What You Say is What You Get

Handsfree Coding in 2026

ICCI'26, Bangalore, India

January 28, 2026

Wolle

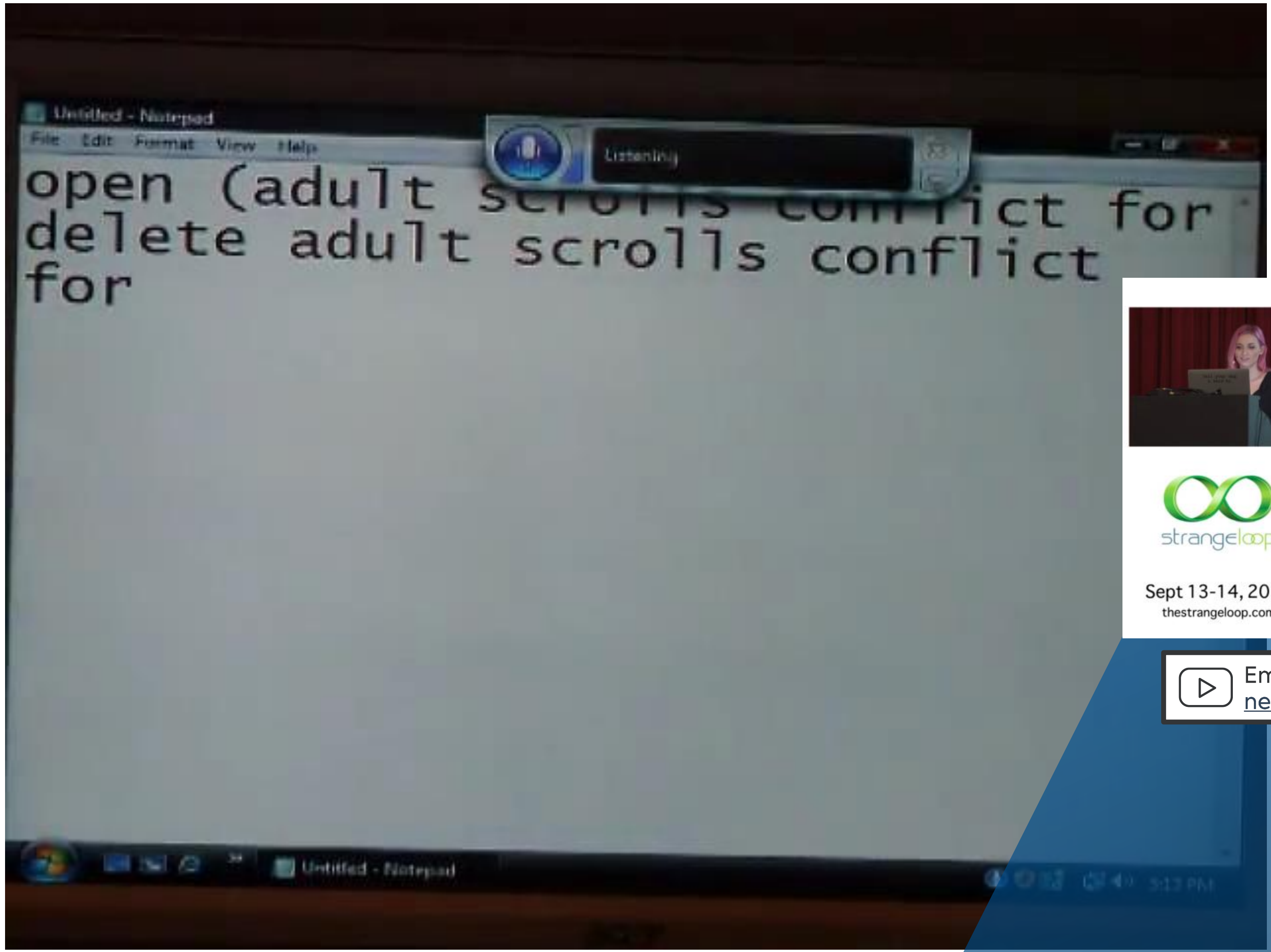
Videos & Slides Available at https://wolle.science

It‘s Simple, Really!

The requirements:

- ✓ **Microphone**: Every notebook has one!
- ✓ **Speech Recognition Software (SR)**: Included in Windows since 2007!
- ✓ **Voice Command Execution**: Available in every SR software!

Let Me Just Show You How Easy It is



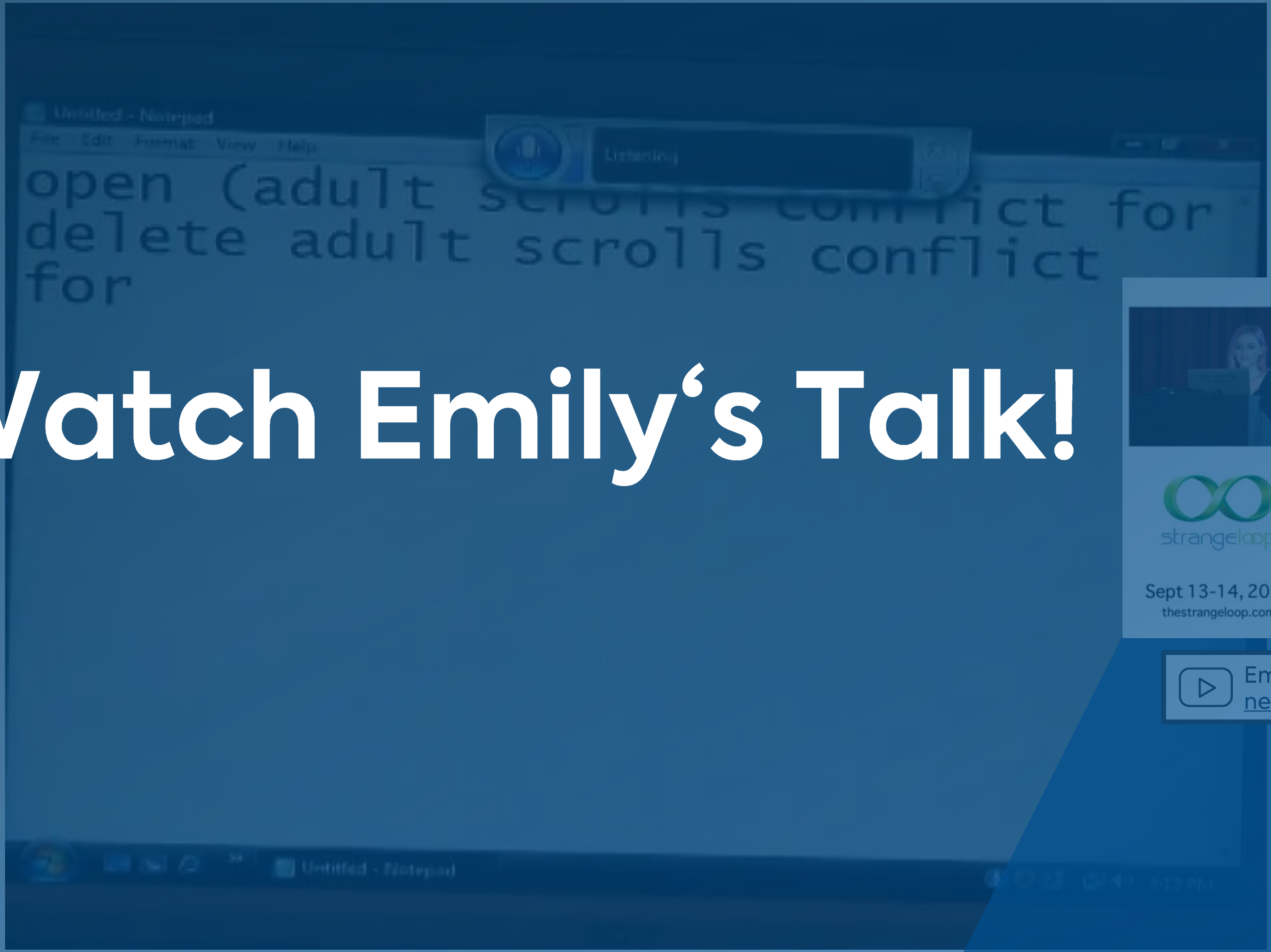
```
open (adult scrolls conflict for
delete adult scrolls conflict
for
```

Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

scrubadub1. Windows Vista Speech Recognition Tested - Perl Scripting, YouTube, 2007

Idea to use this video blatantly stolen from: Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop, 2019

Let Me Just Show You How Easy It is



```
open (adult scrolls conflict for
delete adult scrolls conflict
for
```

Go Watch Emily‘s Talk!

Sept 13-14, 2019
thestrangeloop.com

whois emily

- Software Engineer
- GitHub: @2shea
- Twitter: @yomilly
- I write code for Fastly

Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

scrubadub1. Windows Vista Speech Recognition Tested - Perl Scripting, YouTube, 2007

Idea to use this video blatantly stolen from: Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop, 2019

Where‘s the Challenge?

Where's the Challenge?

WSR, Dragon, ...

- **Automatic Speech Recognition (ASR)**: optimized for <u>natural</u> languages
  1. Signal processing extracts features from audio recording
  2. Acoustic model recognizes phonemes
  3. Language model finds a matching sequence of words:

  → <u>Default</u>: Every utterance is interpreted as (spoken) text

  (Commands only through special keywords)

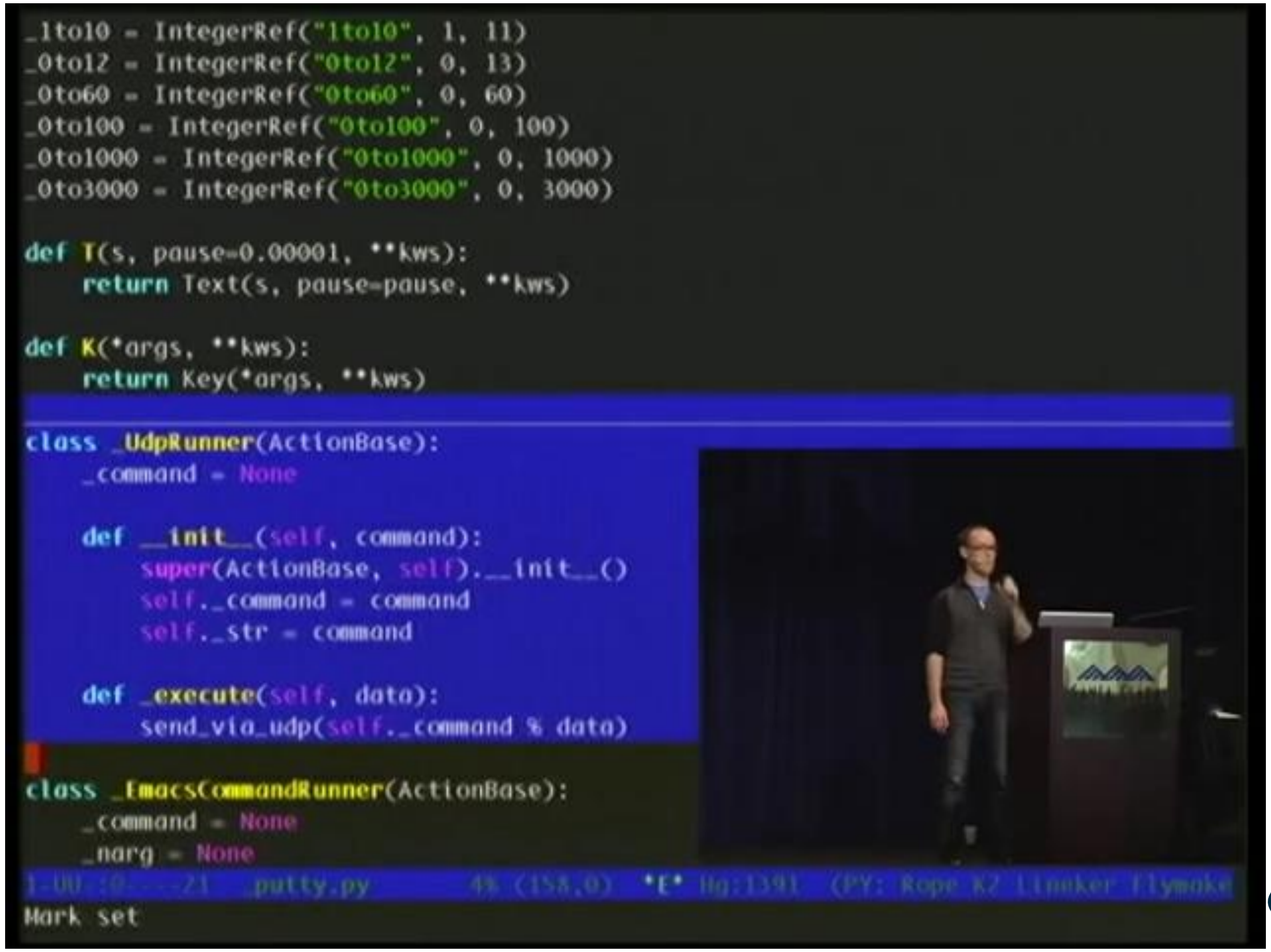
Talon, Dragonfly ...

- **Voice Coding**: optimized for <u>actions & programming</u> languages

  → <u>Default</u>: Everything is interpreted as a command

  (Natural language through special keywords, e.g. `say <utterance>`)

Handsfree Coding: How It Actually Looks



```
_1to10 = IntegerRef("1to10", 1, 11)
_0to12 = IntegerRef("0to12", 0, 13)
_0to60 = IntegerRef("0to60", 0, 60)
_0to100 = IntegerRef("0to100", 0, 100)
_0to1000 = IntegerRef("0to1000", 0, 1000)
_0to3000 = IntegerRef("0to3000", 0, 3000)

def T(s, pause=0.00001, **kws):
    return Text(s, pause=pause, **kws)

def K(*args, **kws):
    return Key(*args, **kws)

class _UdpRunner(ActionBase):
    _command = None

    def __init__(self, command):
        super(ActionBase, self).__init__()
        self._command = command
        self._str = command

    def _execute(self, data):
        send_via_udp(self._command % data)

class _EmacsCommandRunner(ActionBase):
    _command = None
    _narg = None
1-UU-:@---21   putty.py       4% (158,0)  *E* Hg:1391  (PY: Rope K2 Linker Flymake
Mark set
```

Tavis Rudd. Using Python to Code by Voice, PyCon US (2013)

Outline: What This Talk is Going to Cover

1. **My Personal Background**

   As data engineer & scientist, I use handsfree coding every day.

2. **Demo & Usage Examples**

   Handsfree coding is awesome and can be useful for everyone!

3. **Setup & Best Practices**

   No-cost base setup with optional upgrades (e.g. for eye tracking).

4. **How to Get Started**

   Videos, blogs, articles, support & community – engage now!

My Job is Data Science

I Am Wolle

Wolfram Wingerath
Data Science

Research:
- Stream Processing
- Real-Time Databases
- NoSQL & Cloud Systems
- …

+

Practice:
- Web Caching
- Big Data Analytics
- Anger Management
- …

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Carl von Ossietzky
Universität
Oldenburg

Baqend

Look, No Hands!

Basic Voice Control

- **Symbols, Modifiers & Navigation:**
  - Numbers, brackets, etc.: $\overbrace{\texttt{one space air bat shift one}}^{\text{What you say}} \rightarrow \overbrace{\texttt{1 ab!}}^{\text{What you get}}$
- **Spelling** through a phonetic alphabet:
  - NATO alphabet: `alpha bravo charlie delta echo` → `abcde`
  - Optimized alphabet: `air bat cam drum each` → `abcde`
- **Command Management** for efficiency, e.g.:
  - Chaining: $\boxed{\texttt{paren close paren}}\boxed{\texttt{go left}}\boxed{\texttt{say hi}} \rightarrow \texttt{(hi)}$

    1.) type: `()` 2.) move cursor: ⇦ 3.) dictate: `hi`
- **On-the-fly grammar prototyping** through Python live reloading!

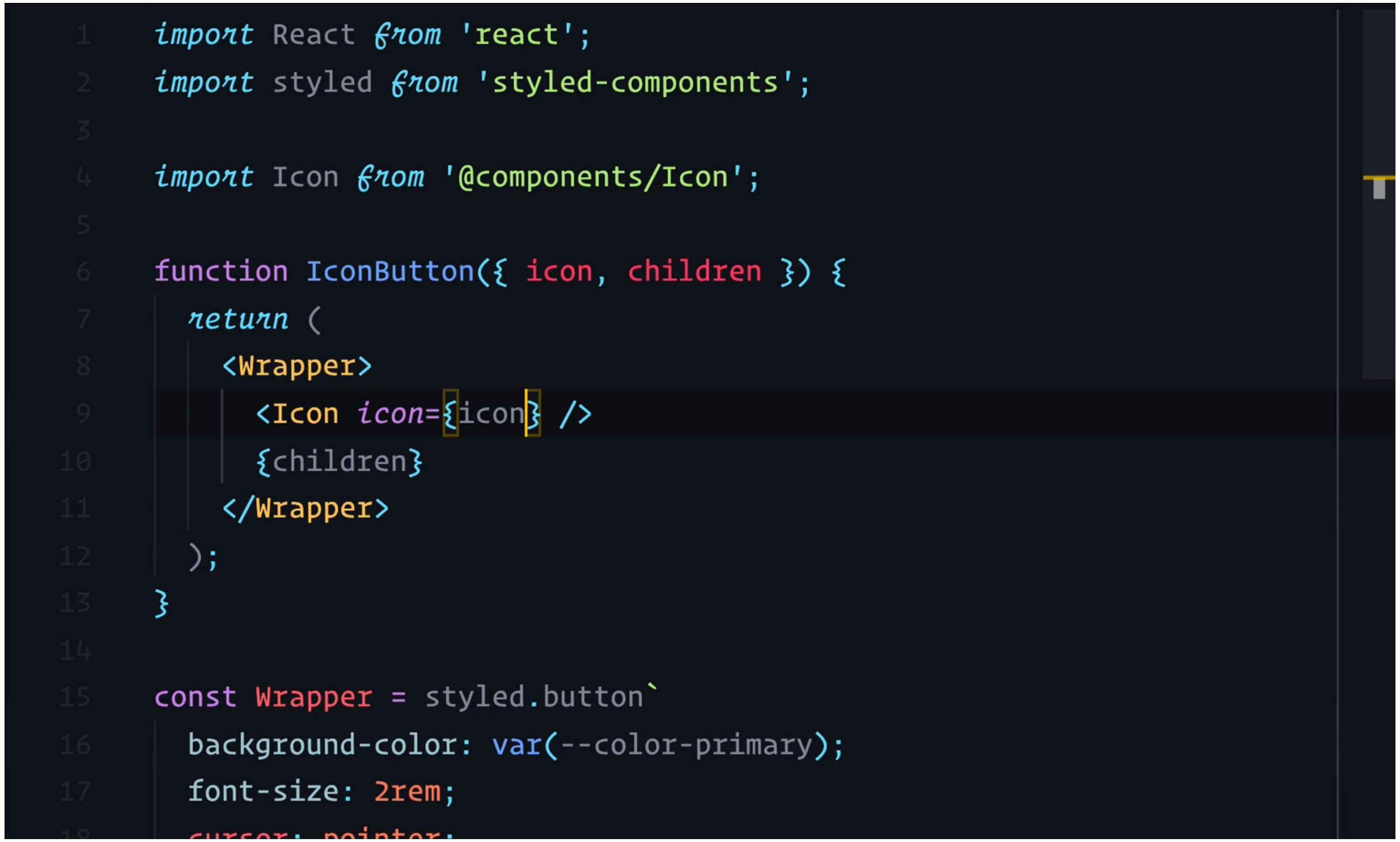
Live Demo

HandsfreeCoding

- **Context-dependent behavior**, for example:
  - C#: `funky test funk` → `private void testFunk()`
  - JavaScript: `funky test funk` → `function testFunk()`
- **Intuitive IDE shortcuts** such as
  - `"run code"` instead of `<shift-f10>`
  - `"find usage"` instead of `<ctrl-alt-f7>`
- **Powerful templates**, `e.g.:`

```
action(user.code_state_if):
  insert("if () {}")
  key(left enter up end left left left)
```

Handsfree Coding: Talon



```
import React from 'react';
import styled from 'styled-components';

import Icon from '@components/Icon';

function IconButton({ icon, children }) {
  return (
    <Wrapper>
      <Icon icon={icon} />
      {children}
    </Wrapper>
  );
}

const Wrapper = styled.button`
  background-color: var(--color-primary);
  font-size: 2rem;
```

Handsfree Coding: Talon + Cursorless

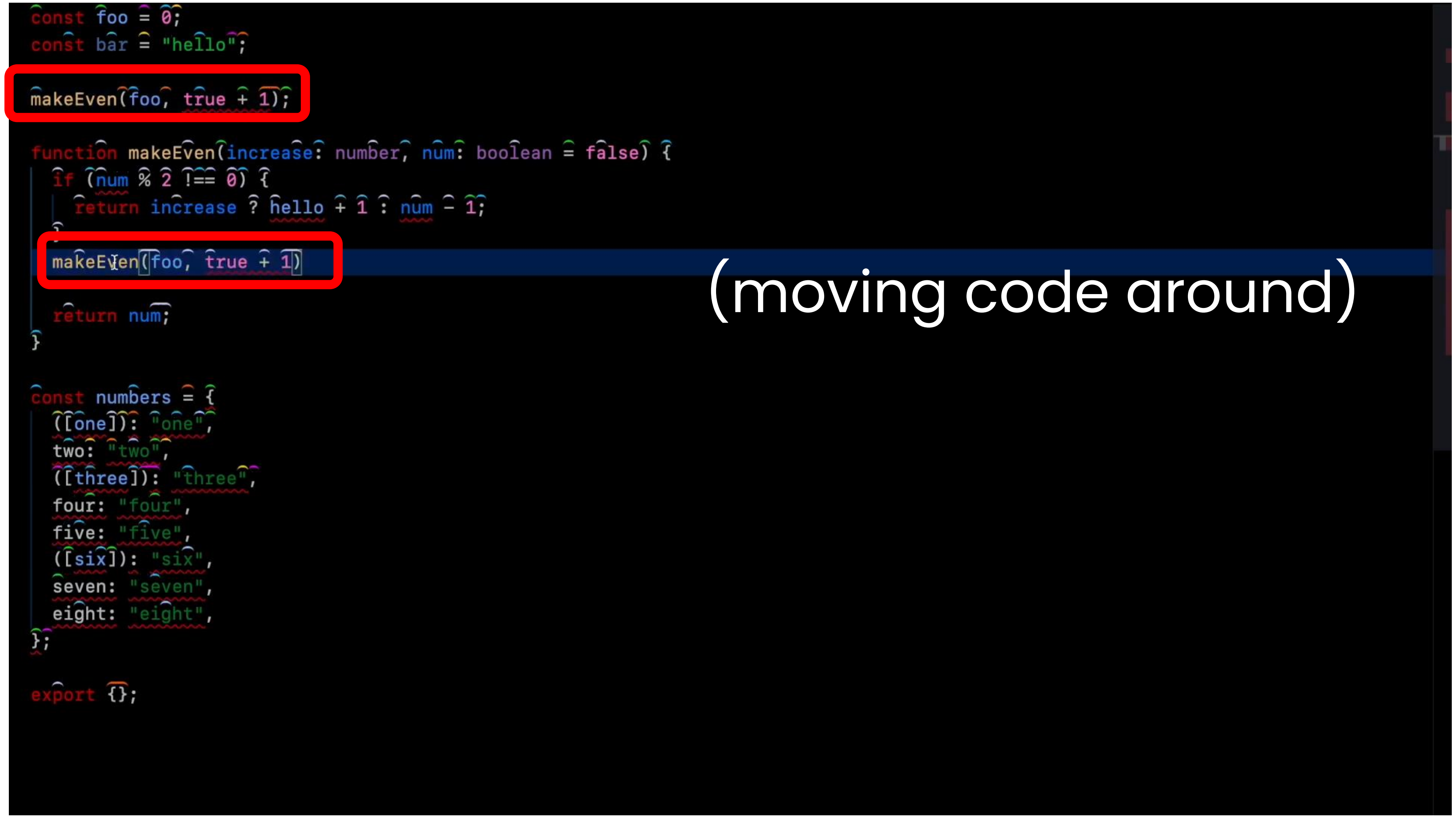
- Available on GitHub: github.com/cursorless-dev/
- VSCode extension
- Spoken language for **structural code editing**
  - Decorates every token on screen with a **mark**
  - tokens can be selected via combination of mark and **scope**
  - **actions** operate on the specified tokens

- Example:

```
bring     call     vest
action    scope    mark
```

(copy the function call with the marked „v“ to where my cursor is)

Handsfree Coding: Talon + Cursorless

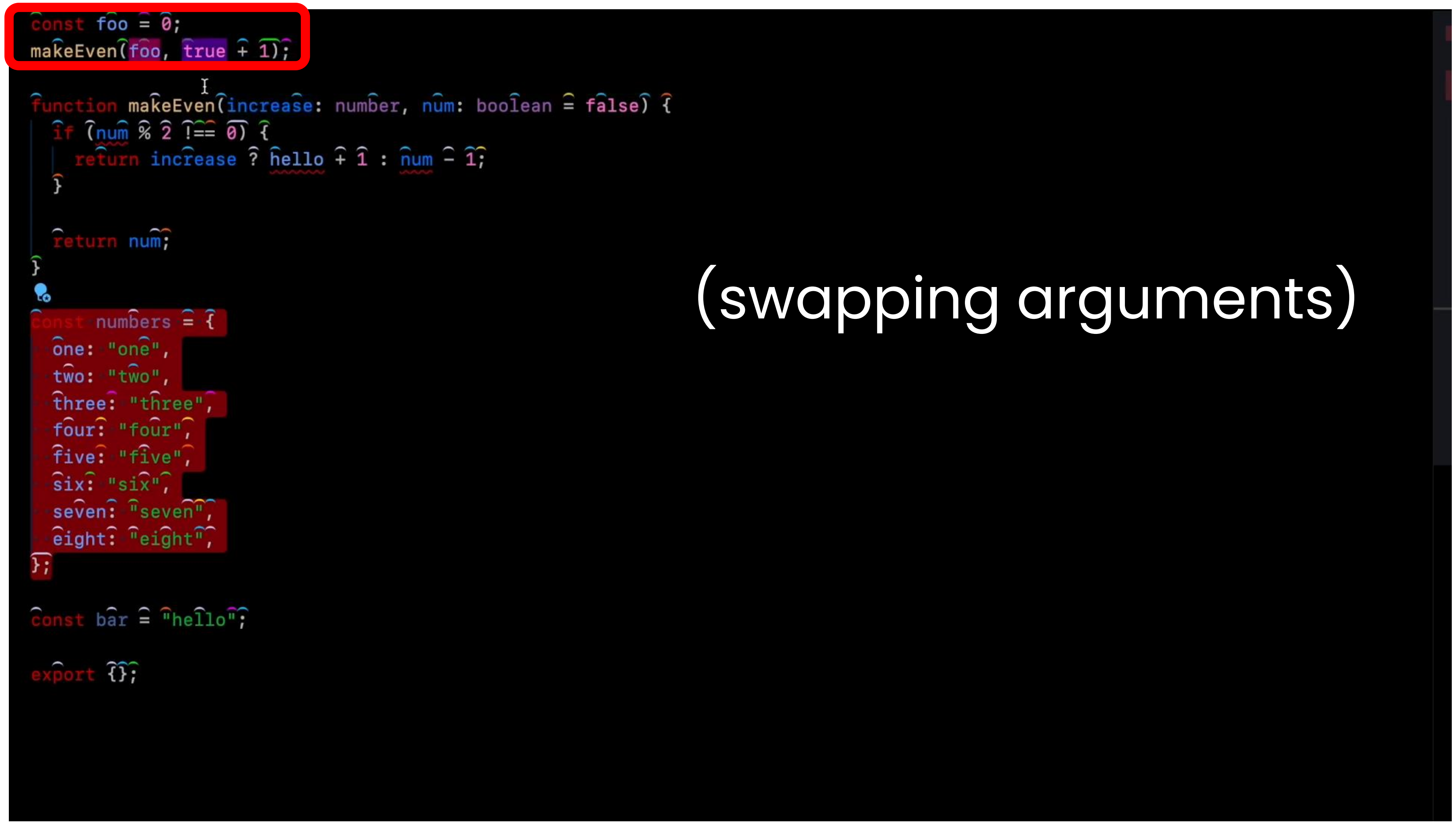


```
const foo = 0;
const bar = "hello";

makeEven(foo, true + 1);

function makeEven(increase: number, num: boolean = false) {
  if (num % 2 !== 0) {
    return increase ? hello + 1 : num - 1;
  }
  makeEven(foo, true + 1)

  return num;
}

const numbers = {
  ([one]): "one",
  two: "two",
  ([three]): "three",
  four: "four",
  five: "five",
  ([six]): "six",
  seven: "seven",
  eight: "eight",
};

export {};
```

(moving code around)

Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Handsfree Coding: Talon + Cursorless

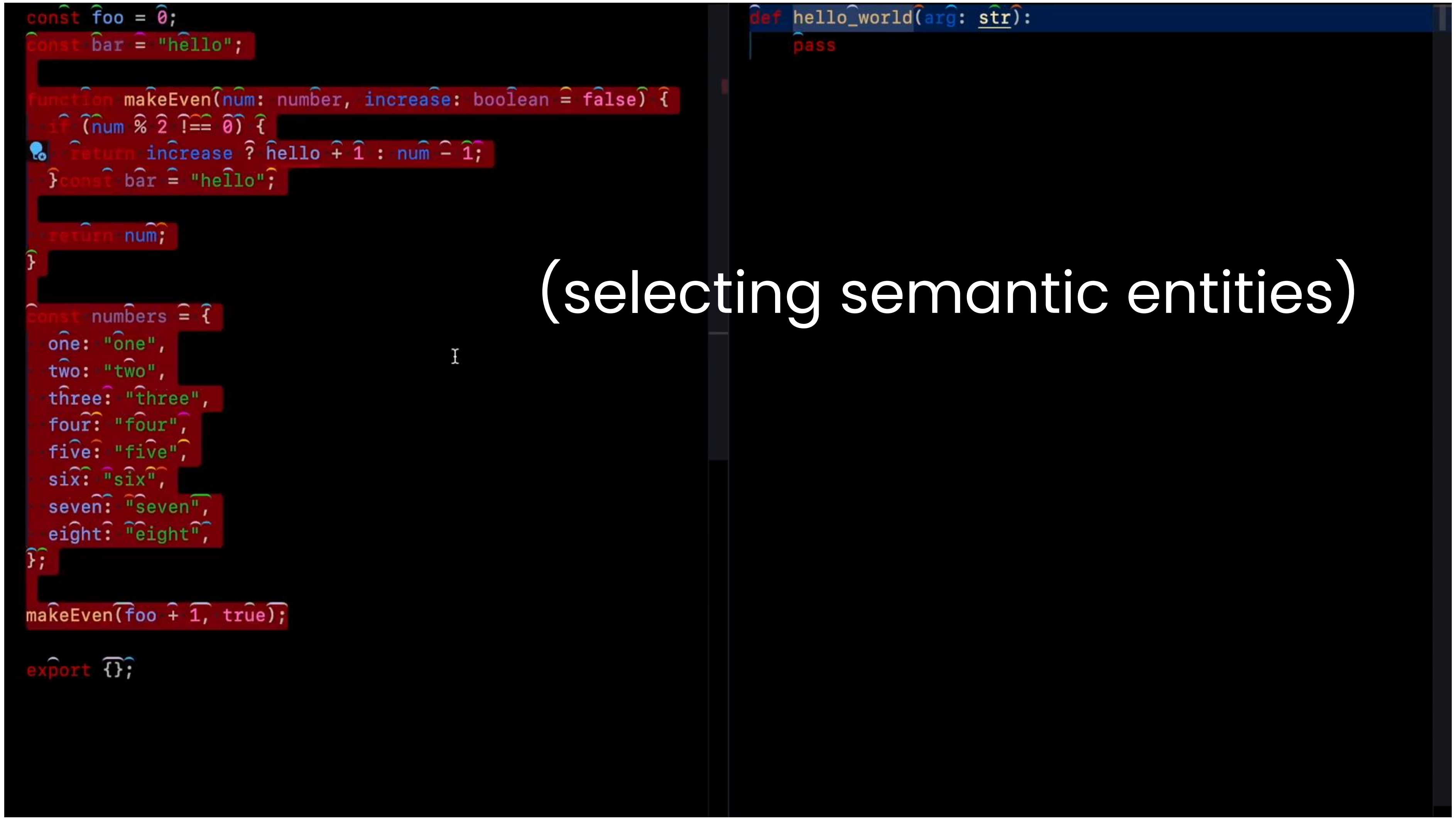


```
const foo = 0;
makeEven(foo, true + 1);

function makeEven(increase: number, num: boolean = false) {
  if (num % 2 !== 0) {
    return increase ? hello + 1 : num - 1;
  }

  return num;
}

const numbers = {
  one: "one",
  two: "two",
  three: "three",
  four: "four",
  five: "five",
  six: "six",
  seven: "seven",
  eight: "eight",
};

const bar = "hello";

export {};
```

(swapping arguments)

Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Handsfree Coding: Talon + Cursorless



```
const foo = 0;
const bar = "hello";

function makeEven(num: number, increase: boolean = false) {
  if (num % 2 !== 0) {
    return increase ? hello + 1 : num - 1;
  }const bar = "hello";

  return num;
}

const numbers = {
  one: "one",
  two: "two",
  three: "three",
  four: "four",
  five: "five",
  six: "six",
  seven: "seven",
  eight: "eight",
};

makeEven(foo + 1, true);

export {};
```

```
def hello_world(arg: str):
    pass
```

(selecting semantic entities)

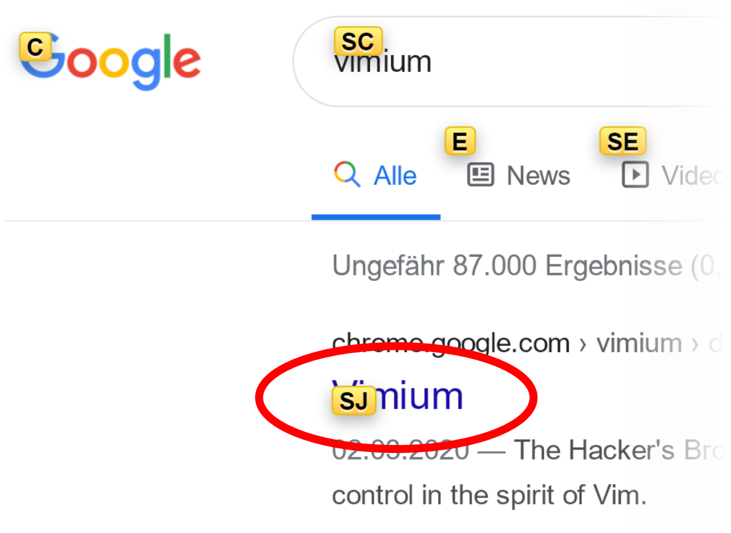
Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Handsfree Browsing

- **Vimium** browser extension: vimium.github.io/

  Example navigation without keyboard:

  1. Show clickable links by pressing the `f` key
  2. Press `sj` keys to click on Vimium link



Ungefähr 87.000 Ergebnisse (0,

chrome.google.com › vimium › c

Vimium

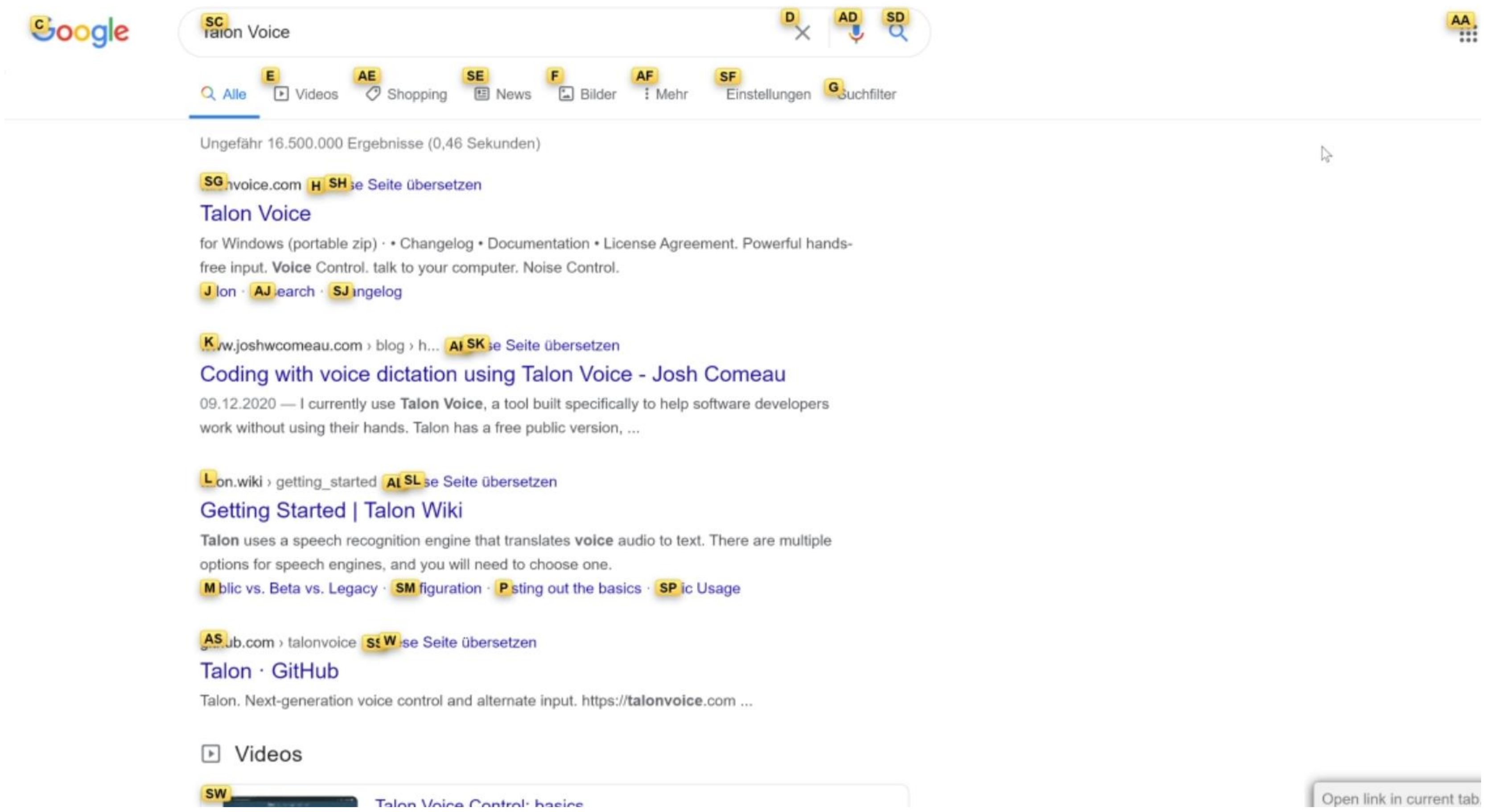
02.03.2020 — The Hacker's Bro

control in the spirit of Vim.

- Natively compatible with tools like **Talon**
  - Simplify shortcuts with easy-to-remember utterances
  - Optional: eye tracking + noise control to select links with your gaze

- **Rango** browser extension for Talon: github.com/david-tejada/rango

Handsfree Browsing: Vimium



Google

Talon Voice

Alle · Videos · Shopping · News · Bilder · Mehr · Einstellungen · Suchfilter

Ungefähr 16.500.000 Ergebnisse (0,46 Sekunden)

voice.com · e Seite übersetzen

Talon Voice

for Windows (portable zip) · • Changelog • Documentation • License Agreement. Powerful hands-free input. **Voice** Control. talk to your computer. Noise Control.

lon · earch · ngelog

w.joshwcomeau.com › blog › h... · e Seite übersetzen

Coding with voice dictation using Talon Voice - Josh Comeau

09.12.2020 — I currently use **Talon Voice**, a tool built specifically to help software developers work without using their hands. Talon has a free public version, ...

on.wiki › getting_started · se Seite übersetzen

Getting Started | Talon Wiki

Talon uses a speech recognition engine that translates **voice** audio to text. There are multiple options for speech engines, and you will need to choose one.

blic vs. Beta vs. Legacy · figuration · sting out the basics · ic Usage

ub.com › talonvoice · se Seite übersetzen

Talon · GitHub

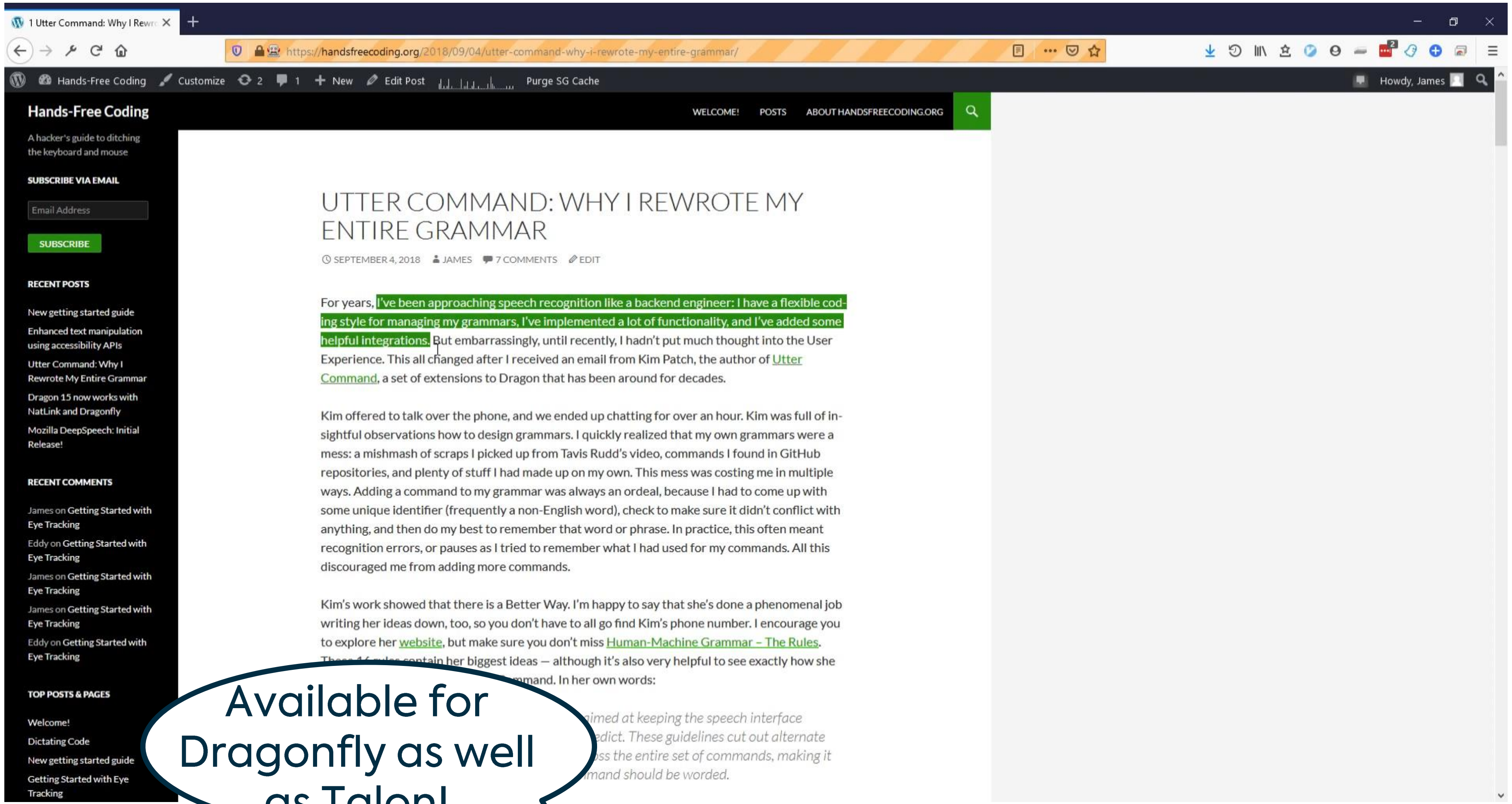
Talon. Next-generation voice control and alternate input. https://**talonvoice**.com ...

Videos

Talon Voice Control: basics

Open link in current tab

Handsfree Browsing: Gaze OCR



1 Utter Command: Why I Rewr

https://handsfreecoding.org/2018/09/04/utter-command-why-i-rewrote-my-entire-grammar/

Hands-Free Coding · Customize · 2 · 1 · New · Edit Post · Purge SG Cache · Howdy, James

Hands-Free Coding

A hacker's guide to ditching the keyboard and mouse

WELCOME! · POSTS · ABOUT HANDSFREECODING.ORG

SUBSCRIBE VIA EMAIL

Email Address

SUBSCRIBE

RECENT POSTS

- New getting started guide
- Enhanced text manipulation using accessibility APIs
- Utter Command: Why I Rewrote My Entire Grammar
- Dragon 15 now works with NatLink and Dragonfly
- Mozilla DeepSpeech: Initial Release!

RECENT COMMENTS

- James on Getting Started with Eye Tracking
- Eddy on Getting Started with Eye Tracking
- James on Getting Started with Eye Tracking
- James on Getting Started with Eye Tracking
- Eddy on Getting Started with Eye Tracking

TOP POSTS & PAGES

- Welcome!
- Dictating Code
- New getting started guide
- Getting Started with Eye Tracking

UTTER COMMAND: WHY I REWROTE MY ENTIRE GRAMMAR

SEPTEMBER 4, 2018 · JAMES · 7 COMMENTS · EDIT

For years, I've been approaching speech recognition like a backend engineer: I have a flexible coding style for managing my grammars, I've implemented a lot of functionality, and I've added some helpful integrations. But embarrassingly, until recently, I hadn't put much thought into the User Experience. This all changed after I received an email from Kim Patch, the author of Utter Command, a set of extensions to Dragon that has been around for decades.

Kim offered to talk over the phone, and we ended up chatting for over an hour. Kim was full of insightful observations how to design grammars. I quickly realized that my own grammars were a mess: a mishmash of scraps I picked up from Tavis Rudd's video, commands I found in GitHub repositories, and plenty of stuff I had made up on my own. This mess was costing me in multiple ways. Adding a command to my grammar was always an ordeal, because I had to come up with some unique identifier (frequently a non-English word), check to make sure it didn't conflict with anything, and then do my best to remember that word or phrase. In practice, this often meant recognition errors, or pauses as I tried to remember what I had used for my commands. All this discouraged me from adding more commands.

Kim's work showed that there is a Better Way. I'm happy to say that she's done a phenomenal job writing her ideas down, too, so you don't have to all go find Kim's phone number. I encourage you to explore her website, but make sure you don't miss Human-Machine Grammar – The Rules. These 14 rules contain her biggest ideas — although it's also very helpful to see exactly how she ...mmand. In her own words:

...imed at keeping the speech interface ...edict. These guidelines cut out alternate ...ss the entire set of commands, making it ...mand should be worded.

Available for Dragonfly as well as Talon!

James Stout, Gaze OCR: Talon support and 10 new features!, Hands-Free Coding Blog (2022)

James Stout. Say what you see: Efficient UI interaction with OCR and gaze tracking, Hands-Free Coding Blog (2020)

Snappy Noise Control With Parrot

- Available on GitHub: github.com/chaosparrot/parrot.py
- Noise-controlled actions with latency <50ms
- Workflow
  (1) Record sounds
  (2) Train model for recognition
  (3) Map sounds to actions
- Compatible (and recommended in combination with) with other tooling:
  - Often used with Project IRIS (eye tracking)
  - Can be used to produce **Talon-compatible** models

Custom noises for your Talon grammar!

Handsfree Gaming: Noise Recognition

Cian Dowd. Speed Runners Hands-Free, YouTube (2021)

Eye Tracking & Noise Recognition

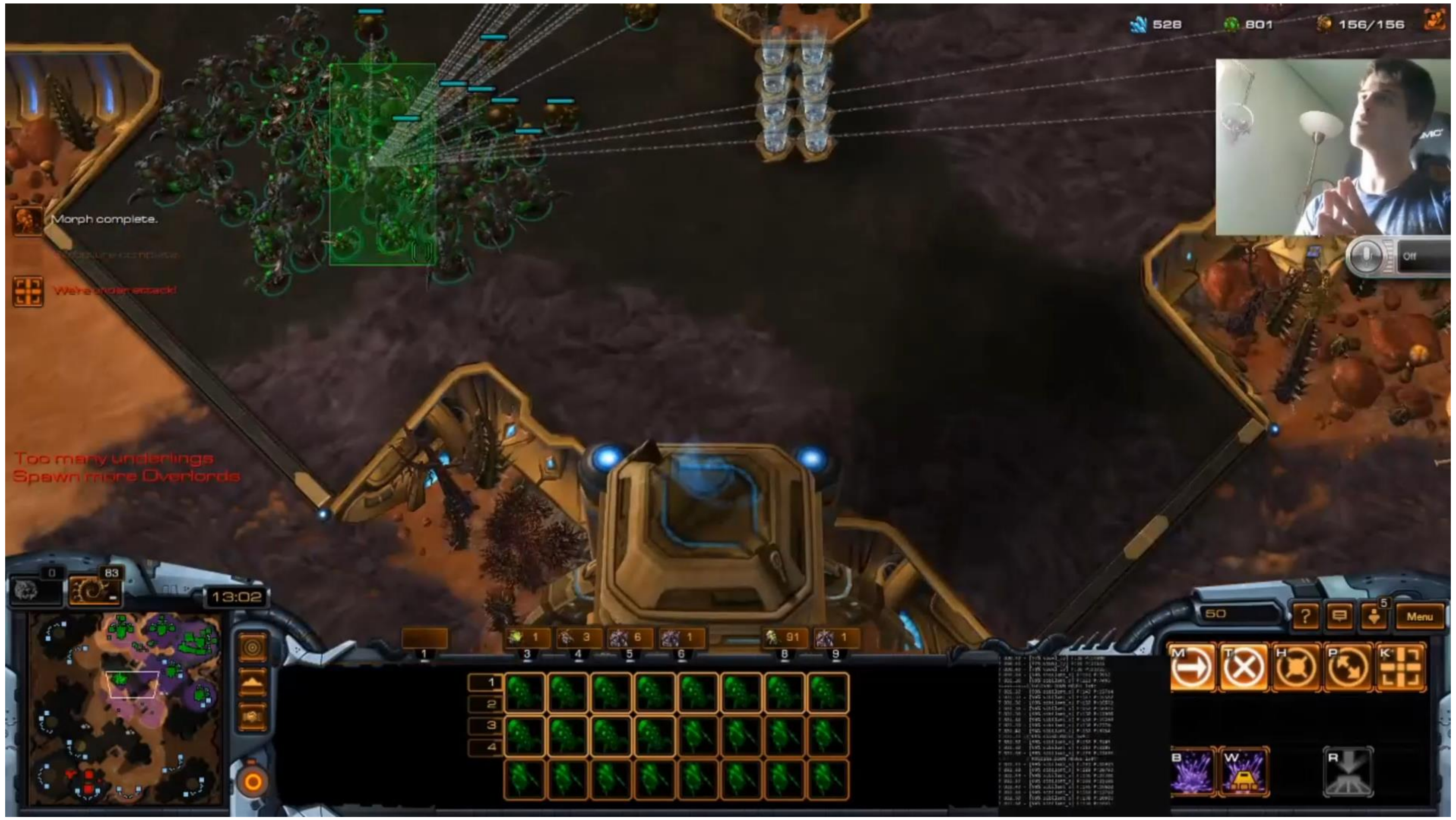
- **Calibration** for adjusting your eye tracker to your current position
- **Noises** for actions (e.g. clicking & right-clicking):
  - Extremely low latency (<50ms)
  - Talon currently supports `*pop*` & `*hiss*`
  - Custom noise models available via Parrot

github.com/chaosparrot/parrot.py

- **Different Modes** for convenience:
  - Zoom: (1) `*pop*` for zooming, (2) `*pop*` for clicking
  - Head tracking: eye gaze (jumps) + head movement (adjustment)
- **Debug** mode & camera overlay

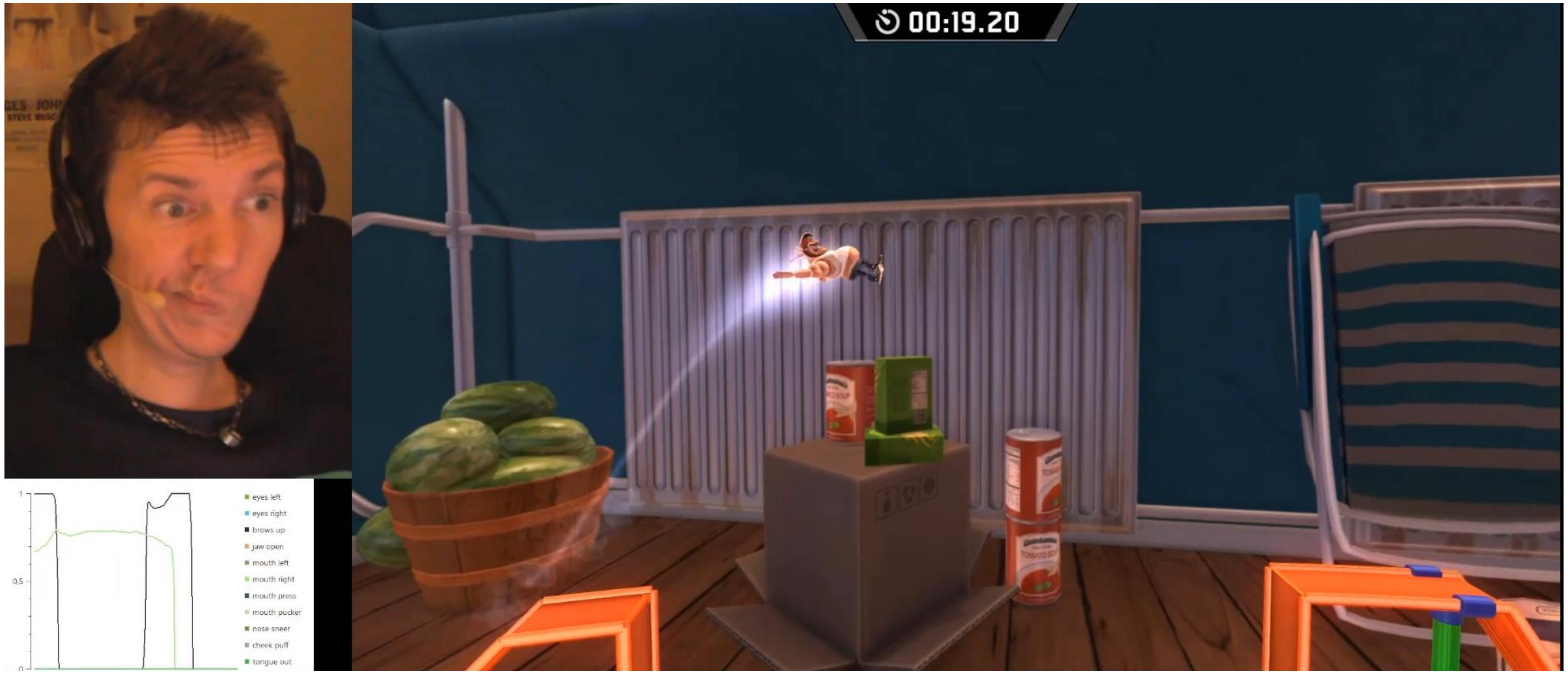
Handsfree Gaming: Head + Eye Tracking

Handsfree Gaming: Noises + Eye Tracking



ParrotPy. Handsfree Starcraft 2 – Diamond Zerg vs Terran, YouTube (2020)

Handsfree Gaming: Facial Actions

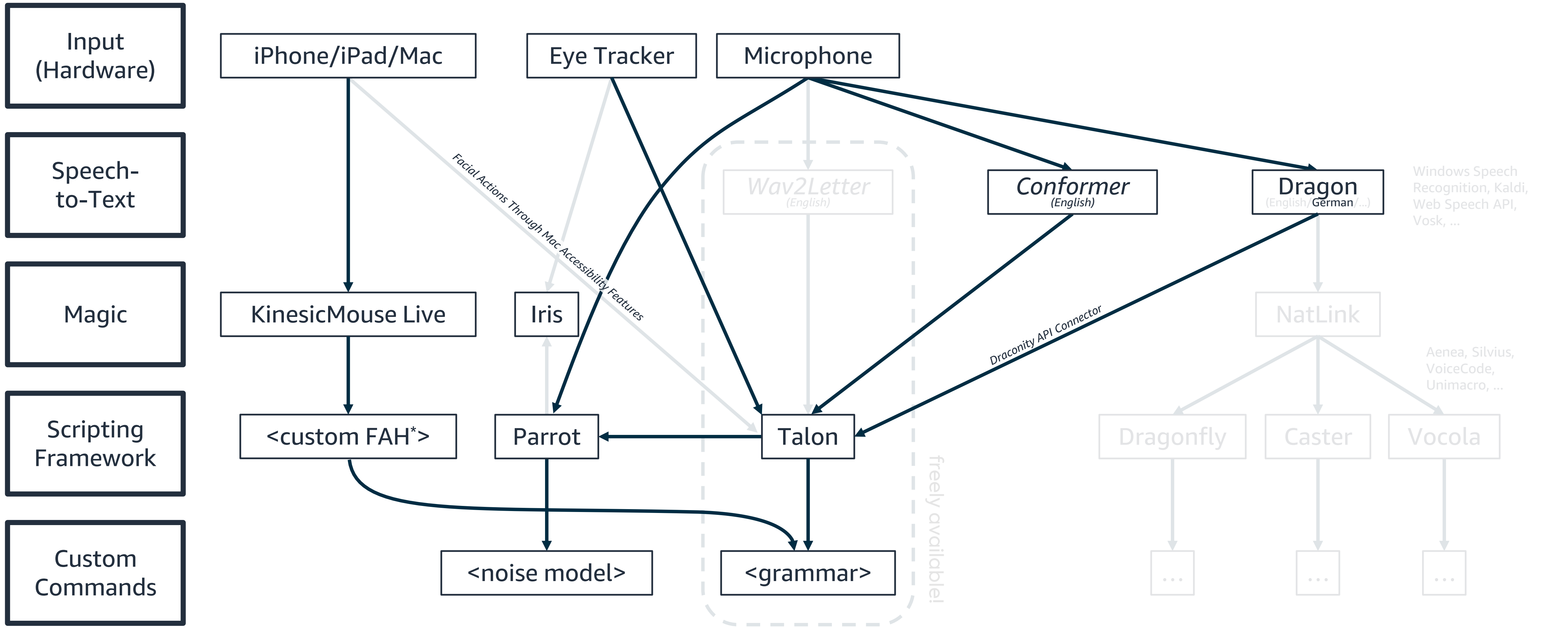


Handsfree Gaming: Eyes + Face + Voice/Noise

wolle.science/twitch

The Base Setup

Popular Handsfree Coding Stacks: Overview



*Facial Action Handling
Please note that this overview is NOT complete: On every level, there are MANY other options!

This overview was inspired by:
https://dictation-toolbox.github.io/dictation-toolbox.org/ (accessed: January 4, 2021)

Upgrades & Add-Ons

Supplementary Equipment

- <u>**High-quality (!)**</u> **microphone**: Get one now!
  - A wired (or good Bluetooth!) headset
  - Steno-mask: For noisy places & special looks
  - XLR mic for maximum accuracy

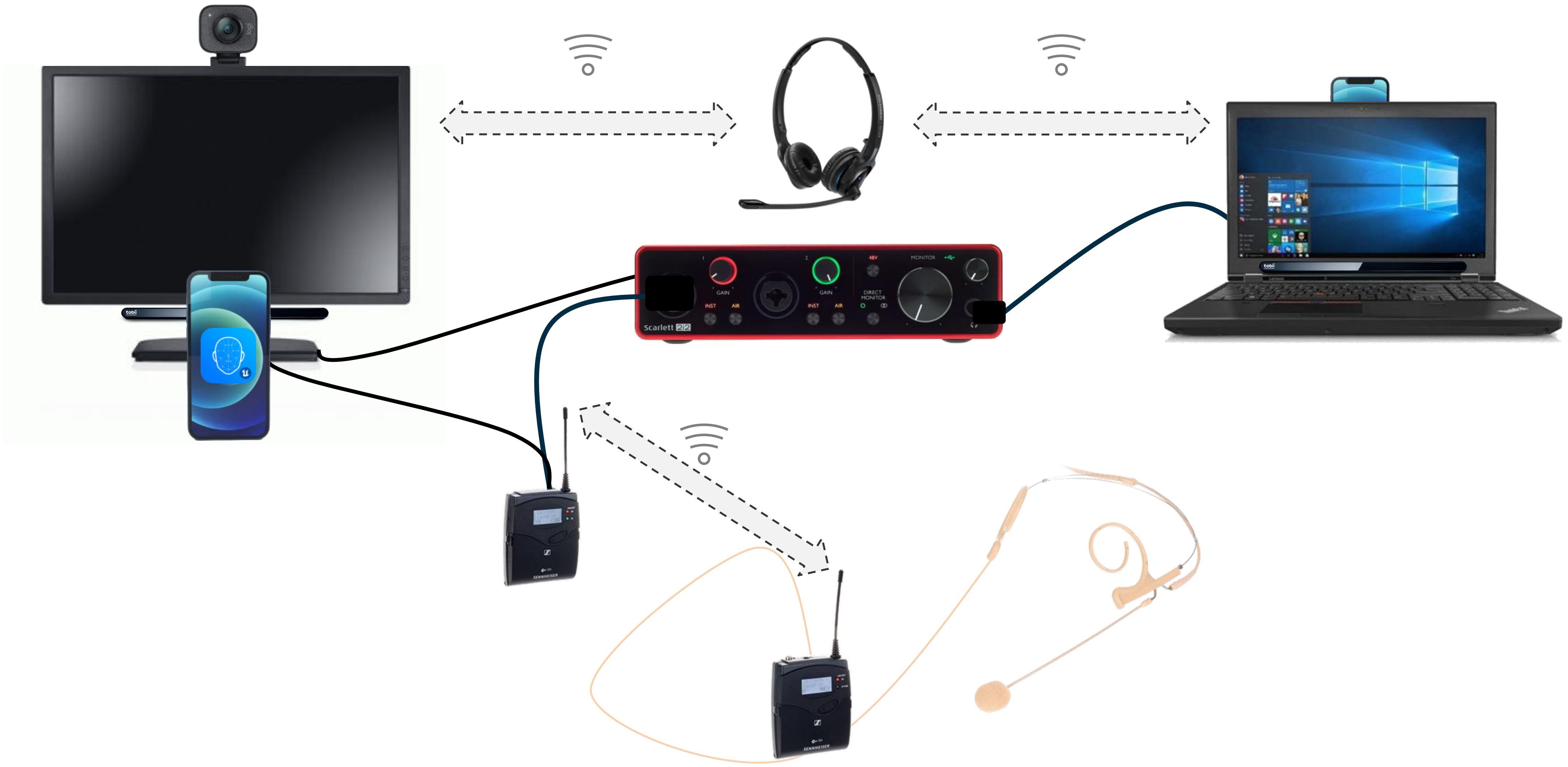
  → + sender/receiver for max. convenience
- **Foot pedal** for push-to-talk
- **Eye tracker**: Tobii 4C & 5 supported by Talon
  - Multi-monitor support coming (?)

DPA 4188

Sennheiser MB Pro 1/2

Sennheiser EW 112P G4 E-Band

Multi-Computer Setup



Pitfalls & Challenges

Recognition Accuracy Issues

- **Microphone** determines accuracy!
  - *Build quality*: built-in < gaming headset < stage mic
  - *Positioning*: consistent, close to your mouth, away from all noise
  - *Mixed bag: Noise canceling* via hardware or software (e.g. RTX Voice)
- **Environment**: Minimize noise for you and annoyance for others!
  - Suspend ASR / mute mic accordingly (e.g. via push-to-talk pedal)
- **Homophones** should be avoided, e.g. through:
  - Grammar optimization to avoid ambiguity
  - Clear pronunciation

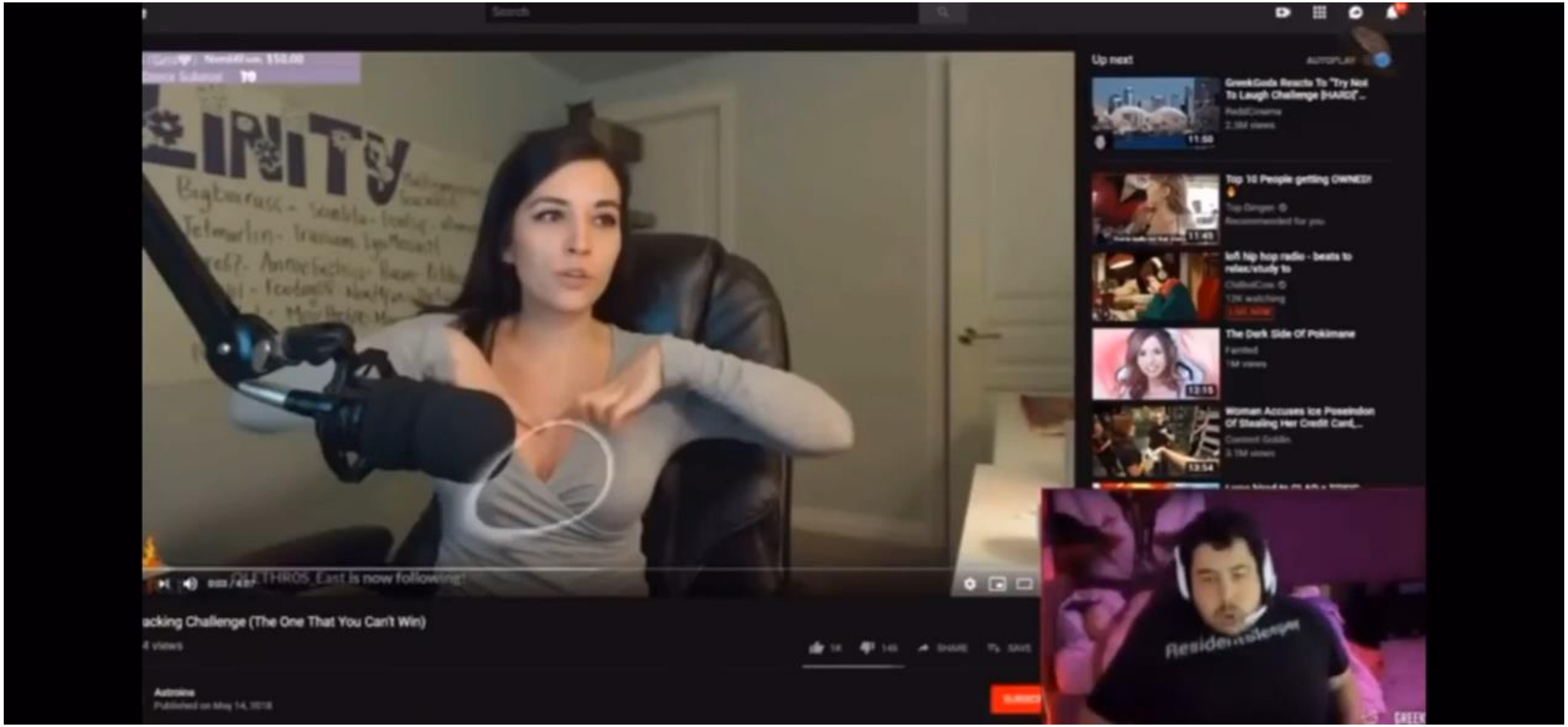
Potential Privacy Issues

- **Watch Your Tongue**: Passwords & confidential info may be leaked …
  - … through plain acoustics (beware *eavesdroppers*!)
  - … as they are stored your *command history*!
  - … to involved third parties (e.g. with *Web Speech*)
- **Watch Your Transmitter**: Wireless solutions are often not encrypted!
- **Watch Your Eyes**: Your eye movement may give away a lot

  → perhaps avoid continuous eye tracking ;-)

<insert eye tracking challenge joke here>

Potential Privacy Issues


Up next
acking Challenge (The One That You Can't Win)
GreekGods Reacts To "Try Not To Laugh Challenge [HARD]"...
Top 10 People getting OWNED!
lofi hip hop radio - beats to relax/study to
The Dark Side Of Pokimane
Woman Accuses Ice Poseidon Of Stealing Her Credit Card...

Greekgodx. Eye Tracking Challenge, Twitch (2019)

Workflow & Anger Management Issues

- **Beware the Trolls**: Having an audience generally does not help!

  → Prepare to hear „Format C“ from your colleagues a lot
- **Keep your calm**: Shouting at the computer will not help, either!

  → Stay in your neutral voice, even when raging inside ...
- **Avoid Voice Strain**: Find a comfortable way to speak A LOT!

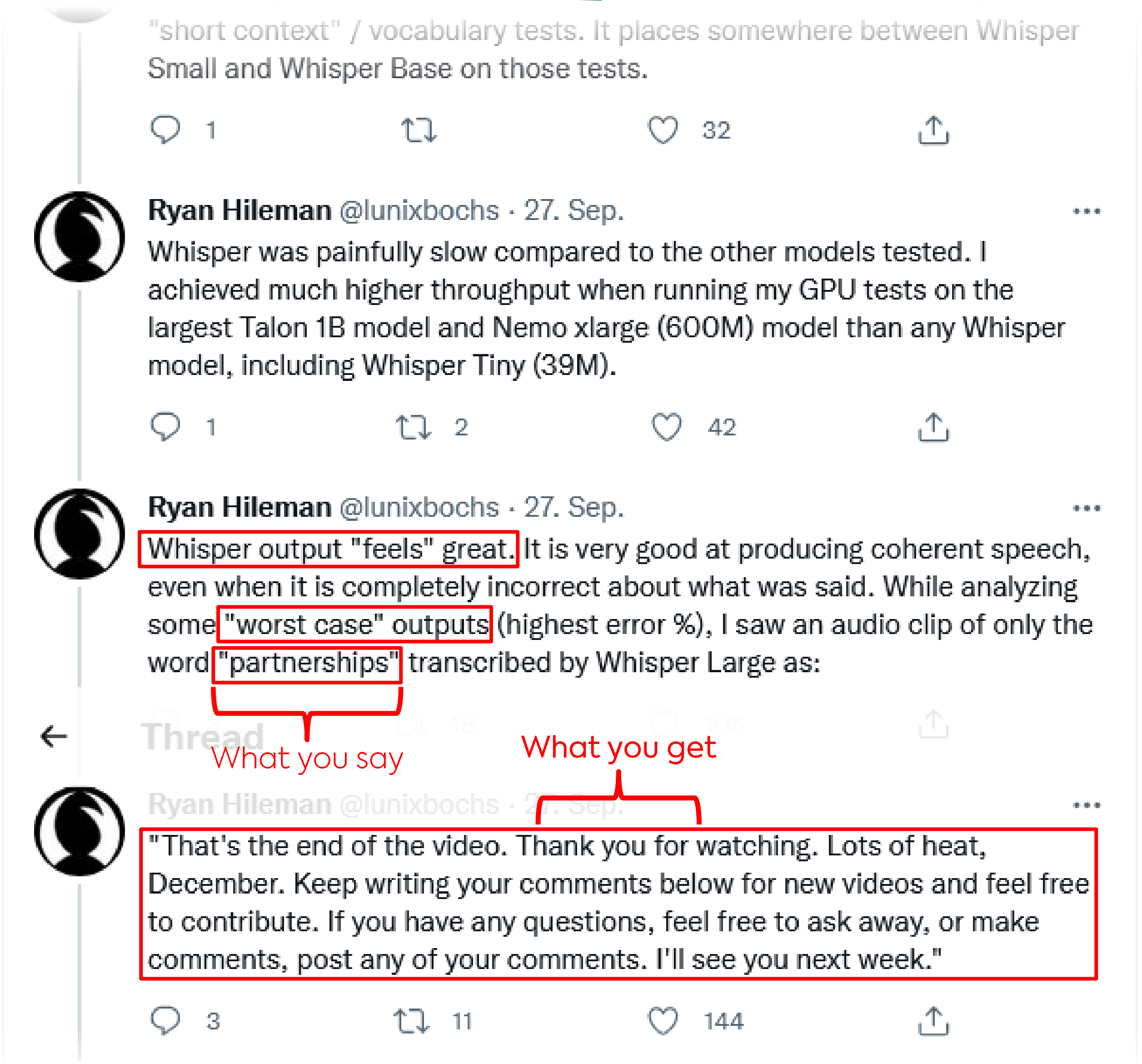
  → e.g. use your natural voice & drink a lot of tea
- **Command chaining**: Anticipate what is going to happen!

  → Practice, practice, practice!

General Issues

- **Multilanguage support** is still in ist infancy
  - → Non-English language models all have their problems
  - → Designing command libraries for different languages means effort
- **Complex setup** with many moving parts:
  - → Random stuff sometimes just happens, get used to it!
  - → Fallback to manual input sometimes necessary …
- **MACHINE LEARNING!!!**
  - → Models often reflect typical issues (data bias, data quality issues, …)
  - → Sometimes you have to just hope for the best …

Model Issues



"short context" / vocabulary tests. It places somewhere between Whisper Small and Whisper Base on those tests.

1 32

Ryan Hileman @lunixbochs · 27. Sep.

Whisper was painfully slow compared to the other models tested. I achieved much higher throughput when running my GPU tests on the largest Talon 1B model and Nemo xlarge (600M) model than any Whisper model, including Whisper Tiny (39M).

1 2 42

Ryan Hileman @lunixbochs · 27. Sep.

Whisper output "feels" great. It is very good at producing coherent speech, even when it is completely incorrect about what was said. While analyzing some "worst case" outputs (highest error %), I saw an audio clip of only the word "partnerships" transcribed by Whisper Large as:

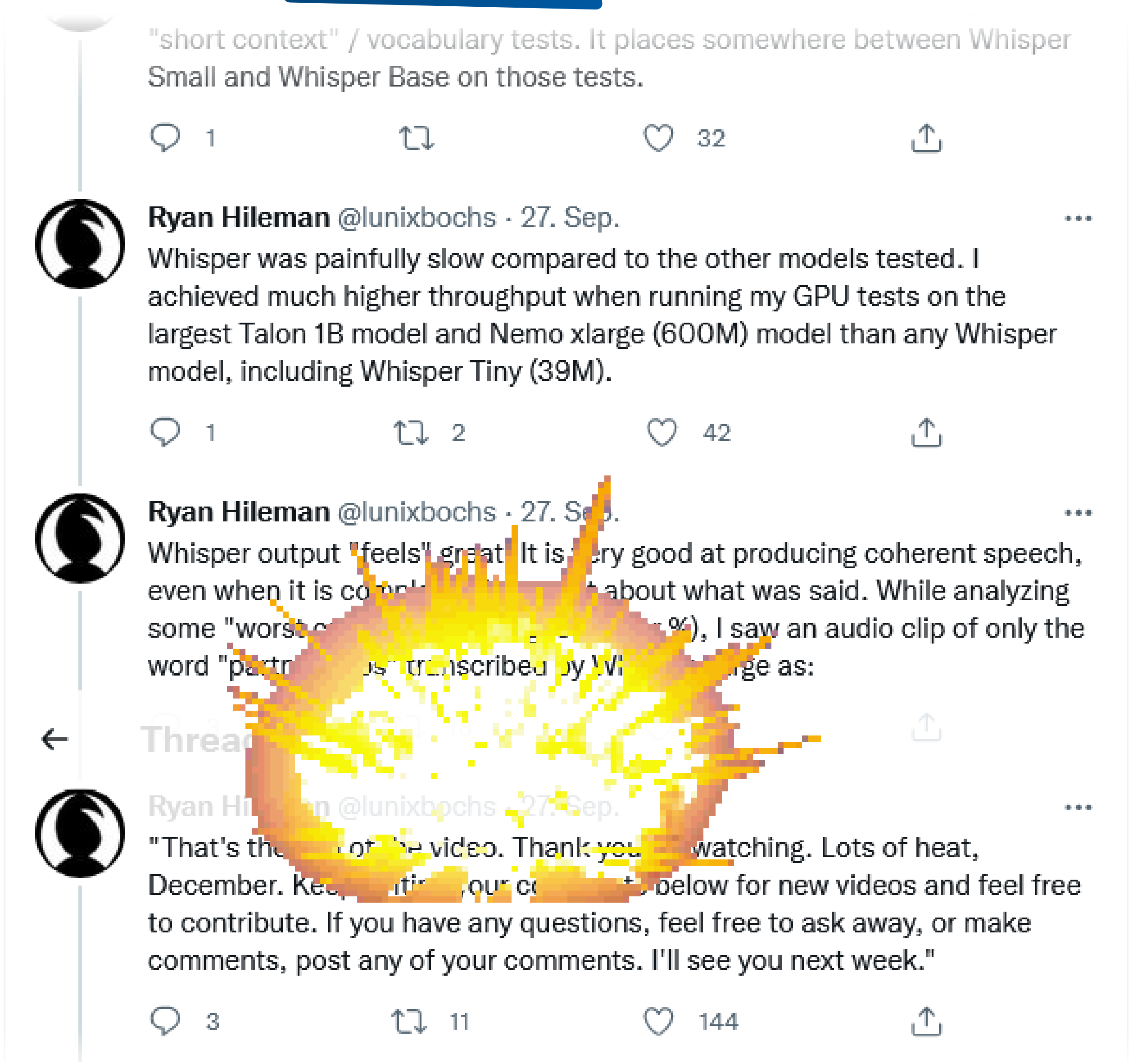
Thread

Ryan Hileman @lunixbochs · 27. Sep.

"That's the end of the video. Thank you for watching. Lots of heat, December. Keep writing your comments below for new videos and feel free to contribute. If you have any questions, feel free to ask away, or make comments, post any of your comments. I'll see you next week."

3 11 144

Model Issues



"short context" / vocabulary tests. It places somewhere between Whisper Small and Whisper Base on those tests.

1 32

Ryan Hileman @lunixbochs · 27. Sep.

Whisper was painfully slow compared to the other models tested. I achieved much higher throughput when running my GPU tests on the largest Talon 1B model and Nemo xlarge (600M) model than any Whisper model, including Whisper Tiny (39M).

1 2 42

Ryan Hileman @lunixbochs · 27. Se[illegible].

Whisper output "feels" great! It is [illegible]ry good at producing coherent speech, even when it is co[illegible] about what was said. While analyzing some "wors[illegible] %), I saw an audio clip of only the word "pa[illegible] transcribed by W[illegible]ge as:

Threa[illegible]

Ryan Hi[illegible]n @lunixbochs · 27. Sep.

"That's th[illegible] video. Thank you [illegible] watching. Lots of heat, December. Ke[illegible] below for new videos and feel free to contribute. If you have any questions, feel free to ask away, or make comments, post any of your comments. I'll see you next week."

3 11 144

Why This is Still Worth All the Hassle

Productivity
- Speed up input-heavy tasks
- Faster navigation through easy-to-remember shortcuts

Convenience
- Intuitive interfaces
- Relieve your hands

Accessibility
Compensate handicaps:
- Injuries (e.g. broken hand)
- Repetitive stress injury (RSI)
- Cubital Tunnel Syndrome
- …

General Awesomeness
- Talk to your computer!!!

Helpful Resources & Outlook

Tooling Recommendations (Incomplete!)

- **Talon** (Free of Charge): talonvoice.com / talon.wiki
  - Voice coding for Win / Linux / Mac!
  - Starter Grammar (English): github.com/knausj85/knausj_talon
- parrot.py (noise control): github.com/chaosparrot/parrot.py
- Cursorless (code editing for VSCode): github.com/cursorless-dev
- Rango (handsfree browsing): github.com/david-tejada/rango
- Paid Upgrades:
  - Talon Premium Support: patreon.com/join/lunixbochs
  - Dragon Speech Recognition: nuance.com/dragon/

Alternatives: Speech Recognition

- Speech Recognition
  - WSR (Windows Speech Recognition): Built into Windows
  - Kaldi: github.com/kaldi-asr/kaldi
  - Vosk (ASR on mobile devices!): github.com/alphacep/vosk-api
  - Web Speech API (compatible with Talon through Chrome or Firefox)
- Scripting:
  - NatLink: sourceforge.net/p/natlink/
  - Dragonfly: github.com/dictation-toolbox/dragonfly
  - Caster: github.com/dictation-toolbox/Caster
  - Vocola (Voice Command Language): vocola.net

Recommended Talks

whois emily

- Software Engineer
- GitHub: @2shea
- Twitter: @yomilly
- I write code for Fastly

2 / 45

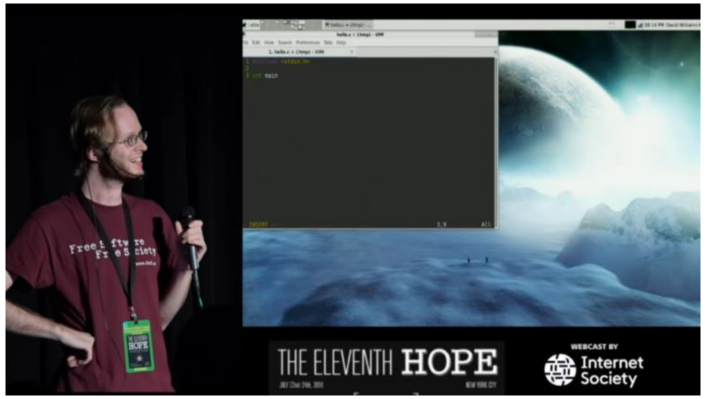
Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

Dragonfly

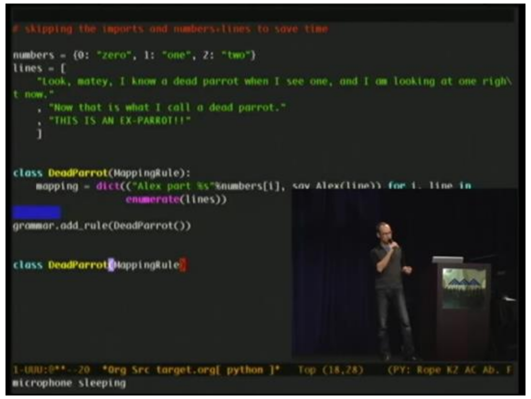
Core Features

- Language Object Model
- Support for multiple speech recognition
  Default supports DNS and WSR
- Built-in action framework
  Keystrokes, text input

Boudewijn Aasman. Coding by Voice with Dragonfly, PyGotham (2018)



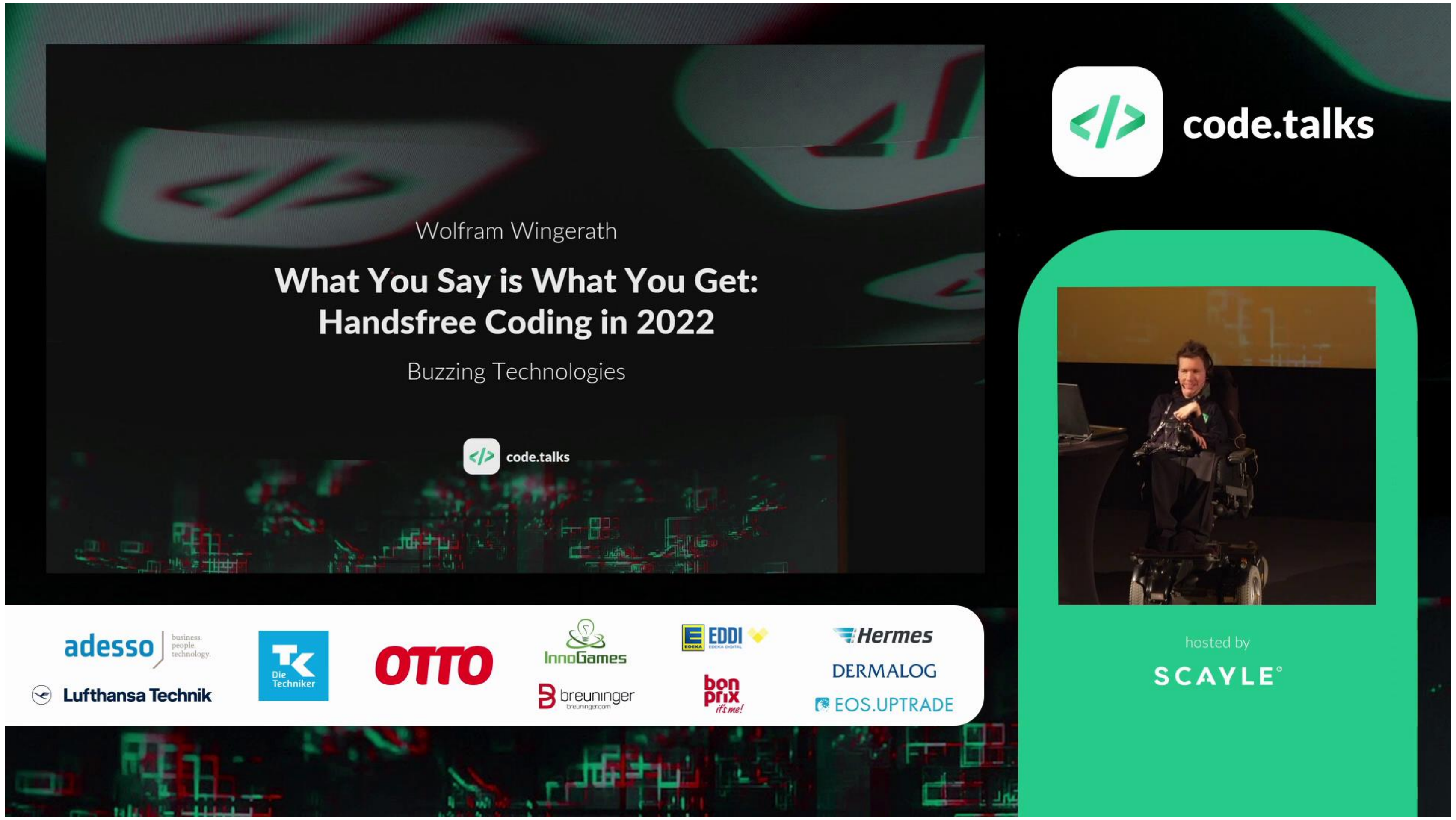
David Williams-King. Coding by Voice with Open Source Speech Recognition, The Eleventh Hope (2016)



```
# skipping the imports and numbers+lines to save time

numbers = {0: "zero", 1: "one", 2: "two"}
lines = [
    "Look, matey, I know a dead parrot when I see one, and I am looking at one righ\
t now."
    , "Now that is what I call a dead parrot."
    , "THIS IS AN EX-PARROT!!"
    ]


class DeadParrot(MappingRule):
    mapping = dict(("Alex part %s"%numbers[i], say Alex(line)) for i, line in
                   enumerate(lines))

grammar.add_rule(DeadParrot())


class DeadParrot(MappingRule)

1-UUU:@**--20  *Org Src target.org[ python ]*   Top (18,28)    (PY: Rope K2 AC Ab. F
microphone sleeping
```

Tavis Rudd. Using Python to Code by Voice, PyCon US (2013)

CodeTalks Video

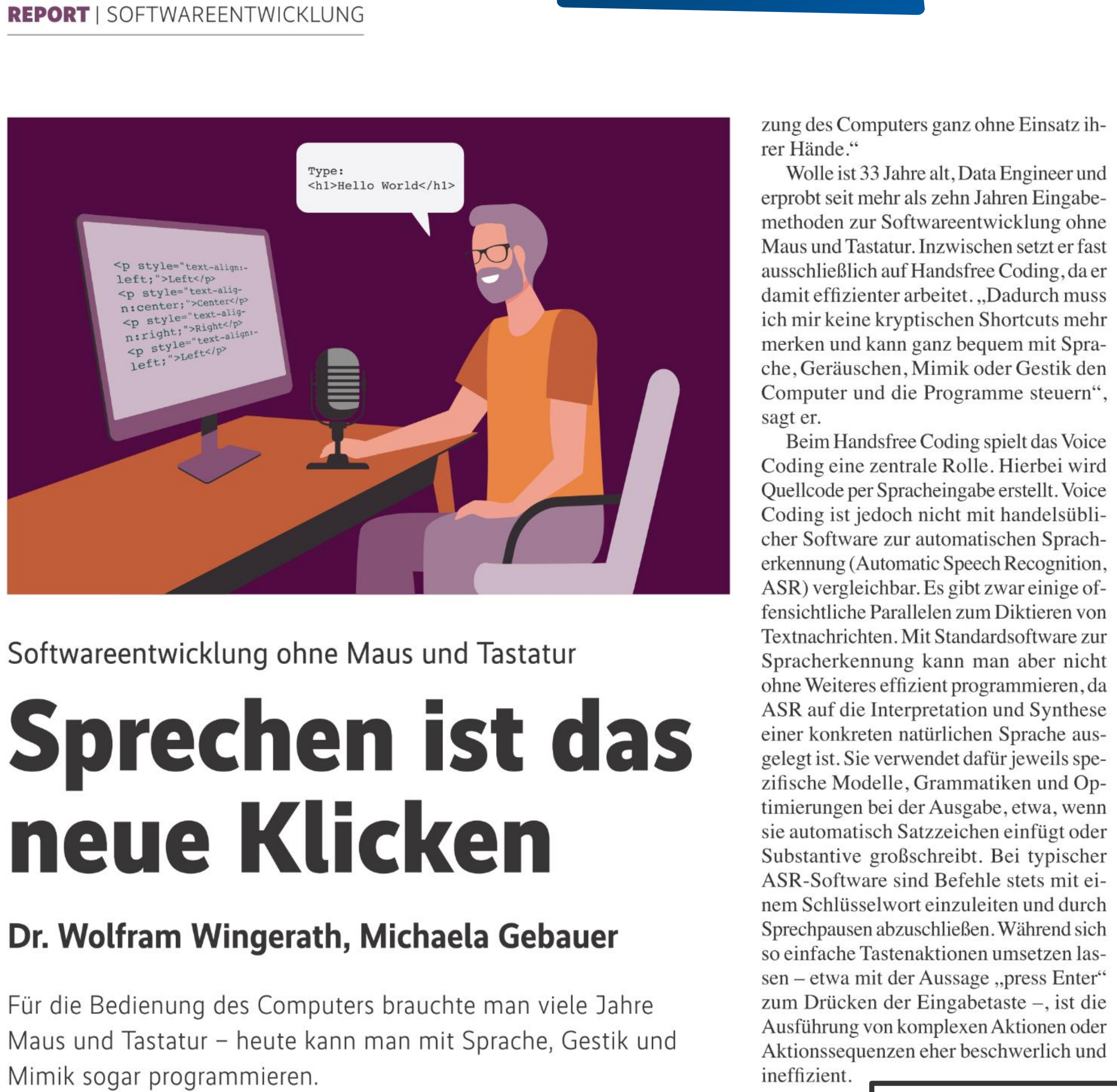


Wolfram Wingerath. What You Say is What You Get: Hands-Free Coding in 2022, CodeTalks (2022)

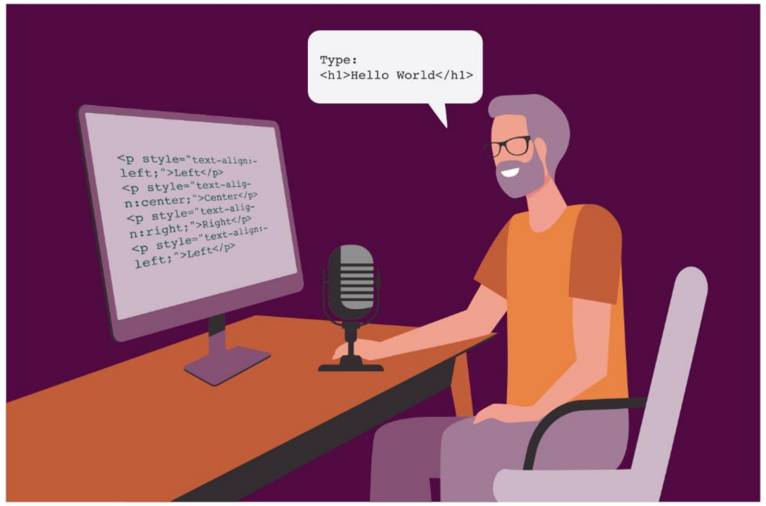
Articles & Blogs

- Emily Shea: whalequench.club/
  - Talon user
  - Very good starter instructions
- James Stout: handsfreecoding.org/
  - Dragonfly user
  - Huge collection of relevant blog posts
- Josh W. Comeau (2020): joshwcomeau.com/blog/hands-free-coding/
- Dusty Phillips (2020): dusty.phillips.codes/2020/02/15/on-voice-coding/
- Max Gravenstein (2018): medium.com/hubabl/handsfree-fe70980f36b/

Heise Article



REPORT | SOFTWAREENTWICKLUNG



```
<p style="text-align:-
left;">Left</p>
<p style="text-alig-
n:center;">Center</p>
<p style="text-alig-
n:right;">Right</p>
<p style="text-align:-
left;">Left</p>
```

Softwareentwicklung ohne Maus und Tastatur

Sprechen ist das neue Klicken

Dr. Wolfram Wingerath, Michaela Gebauer

Für die Bedienung des Computers brauchte man viele Jahre Maus und Tastatur – heute kann man mit Sprache, Gestik und Mimik sogar programmieren.

zung des Computers ganz ohne Einsatz ihrer Hände.“

Wolle ist 33 Jahre alt, Data Engineer und erprobt seit mehr als zehn Jahren Eingabemethoden zur Softwareentwicklung ohne Maus und Tastatur. Inzwischen setzt er fast ausschließlich auf Handsfree Coding, da er damit effizienter arbeitet. „Dadurch muss ich mir keine kryptischen Shortcuts mehr merken und kann ganz bequem mit Sprache, Geräuschen, Mimik oder Gestik den Computer und die Programme steuern“, sagt er.

Beim Handsfree Coding spielt das Voice Coding eine zentrale Rolle. Hierbei wird Quellcode per Spracheingabe erstellt. Voice Coding ist jedoch nicht mit handelsüblicher Software zur automatischen Spracherkennung (Automatic Speech Recognition, ASR) vergleichbar. Es gibt zwar einige offensichtliche Parallelen zum Diktieren von Textnachrichten. Mit Standardsoftware zur Spracherkennung kann man aber nicht ohne Weiteres effizient programmieren, da ASR auf die Interpretation und Synthese einer konkreten natürlichen Sprache ausgelegt ist. Sie verwendet dafür jeweils spezifische Modelle, Grammatiken und Optimierungen bei der Ausgabe, etwa, wenn sie automatisch Satzzeichen einfügt oder Substantive großschreibt. Bei typischer ASR-Software sind Befehle stets mit einem Schlüsselwort einzuleiten und durch Sprechpausen abzuschließen. Während sich so einfache Tastenaktionen umsetzen lassen – etwa mit der Aussage „press Enter“ zum Drücken der Eingabetaste –, ist die Ausführung von komplexen Aktionen oder Aktionssequenzen eher beschwerlich und ineffizient.

Wolfram Wingerath, Michaela Gebauer: Sprechen ist das neue Klicken, iX 9/2021 (https://wingerath.cloud/2021/ix)

Closing Recommendations

- **Keep it simple**: Prioritize ease-of-use over efficiency at the start (in particular: get used to an existing grammar before optimizing it)

- **Keep it reasonable**: Try to find use cases that make sense <u>for you</u> (e.g.: I‘m not giving this talk handsfree, since I can use my index finger)

- **Keep it in mind**: Handsfree coding might save you one day (revisit this talk when you struggle with RSI, broken hand, etc.)

Thanks! So What Now?

Slack talonvoice.slack.com	✓	Ask questions! Enjoy the community!
GI Initiative handsfree-coding.gi.de	✓	Try out handsfree coding!
Patreon patreon.com/lunixbochs	✓	Support Talon Development!

Subscribe to the mailing list!

Videos & Slides Available at https://wolle.science

Wolfram „Wolle“ Wingerath wolle@uol.de