Wolfram Wingerath

What You Say is What You Get: Handsfree Coding in 2022

Buzzing Technologies

code.talks

What You Say is What You Get

Handsfree Coding in 2022

code.talks 2022
Sep. 16, 2022
Wolle

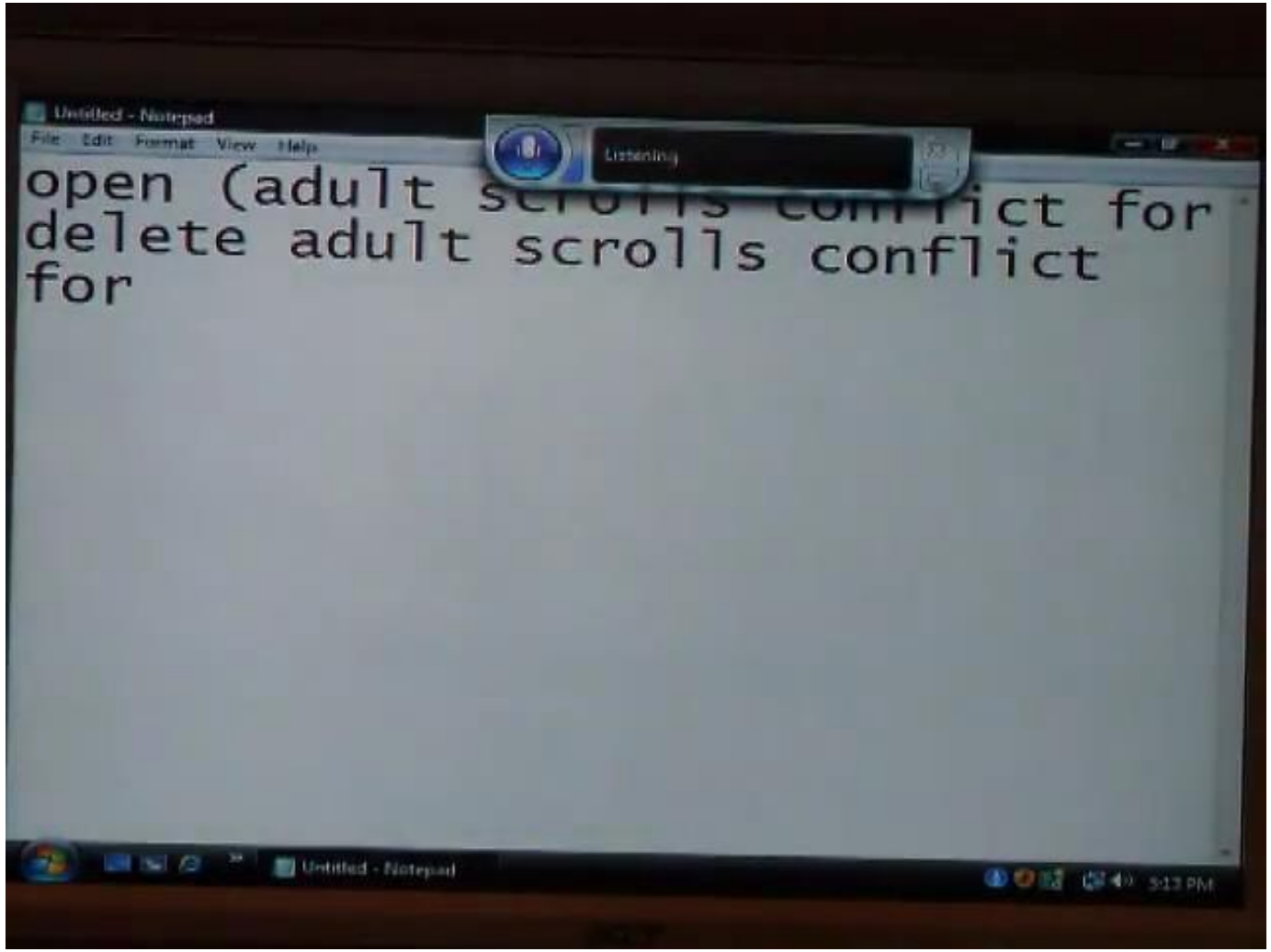
Carl von Ossietzky
Universität
Oldenburg

It's Simple, Really!

The requirements:

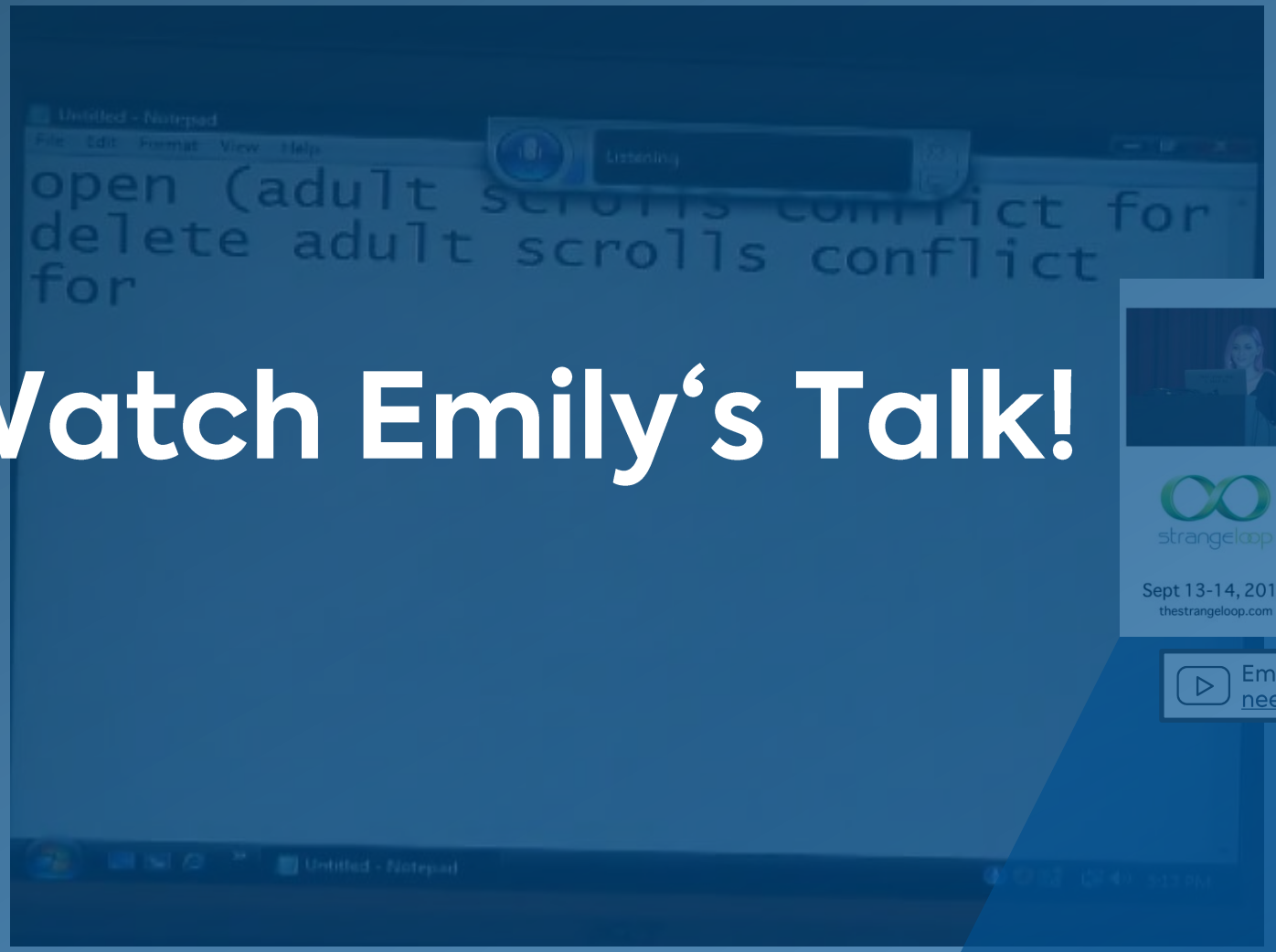
- **Microphone**: Every notebook has one!
- **Speech Recognition Software (SR)**: Included in Windows since 2007!
- **Voice Command Execution**: Available in every SR software!

Let Me Just Show You How Easy It is



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

Let Me Just Show You How Easy It is



Go Watch Emily's Talk!

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?

WSR, Dragon, ...

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

  → Default: Every utterance is interpreted as (spoken) text

  (Commands only through special keywords)

Dragonfly, Talon, ...

- **Voice Coding**: optimized for actions & programming languages

  → Default: Everything is interpreted as a command

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

Where‘s the Challenge?

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

I'm data engineer,
not an ASR or HCI expert!

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

- **Actions & Symbols:**
  - Numbers, brackets, etc.: `one space paren bang slash` → `1_(!/`

    What you say / What you get
  - Modifiers, e.g.: `shift touch` or `hold alt` or `release control`
- **Spelling** through a phonetic alphabet:
  - NATO alphabet: `alpha bravo charlie delta echo` → `abcde`
  - Optimized alphabet: `air bat cam drum each` → `abcde`
- **Navigation**, e.g.:
  - Cursor movement: `go left` or `go way up` or `home`
  - Selection: `mark left` or `mark way up` or `mark all`

Basic Voice Control

- **Typing**:
  - Dictation: `say hey comma you` → `hey, you`
  - Type recognized text: `phrase hey comma you` → `hey comma you`
- **Formatters**, e.g.:
  - Camel case: `camel hey you` → `heyYou`
  - Dashed: `kebab hey you` → `hey-you`
  - Dotted: `dotted hey you` → `hey.you`
  - Without whitspace: `smash hey you` → `heyyou`
  - Uppercase: `uppercase hey you` → `HEY YOU`

Basic Voice Control

- **Command History & Help:**
  - Manage history: `command history` or `command history clear`
  - Show & hide help: `help active` or `help close`
- **Application management**, e.g.:
  - Launch & focus: `launch Firefox` or `focus Firefox`
- **Command Management** for efficiency, e.g.:
  - Repetition: `one third` → `111`
  - Chaining: `paren close paren` `go left` `say hi` → `(hi)`

1.) type: `()` 2.) move cursor: ⇦ 3.) dictate: `hi`

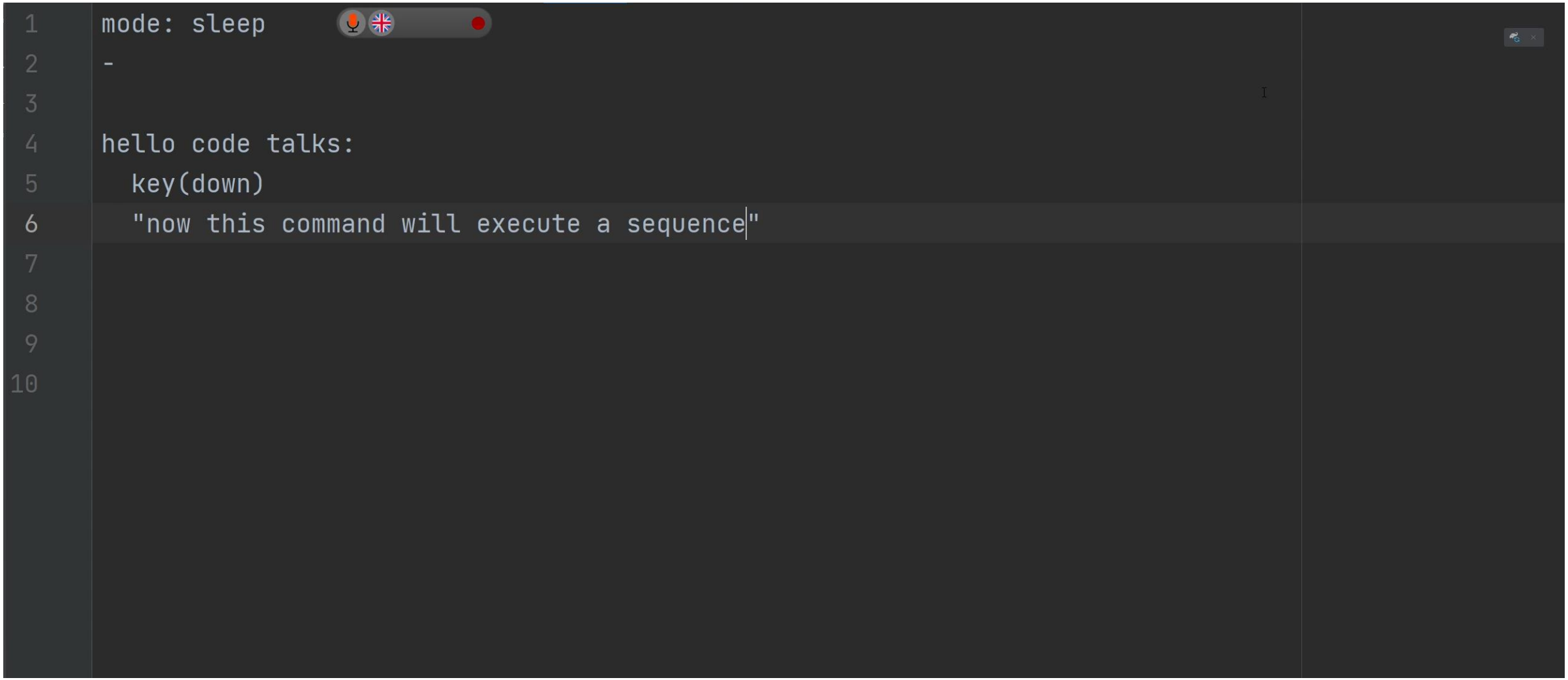
Dynamic Scripting

- **On-the-fly prototyping** for your grammar!
  - Python for live reloading
  - Voice coding inception: Work in the grammar with the grammar
- **Customization** of format and spelling:
  - Example: `etcetera → etc.`
- **Settings**, delays, speech engine, etc.

```
settings():
    key_hold = 150.0
    key_wait = 20.0
    speech.engine = 'dragon'
```


Live Demo

Dynamic Scripting



Handsfree Coding

- **Different behavior** for different semantics, 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;
  cursor: pointer;
```

Josh W. Comeau. Hands-free Coding, Josh W. Comeau's Blog (2020)

Handsfree Coding: 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

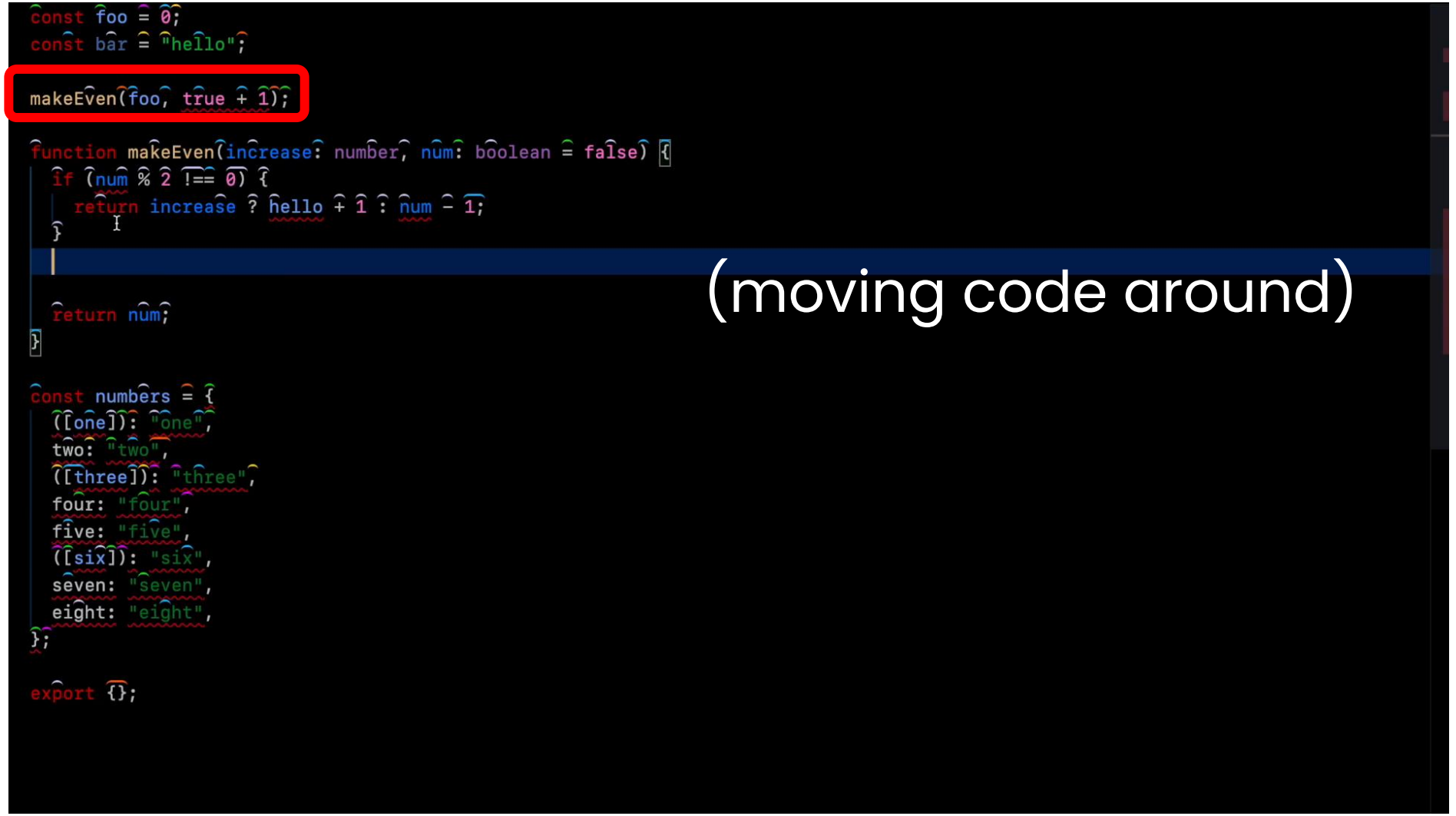
- Example:

```
chuck  funk  red
```

action scope mark

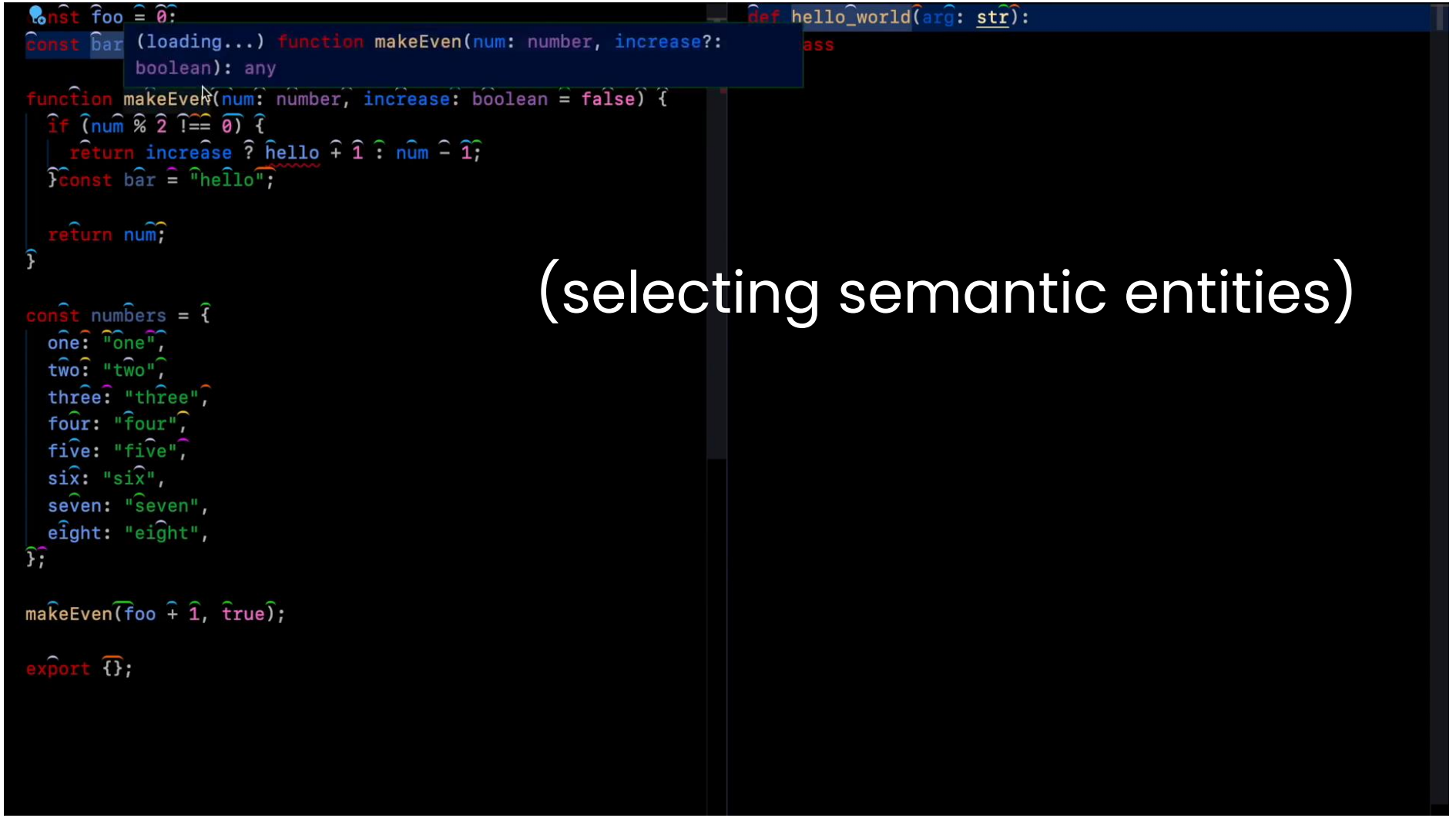
(delete the function marked with „r“)

Handsfree Coding: Cursorless



Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Handsfree Coding: Cursorless



(selecting semantic entities)

Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

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

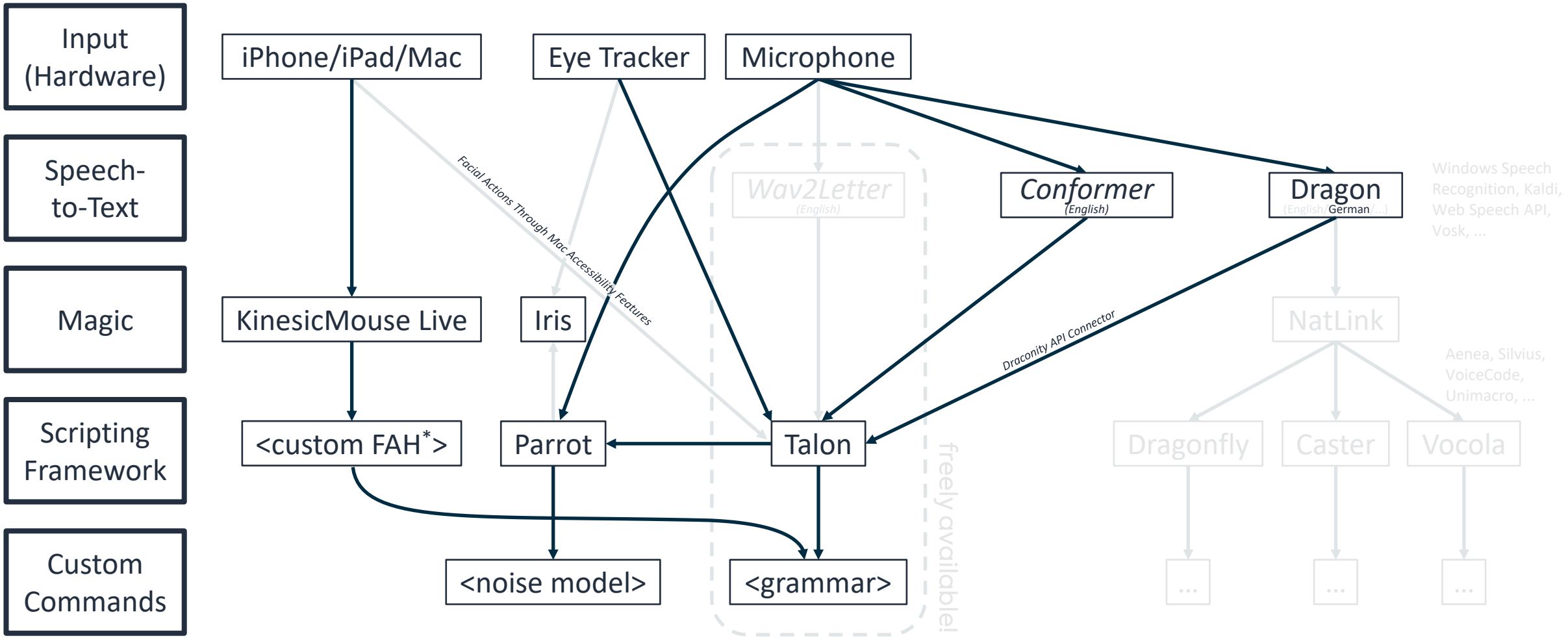
Custom noises for your Talon grammar!

Handsfree Gaming: Eyes + Noise + Face

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

Multi-Computer Setup


Pitfalls &
Challenges

Recognition Accuracy

- **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

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
- **Eye tracker stability** is an issue as it disconnects sporadically
  - → You will have to restart Talon from time to time …
- **Random crashes** are rare, but do occure from time to time:
  - → Fallback to manual input sometimes necessary …

Workflow & Anger Management

- **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!

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 ;-)

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

Recommended Tooling & Documentation

- **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
- 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

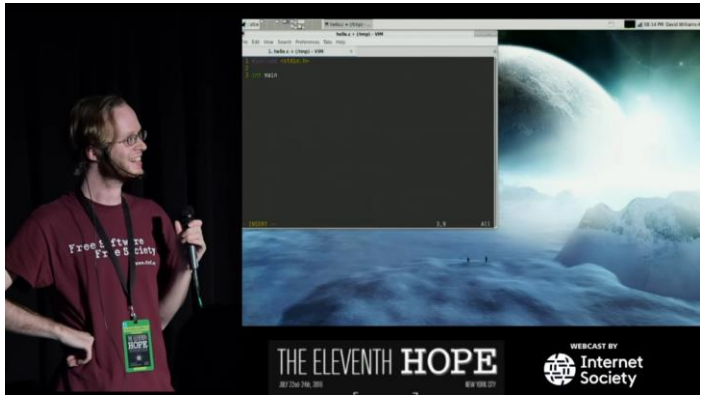
Articles & Blogs

- 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/

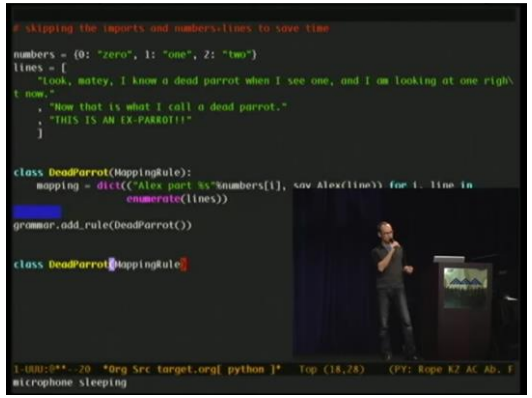
Recommended Talks

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

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



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



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

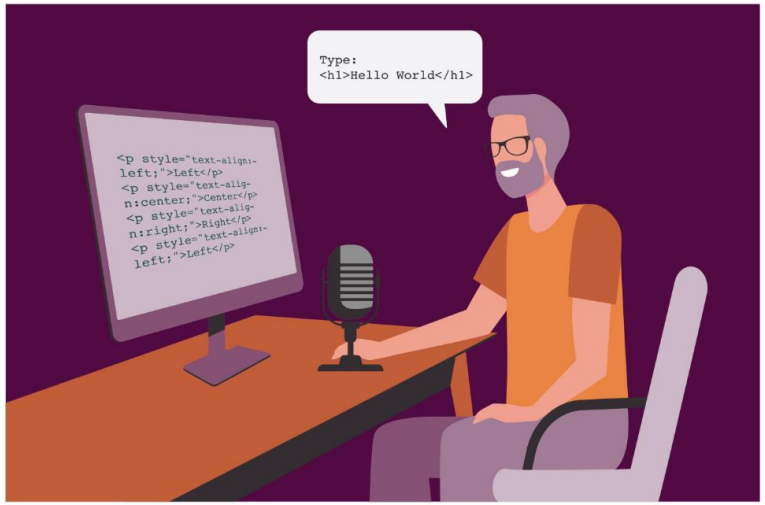
Closing Recommendations

- **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 for you
  (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.)

Read the Article!



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)

Thanks! So What Now?

Slack
talonvoice.slack.com

Ask questions!
Enjoy the community!

Subscribe to the mailing list!

GI Initiative
handsfree-coding.gi.de

Try out handsfree coding!

Patreon
patreon.com/lunixbochs

Support Talon Development!

Wolfram Wingerath wolle@uol.de

Carl von Ossietzky
Universität
Oldenburg