Sound, Code, Class: Live coding as a medium of computer science education in schools and science communication

Martin Sterzel
martin.sterzel@whz.de

Frank Grimm
frank.grimm@whz.de

Abstract

This paper reflects on our experiences with musical live coding as an entry point to computer science education in both school-based and public contexts. Based on formats such as teaching units, teacher training, and an interactive booth at the "Lange Nacht der Technik" we explore how tools like Strudel can foster accessible engagement with algorithmic thinking. We discuss opportunities and challenges of this approach at the intersection of education, art, and technology.

1 Introduction

Although live coding is increasingly seen in the international research landscape as a creative interface between art, technology, and education, there is still a lack of systematic exploration of its potential for computer science education in schools and public spaces – particularly in German-speaking regions. While live coding has been widely explored and documented in artistic and performance contexts [3, 12], empirically grounded examples of its didactic use remain rare [16]. It remains an open question to what extent the creative openness and performative character of live coding can be reconciled with curricular requirements – especially given the typically short and exploratory nature of the interventions described in this paper.

Live coding is often described as a techno-artistic practice in which writing and modifying source code is used to generate music, visual content, or other media processes in real time. This definition captures key characteristics frequently cited in the literature, including the simultaneous execution and modification of code and the interplay of creative design and algorithmic expression. Scholars such as Collins et al. [3], Magnusson [11], and Blackwell et al. [1] emphasize the performative dimension of live coding, in which code is altered during execution and the process of programming is made visible to an audience. Definitions vary depending on context – musical, educational, or technical – but all share a foundation in creativity, real-time feedback, and algorithmic form.

Originally rooted in electronic music culture, live coding has in recent years developed into an interdisciplinary field spanning art, research, and education [1]. Especially from a pedagogical perspective live coding is increasingly recognized as a method for conveying fundamental computing concepts such as abstraction, sequence, iteration, and modularization in a creative way [11, 1]. The combination of real-time feedback, creative expression, and collaborative interaction makes live coding particularly well-suited for formal and informal learning settings. Browser-based tools such as Strudel[1] offer a low-threshold entry, even for those without prior technical experience.

Furthermore, we examine the extent to which live coding can serve as a medium for science communication – through accessible, interactive formats that allow the public to experience computational principles firsthand.

This paper investigates the potential of live coding as a medium for computer science education and engagement by analyzing three central contexts of implementation: school-based teaching, teacher training, and public outreach formats. Our goal is to derive insights from practical experience regarding:

- how live coding can support algorithmic thinking,

[1] Strudel is a port of the Tidal Cycles pattern language to JavaScript to write dynamic music pieces (see https://strudel.cc).

- what motivational and interactive dynamics emerge in different target groups,
- and what didactic and technical challenges arise in these processes.

This contribution represents an exploratory approach to the question of how a performative, creative form of coding can contribute to computer science education and public engagement – both conceptually and in terms of a broader understanding of education.

The paper is structured as follows: after outlining a theoretical framework for live coding and computer science education in Saxony (Germany) as well as science communication in section 2, we present three core contexts of implementation in section 3: school-based teaching, teacher training and workshops, and public outreach formats. Section 4 reflects on opportunities and challenges, and section 5 offers an outlook on future developments.

2 Theoretical Background

2.1 Live Coding as a Techno-Artistic Practice

Live coding is a creative practice in which code is written and modified in real time to generate audiovisual processes. The code is often shown to the audience by using projectors. This practice links algorithmic thinking with performative expression and has its roots in the electronic music scene [3, 11].

In artistic contexts, live coding is not only a means of production but also a socio-technical form of expression that emphasizes liveness, processuality, and algorithmic transparency [1]. This should be distinguished from didactic methods such as "participatory live coding" in university teaching, where instructors live-code to demonstrate concepts [15].

2.2 Live Coding in Educational Contexts

Recent years have seen growing interest in the pedagogical potential of live coding [2, 4, 14, 6]. At its core are creative approaches to computing concepts, supported by real-time feedback, sonic experimentation, and open-ended design, which foster motivation and exploratory learning.

Live coding aligns well with the constructionist learning theory, as formulated by Seymour Papert, which emphasizes active creative engagement with computational artefacts [13]. In this tradition, live coding supports a design-oriented approach to teaching core computing ideas such as sequence, iteration, and parameterization [11, 1]. It also complements ideas from the maker education movement, which focuses on hands-on, interdisciplinary, and self-directed learning. Honey and Kanter (2013) argue that learning environments that incorporate design, making, and play are particularly effective in STEM education [10].

By combining music, code, and direct feedback, live coding facilitates learning processes that address both technical and aesthetic dimensions – with the latter aspect which is often underrepresented in traditional computer science curricula.

Several studies exemplify the educational value of live coding: Diapoulis [6] describes a Strudel-based course for girls aged 10–15, which used dialogic structures and public concerts. Corvi et al. [4] report on curricular integration in Italian music education. The FiLOrk project demonstrates inclusive practices for visually impaired students using TidalCycles [14], while Freeman and Magerko's work on EarSketch links music technology and computing through creative media composition [9].

2.3 Connection to Saxony's Digital Education Strategy

The Saxon initiative "Digitale Schule Sachsen" [8] and the concept of a "Culture of Digitality" [5] outline educational goals such as promoting digital competencies, establishing creative media practices, and supporting innovative teaching and learning formats.

The formats we implemented – *live-coding sprints* in school, teacher training and exchange, and a public outreach booth during the "Lange Nacht der Technik" (english translation: "Long Night of Technology") at the University of Applied Sciences Zwickau – align with these strategic aims. They provide accessible entry points into computer science education, support creative expression, and foster reflection on digital practice across diverse audiences.

2.4 Live Coding and Science Communication

Live coding can also be considered a participatory format of science communication. Literature on public engagement highlights the value of dialogic, aesthetically engaging, and interactive formats in making scientific content more accessible [17].

Live coding makes algorithmic processes visible and tangible, inviting audiences to explore and co-create. This performative practice is not only expressive but also pedagogically powerful.

Programs promoting science communication have shown that interactive formats enhance researchers' communication skills and foster sustainable outreach practices. Staus, Risien, and Cho [18] emphasize that dialogic and participatory formats can help embed communication more deeply into the everyday work of scientists.

Algorave, a format that combines live coding and electronic dance music performance, further illustrates how algorithmic thinking can be experienced collectively. Diapoulis and Carlé [7] analyze over 130 Algorave performances and show how patterns, code visibility, and improvisation enable novel forms of cultural engagement with technology.

Overall, live coding offers a unique contribution to science communication – as a culturally embedded, dialogic, and experiential format that connects technical knowledge with public engagement.

3 Approaches and Results

3.1 Live-Coding Sprints: Classroom Interventions

The so-called *live-coding sprints* were implemented during project days at the Berufliches Schulzentrum für Technik und Wirtschaft "Julius Weisbach" (vocational grammar school) in Freiberg and the Wintersport-Campus Klingenthal (grammar school). Both sessions targeted 11th-grade students and lasted 90 minutes each. The activities were conducted by a research associate from the Faculty of Physical Engineering/Computer Science at the University of Applied Sciences Zwickau and accompanied by computer science teachers from the respective schools. The feedback on the event from the accompanying teachers can be found in Appendix B.1.

After a brief theoretical introduction – supported by multimedia impulses on live coding and a presentation of pattern syntax and the Strudel interface – participants immediately began creating their own musical structures. Following a demonstration of Strudel's core functionality, selected functions and effects were introduced systematically and practiced in self-guided phases. The session concluded with the opportunity for students to present their musical results.

Feedback from accompanying teachers indicated a consistently high level of interest and engagement among students. The combination of music and code opened up a new perspective on programming for many. The sessions were described as engaging, lively, and inspiring. The open-ended structure and immediate auditory feedback encouraged creative experimentation and individual participation.

From a didactic standpoint, the format successfully conveyed core computational thinking concepts such as abstraction, sequencing, iteration, and parameterization. The musical framing enabled many students – especially those less experienced with programming – to engage with the material. Connections to other subjects also became apparent: music (rhythm, notation), physics (sound and vibration), and the arts.

Challenges for the students included unfamiliar syntax and how to respond to unexpected programming errors. However, these challenges were framed as valuable learning opportunities from the respective teacher. The sessions fostered a constructive error culture: mistakes were discussed openly and seen as part of the learning process. Students worked at their own pace, free from judgment.

One recommendation of the observing teacher to the lecturer was to allocate more time for sharing final creations, as the concluding presentations were somewhat rushed. Additional time for peer exchange and showcasing student work was seen as beneficial for future iterations.

Overall, the *live-coding sprints* demonstrated that creative, hands-on approaches can make computer science education more engaging, particularly in interdisciplinary settings.

The following learning objectives were defined in advance for the teaching units: (A) Lowering entry barriers to key computing concepts such as sequence, iteration, and parameterization; (B) Fostering algorithmic thinking through creative, auditory experiments; (C) Designing individual patterns using Strudel; (D) Encouraging positive attitudes toward error handling and technical problem-solving; (E) Supporting interdisciplinary thinking through connections to music and physics.

3.2 Collegial Exchange and Teacher Training

In addition to the classroom sessions, two formats were offered for educators: an online exchange and a three-hour in-person workshop at the Zittau/Görlitz University of Applied Sciences (HSZG) in Germany. Detailed feedback from the teachers on these formats can be found in Appendix B.2.

The *collegial exchange* took place via the BigBlueButton video platform and included a 30-minute presentation followed by a moderated discussion. The talk introduced key ideas behind live coding, demonstrated Strudel, and prompted reflection on its didactic potential.[2]

Teachers discussed various integration scenarios – including the idea of hosting a school-internal *Algorave*, inspired by an English teacher. One participant followed up with the authors and subsequently organized a live coding sprint at the Berufliches Schulzentrum für Technik und Wirtschaft "Julius Weisbach".

Feedback highlighted the practical advantages of browser-based tools like Strudel, especially in schools with limited IT support. The session was perceived as accessible, understandable, and interactive. Potential for interdisciplinary projects and student-centered learning was emphasized, though some teachers noted the importance of having at least a basic musical background.

The *teacher training workshop* at HSZG took place in a computer lab and lasted three hours. After an introduction to Strudel's concepts and syntax, participants explored musical pattern creation through guided exercises using handouts.[3]

Feedback indicated enthusiasm for the immediacy and creativity of live coding but also highlighted initial challenges with syntax complexity. Participants requested simplified materials, more practice time, and a clear strategy for embedding live coding into the computer science curriculum in schools. The strongest outcomes were seen in the links between music and computer science, suitability for project-based learning, and motivation of students less inclined toward conventional programming.

Both formats illustrate that live coding is not only methodologically innovative, but can also be integrated into the school education landscape – provided the right technical conditions and didactic support are in place. The following learning objectives were defined in advance for the teacher formats: (A) Introduction to the concept and educational potential of live coding; (B) Hands-on experience with browser-based tools (Strudel) from a teacher's perspective; (C) Reflecting on didactic applications in classroom, cross-disciplinary, and project contexts; (D) Identifying practical needs for implementation (e.g., materials, infrastructure, curricular links).

3.3 Public Outreach: Lange Nacht der Technik

As part of the "Lange Nacht der Technik" 2025 at the University of Applied Sciences Zwickau, an interactive booth on live coding was installed in the training workshop space. The booth – titled *TENSOR::BYTE CLUB* (Allusion to the techno club *Tresor* in Berlin and the legendary movie *Fight Club*)– was staffed by a computer science professor, one live-coding performer (*MC Algorhythm*) and four student assistants from the Department of Physical Engineering/Computer Science.

The concept combined low-threshold interaction with performative elements: In a *hands-on corner*, visitors could use the browser-based tool *Strudel* to generate sounds, which were amplified via a PA system. In addition, hourly live coding performances – short *Algoraves* – were

[2] `https://slides.com/martinst/bento-6c3dc2`

[3] Handouts are available in Appendix A (German only). They are based on the structure of the TidalCycles documentation.

presented using *Tidal Cycles*, with the code projected onto a wall. A second projection displayed visuals generated with *Hydra*, using live video input from the space. The structure of the stand is shown in Figure 1.

The ambient soundscape, lighting setup, and open layout attracted especially younger audiences and families. Many visitors spontaneously experimented with their own patterns or followed the performances. The combination of sound, visual feedback, and a user-friendly interface was frequently cited as engaging and motivational.

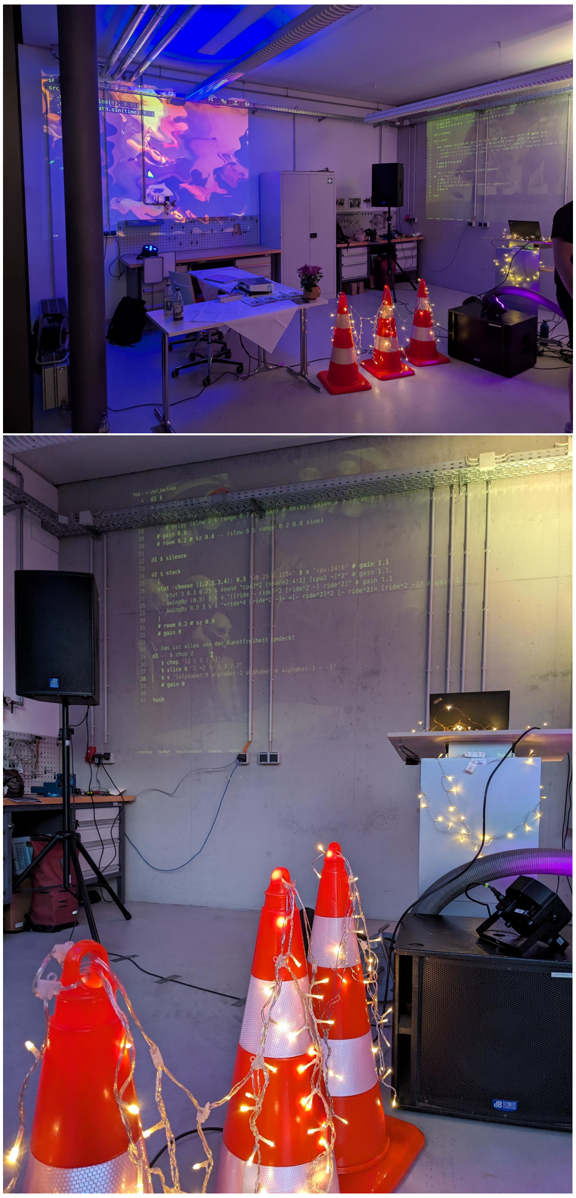

Figure 1: Impressions from the live coding booth at Scheffelberg Campus (photos: Melanie Merkel and Martin Sterzel)

In the context of science communication, the format demonstrated the potential of live coding as a participatory method. Its informal atmosphere and experimental nature enabled encounters with computer science concepts beyond disciplinary boundaries and outside of formal education. An internal review identified areas for improvement, including clearer visibility of the computer science aspect and enhancements to technical infrastructure (e.g., projector brightness, more headphone stations).

The feedback log, summarized in Appendix C, highlighted the following:

- easy access to programming through visual and auditory feedback,
- successful integration of technology, creativity, and spatial design,
- constant musical presence as an attention-drawing element,
- potential for improvement in pedagogical framing, infrastructure, and visibility.

The following learning objectives were defined in advance for the public format: (A) Raising awareness of computational and algorithmic processes through creative experience: (B) Making computer science visible as a cultural and aesthetic practic; (C) Promoting active participation via accessible digital tools; (D) Creating a dialogic space between science, art, and the public.

4 Discussion

The presented formats demonstrate how live coding can effectively bridge computer science, music, and cultural education. The following sections synthesize key insights.

Conveyed Computational Concepts

Across all formats, core ideas of computational thinking—such as sequencing, iteration, and parameterization—were made tangible through immediate auditory feedback and live code output. Musical structures like rhythm and meter supported interdisciplinary reflections on algorithmic thinking.

Performativity and Creative Dynamics

Live coding's performative nature fostered shared engagement and individual exploration. Especially in public settings, its aesthetic and experimental qualities lowered the threshold for engaging with computing, offering an accessible entry point for many participants.

Conditions for Success

In schools, clear structure, stable tech, and a balance between instruction and open exploration were key. Teacher training required combining playful experimentation with conceptual framing. Public formats thrived on multisensory interaction and open, participatory setups supported by a responsive team.

Educational Challenges and Potentials

These activities aligned with constructionist learning: immediate feedback, creative autonomy, and sensory engagement encouraged reflective practice. The musical framing appealed even to those less inclined toward traditional coding. Still, technical complexity and infrastructure posed obstacles. Beyond technical skills, participants developed cross-disciplinary capacities like error tolerance, creative problem-solving, and integrative thinking—qualities often underrepresented in standard curricula.

Live Coding as Science Communication

At the "Lange Nacht der Technik," live coding enabled participatory encounters with computing—through the interplay of sound, visuals, and code. Accessibility without prior knowledge was a key strength. However, success depended on careful didactic design, technical readiness, and dramaturgical sensibility. Live coding in science communication must go beyond spectacle to foster meaningful public engagement with algorithmic thinking.

5 Conclusion and Future Work

Live coding holds significant potential for both computer science education and science communication – provided that appropriate conditions, pedagogical strategies, and institutional openness are in place. It opens up creative learning environments that are interdisciplinary and participatory, but which require thoughtful technical and didactic planning.

Looking ahead, a key question is how such formats can be more systematically integrated into regular educational and communication practices. This includes:

- curricular integration of creative coding approaches into computer science education,
- development of modular training materials for educators,
- long-term inclusion in universities' science communication strategies.

Further empirical evaluation is needed – particularly with regard to learning outcomes, motivation, and sustainability. In the long term, live coding could move from a niche format to an established part of interdisciplinary education and public engagement in science.

References

[1] Alan F. Blackwell et al. *Live Coding: A User's Manual*. The MIT Press, Nov. 22, 2022. ISBN: 978-0-262-37263-3. DOI: 10.7551/mitpress/13770.001.0001. URL: https://direct.mit.edu/books/book/5495/Live-CodingA-User-s-Manual (visited on 06/24/2025).

[2] Hussein Boon. "Live Coding and Music Production as Hybrid Practice". In: *Organised Sound* 28.2 (Aug. 2023), pp. 253–263. ISSN: 1355-7718, 1469-8153. DOI: 10.1017/S1355771823000419. URL: https://www.cambridge.org/core/product/identifier/S1355771823000419/type/journal_article (visited on 07/04/2025).

[3] Nick Collins et al. "Live Coding in Laptop Performance". In: *Organised Sound* 8.3 (Dec. 2003), pp. 321–330. ISSN: 1355-7718, 1469-8153. DOI: 10.1017/S135577180300030X. URL: https://www.cambridge.org/core/product/identifier/S135577180300030X/type/journal_article (visited on 06/24/2025).

[4] Francesco Corvi, Giovanni Mori, and Giovanni Nulli. "Live Coding and Education. A Practical Experience." In: (Apr. 18, 2023). DOI: 10.5281/ZENODO.7843819. URL: https://zenodo.org/record/7843819 (visited on 06/27/2025).

[5] Christine Dallmann. *Kultur Der Digitalität*. June 27, 2025. URL: https://www.medienbildung.sachsen.de/kultur-der-digitalitaet-6755.html.

[6] Georgios Diapoulis. "Teaching Strudel to Young Girls: Realizing Live Coding through Performance Practice". In: (May 27, 2024). DOI: 10.5281/ZENODO.11350109. URL: https://zenodo.org/doi/10.5281/zenodo.11350109 (visited on 06/27/2025).

[7] Georgios Diapoulis and Martin Carlé. "Reproducible Musical Analysis of Live Coding Performances Using Information Retrieval: A Case Study on the Algorave 10th Anniversary". In: (Apr. 18, 2023). DOI: 10.5281/ZENODO.7843813. URL: https://zenodo.org/record/7843813 (visited on 06/30/2025).

[8] Jens Drummer. *Initiative „Digitale Schule Sachsen“*. June 27, 2025. URL: https://medienbildung.sachsen.de/initiative-digitale-schule-sachsen-6550.html.

[9] Jason Freeman and Brian Magerko. "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch". In: *Journal of Music, Technology & Education* 9.1 (May 1, 2016), pp. 57–74. ISSN: 1752-7066, 1752-7074. DOI: 10.1386/jmte.9.1.57_1. URL: https://intellectdiscover.com/content/journals/10.1386/jmte.9.1.57_1 (visited on 06/27/2025).

[10] Margaret Honey and David Kanter, eds. *Design, Make, Play: Growing the next Generation of STEM Innovators*. Online-Ausg. New York: Routledge, 2013. 238 pp. ISBN: 978-0-415-53920-3 978-0-203-10835-2.

[11] Thor Magnusson. "Algorithms as Scores: Coding Live Music". In: *Leonardo Music Journal* 21 (Dec. 2011), pp. 19–23. ISSN: 0961-1215, 1531-4812. DOI: 10.1162/LMJ_a_00056. URL: https://direct.mit.edu/lmj/article/63659 (visited on 06/24/2025).

[12] Alex McLean and Geraint Wiggins. "Tidal–Pattern Language for the Live Coding of Music". In: vol. Proceedings of the 7th sound and music computing conference. 2010, pp. 331–334.

[13] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. 2nd edition. New York, NY: Basic Books, 1993. 230 pp. ISBN: 978-0-465-04627-0 978-0-465-04674-4 978-0-465-04629-4.

[14] William C Payne et al. "Live Coding Ensemble as Accessible Classroom". In: (May 31, 2023). ISSN: 2220-4806. DOI: 10.5281/ZENODO.11189248. URL: https://zenodo.org/doi/10.5281/zenodo.11189248 (visited on 06/27/2025).

[15] Adalbert Gerald Soosai Raj et al. "Role of Live-coding in Learning Introductory Programming". In: *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*. Koli Calling '18: 18th Koli Calling International Conference on Computing Education Research. Koli Finland: ACM, Nov. 22, 2018, pp. 1–8. ISBN: 978-1-4503-6535-2. DOI: 10.1145/3279720.3279725. URL: https://dl.acm.org/doi/10.1145/3279720.3279725 (visited on 06/27/2025).

[16] Ana Selvaraj et al. "Live Coding: A Review of the Literature". In: *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2021: 26th ACM Conference on Innovation and Technology in Computer Science Education. Virtual Event Germany: ACM, June 26, 2021, pp. 164–170. ISBN: 978-1-4503-8214-4. DOI: 10.1145/3430665.3456382. URL: https://dl.acm.org/doi/10.1145/3430665.3456382 (visited on 07/04/2025).

[17] Joana Silva and Karen Bultitude. "Best Practice in Communications Training for Public Engagement with Science, Technology, Engineering and Mathematics". In: *Journal of Science Communication* 08.02 (May 18, 2009), A03. ISSN: 1824-2049. DOI: 10.22323/2.08020203. URL: https://jcom.sissa.it/article/pubid/Jcom0802(2009)A03/ (visited on 06/30/2025).

[18] Nancy L. Staus, Julie Risien, and Holly Cho. "Scientists' Views about Relationship-Based Science Communication Strategies". In: *Journal of Science Communication* 24.01 (Jan. 27, 2025). ISSN: 1824-2049. DOI: 10.22323/2.24010803. URL: https://jcom.sissa.it/article/pubid/JCOM_2401_2025_N03/ (visited on 06/30/2025).

A Handouts

A.1 Pattern Handout

Datei: strudel_workshop_patterns.js Seite 1 von 1

```
/*
Verwendete Samples:
bd = Bass Drum (große Trommel)
sd = Snare Drum (Kleine Trommel)
cp = Clap (Klatschen)
hh = HiHat (übereinanderliegende Becken)

-- Übung zu Pattern und Mini-Notation --
*/

// Mit ~ oder - eine Pause einfügen
sound("bd ~ sd:3 bd sd:5 - bd:2 sd:2")

// Mit eckigen Klammern kann man eine Untersequenz einfügen
sound("bd [bd cp] bd bd")

sound("[bd bd sd:5] [bd sd:3]")

sound("[[bd bd] bd sd:5] [bd sd:3]")

sound("[[bd [bd bd bd bd]] bd sd:5] [bd sd:3]")

// Mit dem * kann man ein Sample wiederholen
sound("bd sd*2")
// ... und ist äquivalent zu
sound("bd [sd sd]")

sound("bd [sd cp]*2")

// Mit / kann man ein Pattern verlangsamen (sd wird nur in jedem zweiten Cycle gespielt)
sound("bd sd/2")

sound("bd [sd cp]/2")
// Mit <> können wir eine Warteschlange erzeugen, die die Samples in jedem Cycle nacheinander
wechselt
sound("bd <sd cp bd>")

sound("<bd sd> <sd [cp cp]> <bd [cp cp]>")

// Mit | können wir zufällig aus zwei Samples auswählen
sound("[bd:0|bd:1]")

sound("[sd|cp]")

// Mit , können wir zwei Samples/Patterns gleichzeitig spielen
sound("[sd,cp]")

// Mit ! können wir ein Sample mehrfach wiederholen (bestimmt durch die Zahl nach dem
Ausrufezeichen)
sound("bd!3 sd")

/*
Beachtet den Unterschied zu "bd*3 sd":
Im ersten Fall gibt es vier Ereignisse, die alle die gleiche Zeit dauern.
Im zweiten Fall dauern die drei bd einen halben Cycle lang, und das sd die andere Hälfte.
"bd!3 sd" ist dasselbe wie "bd bd bd sd".
"bd*3 sd" ist dasselbe wie "[bd bd bd] sd"
*/
```

A.2 Effects Handouts

Datei: strudel_workshop_effects.js Seite 1 von 1

```
// -- Übung zu Effekten --

// Filter um Klänge wie Vokale klingen zu lassen
sound("bd bd bd bd").vowel("a")

sound("bd bd bd bd").vowel("a o e e")

// Mit gain können wir die Lautstärke der Pattern steuern
sound("bd hh sd:1 hh sd:1 hh").gain("0.5")

// Mit speed können wir die Geschwindigkeit mit der ein Sample abgespielt, und damit die
Tonhöhe, ändern
sound("bd bd bd bd").speed("1 1.5 2 0.5")

// Mit pan können wir das Stereo-Verhälten ändern (0 = links, 0.5 = Mitte, 1 = rechts)
sound("bd bd bd bd").pan("0 0.5 1")

// Mit Delay können wir eine Art Echo-Effekt erzeugen
sound("cp").delay("0.8").delaytime("0.15").delayfeedback("0.6")

// lpf ist ein Low-Pass-Filter, der die Lautstärke aller Frequenzanteile oberhalb eines
bestimmten Schwellwerts absenkt
sound("cp cp").lpf("400 1000")
```

A.3 Functions Handouts

Datei: strudel_workshop_functions.js Seite 1 von 2

```
// -- Übung zu Funktionen --

// note bietet die Möglichkeit Melodien zu spielen
note("c a f e").sound("piano")

// slow dehnt das Pattern über mehrere Cycles
sound("bd sd cp sd")

sound("bd sd cp sd").slow(2)

// fast komprimiert das Pattern in weniger als einen Cycle
sound("bd sd cp sd").fast(2)

// Merke: fast 0.5 ist äquivalent zu slow 2
sound("bd sd cp sd").fast(0.5)

// hurry ist ähnlich wie fast. Es spielt ein Pattern schneller ab, aber erhöht auch die
Geschwindigkeit des Samples um den gleichen Faktor
sound("bd sd cp sd").hurry(2)

// Die Funktion jux erzeugt Stereoeffekte, indem sie eine Funktion auf ein Pattern anwendet,
allerdings nur im rechten Kanal.
// In diesem Fall wird das Pattern im rechten Stereo-Kanal in umgekehrter Reihenfolge
abgespielt.
sound("[bd [sd sd] bd sd]*2").jux(rev)

/*
Mit echo können wir einen ähnlich Effekt erzielen wie mit delay. echo hat 3 Parameter:
Der erste parameter steuert wie oft das Sample wiederholt werden soll
Der zweite Parameter gibt an in welchem Abstand das jeweilige Echo zum vorhergehenden
gespielt wird (z.B. im Abstand von 20% eines Cycles)
Der dritte Parameter gibt an um welchen Anteil das jeweils folgende Echo leiser gespielt wird
(z.B. 0.5 = 50% leiser als das vorhergehende Echo)
*/
sound("bd sd").echo(4, 0.125, 0.5)

// Mit stack können wir mehrere Pattern parallel in einem Kanal abspielen und individuell
manipulieren
stack(sound("bd sd [cp hh] sd").pan("0.5 1"),
         sound("bd sd [cp hh] sd").fast(2))

/*
Mit sometimes (und seinen Verwandten) können wir Funktionen anwenden, die nur mit einer
bestimmten Wahrscheinlichkeit ausgeführt werden:
function        likelihood
---------------------------
always      100%
almostAlways    90%
often       75%
sometimes   50%
rarely      25%
almostNever 10%
never       0%
*/

sound("[bd bd] sd*3 bd cp*2").sometimes(x=>x.speed("2"))

// Mit chop können wir Samples in die angegebene Anzahl von Teilen zerstückeln
sound("bd sd cp sd*2").chop(2)

/*
Mit striate können wir einen ähnlichen Effekt wie mit chop erzeugen und die Samples
zerstückeln. In diesem Fall wird die Schleife aber dreimal abgespielt,
beim ersten Mal das erste Drittel jedes Samples, beim zweiten Mal das zweite Drittel jedes
Samples und schließlich das letzte Drittel jedes Samples.
```

B Teacher Feedback (translated from German language)

B.1 Teacher Feedback – Live Coding Sprint

B.1.1 Response Teacher 1

General Observations

- **Questions**
  - How did you perceive the students' interest and participation during the live coding session?
  - Which content-related or methodological aspects did you find particularly engaging?
  - Were there moments when students had particular "aha" experiences? How did you notice?
- **Response**
  - Strong interest, due to the immediate results of generating sounds and rhythms in real time.
  - Student motivation was encouraged by selected introductory videos (e.g., by Dan Gorelick), and a short theoretical introduction (patterns, syntax) enabled a high degree of practical student activity with varied implementation.
  - Students were able to create individual patterns/rhythmic structures (without classification as "right" or "wrong"). They expanded and adapted their code, independently using examples (and excerpts) from the "Getting Started" page on strudel.cc.

Didactic Reflection

- **Questions**
  - How do you assess the use of Strudel compared to other computing approaches in class?
  - Which computational skills or thinking patterns were supported through live coding?
  - Do you see connections to other subjects (e.g., music, art, language, math, physics)?
- **Response**
  - A good way to address student interests and to playfully introduce elements of algorithmic thinking.
  - Breaking down processes into discrete steps (from a sequence of sounds to a pattern and then a rhythm); enforcing syntax rules for generating correct code.
  - Yes – e.g., with Strudel.cc, to music (notation, rhythm, meter, instruments), physics (sound, tones, waves), and potentially also art.

Challenges

- **Questions**
  - Were there challenges or confusion for students (e.g., with syntax, sound output, or tech)?
  - How would you describe the error culture in the live coding session?
- **Response**
  - Definitely—especially with the new syntax, which differed from previously learned programming languages.
  - Positive learning environment: students worked at their own pace and mistakes were treated as natural and constructive.

B.1.2 Response Teacher 2

General Observations

- **Questions**
  - How did you perceive the students' interest and participation during the live coding session?
  - Which content-related or methodological aspects did you find particularly engaging?
  - Were there moments when students had particular "aha" experiences? How did you notice?
- **Response**
  - The "Live Coding" workshop, conducted by the lecturer as part of our Science Day on April 1, was followed with great interest by both the 11th-grade students and the attending teachers—including colleagues from language departments.
  - Particularly noteworthy was the creative connection between coding and music, which opened up a completely new access point to computer science for many participants.
  - The atmosphere was consistently positive: the workshop was engaging, concise, and lively.
  - The students actively participated, showing curiosity and commitment.
  - The lecturer succeeded in involving all participants with motivational and professionally competent guidance.

Didactic Reflection

- **Questions**
  - How do you assess the use of Strudel compared to other computing approaches in class?
  - Which computational skills or thinking patterns were supported through live coding?
  - Do you see connections to other subjects (e.g., music, art, language, math, physics)?
- **Response**
  - From a didactic perspective, the concept of live coding combined with music is particularly promising.
  - It provides a creative and structured way to convey core computer science concepts—such as abstraction, iteration, parameterization, and immediate feedback.
  - Many students, especially girls, were able to gain a new understanding of programming through the musical approach.
  - It also became clear how musical patterns, rhythm, and even elements of music theory can be translated into and transformed by code.
  - The workshop impressively demonstrated how computer science instruction can be linked with artistic subjects such as music.
  - The format holds great potential for interdisciplinary teaching projects.

Challenges

- **Questions**
  - Were there challenges or confusion for students (e.g., with syntax, sound output, or tech)?
  - How would you describe the error culture in the live coding session?
- **Response**
  - Some technical or conceptual stumbling blocks became apparent, especially when dealing with syntax and reacting to unforeseen programming errors.
  - These were used didactically to great effect in the context of live coding.
  - Students were able to experience that errors are an integral part of the programming process.
  - The error culture was constructive throughout: mistakes were reflected upon together,

treated as learning opportunities, and usually addressed with humor and patience.

Conclusion

- **Questions**
  - Additional comments or suggestions?
- **Response**
  - The workshop was a successful example of how creative, hands-on, and interactive formats can enrich computer science instruction.
  - For many participants, it was an unexpectedly inspiring experience to encounter programming in a musical way.
  - We thank the lecturer for the excellent implementation and would welcome a continuation or deepening of this project in the future.
  - One suggestion for improvement relates to time management: the creative closing phase—the presentation of the music pieces created—was somewhat too short.
  - Additional time here would help in appreciating the outcomes and encouraging exchange between student groups.

B.2 Teacher Feedback – Collegial Exchange and Teacher Training

B.2.1 Resonse Teacher 1

Perception of the format

- **Questions**
  - What did you find particularly appealing about the live coding approach—and what was rather irritating?
  - How comprehensible was the example shown in the collegial exchange for you personally (e.g., Strudel syntax, workflow, output)?
  - Did you feel that the format encouraged active participation—or was it more passively consumed?
- **Response**
  - Appealing: browser-based, no additional software needed (Strudel.cc).
  - Also works in school environments, which is not a given (limited IT support).
  - Very comprehensible.
  - However, only a limited scope can be covered in 90 minutes, which is acceptable for a good "extra lesson."
  - Definitely invited participation—clearly hands-on!

Didactic Potential

- **Questions**
  - Where do you see didactic potential for the use of live coding in school or teacher training?
  - Which topics or content areas could be effectively combined with this approach?
- **Response**
  - Offers variety, considers heterogeneity and student interests.
  - Yes – e.g., with Strudel.cc, to music (notation, rhythm, meter, instruments), physics (sound, tones, waves), and potentially also art.
  - Well-suited for interdisciplinary teaching and project-based learning.

Transfer to your own practice

- **Questions**
  - Could you imagine using live coding or elements of it in your own teaching or subject area?

  - What would you still need (e.g., materials, technical equipment, training)?
- **Response**
  - Yes – but rather for interdisciplinary projects, varied lessons (e.g., before holidays), or individual student support.
  - A basic musical education (haha).

B.2.2 Resonse Teacher 2

Perception of the format

- **Questions**
  - What did you find particularly appealing about the live coding approach—and what was rather irritating?
  - How comprehensible was the example shown in the collegial exchange for you personally (e.g., Strudel syntax, workflow, output)?
  - Did you feel that the format encouraged active participation—or was it more passively consumed?
- **Response**
  - I was particularly impressed by the immediacy and the creative aspect of live coding.
  - It is fascinating to see how code generates audible and visible results in real time.
  - The experimental character and the ability to receive direct feedback are highly motivating.
  - However, I was irritated by the complexity of the syntax, especially for students with little programming experience.
  - The learning curve seems to be very steep.
  - The examples were generally understandable for me as a teacher, but I quickly realized that the Strudel syntax and workflow are significantly more complex than they initially appear.
  - For a first impression, they were fine, but a deeper understanding would require more time and practice.
  - In principle, the format invites active participation, but in practice it felt rather passive, as the technical hurdles and the complexity of the syntax were initially overwhelming

Didactic Potential

- **Questions**
  - Where do you see didactic potential for the use of live coding in school or teacher training?
  - Which topics or content areas could be effectively combined with this approach?
- **Response**
  - The didactic potential, in my view, lies in the combination of programming with creative outputs.
  - This is especially valuable for interdisciplinary projects between computer science and music/art.
  - It could motivate students who are usually not interested in pure programming.
  - Music theory and rhythm
  - Mathematical patterns and algorithms
  - Creative media design
  - Project-based learning in cross-curricular approaches

Transfer to Practice

- **Questions**

  - Could you imagine using live coding or elements of it in your own teaching or subject area?
  - What would you still need (e.g., materials, technical equipment, training)?
- **Response**
  - Yes, but only in a simplified form and with significantly more preparation time.
  - Full-fledged live coding is too complex for lower secondary students, but selected elements could be interesting.
  - Time: this is a critical factor. Currently, half of our computer science classes are canceled, leaving little room for experimental approaches.
  - Simplified materials: beginner-friendly tutorials and templates.
  - Technical equipment: reliable computers with appropriate software.
  - Additional training: more hands-on time to use the tools confidently.
  - Curricular integration: clear alignment with curriculum content.

C Review Log

Review "Lange Nacht der Technik" 2025 (english Version)

I liked that

- commitment of all involved
- Easy access to programming
- lighting effects and visualization
- continuous music (from rehearsal corners) --> attracted people
- young people looking after the stand to keep initial contact with other young people low-threshold
- interplay of technology and creativity came into its own
- Organizationally well structured

I would wish for this

- More powerful projectors/daylight projectors
- Separate rehearsal corners for adults and children
  - rehearsal corner via PA and for more intensive sessions at a second workstation with headphones
- Structured, guided and block-based introduction to live coding
- Printout of the drums as a poster and assignment of the samples to the individual parts of the instrument
- Strudel documentary in German
- better announcement of the study application aspect (on posters)/ computer science studies must be better emphasized on posters
- more intensive familiarization with live coding --> briefing on the event sooner than a week
- Stable streaming server --> clarify streaming server hosting with university data center
- Scene on site and Hydra manipulation must be more clearly recognizable --> if necessary, alternate Hydra manipulation based on time
- Interactive-collaborative live coding (e.g. https://youtu.be/iKaoOjlHyqE?si=KiKqGFF_9zEOIrat)
- Place tweeters and subwoofers in a row to improve the sound
- Discuss decoration more intensively
- View the room in advance with all participants
- Bring visualization to the corridor --> e.g. swirl notes/instrument visualization as a video
- LED strips in the corridor as light orientation towards the stand
- Live coding mini diploma/medal (for kids) --> 3D printed
- Stamp booklet for scavenger hunt
- Hourly live show format of the main act, accompanied by continuous live coding performances by individual participants or small groups
- At least two live performances in duo format for more dynamics
- Location should be more central/well-known
- More intensive promotion of the event