CONTENT AUTOMATION · YOUTUBE

# Automated YouTube video production

A pipeline that takes a topic to a finished, narrated, animated video. Research, script, voice, animation, mix and render are each a defined stage with its own tooling and its own pass criteria, so one protocol produces every episode.




| 9 | 75 min | 2 | 99.2% |
|---|---|---|---|
| RENDERED MASTERS | FINISHED 1080P | SERIES FORMATS | SCRIPT TO AUDIO ALIGN |

Two formats run on the same machinery. **Tomorrow Sucks** is a stick-figure documentary series about near-future scenarios, where every absurd detail is anchored to a sourced present-day fact. **Founder Stories** uses the same drawing system with archival photography composited onto the page.

**Why it is built this way**

The slow parts of an animated explainer are syncing art to narration and re-syncing it every time the voice changes. Binding every cue to a sentence rather than a timestamp removes that work, so a script revision costs a re-run instead of a re-edit.

**What a new episode costs**

Research and script are the human hours. Voice, alignment, build, proof, mix and render are scripted steps that run the same way every time, which is what makes a weekly cadence possible for one person.

THE PIPELINE

# Eight stages, topic to master

Each stage writes a file the next stage reads. Nothing is passed by hand, so any stage can be re-run without redoing the ones before it.

**01** **Research**

Every factual claim is collected into a receipts ledger with a source URL, exact date and a confidence flag. Claims that are widely misreported get recorded as myths to avoid, so the same error cannot re-enter a later script.

`research-v3.md`
`per-claim provenance`

**02** **Script**

Written to a fixed house protocol: second person, a cold open that states the premise, and alternating fiction and sourced segments. Output is a single markdown file with explicit section headers that later stages parse.

`script-final.md`
`sectioned markdown`

**03** **Voice**

The script is split into chunks under the synthesis limit, numbers spelled out, and each take is generated separately. Every take is transcribed back and checked against its source text for dropped tails and duplicated phrases before it is accepted.

`ElevenLabs`
`whisper verification`

**04** **Alignment**

Takes are stitched with controlled gaps, then transcribed with word-level timestamps. Script sentences are matched onto that word stream with fuzzy matching and number canonicalisation, producing a start and end time for every sentence.

`mlx_whisper`
`words.json → captimes.json`

**05** **Cue authoring**

Visual and sound cues are written in Python against *sentence index* rather than seconds. A re-recorded take shifts every cue automatically instead of requiring a manual re-sync.

`build.py`
`122 windows, 185 cues`

**06** **Animatic**

A single SVG stage with a reusable rig library. Characters, props and printed cards are drawn in code, so a fix to a rig propagates to every episode that uses it. Sound effects are synthesised rather than sourced.

`SVG stage + engine`
`WebAudio synthesis`

**07** **Quality control**

The animatic is sampled at fixed intervals into still frames and tiled into contact sheets, so empty frames, overlaps and mistimed cues are caught before rendering rather than after.

`spot renders`
`76 checkpoints`

**08** **Mix and render**

Voice, synthesised effects and scored music beds are mixed to a single track. Frames are captured by stepping virtual time deterministically, then encoded to a 1080p master.

`headless capture`
`ffmpeg h.264`

THE SETUP

## Everything is a file, every step is a command

An episode is a folder. Each stage reads the previous stage's output and writes its own, so the state of a build is always inspectable and any step can be re-run on its own.

ONE EPISODE ON DISK

```
ep07/
├─ research.md        sourced receipts ledger
├─ script-final.md    sectioned shooting script
├─ tts/               chunked text for synthesis
├─ vo/                verified takes + stitched master
└─ anim/
   ├─ align.py        audio → words → sentence times
   ├─ words.json      1489 word timestamps
   ├─ captimes.json   124 sentence windows
   ├─ build.py        cue table → animatic
   ├─ t_head.html     rig + prop definitions
   ├─ t_engine.html   cue engine + sfx synthesis
   ├─ score.py        music beds → mix
   ├─ spot/           76 proof frames
   └─ contact_*.png   tiled review sheets
```

ALIGNMENT, ONE COMMAND

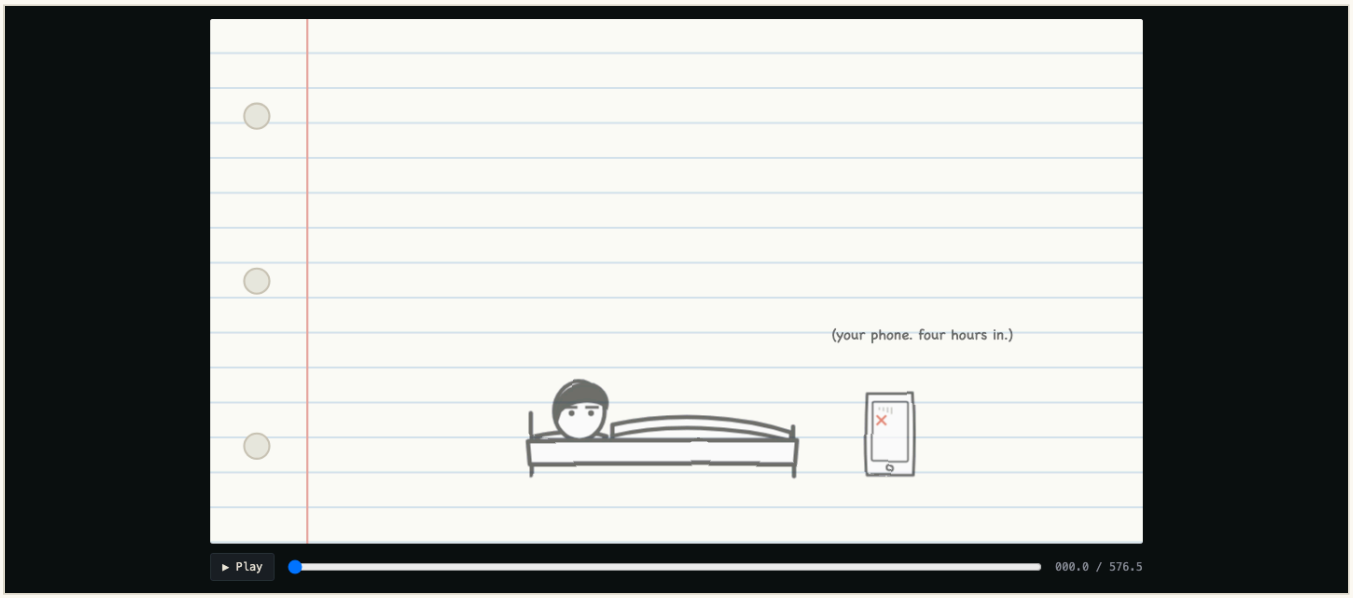
```
$ python3 anim/align.py
words 1489 | sentences 124 | aligned 123/124 = 99.2%
last word ends 573.34s

  coldopen  S  0–S  9     0.00s ->   32.00s
  s1        S 10–S 31    32.84s ->  134.12s
  s2        S 32–S 43   134.12s ->  206.56s
  s3        S 44–S 75   206.56s ->  359.16s
  s4        S 76–S 93   360.90s ->  461.04s
  perks     S 94–S110   461.72s ->  517.72s
  close     S111–S123   519.02s ->  573.34s

$ python3 anim/build.py
ok 122 windows · 185 sfx cues · 0 dead frames
ok wrote ep07_animatic.html
```

### The review harness

The built animatic opens in a browser with a scrub bar and a timecode. Cue timing, page composition and sound placement are checked here, against the real voice track, before a single frame is rendered.




*The stage at 000.0 of 576.5. Scrubbing re-evaluates the whole cue table at that timestamp, so any moment in the episode can be inspected instantly.*

ANIMATION ENGINE

# The timeline is written, not dragged

There is no editing timeline. The whole episode is a declarative cue table compiled to a single HTML stage. `on()` turns an element on between two sentences. `fx()` fires a sound at an offset from a sentence. `stamp()` runs the full present-day interruption ritual as one call.

```
# cues reference sentence numbers, so a re-recorded take re-times itself
def on(el, a, b, cls='on'):
    W.append([el, cls, S(a), S(b)])          # S(i) = start time of sentence i

def stamp(i, card, to):
    """PRESENT DAY slam at sentence i, badge + card until sentence `to`."""
    on('pdstamp', S(i), S(i)+1.6); fx(i, 'rewind'); fx(i, 'stamp', .45)
    on('pd_corner', S(i)+1.6, S(to)); on(card, S(i)+1.7, S(to))

# ===== COLD OPEN =====
on('co_wake', 0, 4);      fx(0, 'coldswell', .1); fx(0, 'softclick', 1.0)
on('co_pile', 6, 7);      fx(6, 'thud', .2); fx(6, 'thud', .5); fx(6, 'thud', .8)
on('co_clock', 7, 9);     fx(7, 'ping', .3)
```

**Drawn in code**

Figures, faces, props and printed cards are SVG definitions in a shared head file. Changing a character means editing one rig, not repainting frames.

**Sound is generated**

Every effect is synthesised from oscillators and filtered noise at build time. No library clearance, and each one is tuned to the beat it lands on.

## The layer stack

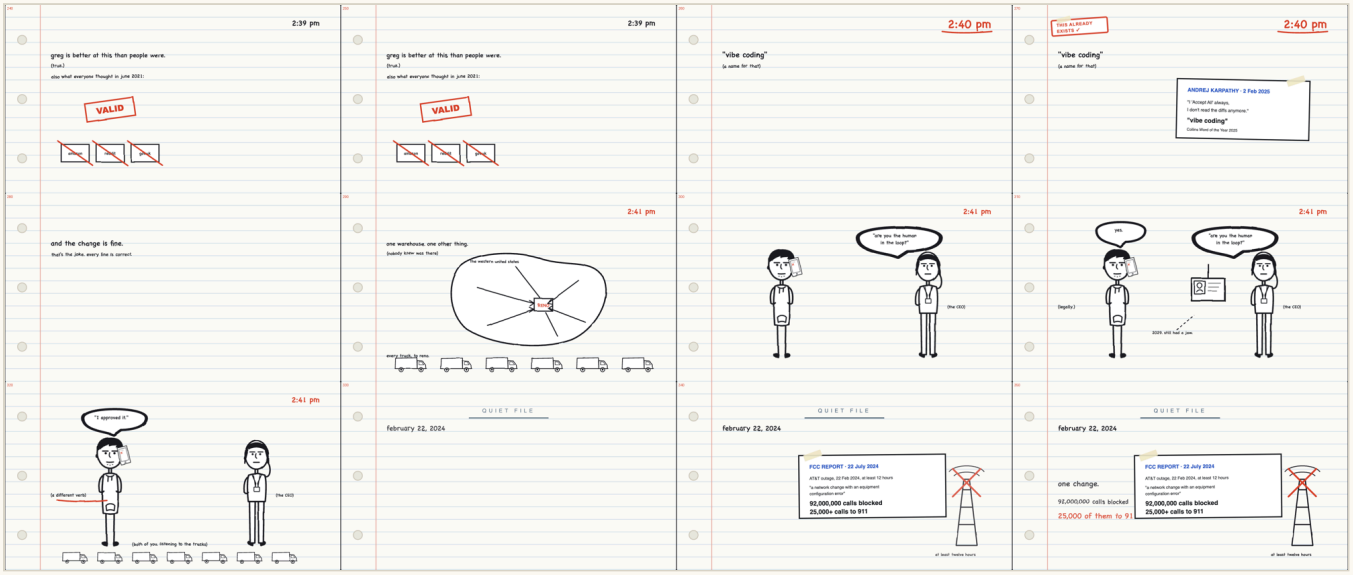
Order matters. The page never moves relative to the drawing, and the printed layer never inherits the wobble, which is what keeps system cards legible against a shaking hand-drawn world.

| | |
|---|---|
| **HUD & stamps** | outside the camera, never scales or shakes |
| **Printed layer** | system cards, taped receipts, crisp type, no boil |
| **Drawn world** | figures, props, handwriting, boil filter cycles here |
| **Notebook page** | rules, red margin, punch holes, fixed |
| **Camera group** | push, jolt and settle transforms applied to everything above |

QUALITY CONTROL

# Every episode is proofed before it renders

Rendering is the expensive step, so the animatic is checked first. The build is sampled into stills and tiled, which makes a dead frame or an overlapping element obvious at a glance.




*Contact sheet from one episode. Each tile is the stage at a fixed timestamp, read left to right, so cue timing and page composition can be verified without scrubbing the video.*

```
words 1489
sentences 124
aligned 123 / 124 = 99.2%
last word ends 573.34s

coldopen S 0–S 9 0.00s → 32.00s
s1 S 10–S 31 32.84s → 134.12s
s2 S 32–S 43 134.12s → 206.56s
close S111–S123 519.02s → 573.34s
```

*Alignment report. Any sentence that fails to match is listed by index and fixed by hand before cue authoring begins.*

## What the checks are looking for

**Dead frames**
A window closing before the next opens leaves the page blank mid-sentence.

**Covered punchlines**
A printed card above handwriting can hide the joke it was meant to support.

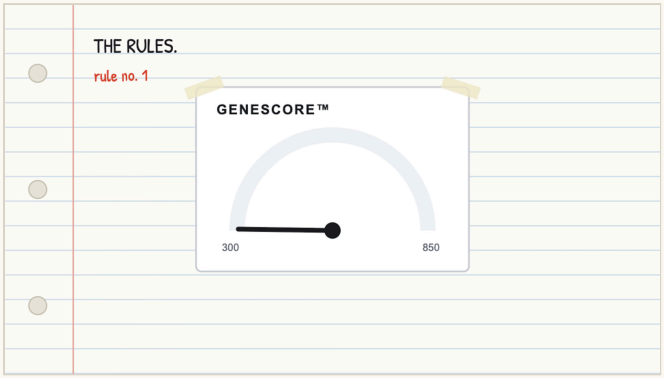
**Early cues**
A sound firing before the word it belongs to kills the beat it was timed for.

**Drift after a retake**
Re-recording a chunk shifts everything after it, so the fix is a re-run.

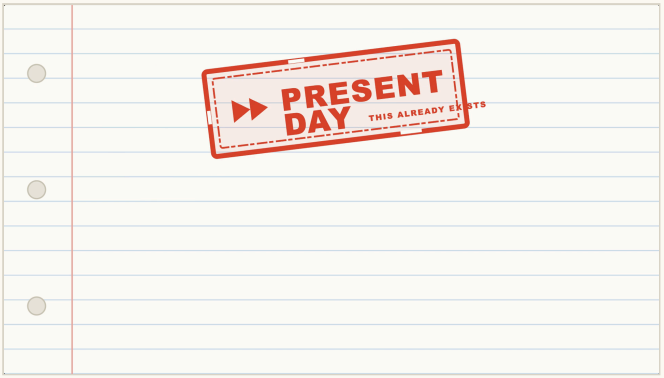
OUTPUT

# What comes out the other end

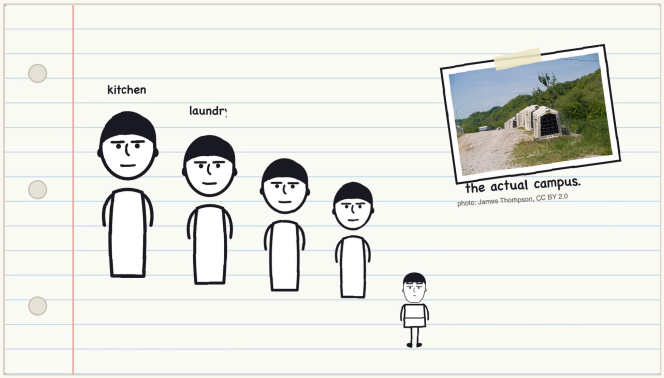
Finished masters at 1920 × 1080, 30fps, h.264 with a mixed AAC track, ready to upload.



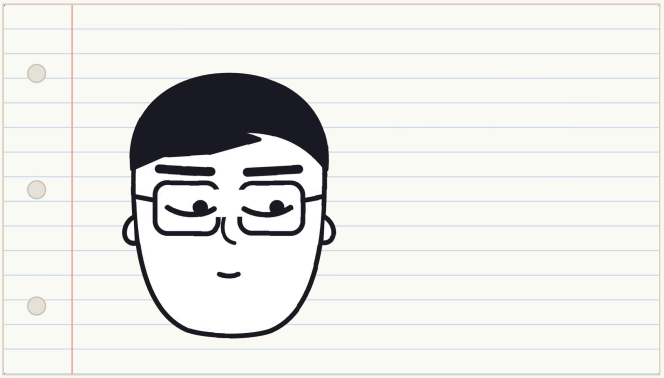

Tomorrow Sucks · cold open




Tomorrow Sucks · sourced segment

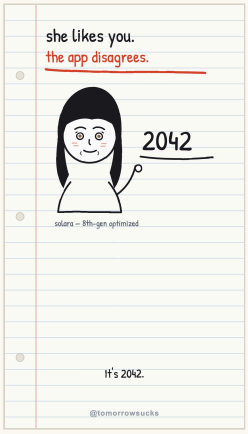



Founder Stories · archival composite

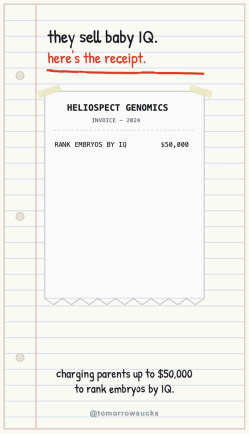


Founder Stories · character rig

**Master** 1920 × 1080, 30 fps, h.264, crf 18
**Audio** AAC 192k, voice + effects + beds
**Runtime** 8 to 17 min per episode
**Vertical** 1080 × 1920 cutdowns, same stage




Short · 9:16




Short · 9:16

REPEATABLE

The same protocol produces every episode, short and reel. Adding a new series means writing a new rig set and a new script format against the existing engine, not rebuilding the pipeline.