Eberswalde University
for Sustainable
Development

SIDT
SOCIAL INNOVATION & DIGITAL TRANSFORMATION

Unit 1 – Basics in computer sciences

Prof. Dr. Britta M. Gossel

Unit 1 – Basics in computer sciences

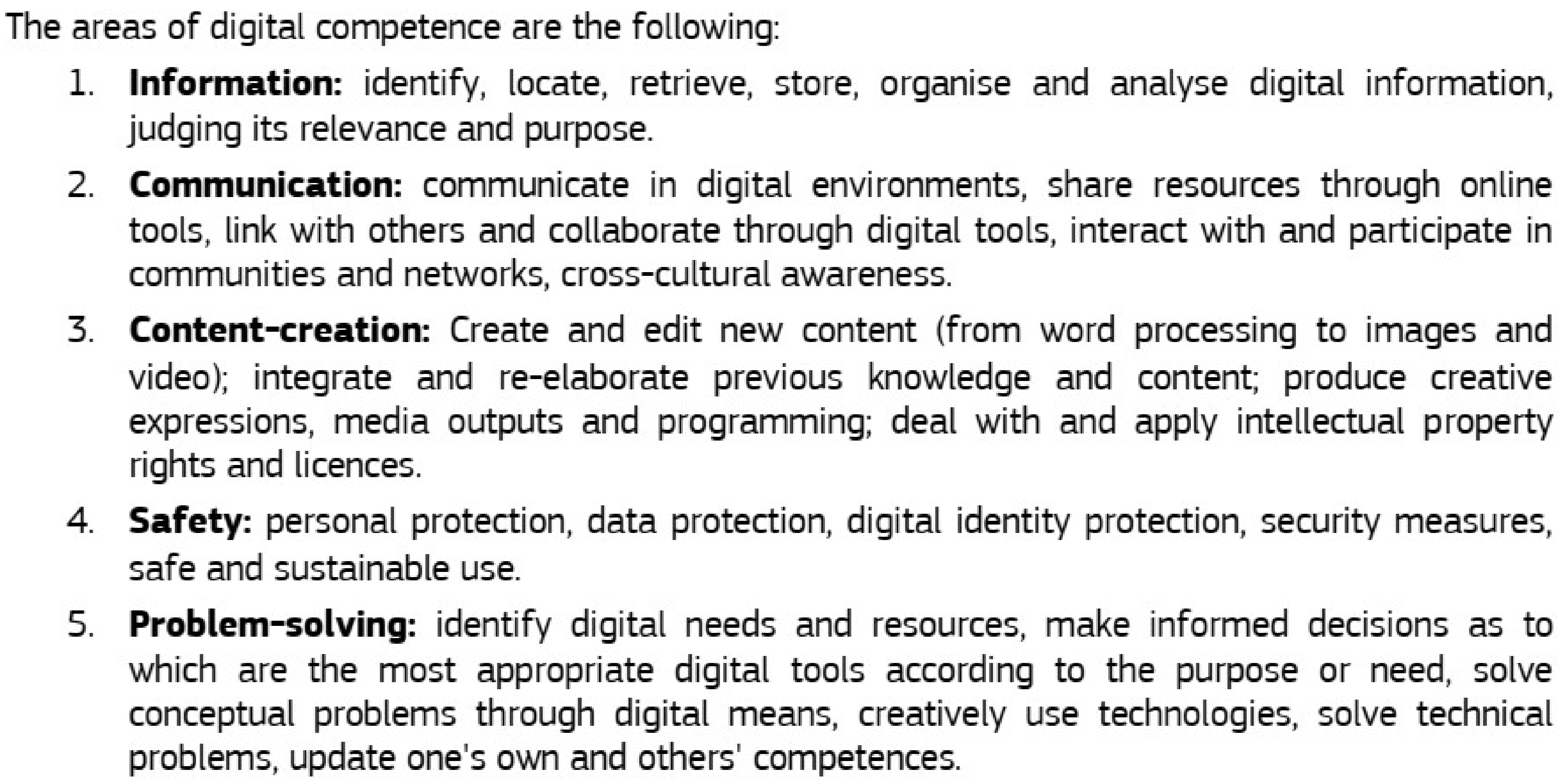
1. Digital Literacy – Why to understand those basics?
2. The computer: the tool that can do more
3. In here, out there: Information processing
4. … a sequence of 0001110101111 ?
5. All follows code

Digital Literacy – Why to understand those basics?

Digital Competence

“Digital Competence is the set of knowledge, skills and attitudes […] that are required when using ICT and digital media to perform tasks, solve problems, communicate, manage information, collaborate, create and share content and build knowledge effectively, efficiently, appropriately, critically, creatively, autonomously, flexibly, ethically, reflectively for work, leisure, participation, learning, socializing, consuming and empowerment.” (Ferrari 2012, 3ff.)

Digital Literacy – Why to understand those basics?

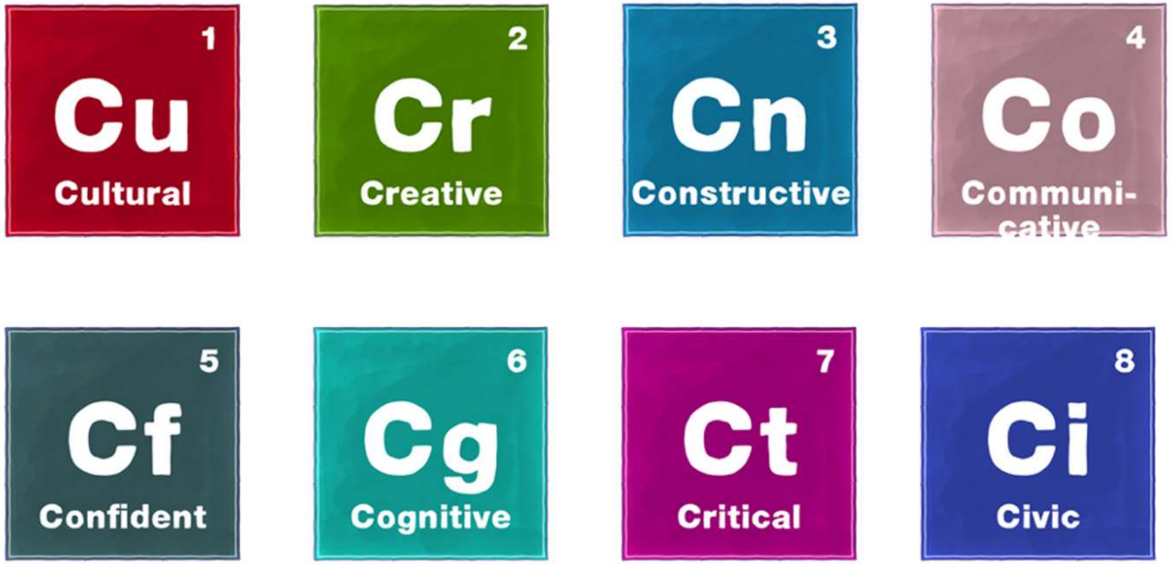


The areas of digital competence are the following:

1. **Information:** identify, locate, retrieve, store, organise and analyse digital information, judging its relevance and purpose.
2. **Communication:** communicate in digital environments, share resources through online tools, link with others and collaborate through digital tools, interact with and participate in communities and networks, cross-cultural awareness.
3. **Content-creation:** Create and edit new content (from word processing to images and video); integrate and re-elaborate previous knowledge and content; produce creative expressions, media outputs and programming; deal with and apply intellectual property rights and licences.
4. **Safety:** personal protection, data protection, digital identity protection, security measures, safe and sustainable use.
5. **Problem-solving:** identify digital needs and resources, make informed decisions as to which are the most appropriate digital tools according to the purpose or need, solve conceptual problems through digital means, creatively use technologies, solve technical problems, update one's own and others' competences.

Ferrari (2013)

Digital Literacy – Why to understand those basics?



Acht Elemente von Digital Literacy. Erstellt von Christian Friedrich mit dem Externer Link: Remixer der Externer Link: Visual Thinkery, der unter CC-BY-SA 4.0 lizenziert ist. Lizenz: cc by-sa/4.0/deed.de

Digital Literacy – Why to understand those basics?

D21-
Digital-Index
2022/23

Jährliches Lagebild zur
Digitalen Gesellschaft

German study about digital society

Quelle: DI21 Index 2022/2023

Digital Literacy – Why to understand those basics?

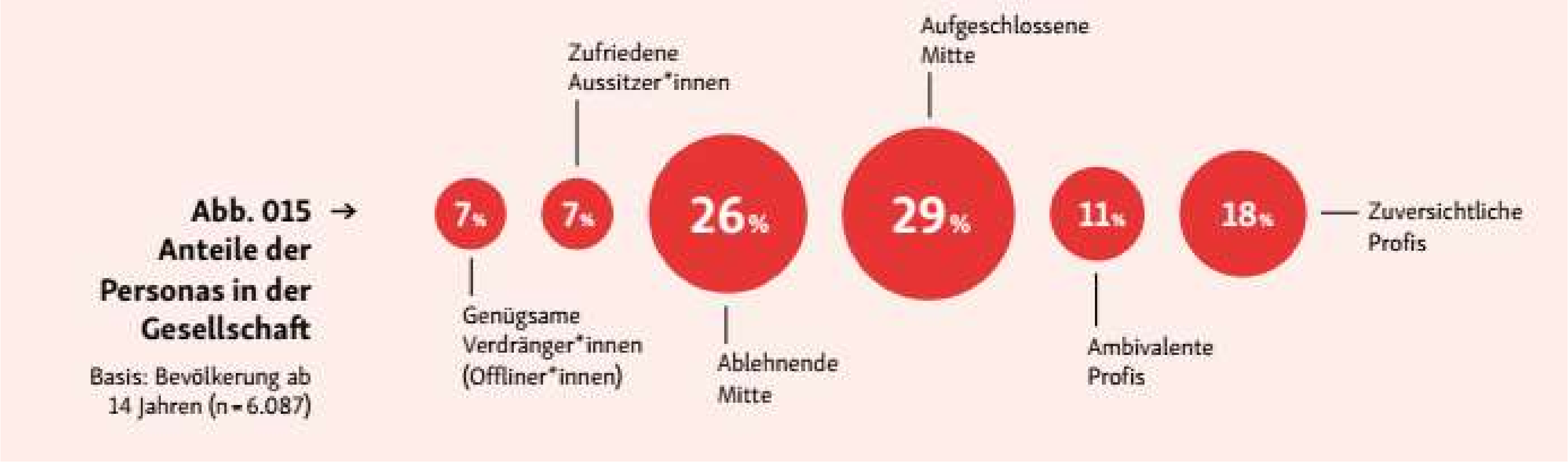


Abb. 015 →
Anteile der Personas in der Gesellschaft

Basis: Bevölkerung ab 14 Jahren (n = 6.087)

Quelle: DI21 Index 2022/2023

Eberswalde University
for Sustainable
Development

Digital Literacy – Why to understand those basics?

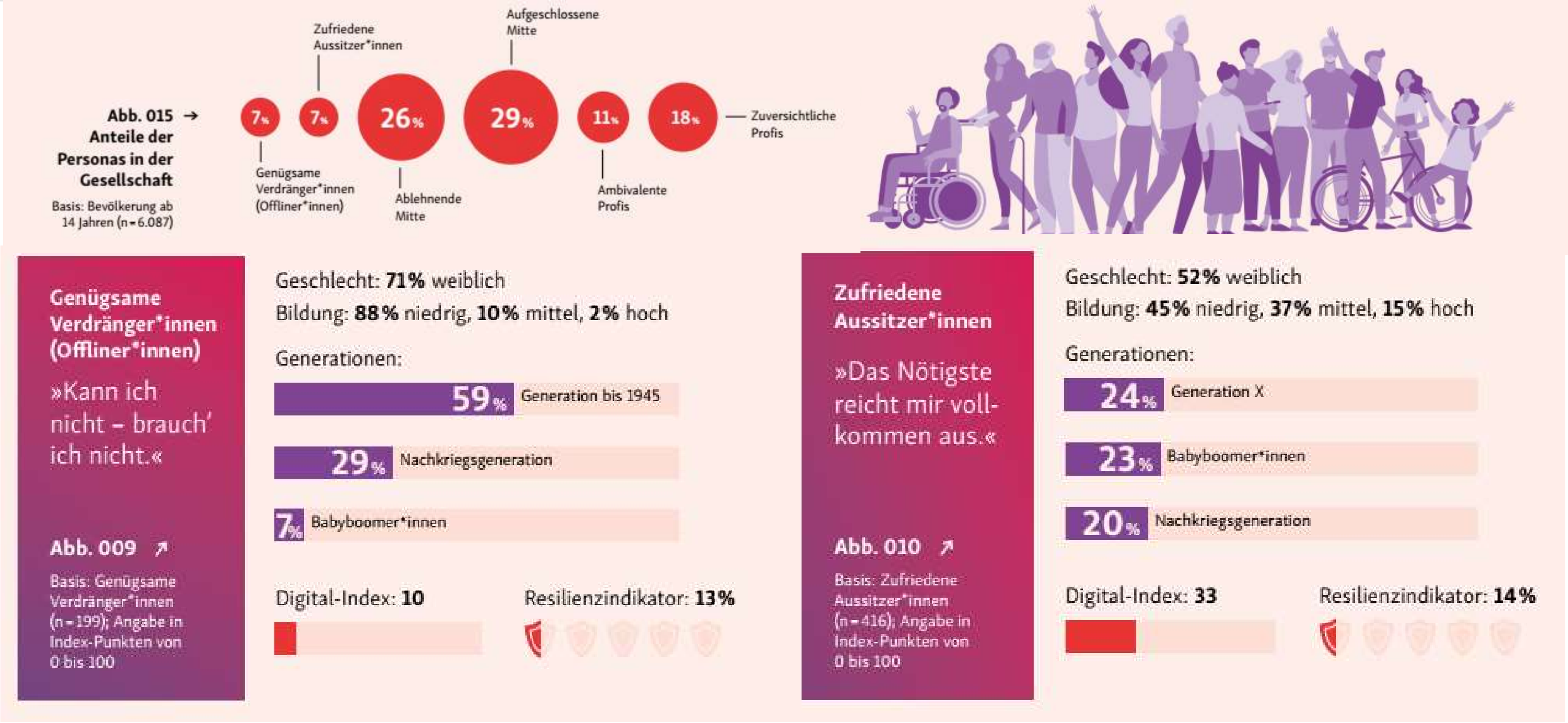


Abb. 015 →
Anteile der Personas in der Gesellschaft
Basis: Bevölkerung ab 14 Jahren (n=6.087)

Persona	Anteil
Genügsame Verdränger*innen (Offliner*innen)	7%
Zufriedene Aussitzer*innen	7%
Ablehnende Mitte	26%
Aufgeschlossene Mitte	29%
Ambivalente Profis	11%
Zuversichtliche Profis	18%

Genügsame Verdränger*innen (Offliner*innen)

»Kann ich nicht – brauch' ich nicht.«

Abb. 009 ↗

Basis: Genügsame Verdränger*innen (n=199); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **71%** weiblich
Bildung: **88%** niedrig, **10%** mittel, **2%** hoch

Generationen:

59%	Generation bis 1945
29%	Nachkriegsgeneration
7%	Babyboomer*innen

Digital-Index: **10**

Resilienzindikator: **13%**

Zufriedene Aussitzer*innen

»Das Nötigste reicht mir vollkommen aus.«

Abb. 010 ↗

Basis: Zufriedene Aussitzer*innen (n=416); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **52%** weiblich
Bildung: **45%** niedrig, **37%** mittel, **15%** hoch

Generationen:

24%	Generation X
23%	Babyboomer*innen
20%	Nachkriegsgeneration

Digital-Index: **33**

Resilienzindikator: **14%**

Quelle: DI21 Index 2022/2023

Digital Literacy – Why to understand those basics?

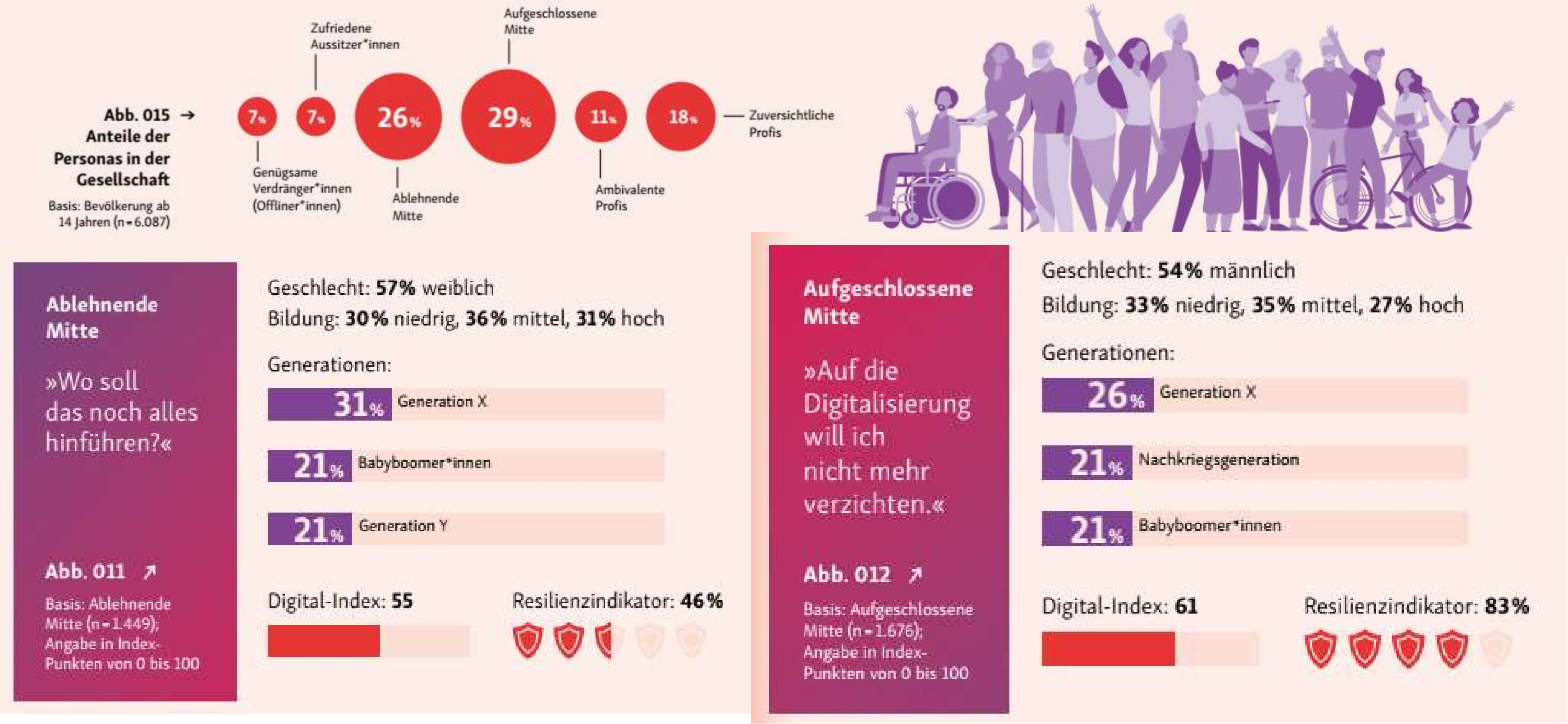


Abb. 015 →
Anteile der Personas in der Gesellschaft
Basis: Bevölkerung ab 14 Jahren (n=6.087)

Ablehnende Mitte

»Wo soll das noch alles hinführen?«

Abb. 011 ↗
Basis: Ablehnende Mitte (n=1.449); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **57%** weiblich
Bildung: **30%** niedrig, **36%** mittel, **31%** hoch

Generationen:

31%	Generation X
21%	Babyboomer*innen
21%	Generation Y

Digital-Index: **55**

Resilienzindikator: **46%**

Aufgeschlossene Mitte

»Auf die Digitalisierung will ich nicht mehr verzichten.«

Abb. 012 ↗
Basis: Aufgeschlossene Mitte (n=1.676); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **54%** männlich
Bildung: **33%** niedrig, **35%** mittel, **27%** hoch

Generationen:

26%	Generation X
21%	Nachkriegsgeneration
21%	Babyboomer*innen

Digital-Index: **61**

Resilienzindikator: **83%**

Quelle: DI21 Index 2022/2023

Digital Literacy – Why to understand those basics?

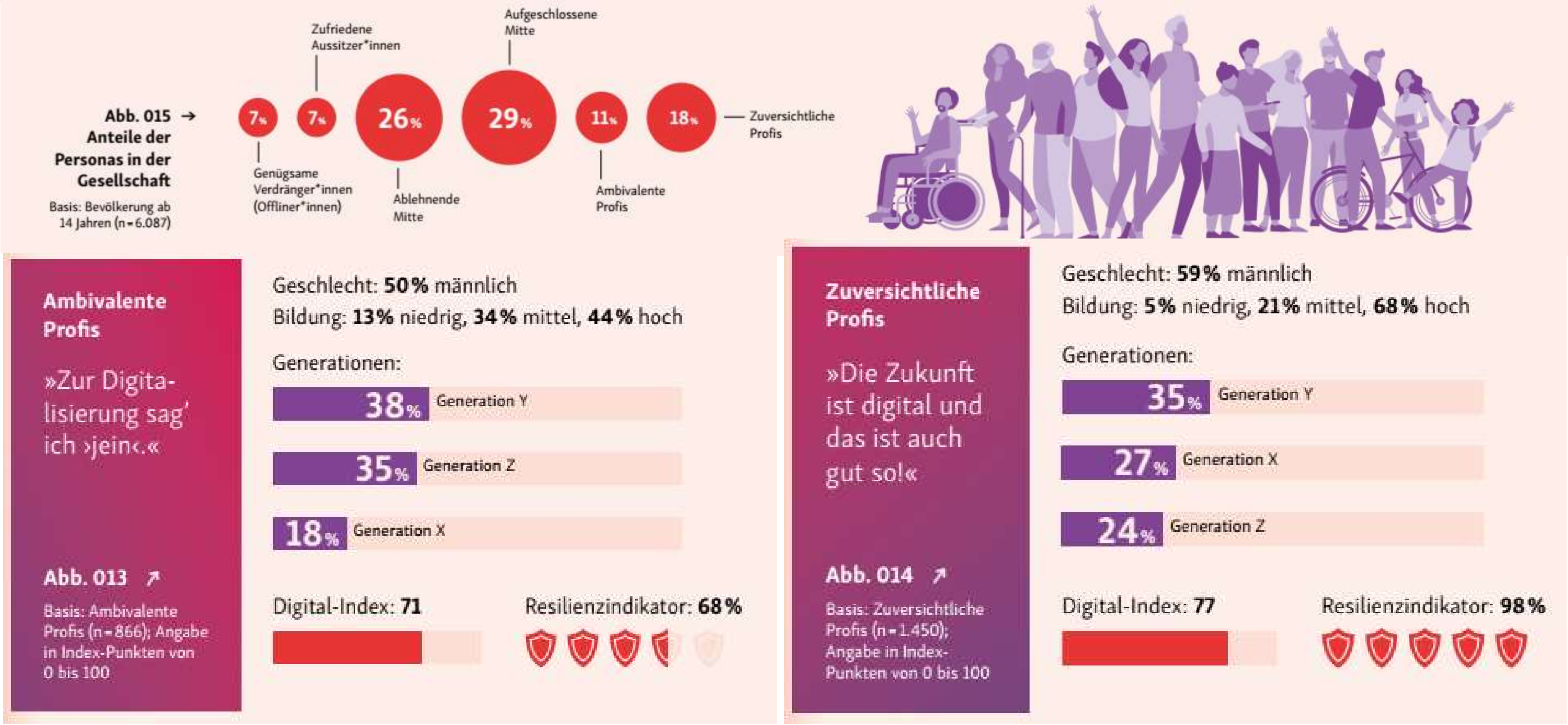


Abb. 015 →
Anteile der Personas in der Gesellschaft
Basis: Bevölkerung ab 14 Jahren (n=6.087)

Ambivalente Profis

»Zur Digitalisierung sag' ich ›jein‹.«

Abb. 013 ↗

Basis: Ambivalente Profis (n=866); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **50%** männlich
Bildung: **13%** niedrig, **34%** mittel, **44%** hoch

Generationen:

Digital-Index: **71**

Resilienzindikator: **68%**

Zuversichtliche Profis

»Die Zukunft ist digital und das ist auch gut so!«

Abb. 014 ↗

Basis: Zuversichtliche Profis (n=1.450); Angabe in Index-Punkten von 0 bis 100

Geschlecht: **59%** männlich
Bildung: **5%** niedrig, **21%** mittel, **68%** hoch

Generationen:

Digital-Index: **77**

Resilienzindikator: **98%**

Quelle: DI21 Index 2022/2023

Digital Literacy – Why to understand those basics?

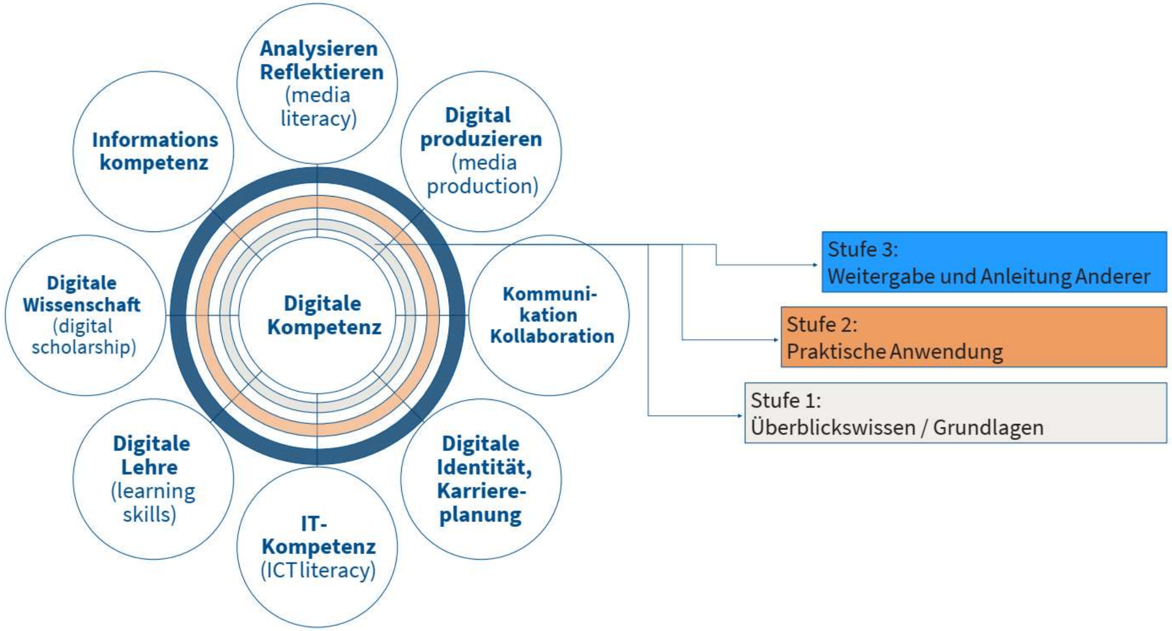
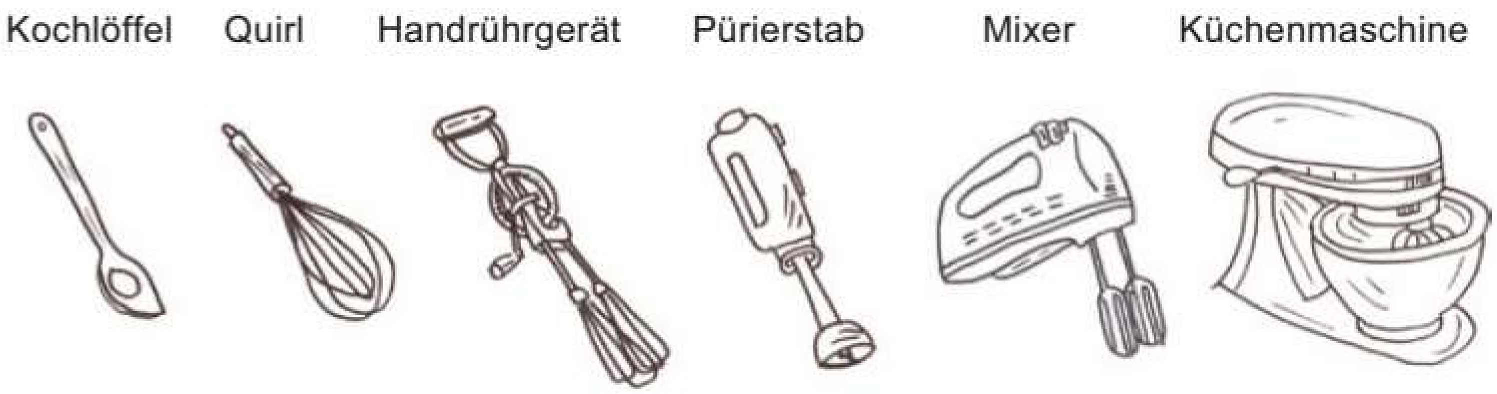


Abb. 1.: Digitale Kompetenz von Hochschullehrenden: Schematische Darstellung des Kompetenzmodells mit acht Dimensionen und drei Kompetenzstufen (In: Eichhorn, Müller, und Tillmann 2017, 214).

Eichhorn, 2019, S.69

The computer: the tool that can do more

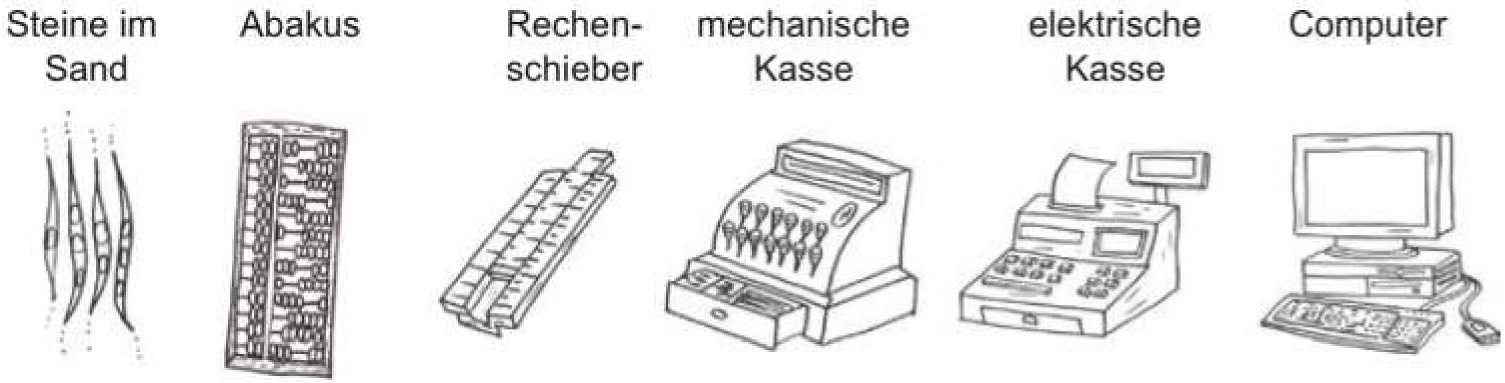
(Zitzler, 2019, own translation)



Development from the wooden spoon to the food processor (Zitzler, 2019, S.4)

The computer: the tool that can do more

(Zitzler, 2019, own translation)



Development of calculating aids up to the computer (Zitzler, 2019, S.5)

The computer: the tool that can do more

(Zitzler, 2019, own translation)

A computer as a mini-factory

- „ To put it simply, a computer is a processing system for numbers in which the processing procedure itself can be variably adjusted. [...] it is rather a data kitchen including a cook, which can independently manage the entire process starting from the initial numbers to the final product.“ (Zitzler, 2019, S.6, own translation)
  - Input: Source material, i.e. store data, elementary data processing.
  - Processing: Implement recipe - the programme - as a sequence of processing steps.
  - Output: Final product

The computer: the tool that can do more

(Zitzler, 2019, own translation)

A computer as a mini-factory

- „ To put it simply, a computer is a processing system for numbers in which the processing procedure itself can be variably adjusted. [...] it is rather a data kitchen including a cook, which can independently manage the entire process starting from the initial numbers to the final product.“ (Zitzler, 2019, S.6, own translation)

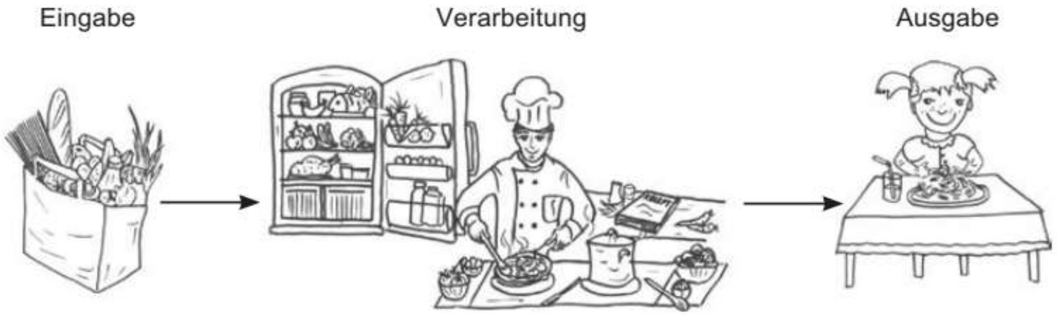


Abb. 1.3 Ein Computer ist wie eine Küche eine Verarbeitungsanlage, nur dass dort Zahlen und nicht Speisen verarbeitet werden. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

The computer: the tool that can do more

(Zitzler, 2019, own translation)

Types of computers

- „ A computer can process data of any kind: Images, texts, sounds, etc. - not just numbers. Therefore, a computer can be described as a universally applicable tool for automated information processing that can receive, store, process and forward data.“ (Zitzler, 2019, S.7, own translation).

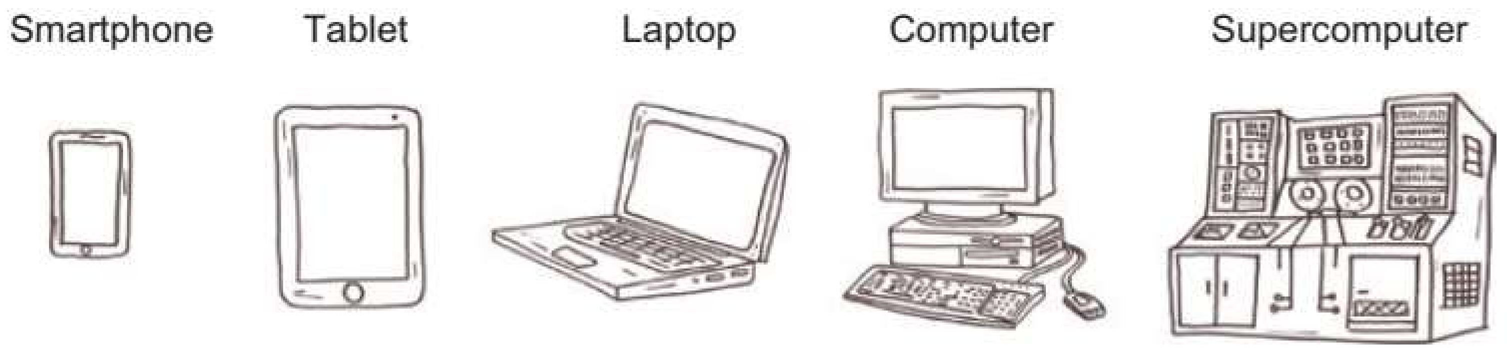


Abb. 1.4 Spielarten von Computern. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

The computer: the tool that can do more

(Zitzler, 2019, own translation)

Never alone

- „ You hardly ever come across a computer as a single, isolated device […]. When we surf on our computer, we actually use many other computers worldwide in parallel […]. This is why we often speak of computer systems.“ (Zitzler, 2019, S.7-8, own translation).

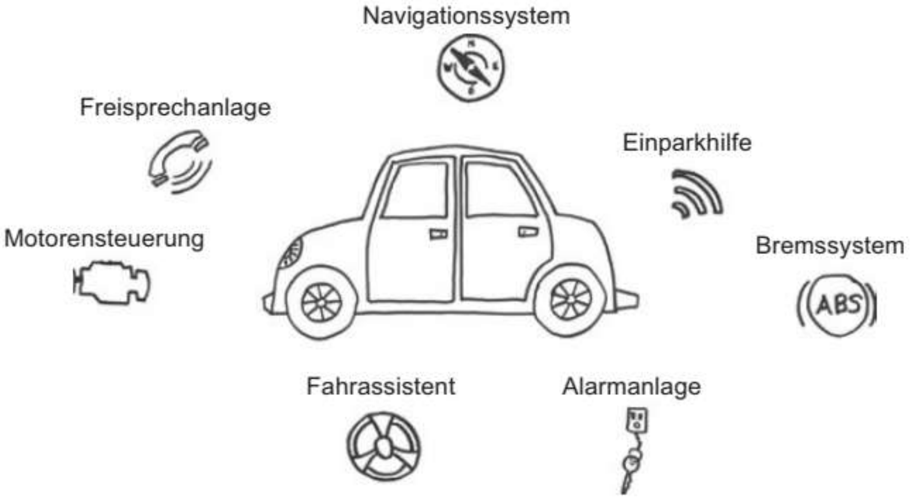


Abb. 1.5 In einem Auto werden mehrere Computer kombiniert und bilden ein Informatiksystem. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

The computer: the tool that can do more

(Zitzler, 2019, own translation)

… and why information sciences (or in German: „Informatik“)?

- Informatik = A made-up word from information and automatic
- „It refers to the science of automated information processing and deals not only with how to build and use computers and computer networks, but also what can generally be done with them - and what not, because even computers have limits: Not every task can be accomplished by a computer system.“ (Zitzler, 2019, S.8, own translation).

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

The human being as an information processor

- People are constantly processing information
  - Reading a book
  - Shopping
  - Jogging

- „ We are constantly taking in impressions from the environment, evaluating them and trying to react to them appropriately.“ (Zitzler, 2019, S.9; own translation)
  - Touch hot cooker top → Take hand away
  - Tapping on shoulder → Showing train ticket
  - Feeling hungry → walk into the kitchen

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

The human being as an information processor

Perception Processing Action

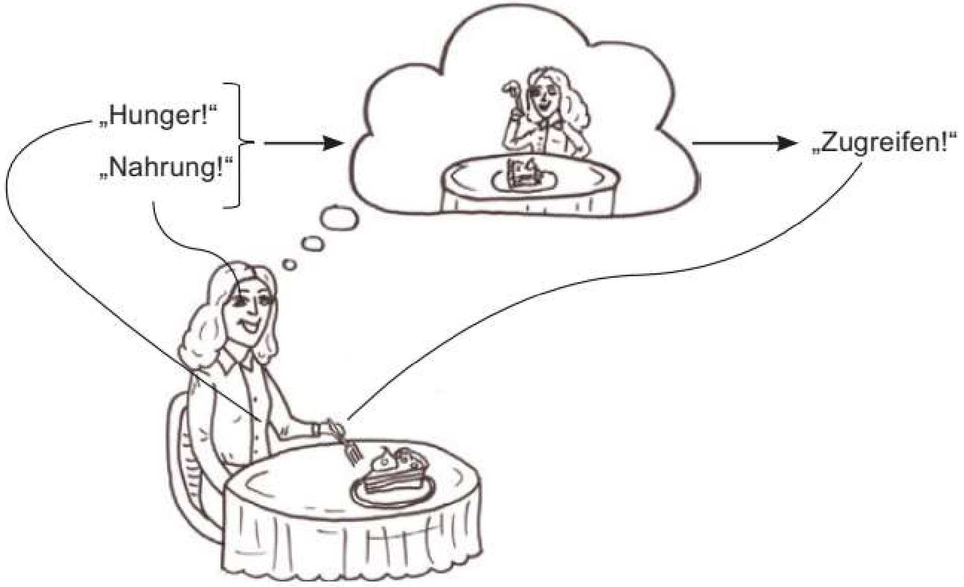


Abb. 2.1 Der Mensch nimmt seine Um- und Innenwelt wahr, verarbeitet die resultierenden Empfindungen und reagiert anschließend, z. B. in Form einer Handlung. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

IPO – Back to the roots

- IPO: Input – Process – Output (German: EVA: Eingabe – Verarbeitung – Ausgabe)
- Example - IPO in the past:
  - Number input by punch card
  - Programme processing in the computer (... very long!)
  - Output result on a printer

Input process output

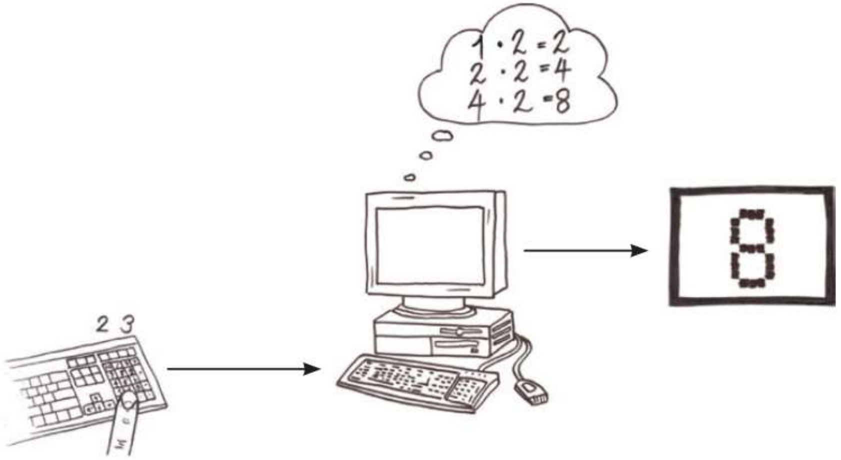


Abb. 2.2 Das Prinzip von Eingabe – Verarbeitung – Ausgabe, abgekürzt als EVA. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

- Programme: rigid if-then scheme (but e.g. web search: processing can be adapted based on new information)

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

Fast and faster: IPO all the time

- IPO today is much more complex
- For example, the control computer of an autonomous car: "The computer cannot lie down on its lazy skin after a calculation, it must constantly perceive the environment and recalculate and adapt its timetable as well as the control signals" (Zitzler, 2019, S.12, own translation)

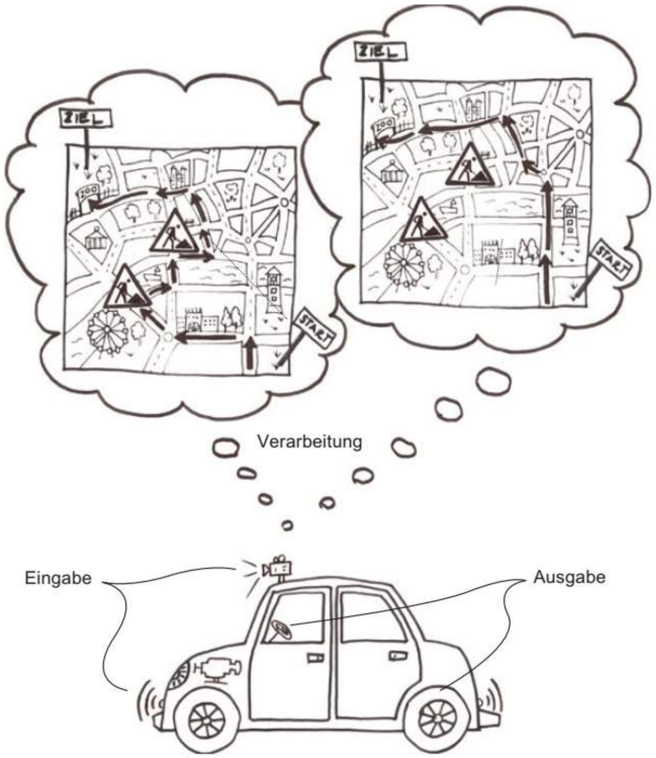


Abb. 2.3 Ein eingebetteter Computer muss ständig auf Signale aus der Umwelt reagieren, um ein Auto durch den Verkehr zu steuern und die Fahrstrecke je nach Gegebenheiten anzupassen. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

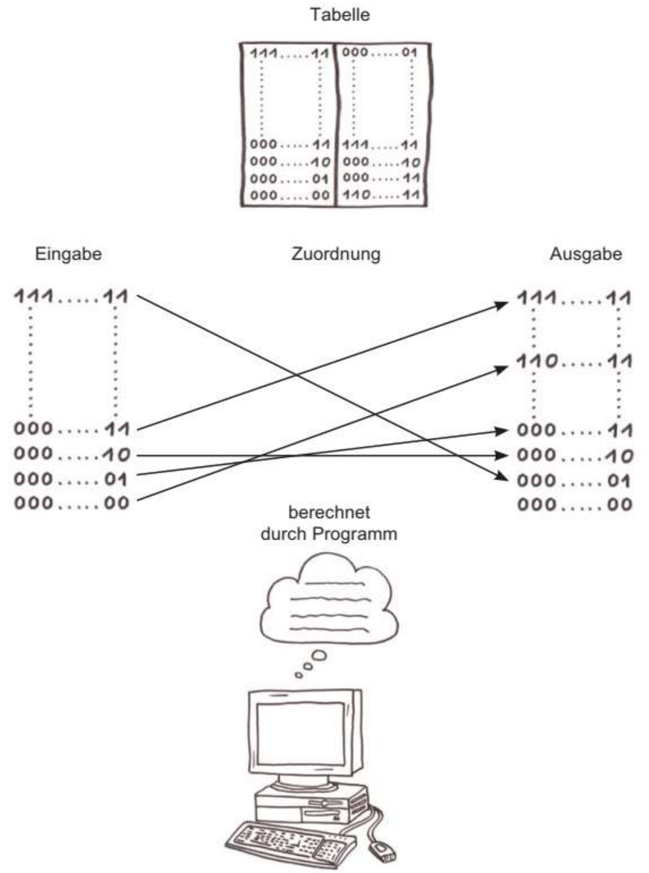
In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

Which pattern belongs to which?

- Information processing: Assignment Input Output
- Idea: Assignment is a huge table. Information processing then means looking up the table to see what output needs to be produced for the current input.
- Problem with this idea:
  - Table much too large for available memory.
  - Table does not exist.

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

Which pattern belongs to which?



111.....11	000.....01
000.....11	111.....11
000.....10	000.....10
000.....01	000.....11
000.....00	110.....11

Abb. 2.4 Bei der Informationsverarbeitung werden Eingaben entsprechende Ausgaben zugeordnet, die Zuordnung wird über das Programm realisiert und kann in Form einer Tabelle beschrieben werden. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

In here, out there: Information processing (Zitzler, 2019, S.9-15, own translation)

Which pattern belongs to which?

- Reality: Assignment input output is done via **programme**: "So instead of listing all input-output assignments individually, one rather describes the way how the corresponding output can be determined from an input".(Zitzler, 2019, S.15, own translation)
- A computer uses a programme to **calculate** which output belongs to an input.
  - Lowest level: zeros & ones (current on | current off).
  - Any kind of information (numbers, pictures, videos...) are mapped over a sequence (a pattern) of 00011101011001111.

… a sequence of 000101110111…?

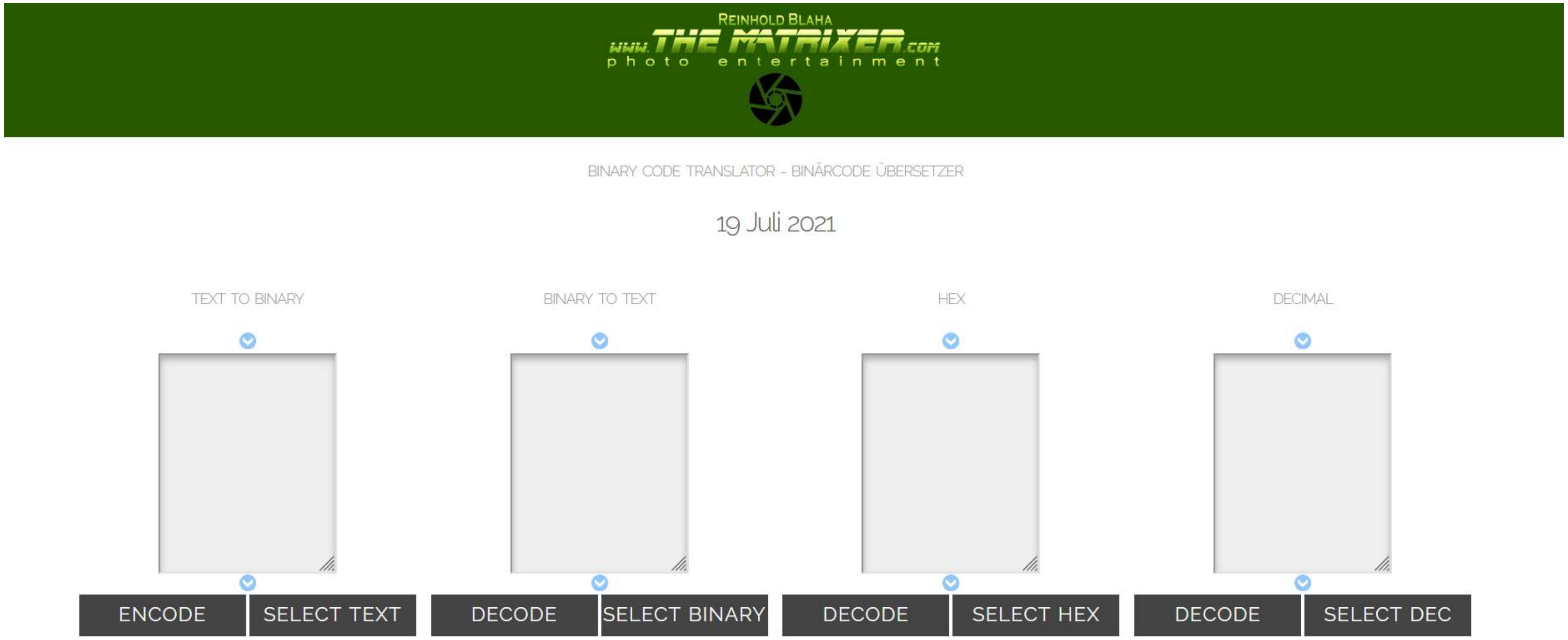
The power of digitalisation (Kollmann, 2020, S.6-7, own translation)

- The digitisation of information in the software sector is a basic prerequisite for the Digital Economy.
- Digitisation makes it possible to edit, copy, transmit and display large amounts of data of text, images and other information without loss of quality and at high speed.

Basic data types:

- Text: ASCII code (American Standard Code for Information Interchange (e.g. 1000001 = "A").
- Image: decomposition into rows and columns → elements of a matrix
  (pixel: 16/32-digit sequence of digits) → row of digits
- Sound: analogue-digital converter digital data stream

… a sequence of 000101110111…?

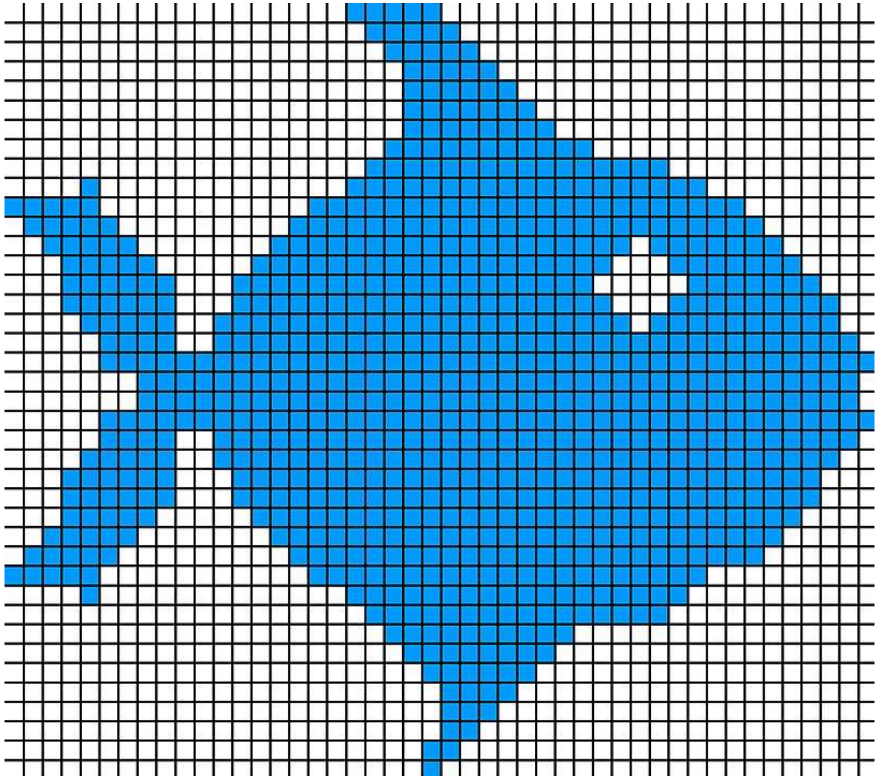


BINARY CODE TRANSLATOR - BINÄRCODE ÜBERSETZER

19 Juli 2021

http://www.thematrixer.com/binary.php

… a sequence of 000101110111…?

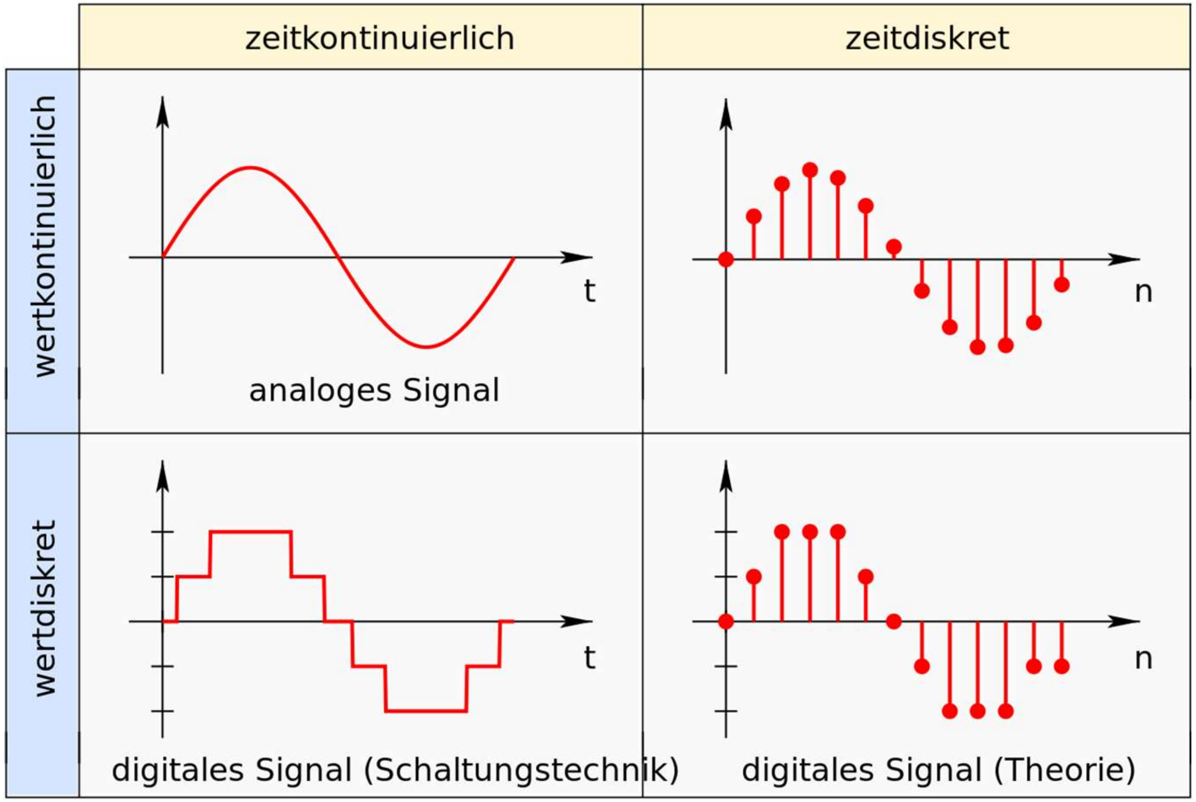


Von Andreas -horn- Hornig, de:Benutzer:Sjr - Erstellt von mir, Andreas -horn- Hornig, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=286813

… a sequence of 000101110111…?

https://www.youtube.com/watch?v=WbIPwVq9KnU

… a sequence of 000101110111…?



Von wdwd - Eigenes Werk, CC BY 3.0,
https://commons.wikimedia.org/w/index.php?curid=17108877

All follows code (Zitzler, 2019, S.17-24, own translation)

Wherever you look: it's all code!

Our everyday life consists of processes.

- "Programmes [or code is] are flow charts of how what has to work, in what order, possibly even exactly when. They describe steps that are to be carried out one after the other or in parallel. And there are very different ways of describing such processes" (Zitzler, 2019, p.18, own translation).

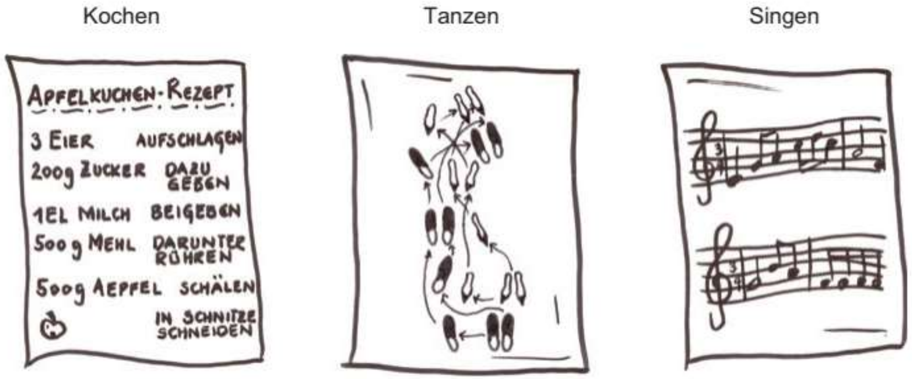


Abb. 3.1 Beispiele für die Darstellung von Abläufen aus unserer Lebenswelt. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

All follows code (Zitzler, 2019, S.17-24, own translation)

What is a computer programme?

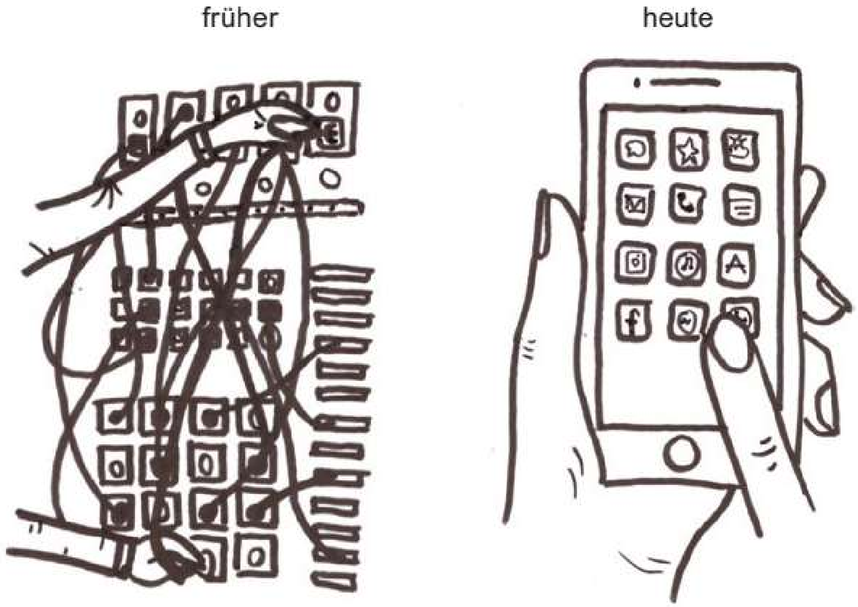


Abb. 3.2 Bei den ersten Computern in den 1940er-Jahren mussten Kabel umgesteckt werden, um das Programm zu wechseln; bei einem heutigen Smartphone reicht ein Antippen des Programmsymbols auf dem Bildschirm. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

All follows code (Zitzler, 2019, S.17-24, own translation)

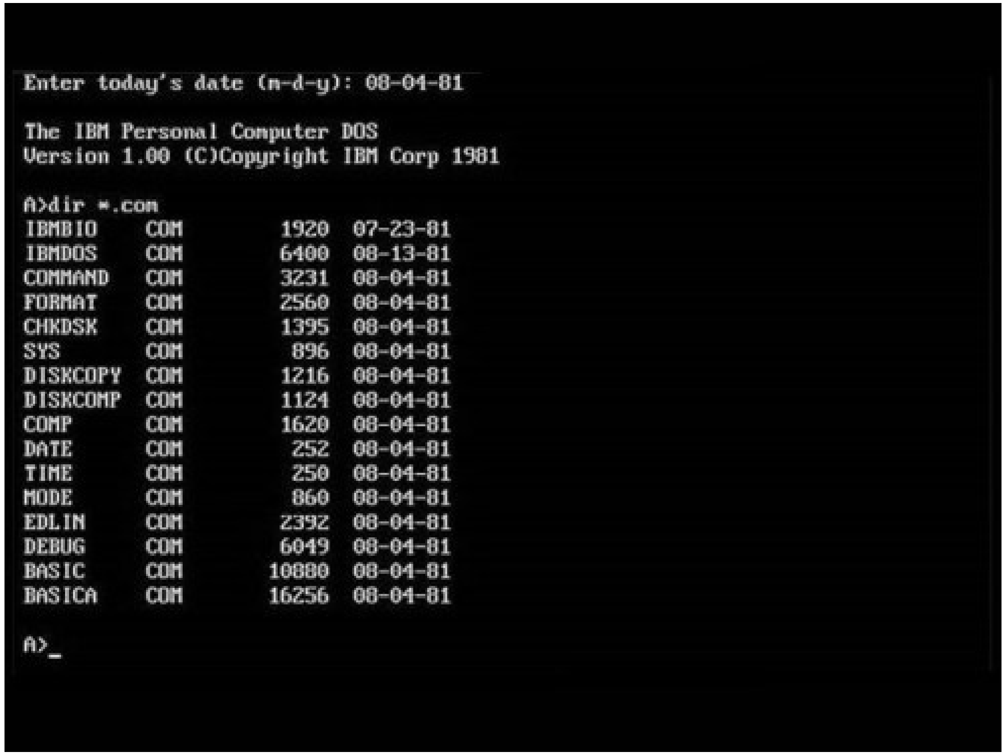
What is a computer programme?

- "A programme determines the processing procedure, one could also say it determines the flexible parts of the device computer." (Zitzler, 2019, p.18, own translation)
- Programme specifies how computer should **solve a task** (e.g. write text, convert image, play music...)
- "A programme is a sequence of elementary processing operations that the computer masters and makes available. You can think of it like the tango: There, basic steps are fixed, but how we string them together can vary; each dance (each programme) can look different" (Zitzler, 2019, p.19, own translation).
- Nothing works without a programme:

  (a) programme for operational readiness (operation, loading, installing etc...) = **operating system**,

  (b) programmes for functionalities (e.g. apps, word processing, e-mails...) = **programme**

All follows code (Zitzler, 2019, S.17-24, own translation)



```
Enter today's date (m-d-y): 08-04-81

The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981

A>dir *.com
IBMBIO    COM      1920  07-23-81
IBMDOS    COM      6400  08-13-81
COMMAND   COM      3231  08-04-81
FORMAT    COM      2560  08-04-81
CHKDSK    COM      1395  08-04-81
SYS       COM       896  08-04-81
DISKCOPY  COM      1216  08-04-81
DISKCOMP  COM      1124  08-04-81
COMP      COM      1620  08-04-81
DATE      COM       252  08-04-81
TIME      COM       250  08-04-81
MODE      COM       860  08-04-81
EDLIN     COM      2392  08-04-81
DEBUG     COM      6049  08-04-81
BASIC     COM     10880  08-04-81
BASICA    COM     16256  08-04-81

A>_
```

MS DOS Startscreen (1981)

Quelle: https://www.computerbild.de/fotos/25-Jahre-Windows-5722893.html#2

All follows code (Zitzler, 2019, S.17-24, own translation)

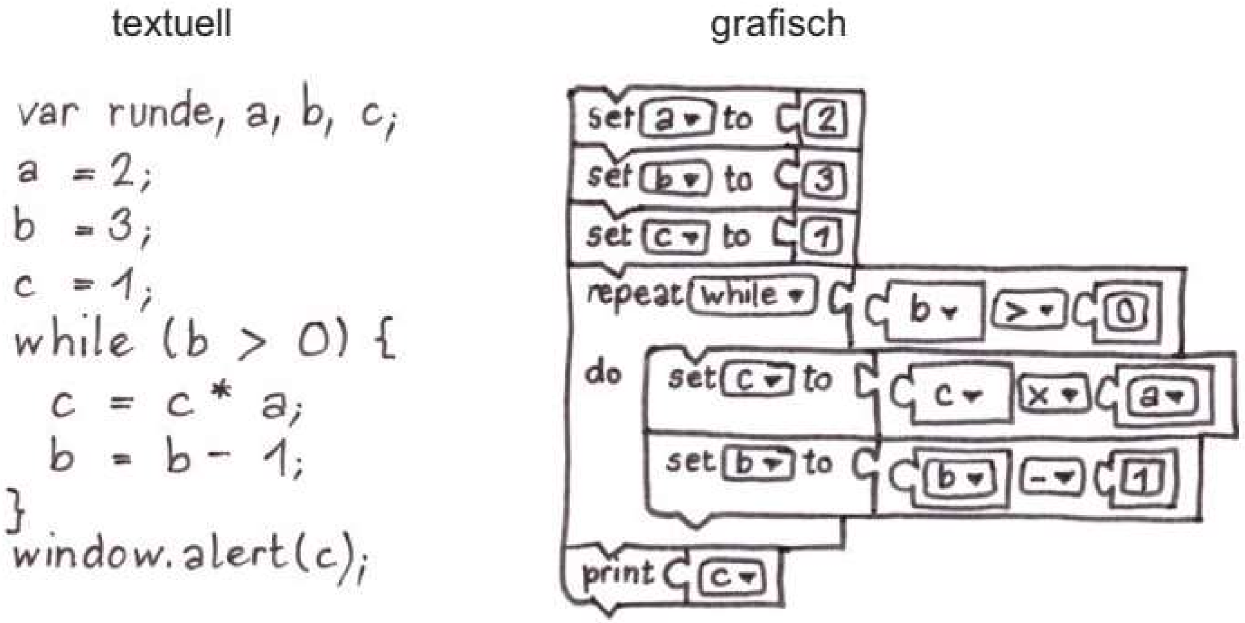
What programmes have to do with poems

- Programmes are formulated in a programming language.
- Programming languages → give computers orders that are also readable by ourselves
- "A programming language defines a basic set of elementary processing operations that can vary in power. Adding two numbers represents a simple command, while drawing a line on the screen is a complex instruction involving many individual steps." (Zitzler, 2019, p.20, own translation)
- Examples:

• Website programming:	HTML, CSS, JavaScript
• Easy to learn:	Python
• Visual building blocks:	Scratch
• Statistics:	R

All follows code (Zitzler, 2019, S.17-24, own translation)

What programmes have to do with poems



```
var runde, a, b, c;
a = 2;
b = 3;
c = 1;
while (b > 0) {
  c = c * a;
  b = b - 1;
}
window.alert(c);
```

```
set a ▾ to 2
set b ▾ to 3
set c ▾ to 1
repeat while ▾ b ▾ > ▾ 0
do  set c ▾ to c ▾ x ▾ a ▾
    set b ▾ to b ▾ - ▾ 1
print c ▾
```

Abb. 3.3 Wie derselbe Ablauf – die Berechnung von 2 hoch 3 – in zwei unterschiedlichen Programmiersprachen formuliert werden kann. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

All follows code (Zitzler, 2019, S.17-24, own translation)

Ingrediences for a computer programme

- All programming languages: Sequence of elementary processing steps.
- Existing processing steps (= instructions) depend on concrete programming language.
- Basic concepts:
  - **Variables** = written containers for data (e.g. a, b, c).
  - **Instructions** = elementary processing steps (e.g. add)
  - **Loops** = repeated execution of instructions (e.g. execute z again and again).
  - **Branches** = conditional execution of instructions (e.g. do x until y occurs)
- Instruction blocks = grouping of processing steps into new instructions

All follows code (Zitzler, 2019, S.17-24, own translation)

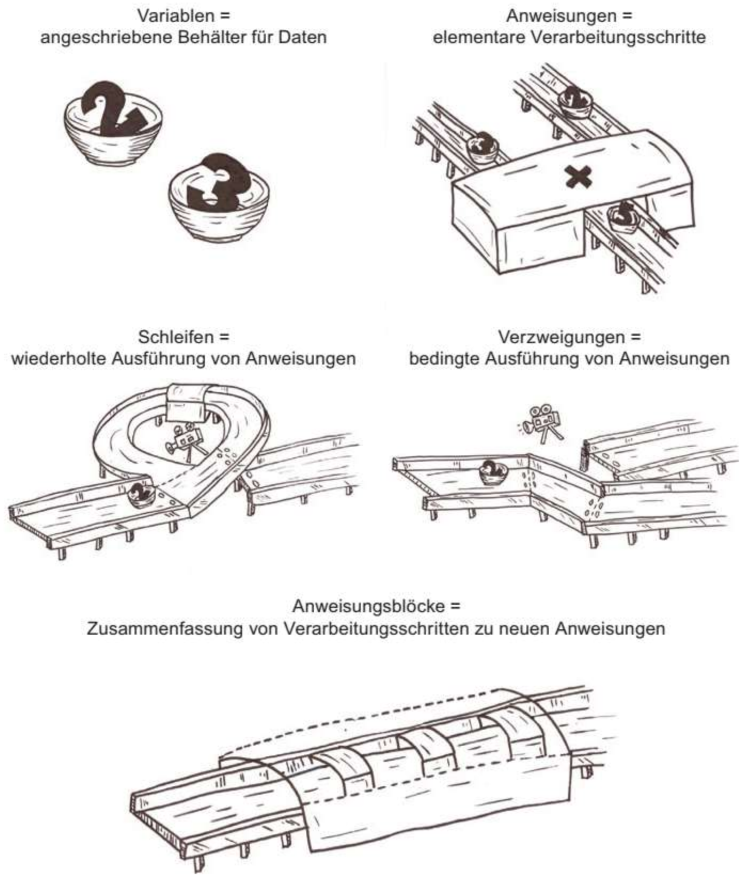

Variablen =
angeschriebene Behälter für Daten
Anweisungen =
elementare Verarbeitungsschritte
Schleifen =
wiederholte Ausführung von Anweisungen
Verzweigungen =
bedingte Ausführung von Anweisungen
Anweisungsblöcke =
Zusammenfassung von Verarbeitungsschritten zu neuen Anweisungen

Abb. 3.4 Grundkonzepte, die Programmiersprachen typischerweise zur Verfügung stellen. (© PHBern: Magdalena Siegenthaler, Eckart Zitzler)

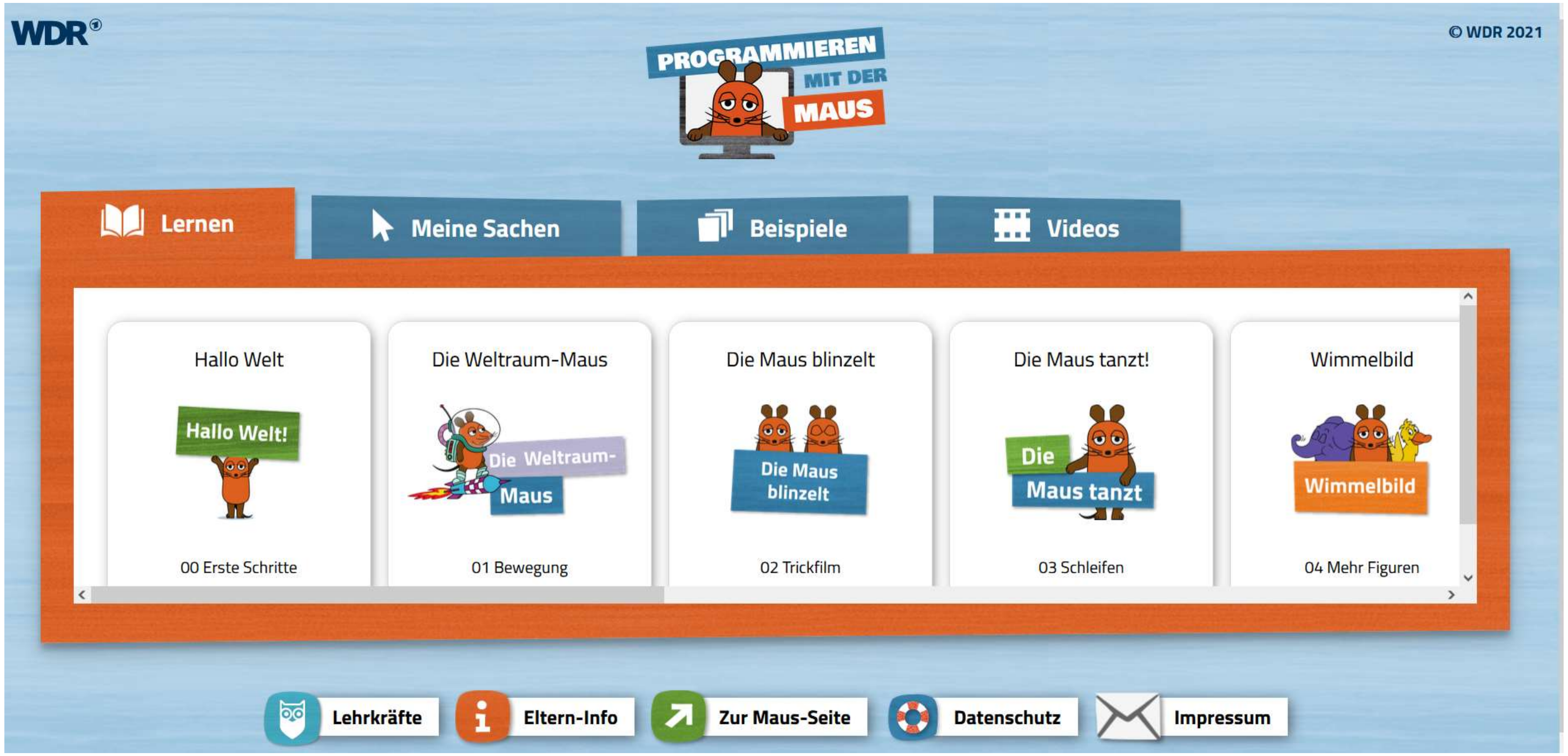
… and by the way

The mathematician **Ada Lovelace** wrote the first computer program in 1843, an algorithm to calculate a sequence of Bernoulli numbers.

https://commons.wikimedia.org/wiki/File:Ada_lovelace.jpg#/media/File:Ada_lovelace.jpg

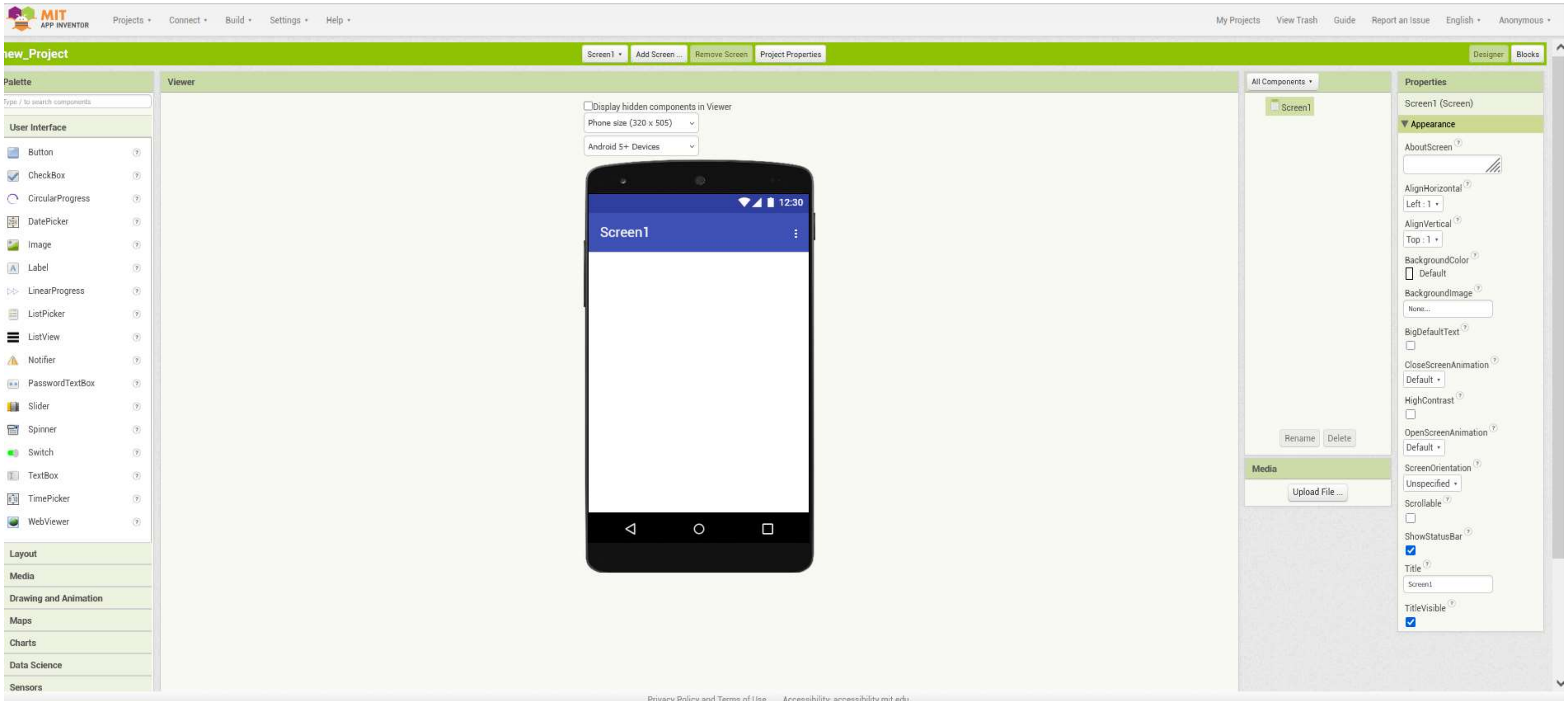
Eberswalde University for Sustainable Development

Let‘s do some practice



https://programmieren.wdrmaus.de/

Let´s do some practice



□ Display hidden components in Viewer
Phone size (320 x 505)
Android 5+ Devices

Screen1 (Screen)
Appearance
AboutScreen
AlignHorizontal: Left : 1
AlignVertical: Top : 1
BackgroundColor: Default
BackgroundImage: None...
BigDefaultText: □
CloseScreenAnimation: Default
HighContrast: □
OpenScreenAnimation: Default
ScreenOrientation: Unspecified
Scrollable: □
ShowStatusBar: ☑
Title: Screen1
TitleVisible: ☑

https://code.appinventor.mit.edu/login/