

Barefoot would like to acknowledge the work of Julia Briggs and the eLIM team at Somerset County Council for their contribution to this poster.

## Inhaltsverzeichnis

- 1 Präludium
- 2 Schlaglicht: Fachwissenschaft
  - Informatisches Denken
  - Beispielhafte Fachwissenschaften
- 3 Schlaglicht: Allgemeinbildung
  - Schulfächer und Denken
  - Allgemeinbildung nach Heymann
  - Fachliches Arbeiten
  - Programmieren für alle
  - Allgemeinbildungsansätze der letzten 50 Jahre
- 4 Schlaglicht: Das Konzept der Roten Fäden
  - Definition von Roten Fäden
  - Beispielhafte Rote Fäden
- 5 Kontext Webdatenbanken
  - Programmiersprachen
  - Datenbanken
  - Netze
  - HTML
  - Betriebssysteme
  - Unterrichtsverlauf
  - Webdatenbanken
- 6 Kompetenzen



**Wirbeltiere: Fische**

Eine Gruppe der Wirbeltiere sind die Fische.  
Sie schwimmen im Wasser.  
Ihr Skelett besteht aus Gräten.  
Die Schwimmblase hilft beim Schwimmen.  
Sie atmen mit Kiemen.  
Fische haben Flossen und Schuppen.

Tage	Menge
0	1000
3	500
6	250
9	125
12	62,5

Reihe1

Tage	Menge
0	1000
-3	2000
-6	4000
-9	8000
-12	16000
-15	32000

Reihe1

www.leer.de

Leben in Leer  
Tourismus & Kultur  
Wirtschaft  
Politik & Verwaltung

Stadt Leer (Ostmeinsland)

Menschen wie sympathisch  
dabei zu sein

• Anzeigen, Besondere  
• Ausstellungen, Veranstaltungen  
• Bücherei  
• Bürgerbüro  
• Bürgerhaushalt  
• Bürgerhaushalt  
• Bürgerhaushalt  
• Bürgerhaushalt

Buchbegleit

Veranstaltungen

Aktuelle aus Leer

- ...
- ...
- ...
- ...
- ...
- ...

Komparieren gesucht

Am Sonntag, den 27. April 2012, 10:30 Uhr ...

Leer best

**INFORMATIK-UNTERRICHT**  
ist ungleich  
Medienkunde oder Digitalkunde  
oder Weiss-der-Teufel-was-kunde

**ABER WAS IST**  
**INFORMATIK-UNTERRICHT**  
**DANN?**

## Computational Thinking

Jeannette Wing 2006



**Computational thinking**  
is a fundamental skill for everyone,  
not just for computer scientists.

## Two A's to C.T. Combined

- Computing is the **automation** of our **abstractions**
  - They give us the audacity and ability to scale.
- Computational thinking
  - choosing the right abstractions, etc.
  - choosing the right "computer" for the task

## Informatisches Denken

**Computational thinking**  
is a fundamental skill for everyone,  
not just for computer scientists.

**Computational thinking** is thinking **recursively**. It is **parallel processing**. It is **interpreting** code as data and data as code. . . .

Computational thinking is **using abstraction and decomposition** when attacking a large complex task . . . .

It is choosing an **appropriate representation** for a problem . . . .

[Wing, 2006, S.33]

## Mathematik

- Numerik, symbolische Berechnungen
- Statistik
- Graphentheorie
- **Diskrete Mathematik** (Aussagenlogik, Kryptologie, Codierungstheorie, Informationstheorie,
- **Algorithmik**

## Physik

- **Elektrik**  
Telefon: Bandbreite/Nyquist, analoge/digitale Übertragung
- **Elektromagnetische Wellen**, Modulationsarten (alte wie neue), Bandbreite
- **Digitalelektronik**
- **Digitalisierung**
- **Informationstheorie** Informationsbegriff, Entropie

## Biologie

- Genetik
- Neuronale Netzwerke
- Künstliches Leben

## Philosophie

- **Mechanistisches Weltbild und Intelligenz**
- **Das Gehirn als informationsverarbeitende Maschine**  
(Philosophie des Geistes, Maschine und Geist, Turingtest)
- **Künstliche Intelligenz** (Elektronengehirn, Supergehirn)
- **Informationstheorie und Informationsbegriff**
- **Ethik**

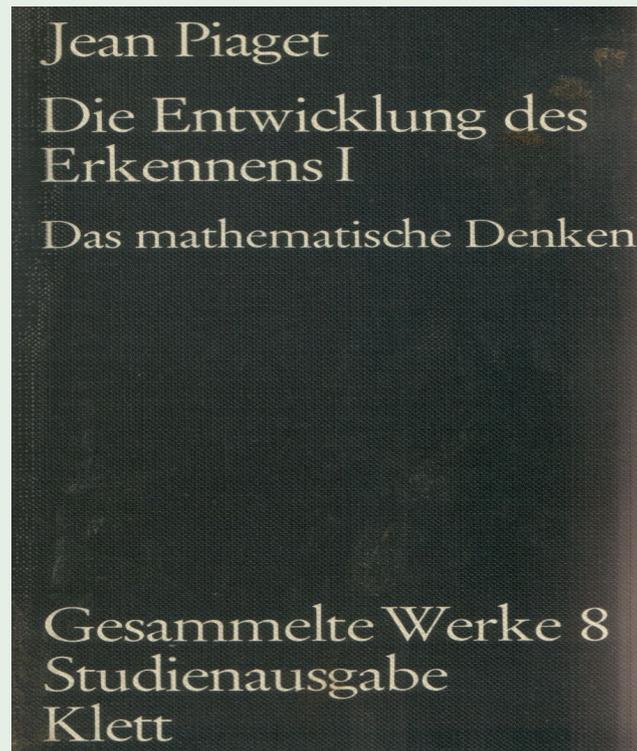
## Sprachwissenschaften

- o **Formale Sprachen**  
*Chomsky-Hierarchie* [Chomsky, 1956]
- o **Natürliche Sprachen**  
... deskriptive Grammatiken und  
... normative Grammatiken
- o **Generative Grammatik** [Chomsky, 1973, S.29ff]  
Eine Sprache zu lernen, erfolgt nicht durch Training, sondern durch kognitive Strukturen im Gehirn. Damit ist der *Behaviorismus* widerlegt durch die **kognitive Wende**.  
(schulisch relevant für die Sprachdidaktik [Aebli, 1976])
- o **Darstellungen von Grammatiken**  
(BNF, [...])

(Fast)

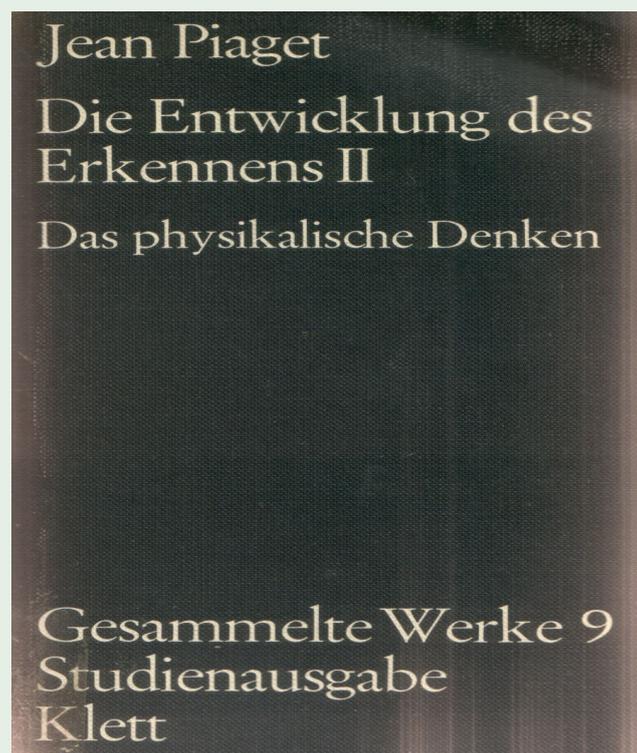
**Alle**  
brauchen  
informatisches Denken!

## Mathematisches Denken



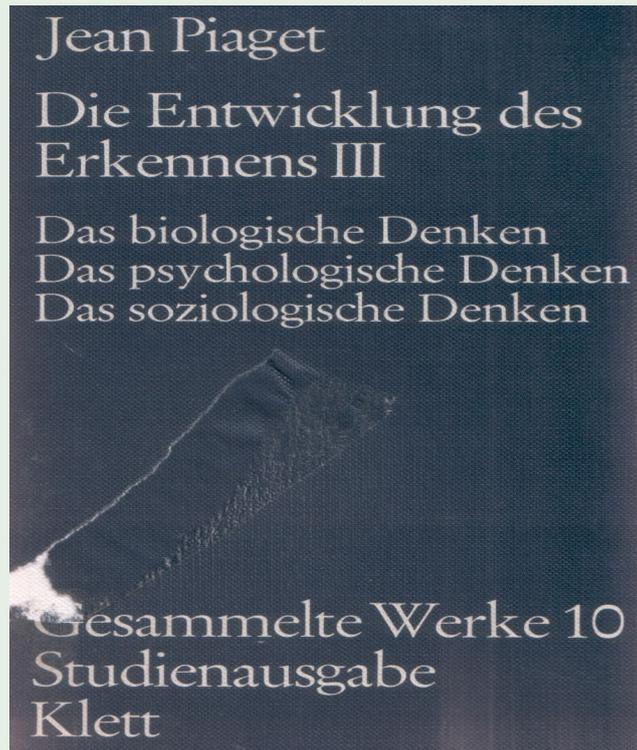
[Piaget, 1975a]

## Physikalisches Denken



[Piaget, 1975b]

## Biologisches und soziale Denken



[Piaget, 1975c]

## Vom (informatischen) Denken zum Curriculum

### Curricula

Aus dieser Begründung für das Fach wird dann alles abgeleitet, was wir für richtig und wichtig für das

**allgemeinbildende Fach Informatik** halten:

- o **Modellierung**
- o ... stellvertretend für die **Abstraktion**
- o **Programmierung/Implementierung**
- o ... stellvertretend für die **Automatisierung**
- o ...

## Curricula enthalten ...

- ... Anteile als Orientierung auf einen möglichen **Beruf**
- ... Anteile im Rahmen von **Weltorientierung**
- ... Anteile zur Förderung und Entwicklung des **demokratischen Bürgers**
- ... Anteile als Anreiz für eine sinnvolle Beschäftigung in der **Freizeit**
- ...

## Curricula enthalten ...

- Anteile als Anforderung aus dem **eigenen Fach**
- Anteile als **Dienstleistung** von und für **andere Fächer**

## Curricula enthalten heute:

- Grosse Anteile **Rechnen** als Dienstleistung für **alle(!) Fächer** und darüber hinaus.
- Anteile in der **Sek II** für die **Physik** (Teil der Analysis) (Klein'sches Programm)
- ...

## ... und müssten heute enthalten:

- Anteil **Graphentheorie** als Dienstleistung für **Informatik** und darüber hinaus.
- **Rechnen in Stellenwertsystemen**  
— vor allem im 2- und 16-System —  
für die **Informatik**.
- **Boolsche Algebra** (??) für die Informatik.
- ...

# Dienstleistung der Schulinformatik

## Curricula enthalten heute:

- Anteile **Programmierung** als Dienstleistung für **Informatik**.
- Anteile **Arbeit** mit und in **Informatiksystemen** (Medienkunde!?)
- ...

## ... und müsste heute enthalten:

- Anteile **Programmierung** als Dienstleistung für **alle (!) Fächer**.

● ...

WIE BITTE???

**Programmierung**  
braucht doch keiner!!!

(schallt es doch überall!)

\*\*\* Nein: \*\*\* Stimmt nicht \*\*\* Überraschende THESE \*\*\*

**Programmierung**  
braucht jeder

(ist die klügere Einsicht!)

## Der Informatiker

Der Informatiker entwickelt:

(oft/zumeist) **Informatiksysteme.**

Dazu benötigt er/sie:

- **Modellierungskompetenzen**
- **Programmierkompetenzen**
- ...

Der Mathematiker löst:

(oft/zumeist) **Probleme** mit Funktionen und Gleichungen. Dazu benötigt er/sie als flexible Tools mit **vorgegebenen Modellierungen**:

- also keine Modellierungskompetenzen
- aber **Programmierkompetenzen**
- ...

Beispiele:

- GnuPlot
- CAS, MaPLa, Mathematica, ...
- Statistik-Programm wie R
- ...

## Beispiel für Mathe-Scripte

GnuPlot-Script:

GnuPlot

Maxima-Script:

Maxima

R-Script:

R

Der Physiker führt:

(oft/zumeist) **Versuche** aus und wertet diese mit Grafiken aus.

Dazu benötigt er/sie als flexible Tools mit **vorgegebenen**

**Modellierungen**:

- also keine Modellierungskompetenzen
- aber **Programmierkompetenzen**
- ...

Beispiele:

- GnuPlot
- ...

Der Biologe oder auch Mediziner führt:

(oft/zumeist) **Versuche** mit vielen Daten aus und wertet diese mit

Grafiken aus und erstellt Statistiken. Dazu benötigt er/sie als

flexible Tools mit **vorgegebenen Modellierungen**:

- also keine Modellierungskompetenzen
- aber **Programmierkompetenzen**
- ...

Beispiele:

- GnuPlot
- Statistik-Programme wie R
- ...

Der Sozialwissenschaftler führt:

(oft/zumeist) **empirische Befragungen** mit vielen Daten aus und wertet diese mit Grafiken aus und erstellt Statistiken. Dazu benötigt er/sie als flexible Tools mit **vorgegebenen Modellierungen**:

- also keine Modellierungskompetenzen
- aber **Programmierkompetenzen**
- ...

Beispiele:

- GnuPlot
- Statistik-Programme wie R
- ...

(Fast)

**Alle**  
können gut  
Programmierkompetenzen  
verwenden.

## Qualifikationen

„Wir gehen also von den Annahmen aus, daß **in der Erziehung** Ausstattung zur **Bewältigung von Lebenssituationen** geleistet wird;

daß diese Ausstattung geschieht ... durch die **Aneignung von Kenntnissen, Einsichten, Haltungen und Fertigkeiten** ... ;

und daß eben die **Curricula** ... zur Vermittlung derartiger **Qualifikationen** bestimmt sind.“

*[Robinsohn, 1967, S.45]*

## Experten und Laien

„Der **Dialog** zwischen Experten und Laien setzt

... **allgemeingebildete Experten** voraus,

die über den Zaun ihres Spezialgebiets hinüberzuschauen vermögen,

und **allgemeingebildete Laien** , die zumindest soviel von den Fragen verstehen, für deren Lösung sie Experten brauchen, daß sie ihnen nicht blind vertrauen müssen.“

*[Heymann, 1996, S.113/114]*

### Kompetenzen

„Dabei versteht man unter **Kompetenzen** die bei Individuen verfügbaren oder durch sie erlernbaren **kognitiven Fähigkeiten und Fertigkeiten** ,

um bestimmte Probleme zu lösen, sowie die damit verbundenen **motivationalen, volitionalen und sozialen** Bereitschaften und **Fähigkeiten**

um die **Problemlösungen in variablen Situationen** erfolgreich und verantwortungsvoll nutzen zu können“.

[Weinert, 2001, S.27/28]

## Informatik für alle

Ja und?

Was bedeutet das alles für das Schulfach Informatik?

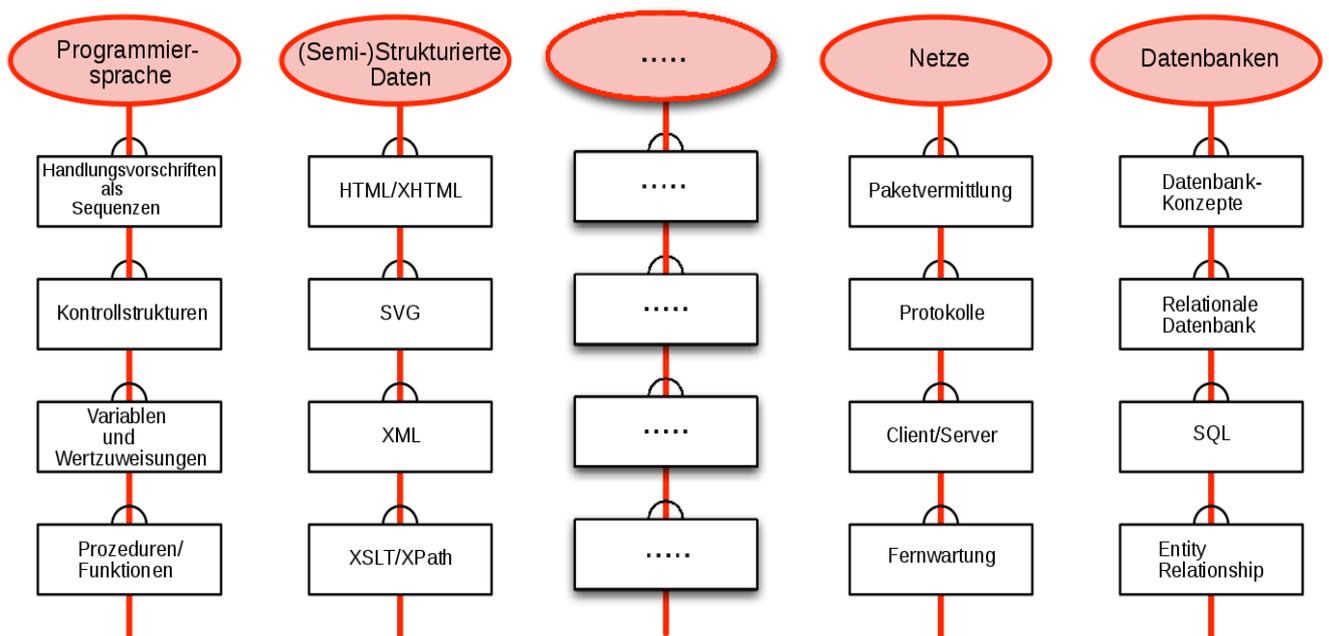


## Definition **Roter Faden**

Ein **Roter Faden** ist eine Anordnung unterrichtlicher Gegenstände, die den folgenden Kriterien genügt:

- Kohärenz
- Transparenz
- Varianz
- Redundanz

## Beispielhafte **Rote Fäden**



Beispielhafte **Rote Fäden** im Unterricht

Wichtig: Die Auswahl der **Roten Fäden** obliegt z.B.(?) dem Fachkollegen bei der Erstellung des Curriculums.

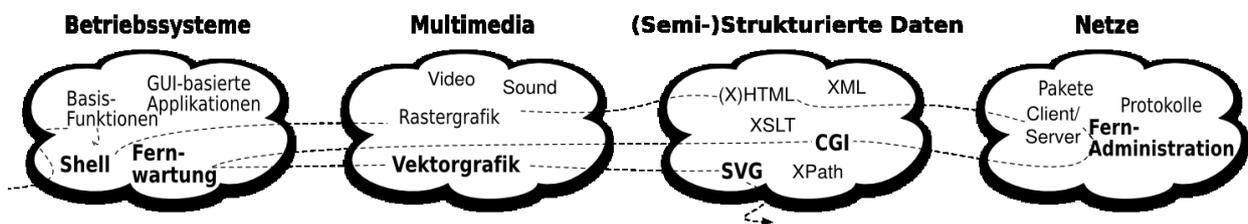
[Pasternak and Vahrenhold, 2009], [Pasternak, 2013, S.138]

## Definition Verwobenes Unterrichten

**Verwobenes Unterrichten** ist das Unterrichten eines gegebenen Curriculums durch eine Auswahl an *Roten Fäden*, die so häufig wie möglich in Unterrichtseinheiten oder Konzepten (teilweise) *gemeinsam* betrachtet und untersucht werden.

[Pasternak and Vahrenhold, 2010], [Pasternak, 2013, S.141]

## Beispiel:



Der verwobene **Rote Faden** *Programmierung* im Geflecht anderer Fäden.

## Ein Beispiel aus dem 6. Jahrgang

### Grafiken

- **Multimedia (Grafik):**  
Arbeiten mit Rastergrafiken im XPM-Format
- **(Semi-)strukturierte Daten:**  
Integration derartiger Grafiken in eine kleine Webseite
- **Betriebssysteme:**  
Konvertieren zwischen Formaten.

### Variablen

- **Programmiersprache:**  
Erzeugen andersfarbiger Autos

## Prozessbereiche

Inhaltsbereiche	Modellieren und Implementieren	Begründen und Bewerten	Strukturieren und Vernetzen	Kommunizieren und Kooperieren	Darstellen und Interpretieren
	Information und Daten	shell-Skripte			Kommandozeile
	Algorithmen	shell-Skripte			
	Sprachen und Automaten			Installation Fernzugriff	
	Informatiksysteme	Installation		Webserver Installation Fernzugriff	
	Informatik, Mensch und Gesellschaft				Webserver

Inhalts- und Prozessbereiche entlang des **Roten Fadens Betriebssysteme** realisiert in verschiedenen Einheiten. [Pasternak, 2013, S.140]

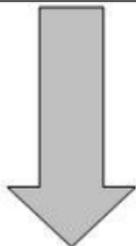
Arno Pasternak

Computational Thinking

(37/65)



Benutzer  
Resucher öffnet Ihre



Anfrage an Webdatenbank

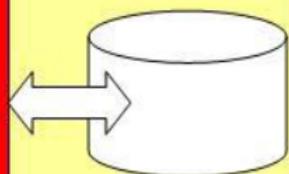
Aufbereitete Daten

## WEBDATENBANKEN

als Beispiel für einen  
(kompetenzorientierten)  
(Gesamt-) Kontext

für den Unterricht in der Sek II

## Webdatenbank



## Notwendig

### „Übliche“ Kenntnisse über

- Variablen
- Prozedurkonzept
- Objekte, Klassen, Module
- Kontrollstrukturen
- ...

## Ergänzend

### Kenntnisse über

- Dateiverwaltung
- Prozesssteuerung
- ...

Es muss **dieselbe Programmiersprache** verwendet werden wie im sonst **üblichen Unterricht** (z.B. Java/Groovy).

## Notwendig

### „Übliche“ Kenntnisse über

- Relationale Datenbanken
- SQL
- speziell: Select-Anweisung
- ...

## Evtl. Ergänzend

### Kenntnisse über

- Eingabe per Script
- Ausgabe als HTML-Fragment
- ...

Es muss **dieselbe Datenbank** verwendet werden wie im sonst **üblichen Unterricht** (z.B. SQLite).

## Notwendig

### „Übliche“ Kenntnisse über

- Aufbau des Internets
- Client-Server-Prinzip
- Remote-Zugriff
- ...

## Evtl. Ergänzend

### Kenntnisse über

- Rolle von IP und Port
- http-Protokoll
- Aufruf von Servern bzw. Daemonen
- ...

## Notwendig

### „Übliche“ Kenntnisse über

- Aufbau einer HTML-Seite (Head, Body)
- Einfache Dokumentenstrukturmittel (Überschriften, ...)
- Unterschied: Aufruf lokal/remote
- ...

## (Evtl.) Ergänzend

### Kenntnisse über

- Formulare
- Statische und dynamische Seiten
- ...

**CSS, Javascript, Ajax** etc. sind in diesem Kontext **nicht nötig**, sie verwirren hier nur.

## Notwendig

### „Übliche“ Kenntnisse über

- Arbeit mit der Konsole
- Fernkonsole
- ...

## (Evtl.) Ergänzend

### Kenntnisse über

- Parallele Prozesse
- Umgebungsvariablen
- ...

Es zeigt sich auch hier, dass es **sinnvoll** ist, **direkt auf der Konsole arbeiten** zu können.

(Beispiele: Aufruf des Java-Compilers, Start eines Java-Programmes)

# HTML statisch

## Erste HTML-Seite

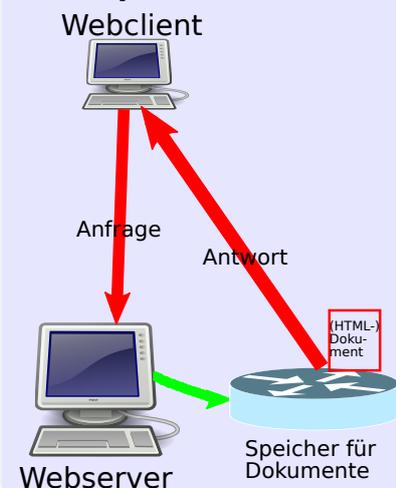
```
<html>

<head>
<title>Unsere erste Seite</title>
</head>

<body>
<h1>Unsere tolle erste Seite</h1>
Die Sch&uuml;ler waren faul.
</body>
</html>
```

## Statische Webseite

### Arbeitsweise des http - Protokolls



Die **Rolle des Browsers** und der Unterschied zwischen Laden einer Datei auf dem **lokalen Computer** und per **http-Protokoll** muss klar werden. (Das ist nicht selbstverständlich.)

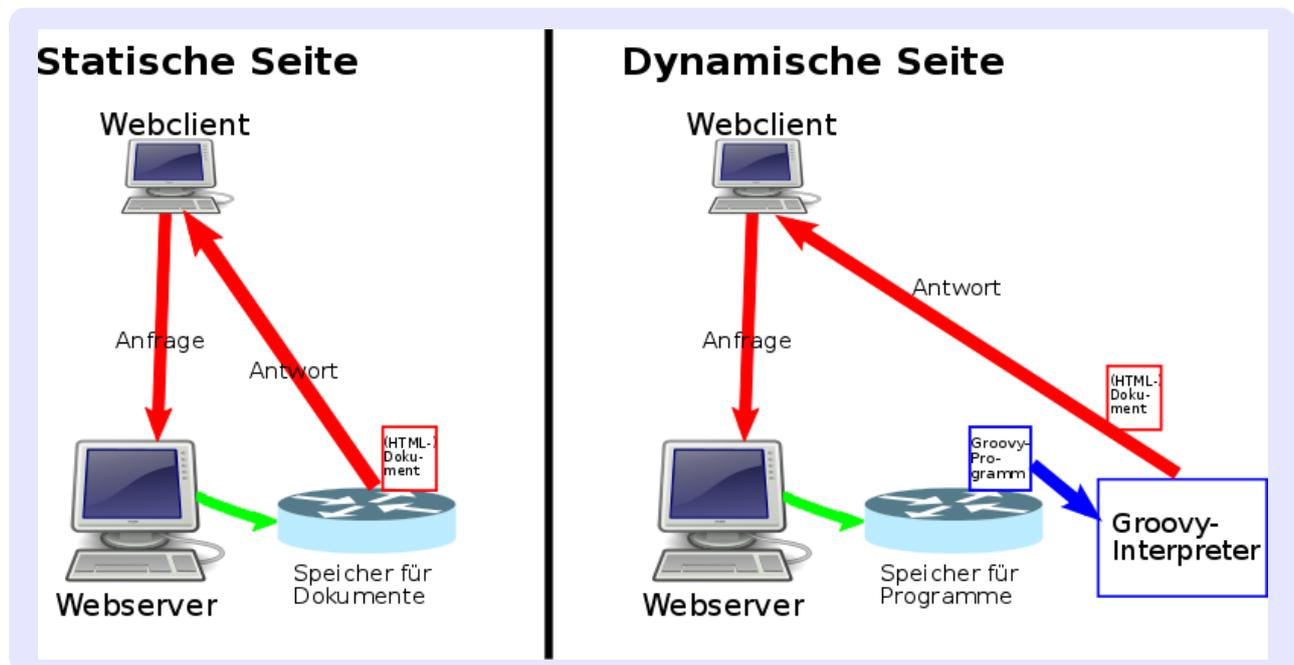
Notwendig: **Start** des (didaktischen) http-Servers: `maya.tcl` bzw. `maya.exe`

Andere Alternativen: `thttpd` (Linux), `mongoose` (Windows)

## Erste dynamische HTML-Seite

```
// PROGRAMM moin0 (in Groovy);
println("Content-type: text/html");
println();
println("<html>");
println("<head>");
println("<title>Guten Tag</title>");
println("</head>");
println("<body>");
println("Hallo allerseits!");
println("</body>");
println("</html>");
// ENDE
```

## Statische und dynamische Webseiten



```
// PROGRAMM farbseite1
String parameterzeile
String farbe
int stelle

parameterzeile = System.getenv("QUERY_STRING")
println ("Parameterzeile: "+parameterzeile)

farbe=""; stelle=6;
while (stelle < parameterzeile.length()){
    buchstabe = parameterzeile.charAt(stelle);
    farbe = farbe + buchstabe;
    stelle = stelle + 1;
}
println ("Farbe ist: "+farbe)

println ("Content-type: text/html");
println ();
println("<html>");
println("<head>");
println("<title>Guten Tag</title>");
println("</head>");
println("<body bgcolor=#"+farbe+">");
println("Hallo allerseits!");
println("</body>");
println("</html>");
// ENDE
```

## HTML Formular

```
<html>
<head>
<title>Seite mit CGI-Groovy-Aufruf</title>
</head>
<body>
<h1>Farbseite – Groovy-Version</h1>
<h2>Es kann der Farbton als hexadezimale Zahl
mit sechs Ziffern eingegeben werden.</h2>
<hr>

<form method="get" action="/cgi/farbseite1.groovy">
    RGB-Farbe: <input name="farbe" type="text"
                "maxlength="6"><br>
    <input type="submit" value="Anfragen">
</form>

</body>
</html>
```

## Ein Skript für SQLite

```
.open adressen1a.db
.output antwort.sql
.header on
.mode html

select * from adressentabelle where stadt='Hagen';
```

# Web-Datenbanken mit SQLite

## Erforderliche Schritte

```
//Programm Datenbank2b
String stadt

stadt = parameterzeile_lesen()
anfrage_schreiben(stadt)
sqlite_aufruf_schreiben()
sqlite_aufrufen()
dokument_ausgeben()
```

## Formular-HTML-Seite

```
<html>
<head>
<title>Seite mit Groovy-Aufruf unserer Datenbank</title>
</head>
<body>
<h1>Datenbank Beispiel 2gb</h1>
<h2>Es wird der Inhalt der Datenbank
    f&uuml;r eine Stadt ausgegeben.</h2>

<form method="get"
    action="/cgi/sql/datenbank2b.groovy">
    Stadt: <input name="stadt" type="text"
        size="15" maxlength="30"><br>
    <input type="submit" value="Anfragen">
</form>
</body>
</html>
```

**Kompetenzen**

**Hard Skills**

**Fachkompetenz**

→ Berufstypische Qualifikationen

**Individualkompetenz**

→ Selbstbeobachtung  
→ Selbstdisziplin

**Sozialkompetenzen**

→ Teamfähigkeit  
→ Empathie

**Methodenkompetenzen**

→ Probleme strukturiert angehen  
→ Zeitmanagement

**Soft Skills**

## Kompetenzen

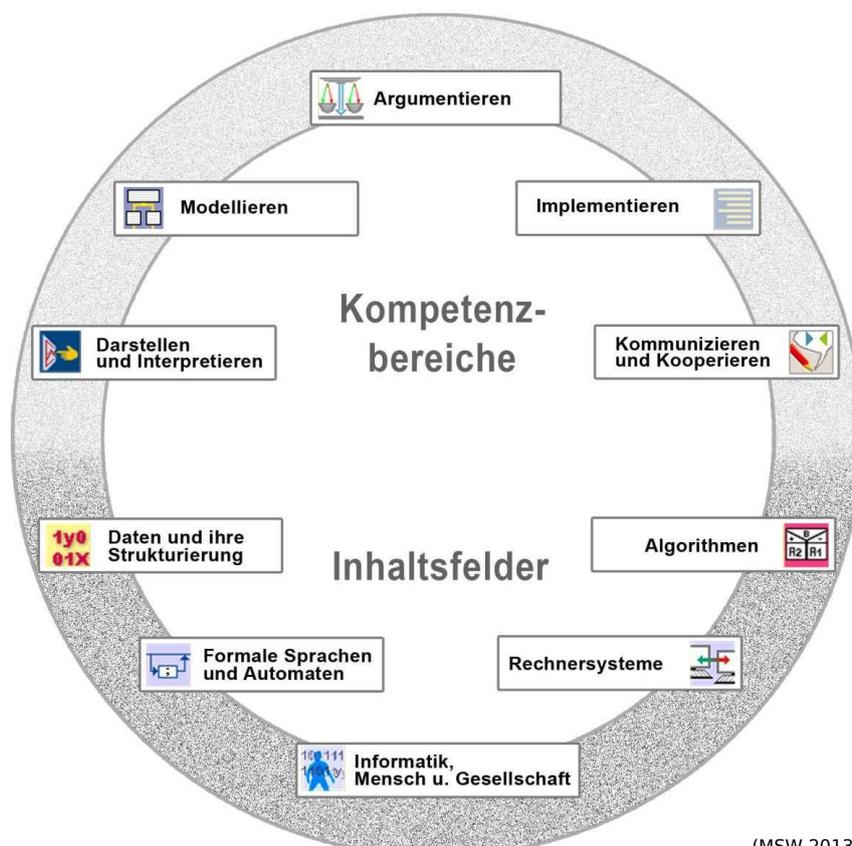
„Dabei versteht man unter **Kompetenzen** die bei Individuen verfügbaren oder durch sie erlernbaren **kognitiven Fähigkeiten und Fertigkeiten** ,

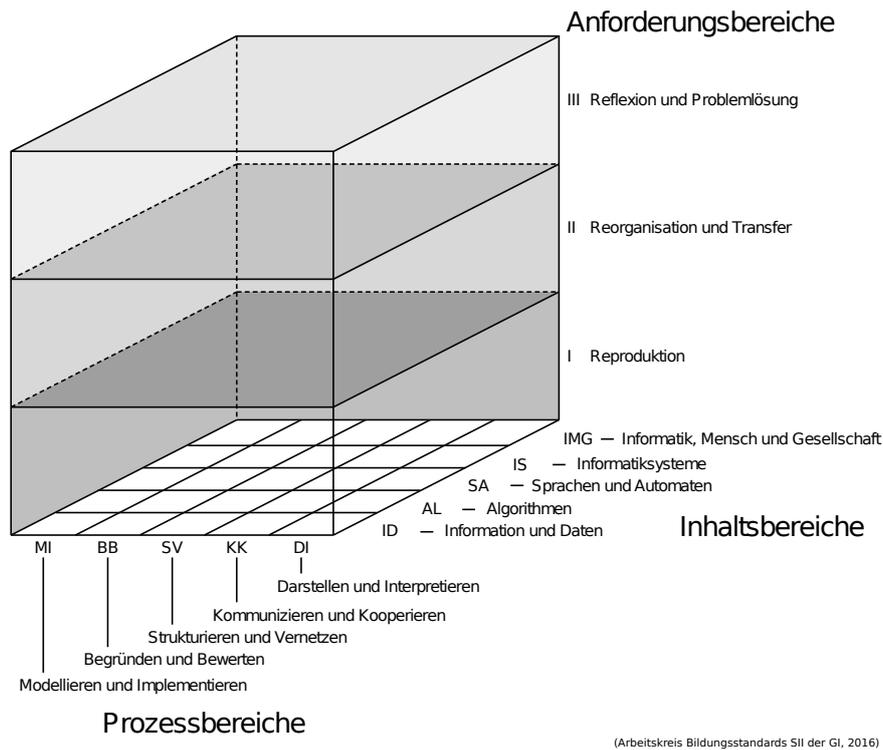
um bestimmte Probleme zu lösen, sowie die damit verbundenen **motivationalen, volitionalen und sozialen** Bereitschaften und **Fähigkeiten**

um die **Problemlösungen in variablen Situationen** erfolgreich und verantwortungsvoll nutzen zu können“.

[Weinert, 2001, S.27/28]

## NRW 2013: Kompetenz-Lehrplan SII





## Computational Thinking

Jeannette Wing 2006



**Computational thinking** is a fundamental skill for everyone, not just for computer scientists.

Jeannette Wing 2006



**Computational thinking** is the **automation** of our **abstractions**.

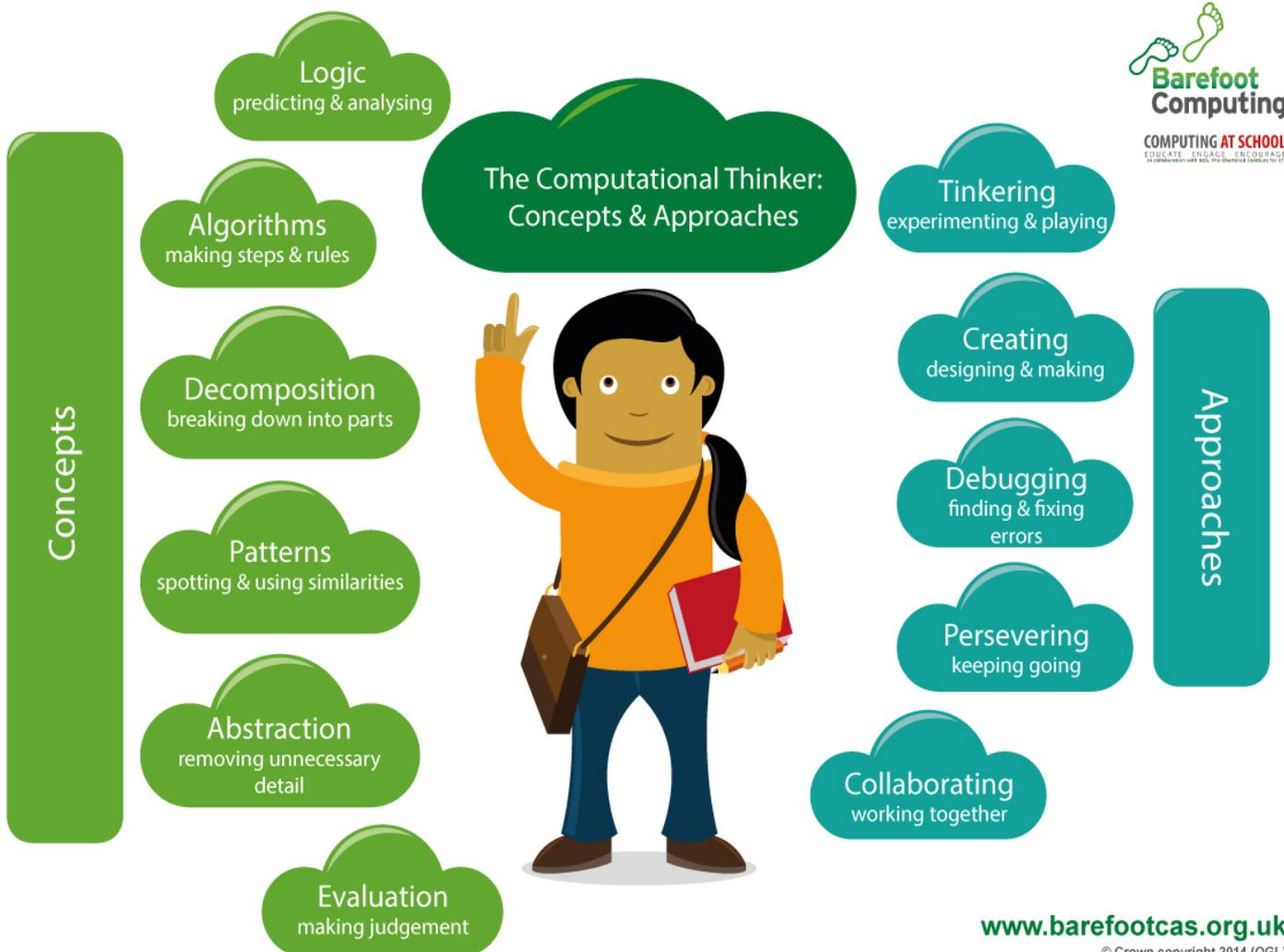
# HOLZHACKEN

ist deshalb so beliebt,  
weil man bei dieser Tätigkeit den

## ERFOLG

sofort sieht.

*Albert Einstein*





Aebli, H. (1976).

*Grundformen des Lehrens: Eine allgemeine Didaktik auf Kognitionspsychologischer Grundlage.*  
Pädagogik - Einführung. Klett, 9. stark erweiterte und umgearbeitete auflage edition.



Arbeitskreis „Bildungsstandards SII“ der GI (2016).

*Bildungsstandards Informatik für die Sekundarstufe II.*

LOGIN-Verlag, Berlin.

Beilage zur LOGIN Nr. 183/184

Im Internet [Stand: 2016-06-09]:

[http://informatikstandards.de/docs/Bildungsstandards\\_SII.pdf](http://informatikstandards.de/docs/Bildungsstandards_SII.pdf).



Chomsky, N. (1956).

Three models for the description of language.

*IRE Transactions on Information Theory*, 2:113–124.



Chomsky, N. (1973).

*Aspekte der Syntax-Theorie.*

Suhrkamp, Frankfurt.



Heymann, H. W. (1996).

*Allgemeinbildung und Mathematik.*

Studien zur Schulpädagogik und Didaktik ; 13. Beltz, Weinheim [u.a.].



MSW NRW (2013).

*Kernlehrplan für die Sekundarstufe II, Gymnasium/Gesamtschule in  
Nordrhein-Westfalen, Informatik.*

Düsseldorf.



Pasternak, A. (2013).

*Fach- und bildungswissenschaftliche Grundlagen für den Informatikunterricht in  
der Sekundarstufe I.*

PhD thesis, Westfälische Wilhelms-Universität, Münster.



Pasternak, A. and Vahrenhold, J. (2009).

Rote fäden und kontextorientierung im informatikunterricht.

Arno Pasternak

Computational Thinking

(65/65)

In Peters, I.-R., editor, *Informatische Bildung in Theorie und Praxis*, pages 45–56,  
Berlin. LOG IN Verlag.



Pasternak, A. and Vahrenhold, J. (2010).

Braided teaching in secondary cs education: Contexts, continuity, and the role of  
programming.

In Cortina, T. and Walker, E., editors, *Proceedings of the 41st ACM Technical  
Symposium on Computer Science Education (SIGCSE 2010)*, pages 204–208. ACM,  
ACM Press.



Piaget, J. (1975a).

*Die Entwicklung des Erkennens I. Das mathematische Denken.*

Gesammelte Werke (Studienausgabe). Klett Cotta Verlag.



Piaget, J. (1975b).

*Die Entwicklung des Erkennens II: Das physikalische Denken.*

Gesammelte Werke (Studienausgabe). Klett Cotta Verlag.



Piaget, J. (1975c).

*Die Entwicklung des Erkennens III.: Das biologische Denken. Das psychologische  
Denken. Das soziologische Denken.*

Gesammelte Werke (Studienausgabe). Klett Cotta Verlag.



Robinsohn, S. B. (1967).

*Bildungsreform als Revision des Curriuculum.*

Luchterhand Verlag, Neuwied.



Weinert, F. (2001).

Vergleichende leistungsmessung in schulen - eine umstrittene selbstverständlichkeit.

*Leistungsmessung in Schulen*, pages 17–31.



Wing, J. M. (2006).

Computational thinking.

*Communications of the ACM*, 49(3):33–35.

Arno Pasternak

Computational Thinking

(65/65)



Wing, J. M. (2008).

Computational thinking and thinking about computing.

*Philosophical Transactions of The Royal Society*, (366):3717–3725.