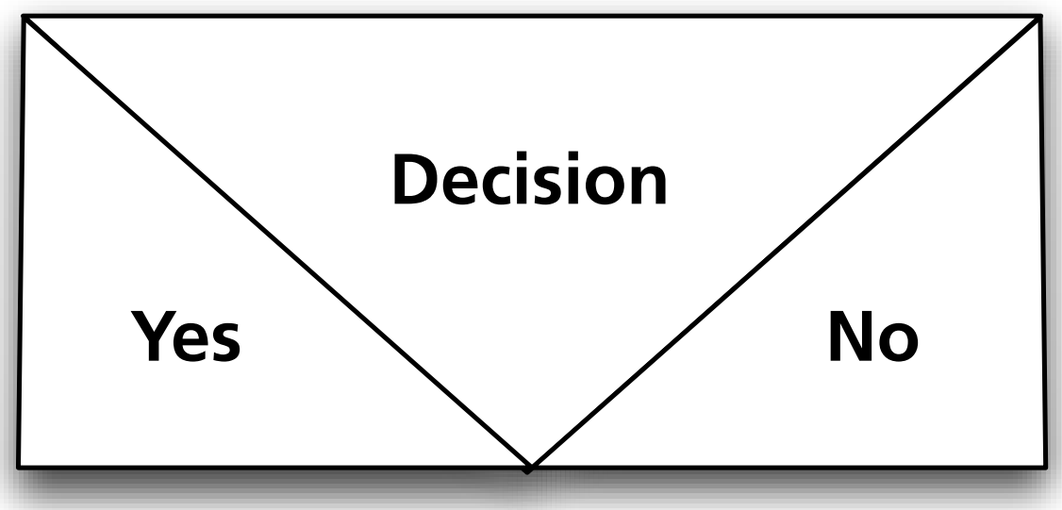
http://scratch.mit.edu

Algorithmen in Scratch: Statements

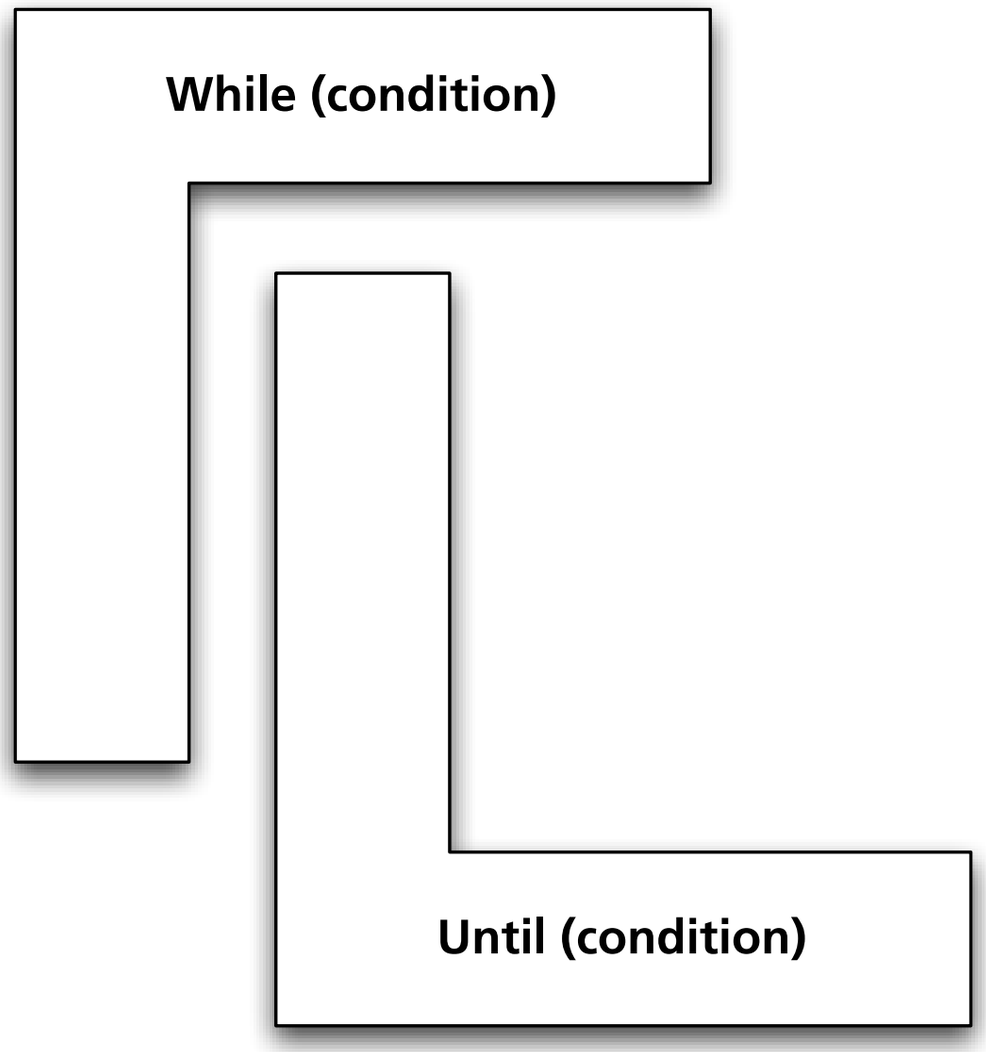
Statement

Algorithmen in Scratch: Decisions

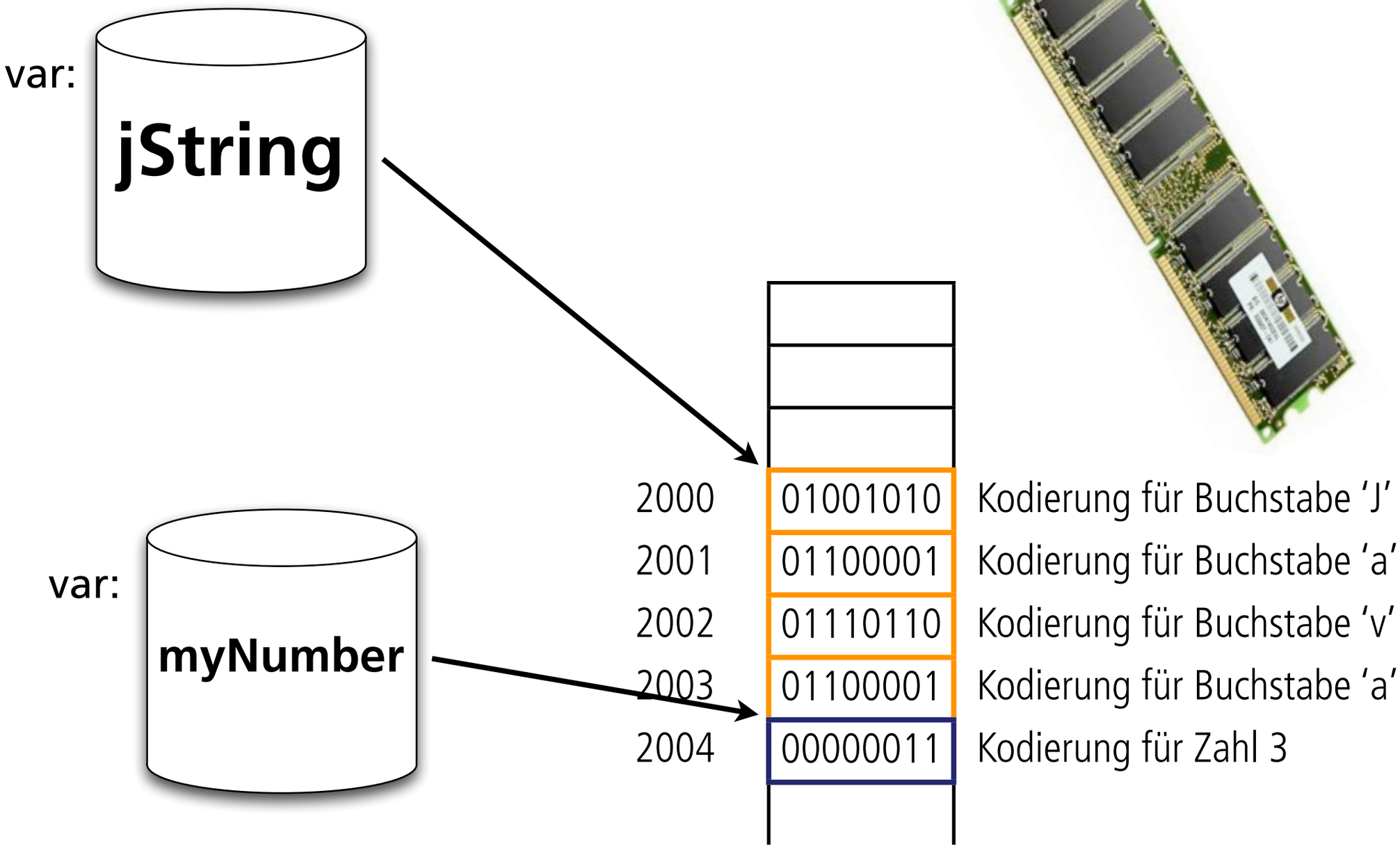




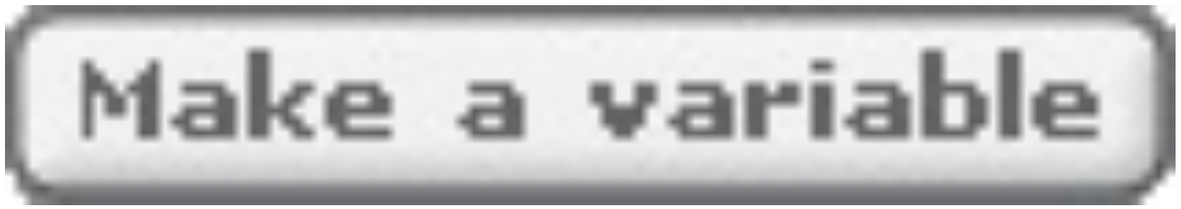
Algorithmen in Scratch: Loops

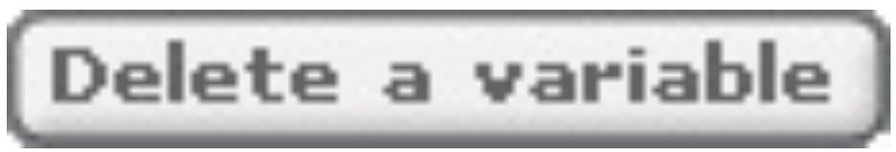


Variablen in einem Programm

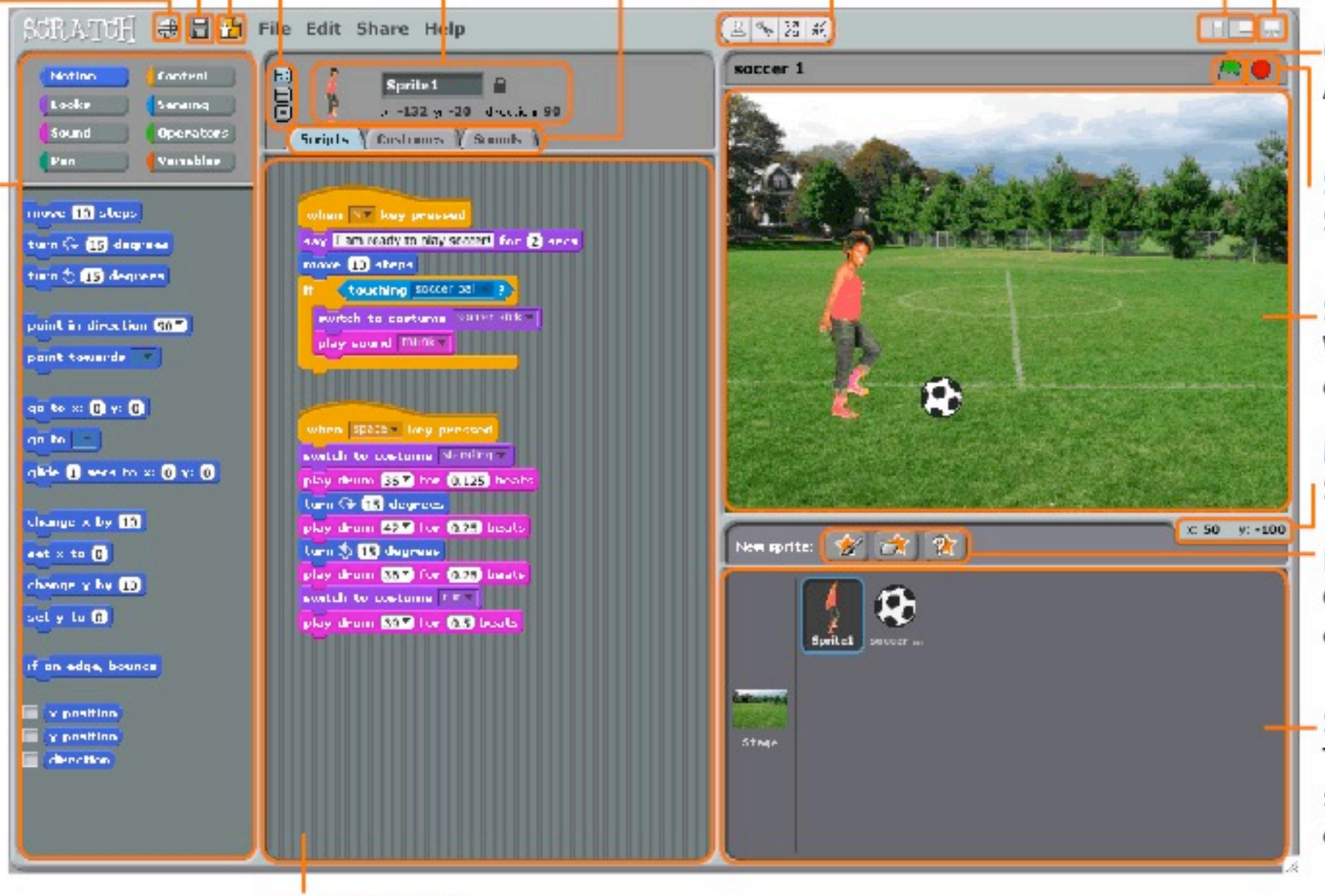


Variablen in Scratch





SCRATCH 1.4 INTERFACE



SAVE

LANGUAGE

SHARE

SPRITE ROTATION STYLE

CURRENT SPRITE INFO

TABS
Edit scripts, costumes, or sounds.

TOOLBAR

VIEW MODE
Change to large or small stage view.

PRESENTATION MODE
Present your project.

BLOCKS PALETTE
Blocks for programming your sprites.

GREEN FLAG
A way to start scripts.

STOP SIGN
Stops all scripts.

STAGE
Where your Scratch creations come to life.

MOUSE X-Y DISPLAY
Shows location of cursor.

NEW SPRITE BUTTONS
Create a new character or object for your project.

SPRITE LIST
Thumbnails of all your sprites. Click to select and edit a sprite.

SCRIPTS AREA
Drag blocks in, snap them together into scripts.