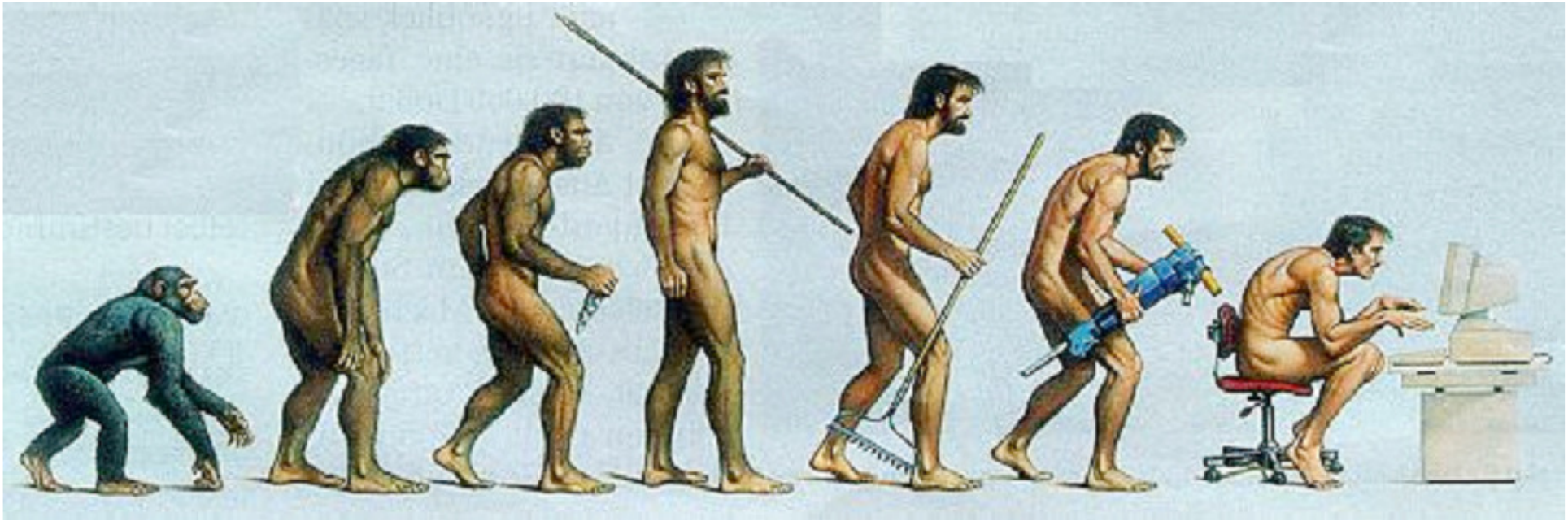
Was ist

Informatik?



Was machen

Informatiker?


„Computer science
is no more about
computers than
astronomy is about
telescopes.“
Edsger W. Dijkstra, niederländischer Informatiker (1930-2002).


Great Principles
of Computing
http://cs.gmu.edu/cne/pjd/GP/

Table 2: The Five Windows of Computing Mechanics

Window	Central Concern	Principal Stories
Computation	What can be computed and how; limits of computing.	Algorithm, data structure, automata, languages, Turing machine, universal computer, Turing complexity, Chaitin complexity, self reference, approximations, heuristics, non-computability, translations, compilations, physical realizations
Communication	Sending messages from one point to another.	Data transmission, Shannon entropy, encoding to medium, channel capacity, noise suppression, error correcting codes, end-to-end-error correction, Huffman and Reed-Solomon codes, file compression, cryptography, packet networking
Coordination	Multiple entities cooperating toward a single result.	Human-to-human (action loops, workflows as supported by communicating computers), human-computer (interface, input, output, response time, data visualization); computer-computer (concurrency control, races, synchronization, deadlock, serializability, atomic actions)
Automation	Performing cognitive tasks by computer.	Simulation and machine performance of cognitive tasks, philosophical distinctions about automation, expertise and expert systems, enhancement of intelligence, Turing tests, machine learning and recognition, bionics
Recollection	Storing and retrieving information.	Hierarchies of storage, locality of reference, caching, address space and mapping, bindings, naming, sharing, thrashing, retrieval by name, retrieval by content

Peter J. Denning. Great Principles in Computing Curricula. SIGCSE 2004, Norfolk, Virginia, USA.

Area	Problem
Computation	• Unbounded error accumulation on finite machines • Non-computability of some important problems • Intractability of thousands of common problems • Optimal algorithms for some common problems • Production quality compilers
Communication	• Lossless file compression • Lossy but high-fidelity audio and video compression • Error correction codes for high, bursty noise channels • Secure cryptographic key exchange in open networks
Interaction	• Arbitration problem • Timing-dependent (race-conditioned) bug problem • Deadlock problem • Fast algorithms for predicting throughput and response time • Internet protocols • Cryptographic authentication protocols
Recollection	• Locality • Thrashing • Search • Two-level mapping for access to shared objects
Automation	• Simulations of focused cognitive tasks • Limits on expert systems • Reverse Turing tests
Design	• Objects and information hiding • Levels • Throughput and response time prediction networks of servers

Peter J. Denning. Is Computer Science Science? Communications of the ACM, April 2005, Vol. 48, No. 4, pp. 27-31.

Principle	Summary	Computing Examples	
Intractability (Computation)	Over 3,000 key problems in science, engineering, and commerce require more computation, even for small inputs, than can be done in the life of the universe.	Searching for optimal solutions. Traveling salesman. Knapsack packing. Bin packing. Tiling a plane.	Parcel delivery. Truck transportation. Taxi routing. Airline routing. Scheduling (industrial engineering).
Compression (Communication)	Representations of data and algorithms can be significantly compressed and the most valuable information recovered later.	Compression of voice (MP3, MP4, ACC), images (JPEG, GIF), files (Zip). Fourier transform.	Operation of cochlea in the ear. Morse code.
Choosing (Coordination)	An uncertainty principle: it is not possible to make an unambiguous choice of one of several alternatives within a fixed deadline.	Hardware that never crashes while responding to interrupts. Mutual exclusion. Deadlocks.	Traffic control. Telephone and network routers. DNA sequencing. Free will (psychology).
Locality (Recollection)	Computations cluster their information recall actions into hierarchically aggregated regions of space and time for extended periods.	Virtual memory. Hardware caching. Web caching. Interconnection structures in parallel machines.	Functional brain cell clusters. Near decomposable economic systems. Punctuated equilibrium (biology).
Search (Automation)	Finding a pattern or configuration in a very large space of possibilities.	Genetic algorithms. Evolutionary computing. Branch and bound. Gradient search.	Genetic evolution. Passing of genes to descendents.
Bottlenecks (Evaluation)	Forced flow laws: in any network, the throughput at any node is the product of the network throughput and the visits per task to the node.	Saturation and bottlenecks in communication networks.	Fast propagating urban gridlock. Assembly lines (industrial engineering).
Hierarchical Aggregation (Design)	Larger entities are composed of many smaller ones.	OS and network software levels. Information hiding. Modularity. Abstraction.	Ladder of scale (astronomy and physics). Functional organs (biology). Fractals.

Peter J. Denning. Computing is a Natural Science. Communications of the ACM, July 2007, Vol. 50, No. 7, pp. 13-18.

Principles of Computing

- Computation
- Communication
- Coordination
- Recollection
- Automation
- Evaluation
- Design

Peter J. Denning. Computing is a Natural Science.
Communications of the ACM, July 2007, Vol. 50, No. 7, pp. 13-18.

Principle

Computation

Wie werden Informationen repräsentiert?
Wie werden Berechnungen repräsentiert?

Was ist berechenbar?
Was ist nicht berechenbar?

Jedes Principle hat eine Story.

Computer science was born in 1936, at the dawn of the electronic digital computer, when Alan Turing wrote about the capabilities of computing machines. [...]

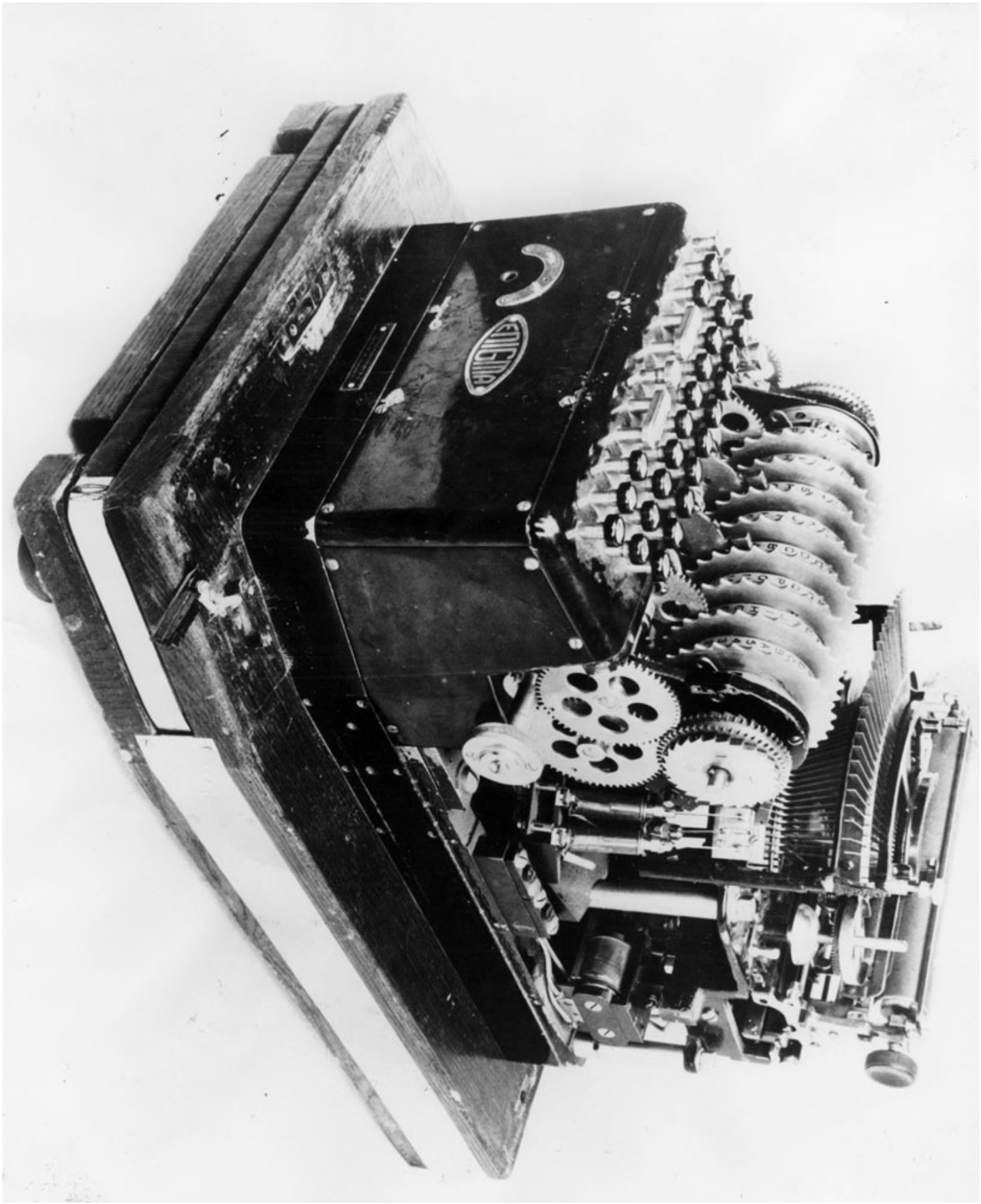
In 1936, Alan Turing published a famous paper, "On the computable numbers, with an application to the Entscheidungsproblem." Turing defined computation, computing machines, and universal machines, and he nonchalantly showed that the halting problem for machines was not computable. From that conclusion he demonstrated that the century-old "Entscheidungsproblem" (German for decision problem) had no solution. That problem posited a complete and consistent universal logic system that would be able to tell if a proposition from any other logic system is true. It was on the mathematician David Hilbert's 1928 list of challenge problems in mathematics. It dreamt of a method to tell "by inspection" whether a computation halted. Turing showed that if such a logic system existed, it would be able to answer the halting question. He concluded that the dream of a "by inspection" method to answer halting questions was impossible. He showed that the very steps a mathematician might use to apply a "by inspection" method were fundamentally computational. Therefore the only general method of approaching the halting question is to run the computations and see what happens.

Turing thus showed that computation is unavoidable. This truly was the birth of computer science.

Peter J. Denning, Craig Martell.
http://cs.gmu.edu/cne/pjd/GP/overviews/ov_computation.pdf


Alan Turing

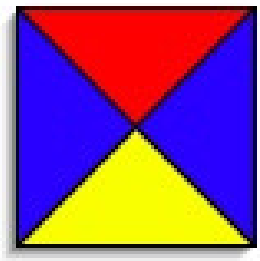
http://en.wikipedia.org/wiki/Alan_Turing

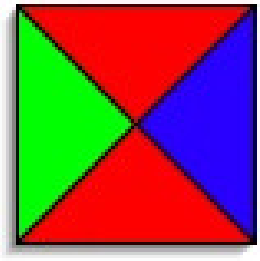


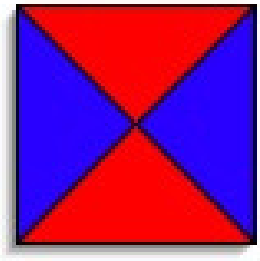
http://www.cryptomuseum.com/crypto/enigma/

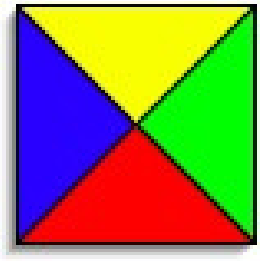
http://www.cs4fn.org/algorithms/uncomputable.php

http://www.cs4fn.org/algorithms/uncomputable.php

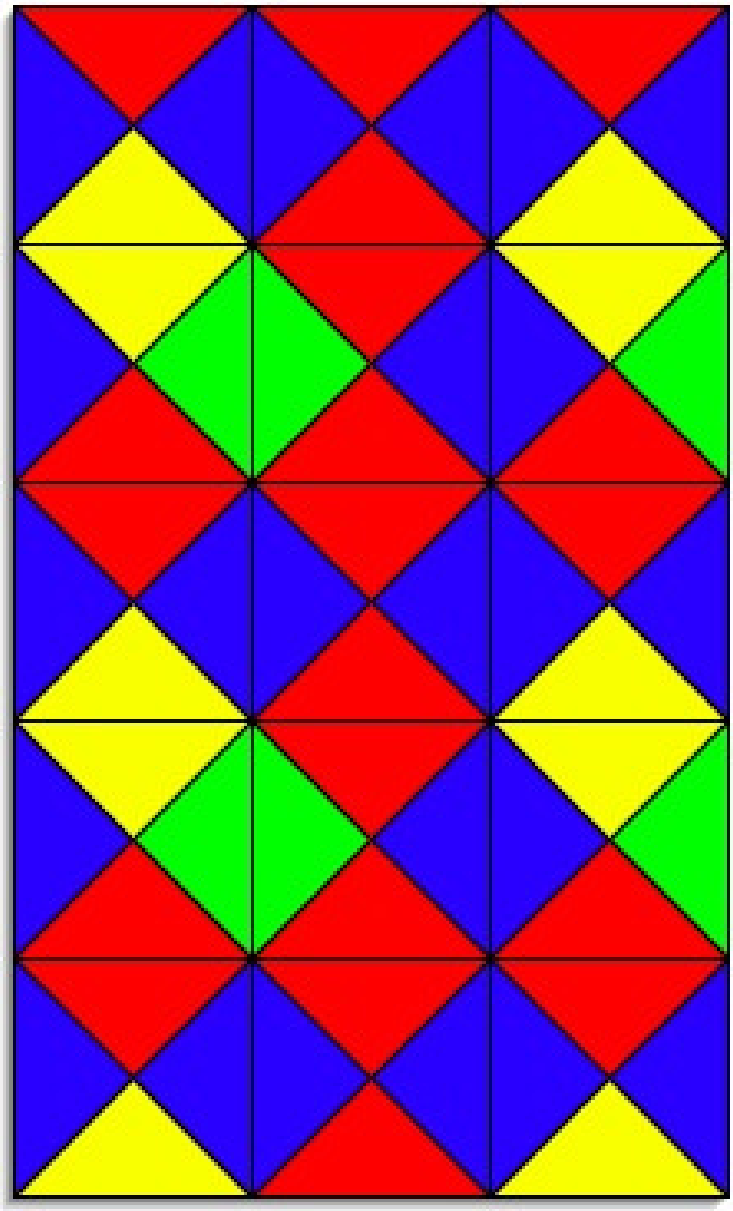


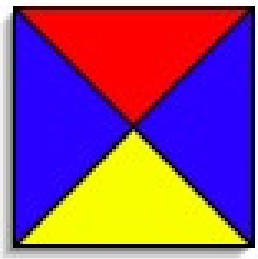


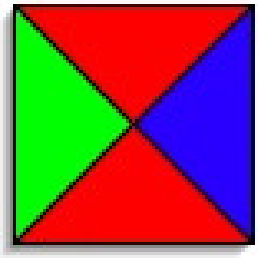


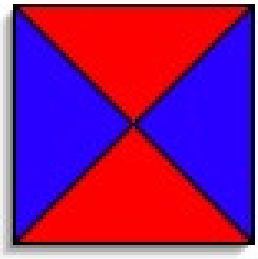


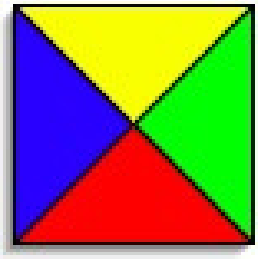
http://www.cs4fn.org/algorithms/tiles.php





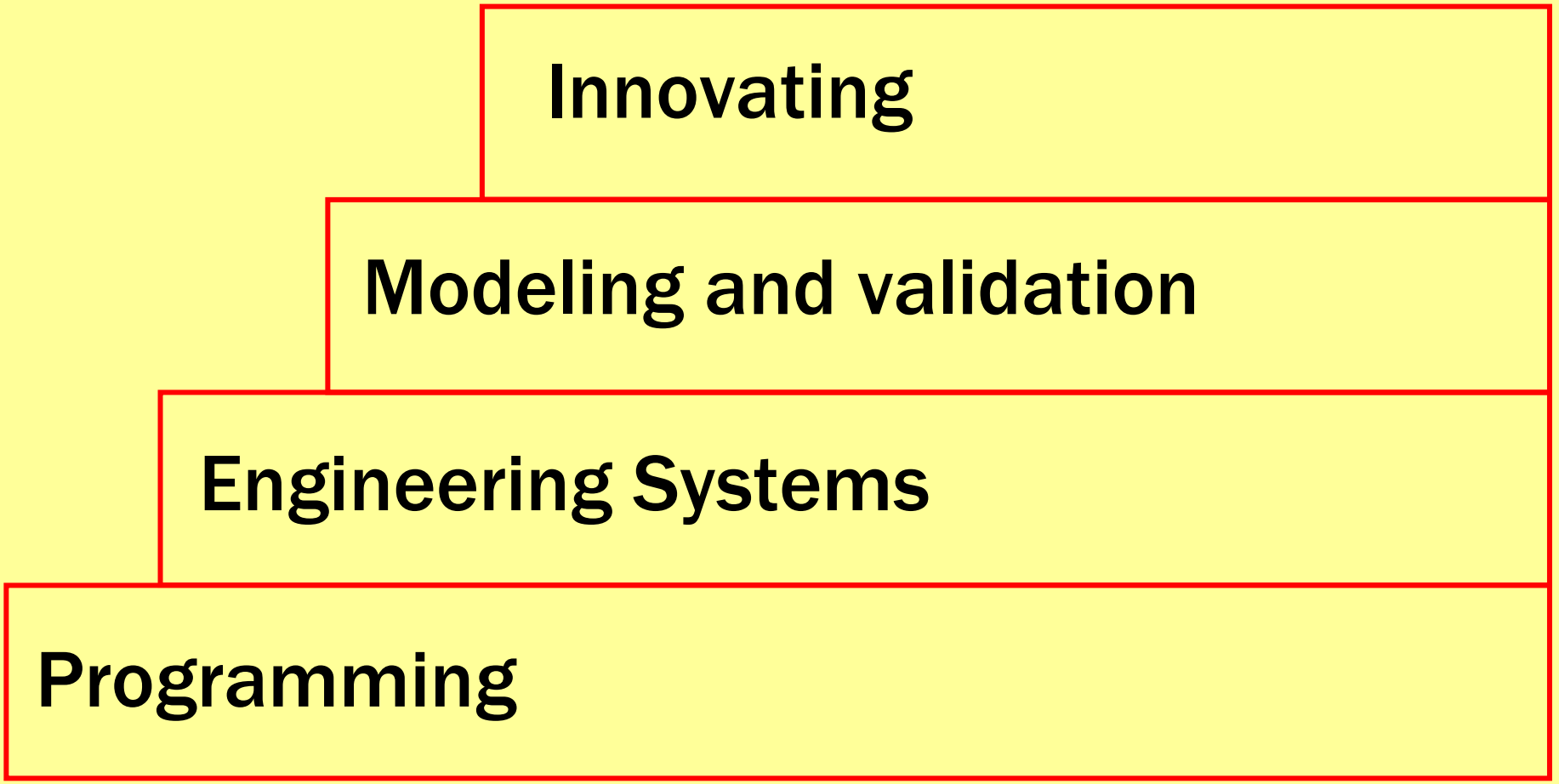






http://www.cs4fn.org/algorithms/tiles.php

Practices of Computing



Great Practices of Computing

Programming

Using programming languages to build software systems that meet specifications created in cooperation with the users of those systems. Computing professionals must be multilingual, facile with the numerous programming languages, each attuned to its own strategies for solving problems.

Engineering Systems

Designing and constructing systems of software and hardware components running on servers connected by networks. These practices include a design component concerned with organizing a system to produce valuable and tangible benefits for the users; and an engineering component concerned with the modules, abstractions, revisions, design decisions, and risks in the system; and an operations component concerned with configuration, management, and maintenance of the system. High levels of skill are needed for large programmed systems encompassing thousands of modules and millions of lines of code.

Great Practices of Computing

Modeling and validation

Building models of systems to make predictions about their behavior under various conditions; and designing experiments to validate algorithms and systems.

Innovating

Bringing about lasting changes in the ways groups and communities operate by exercising technical leadership. Innovators watch for and analyze opportunities, listen to customers, formulate offers customers see as valuable, and manage commitments to deliver the promised results. Innovators are history-makers who have strong historical sensibilities.

Practice

Programming

Bash

```
# Bash: "Hello, world!"
echo Hello, world!
```

BASIC

```
10 REM BASIC: "Hello, world!"
20 PRINT "Hello, world!"
```

Brainfuck

```
Brainfuck: "HelloKOMMA World!" [-]++++++++++[>++++++
+>++++++++++>++++>+++>+<<<<<-]>++.>+.+++++++..+++.>+
+++.>++.<<++++++++.--------.+++.------.--------.>>+.
```

BS2000

```
/BEGIN-PROCEDURE LOGGING=N
/REMARK BS2000 (SDF): "Hello, world!"
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/WRITE-TEXT 'Hello, world!'
/SET-JOB-STEP
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROCEDURE
```

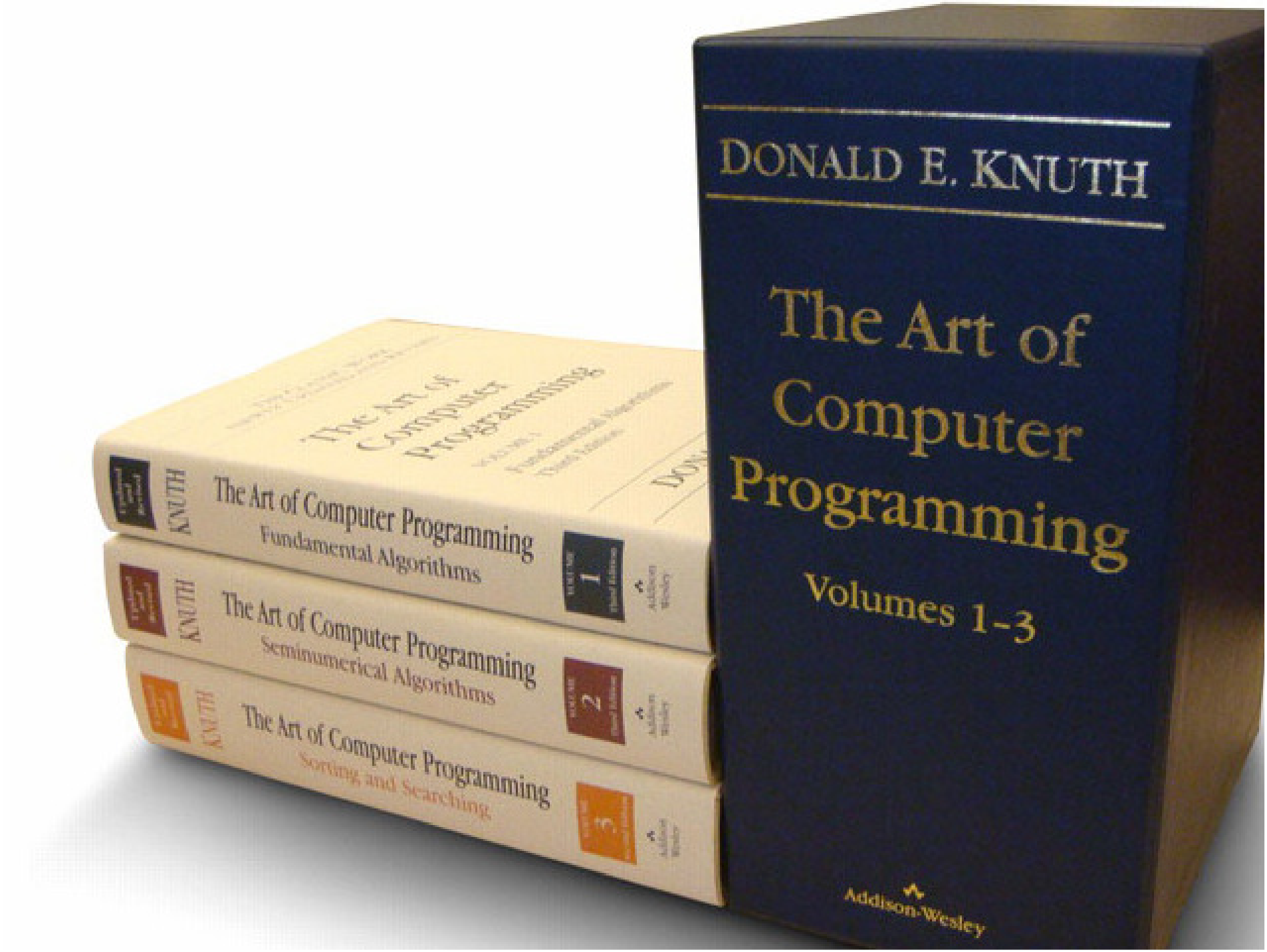
Knackpunkt – das umstrittene Wesen der Informatik – nur schon das Wesen des Programmierens ist umstritten.

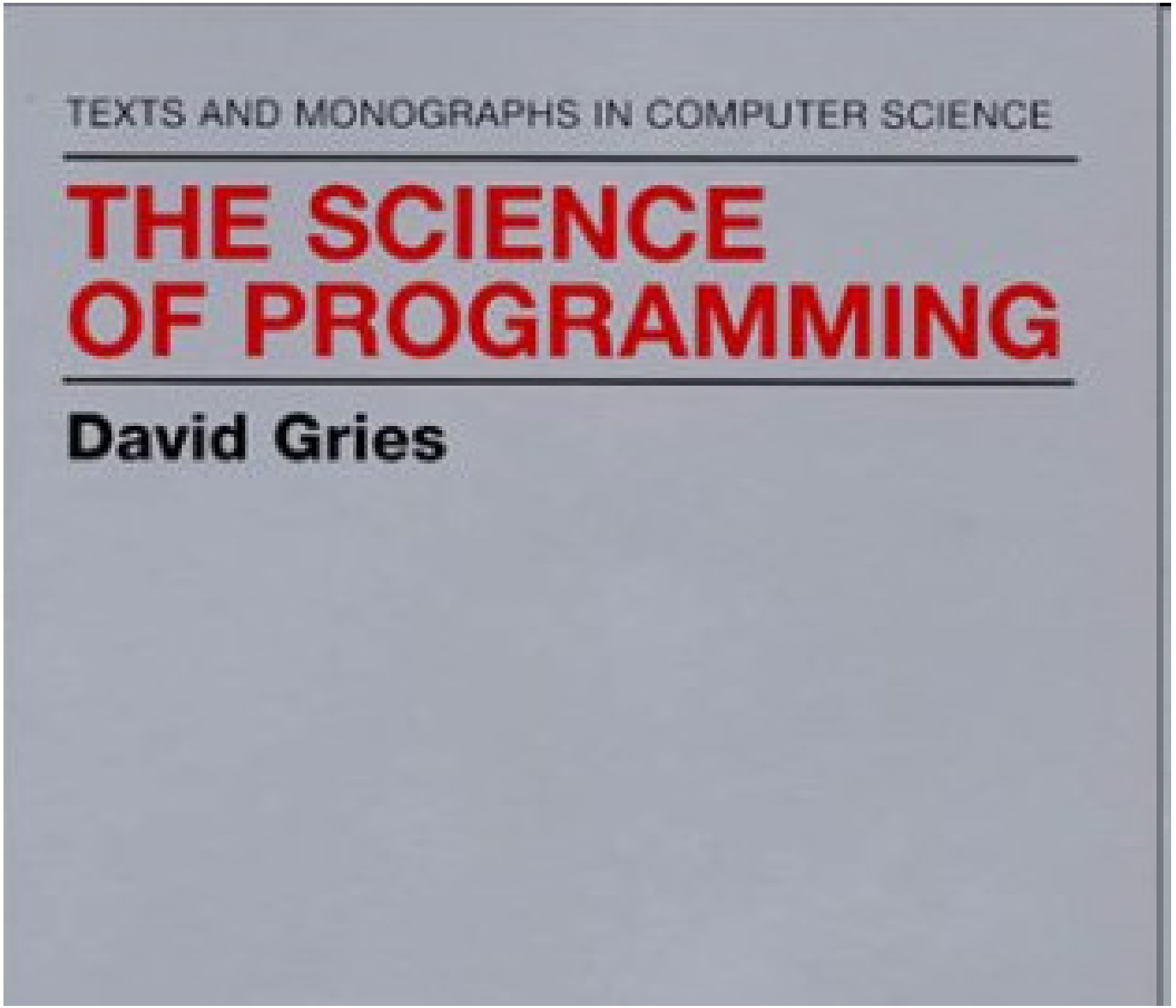
Für manche ist Programmieren eine Art Kunst.

Für manche ist es eine wissenschaftliche Tätigkeit.

Für manche ist es ein Handwerk.

Für manche ist es eine Ingenieur-Tätigkeit.


DONALD E. KNUTH
The Art of Computer Programming
Volumes 1-3
Addison-Wesley
KNUTH
The Art of Computer Programming
Fundamental Algorithms
KNUTH
The Art of Computer Programming
Seminumerical Algorithms
KNUTH
The Art of Computer Programming
Sorting and Searching


TEXTS AND MONOGRAPHS IN COMPUTER SCIENCE
THE SCIENCE OF PROGRAMMING
David Gries

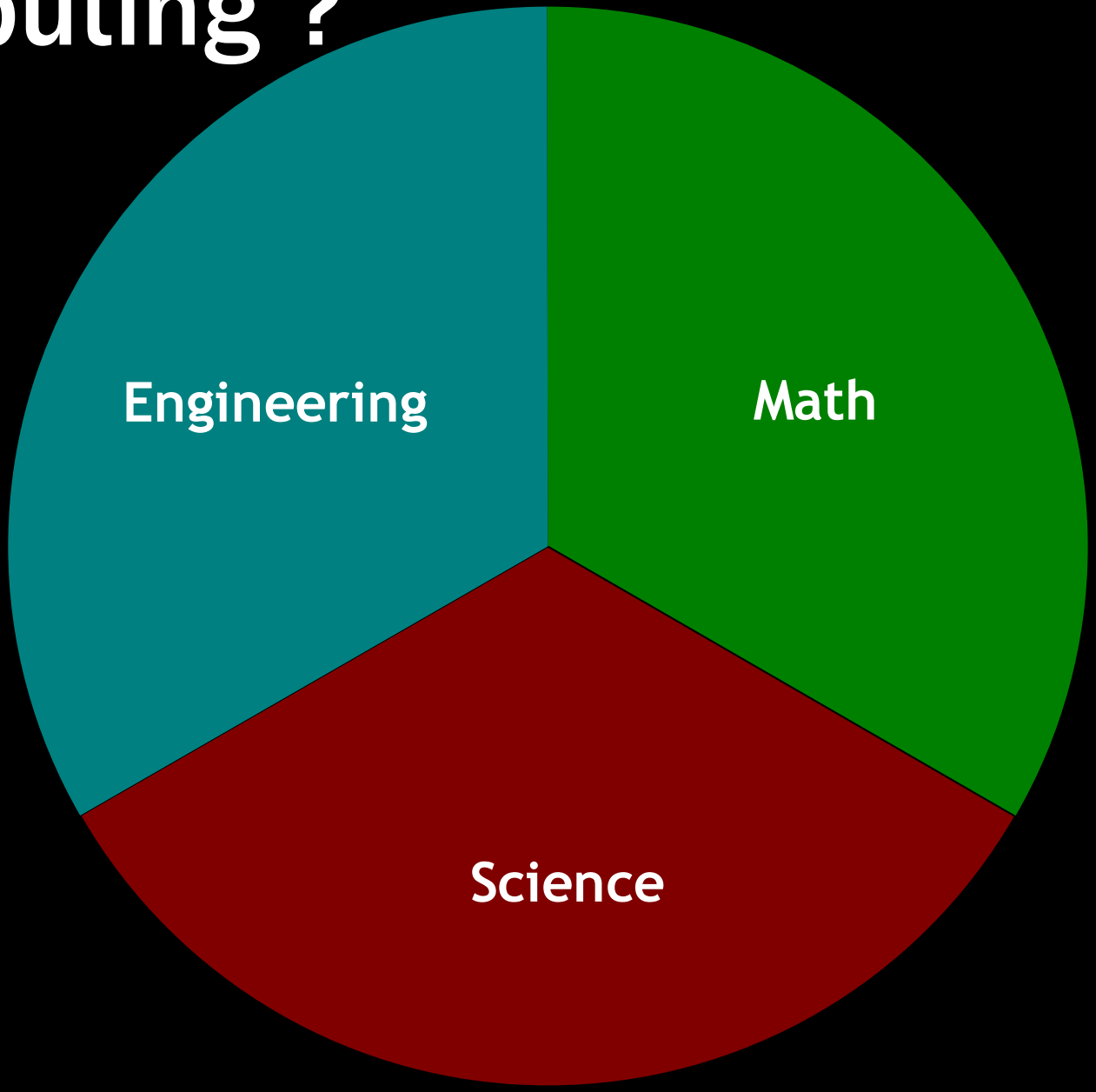
Software Craftsmanship

The New Imperative

SOFTWARE ENGINEERING

Barry W. Boehm's

Computing ?



Peter J. Denning, Peter A. Freeman. The Profession of IT: Computing's Paradigm. Communications of the ACM, Vol. 52, No.12, 2009, pp. pp. 28-30.

Computing !

Peter J. Denning, Peter A. Freeman. The Profession of IT: Computing's Paradigm.
Communications of the ACM, Vol. 52, No.12, 2009, pp. pp. 28-30.

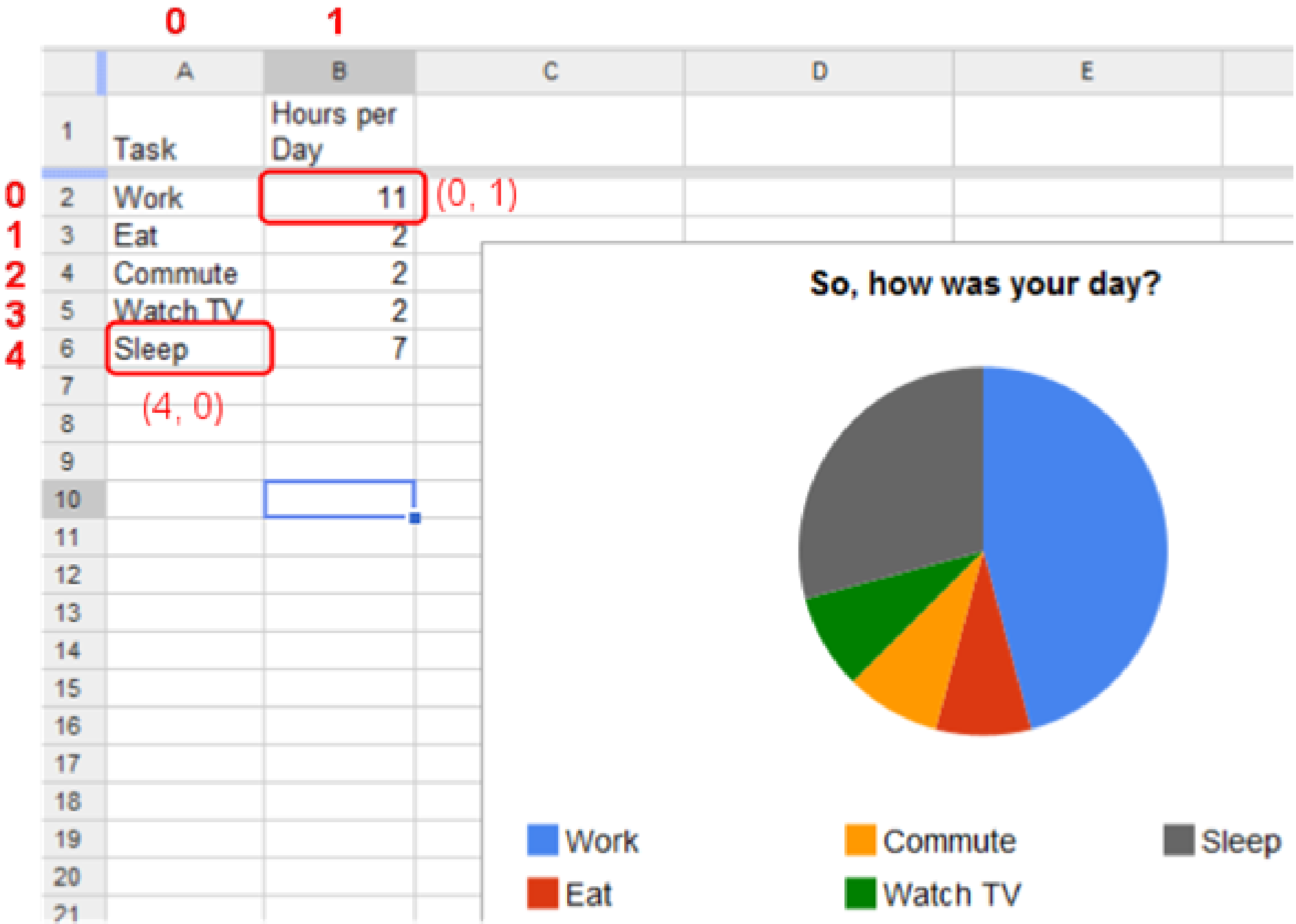
Knackpunkt – das umstrittene Wesen der Informatik – nur schon das Wesen des Programmierens ist umstritten.

Was macht der Praktiker, wenn die Theoretiker sich nicht einig sind?

Er praktiziert ganz pragmatisch.

The Pragmatic Programmer

from journeyman
to master


0
1
A
B
C
D
E
1
Task
Hours per Day
0
2
Work
11
(0, 1)
1
3
Eat
2
2
4
Commute
2
3
5
Watch TV
2
4
6
Sleep
7
7
8
(4, 0)
9
10
11
12
13
14
15
16
17
18
19
20
21
So, how was your day?
Work
Eat
Commute
Watch TV
Sleep

Google code **Code Playground**

Pick an API Search

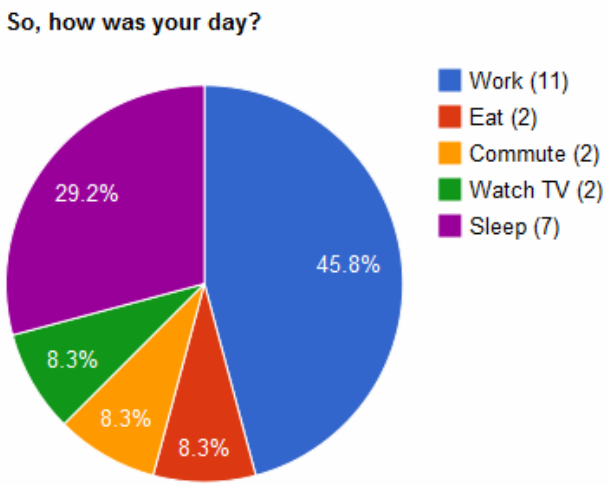
- Area Chart
- Bar Chart
- Column Chart
- Gauge
- Geo Map
- Image Chart
- Image Area Chart
- Image Bar Chart
- Image Candlestick Chart
- Image Line Chart
- Image Pie Chart
- Intensity Map
- Line Chart
- Motion Chart
- Org Chart
- Pie Chart
- Scatter Chart
- Sparkline
- Table
- Tree Map
- Advanced
  - Annotated Time Line Configuration
  - Image Multicolor Bar Chart
  - Motion Chart Time Formats
  - Table Dimensions
  - Table Paging

Edit Code Basics » Pie Chart — View Docs | Edit HTML

```
function drawVisualization() {
  // Create and populate the data table.
  var data = new google.visualization.DataTable();
  data.addColumn('string', 'Task');
  data.addColumn('number', 'Hours per Day');
  data.addRows(5);
  data.setValue(0, 0, 'Work');
  data.setValue(0, 1, 11);
  data.setValue(1, 0, 'Eat');
  data.setValue(1, 1, 2);
  data.setValue(2, 0, 'Commute');
  data.setValue(2, 1, 2);
  data.setValue(3, 0, 'Watch TV');
  data.setValue(3, 1, 2);
  data.setValue(4, 0, 'Sleep');
  data.setValue(4, 1, 7);

  // Create and draw the visualization.
  new google.visualization.PieChart(document.getElementById('visualization')).
      draw(data, {title:"So, how was your day?"});
}
```

Output Debug Code | Run Code

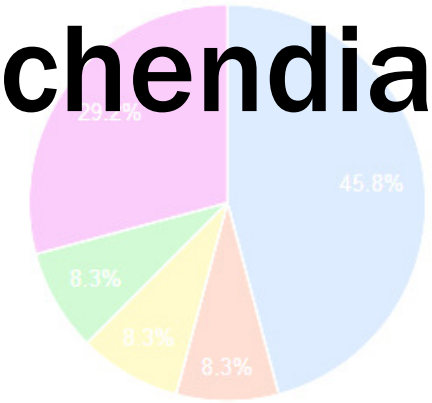


Practice

Programming

Übung am PC

Kuchendiagramme



Practice

Engineering Systems

Software Entwickler:
Alles Einzelkämpfer?

U.S. Department of Justice
United States Marshals Service

WANTED
BY U.S. MARSHALS

AGENCY: Before arrest, validate warrant through National Crime Information Center (NCIC).

Use NCIC entry number: (NIC/ W721460021).

.....MITNICK, KEVIN DAVID

AKS (S): .....MITNIK, KEVIN DAVID
MERRILL, BRIAN ALLEN

DESCRIPTION:

Sex:.....MALE
Race:.....WHITE
Place of Birth:.....VAN NUYS, CALIFORNIA
Date(s) of Birth:.....08/06/63; 10/18/70
Height:.....5'11"
Weight:.....190
Eyes:.....BLUE
Hair:.....BROWN
Skintone:.....LIGHT
Scars, Marks, Tattoos:.....NONE KNOWN
Social Security Number (s):.....550-39-5695
NCIC Fingerprint Classification:...DOPM20PM13DIPM19PM09

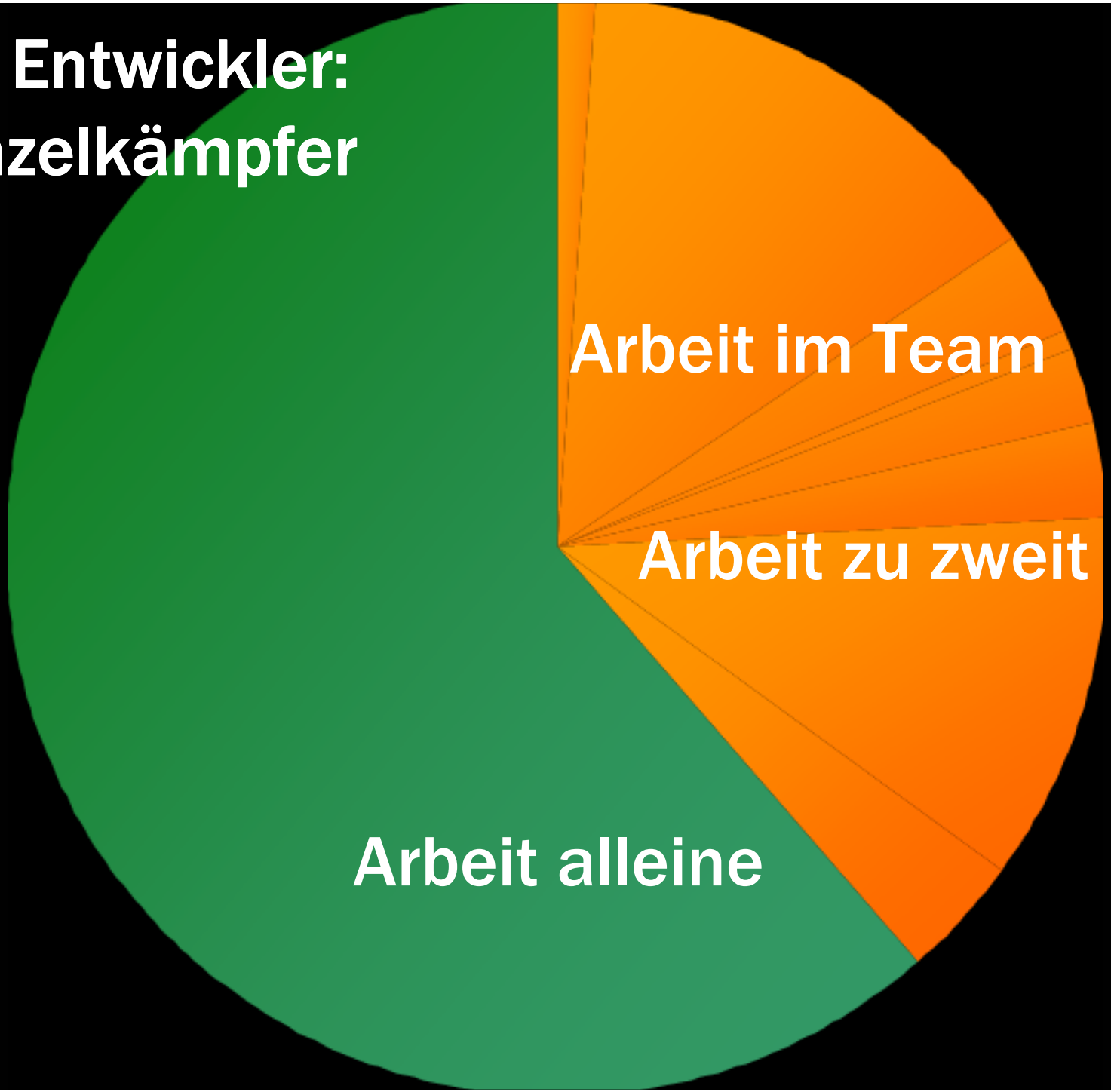
ADDRESS AND LOCALE: KNOWN TO RESIDE IN THE SAN FERNANDO VALLEY AREA OF CALIFORNIA AND LAS VEGAS, NEVADA

Wanted for: Computer fraud.

http://webupon.com/security/convicted-nine-notorious-hackers-of-our-time/

Software Entwickler: Keine Einzelkämpfer

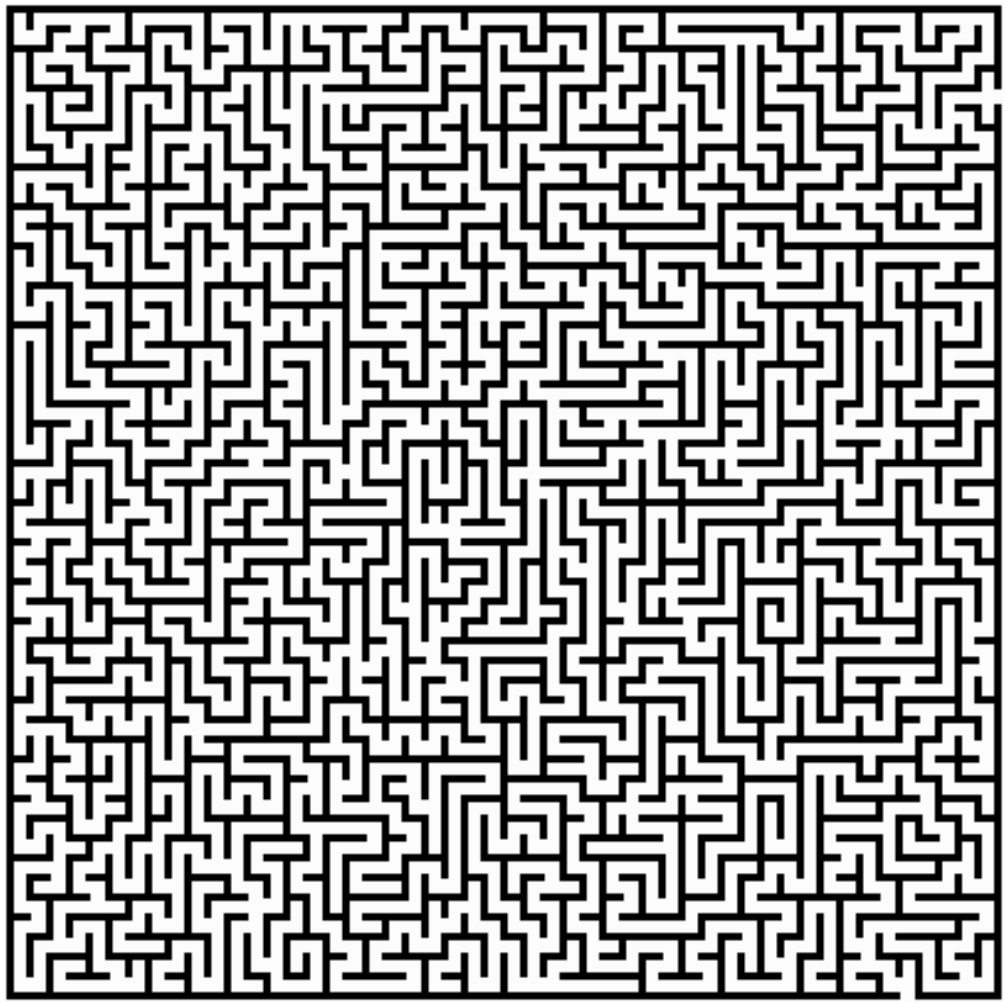
Software Entwickler:
Keine Einzelkämpfer



Practice

Engineering Systems

Ein Beispiel



Practice

Engineering Systems

Information Hinding

David Parnas. On the Criteria to Be Used in Decomposing Systems Into Modules, Communications of the ACM, 1972.

Practice

Engineering Systems

Konzept Schnittstellen

Google code Code Playground

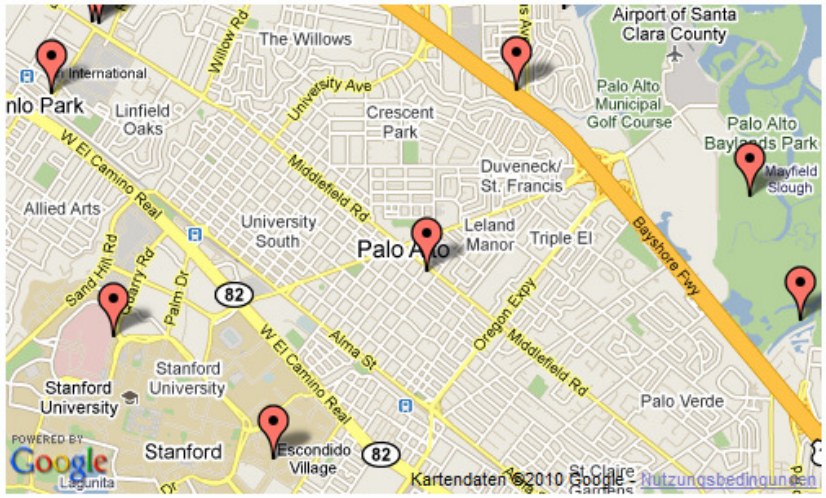
Pick an API Search

- Language
- Blogger
- Blog Search
- Book Search
- Calendar
- Earth New Samples!
- Feeds New Samples!
- Friend Connect
- Image Search New Samples!
- Javascript
- Libraries
- Local Search New Samples!
- Maps
  - Map Examples
    - Map Simple
    - Markers
    - Animate
    - Info Window
  - Event Examples
    - Event Simple
    - Event Context

Edit Code Map Examples » Markers — View Docs | Edit HTML

```
1  function initialize() {
2    if (GBrowserIsCompatible()) {
3      var map = new GMap2(document.getElementById("map_canvas"));
4      map.setCenter(new GLatLng(37.4419, -122.1419), 13);
5
6      // Add 10 markers to the map at random locations
7      var bounds = map.getBounds();
8      var southWest = bounds.getSouthWest();
9      var northEast = bounds.getNorthEast();
10     var lngSpan = northEast.lng() - southWest.lng();
11     var latSpan = northEast.lat() - southWest.lat();
12     for (var i = 0; i < 10; i++) {
13       var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
14                               southWest.lng() + lngSpan * Math.random());
15       map.addOverlay(new GMarker(point));
16     }
17   }
18 }
19
20
```

Output — Debug Code | Run Code



Practice

Engineering Systems

Übung am PC

Maps für die eigene Webseite

Principle

Recollection: Search

Principle

Recollection: Search

Einschränken der Suche

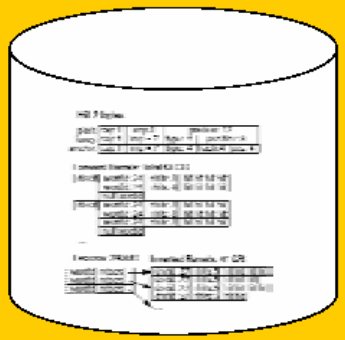
Suchmaschine

1. Crawling/ Spidering

Download und Speicherung aller Dokumente der Kollektion

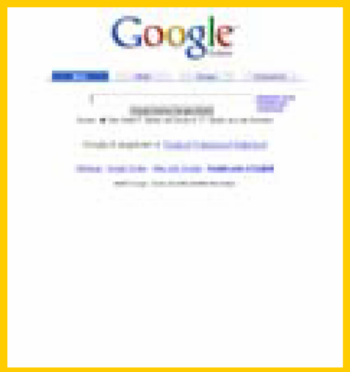
2. Erstellen Index

Erstellen einer effizienten Datenstruktur für die Suche



3. Benutzer-anfrage

Finden passender Dokumente auf eine Benutzeranfrage



4. Resultate

Präsentation der Treffer in der richtigen Reihenfolge

J. Stuker, namics ag

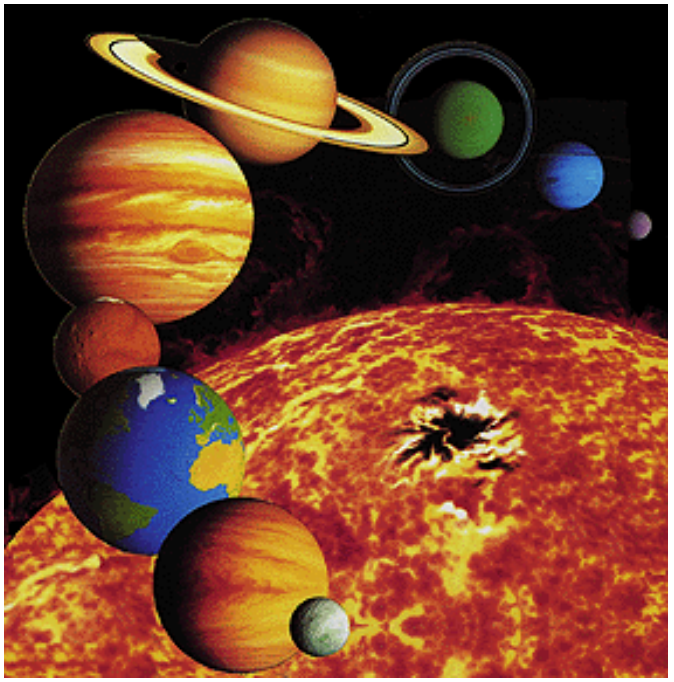
Der Index einer Suchmaschine

Begriffe Vorkommen

mars	chocolate.com/mars.html	geschichte.de/mars.htm	planets.org/list.htm	...	URL
	120	36	3	...	Häufigkeit
	7, 12, 51, ...	21, 33, ...	12, 33, 40, ...		Position
pluto	disney.com/comics.html	planets.org/list.htm	...	...	
	78	15	...	...	
	1, 8, ...	67, 73, ...	...	...	
saturn	sega.com/consoles.html	cars.uk/dealers.html	planets.org/list.htm	...	
	99	10	8	...	
	1, 4, 9, ...	51, 126, ...	80, 85, ...	...	

Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

So arbeitet die Suchmaschine mit dem Index



Gesucht

Übersicht über die Planeten unseres Sonnensystems

Anfrage

mars

Resultat

Alle Webseiten zum Begriff „mars“ aus dem Index, kaum etwas zu Planeten

Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

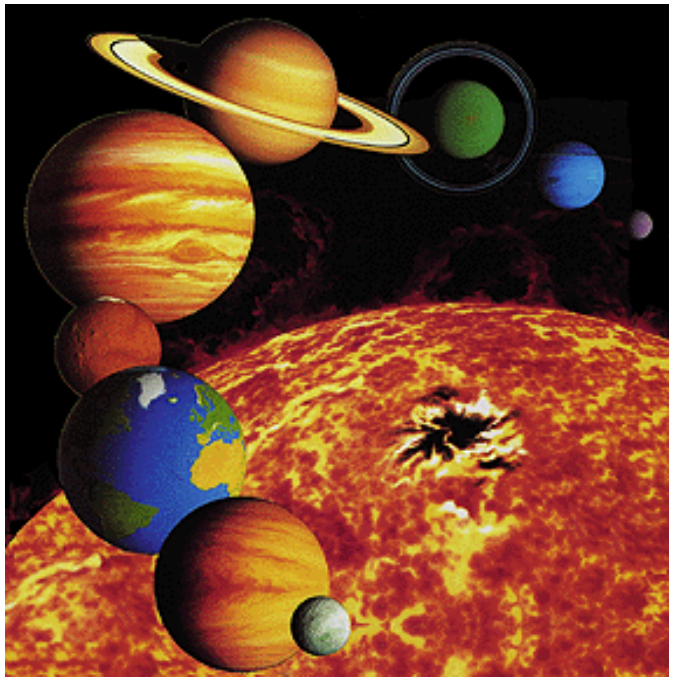
So arbeitet die Suchmaschine mit dem Index

Anfrage: *mars*

Begriffe	Vorkommen			
mars	chocolate.com/mars.html	geschichte.de/mars.htm	planets.org/list.htm	...
	120	36	3	...
	7, 12, 51, ...	21, 33, ...	12, 33, 40, ...	
pluto	disney.com/comics.html	planets.org/list.htm	...	...
	78	15	...	...
	1, 8, ...	67, 73, ...	...	...
saturn	sega.com/consoles.html	cars.uk/dealers.html	planets.org/list.htm	...
	99	10	8	...
	1, 4, 9, ...	51, 126, ...	80, 85, ...	...

Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

So arbeitet die Suchmaschine mit dem Index



Gesucht

Übersicht über die Planeten unseres Sonnensystems

Anfrage

mars pluto saturn

Resultat

Suchmaschine kombiniert Listen zu „mars“, „saturn“ und „pluto“ aus dem Index: Bessere Resultate.

Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

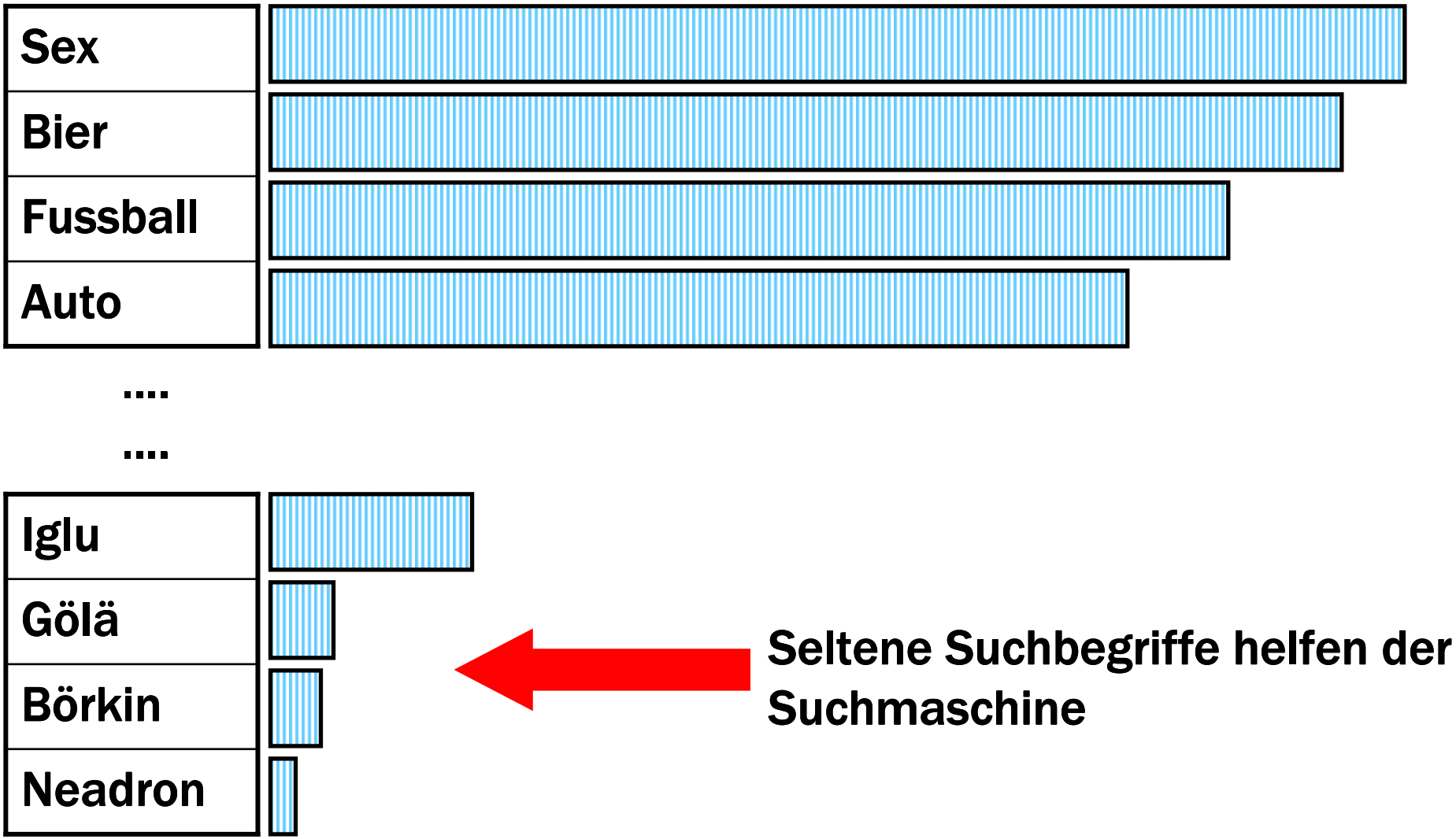
So arbeitet die Suchmaschine mit dem Index

Anfrage: *mars pluto saturn*

Begriffe	Vorkommen			
mars	chocolate.com/mars.html	geschichte.de/mars.htm	planets.org/list.htm	...
	120	36	3	...
	7, 12, 51, ...	21, 33, ...	12, 33, 40, ...	
pluto	disney.com/comics.html	planets.org/list.htm	...	...
	78	15	...	...
	1, 8, ...	67, 73, ...	...	...
saturn	sega.com/consoles.html	cars.uk/dealers.html	planets.org/list.htm	...
	99	10	8	...
	1, 4, 9, ...	51, 126, ...	80, 85, ...	...

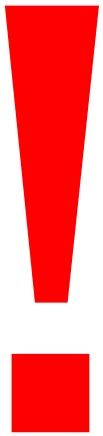
Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

Der Index: wenig häufige Worte



Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

Sie wollen bessere Suchresultate?



Verwenden Sie viele und spezifische Suchbegriffe!

Werner Hartmann, Michael Näf, Peter Schäuble (2000). Informationsbeschaffung im Internet; Grundlegende Konzepte verstehen und umsetzen. Orell Füssli Verlag Zürich.

Tell us what you think | Sign

Google code **Code Playground**

Pick an API Search

- Friend Connect
- Image Search New Samples!
- Javascript
- Libraries
- Local Search New Samples!
- Maps
- Maps V3 New Samples!
- News Search New Samples!
- Patent Search
- Search
- Video Search
- Visualization New Samples!
- Web Search
  - Restrict by CSE
  - Set Google Locale Using Hostname
  - Set Google Locale Restriction
  - Set Language Restriction
  - Restrict by CSE
  - Set Google Locale Using Hostname
  - Set Google Locale Restriction
  - Set Language Restriction
- YouTube

Edit Code Web Search » Restrict by CSE — View Docs | Edit HTML

```
1  /*
2  *  How to restrict a search to a Custom Search Engine
3  *  http://www.google.com/cse/
4  */
5
6  google.load('search', '1');
7
8  function OnLoad() {
9    // Create a search control
10   var searchControl = new google.search.SearchControl();
11
12   // Add in a WebSearch
13   var webSearch = new google.search.WebSearch();
14
15   // Restrict our search to pages from our CSE
16   webSearch.setSiteRestriction('017576662512468239146:omuauf_lfve', 'lectures');
17
18   // Add the searcher to the SearchControl
19   searchControl.addSearcher(webSearch);
20
```

Output — Debug Code | Run Code

design Suchen ×

powered by Google™

Internet

CS 250 VLSI System **Design Lecture** 11

Oct 1, 2009 ... John Wawrzynek and Krste Asanović with John Lazzaro. CS 250. VLSI System **Design**. **Lecture** 11 – DRAM

www-inst.eecs.berkeley.edu/~cs250/ ...

www-inst.eecs.berkeley.edu

Principle

Recollection: Search

Übung am PC

Eine eigene Google Suchmaschine