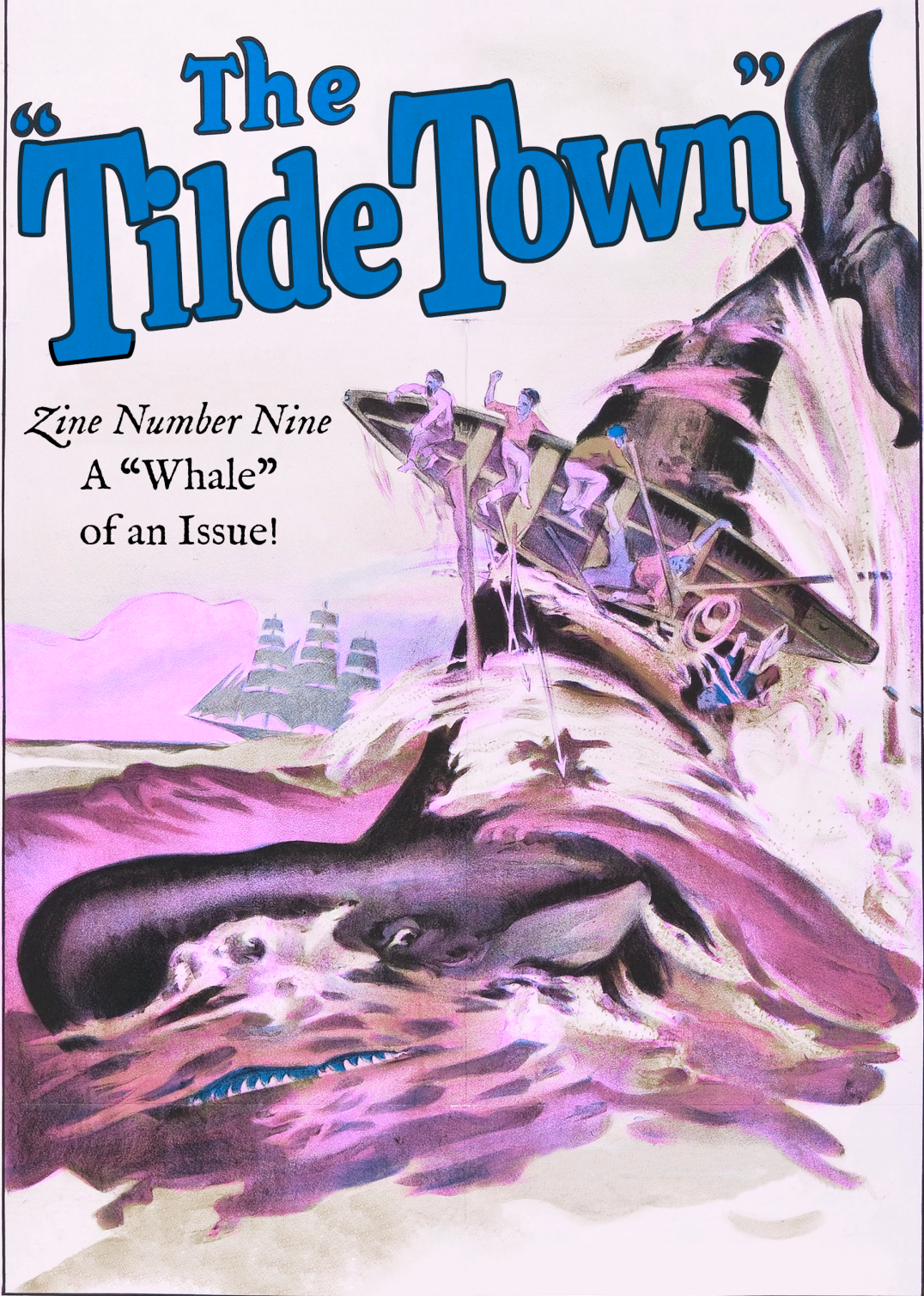


"The Tilde Town"

Zine Number Nine

A "Whale"
of an Issue!



TILDE TOWN ZINE issue #9

is © 2025, 2026 Tilde Town Zine contributors
see end of document for license information

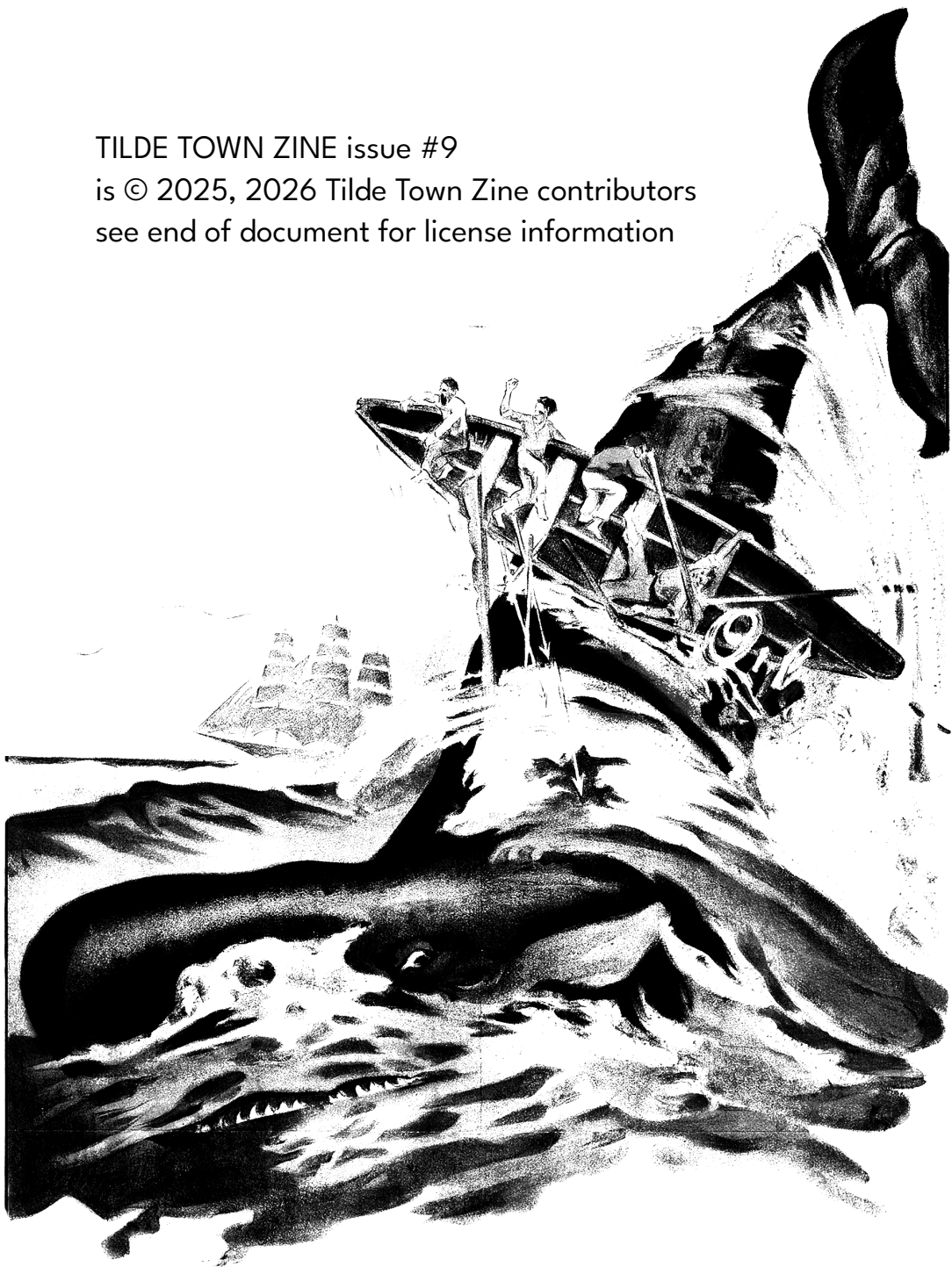


table of contents

4	Editorial ~ <i>agafnd</i>
5	new around town ~ <i>agafnd</i>
8	reflections on tilde.town's relocation ~ <i>vilmibm</i>
12	We Lean in to Listen ~ <i>woodsii</i>
16	Cookbook
17	Sourdough ~ <i>woodsii</i>
23	Bolognese ~ <i>nbsp</i>
24	Latkes ~ <i>nbsp</i>
26	Brunkager ~ <i>fruit</i>
33	Hot Coco ~ <i>grendel</i>
34	camera obscura ~ <i>fruit</i>
38	Writing my own markup language ~ <i>acdw</i>
47	north of Lietzow ~ <i>kls0e</i>
48	Portraits in Ableism ~ <i>hush</i>
50	Haikus ~ <i>kind</i>
53	Tilde Town Free Jazz Band: 2025 Edition ~ <i>agafnd</i>
56	Papillon ~ <i>mio</i>
57	Mail::RFC822::Address <i>Paul Warren</i>
60	ASCII Bookshelf ~ <i>oskar</i>
62	Crossword: Hashmap ~ <i>pilosophos</i>
63	A Monastic Look at Perl ~ <i>kirch</i>
68	TiddlyWiki on Tilde Town ~ <i>basslass</i>
70	only the forest can hear me cry ~ <i>m455</i>
71	Pius Kills the Music Industry ~ <i>piusbird</i>
80	Un soir d'hiver ~ <i>oskar</i>
82	color ~ <i>lateen</i>
84	Only Think This Now ~ <i>malvarma</i>
85	Ivy Blossom Remix ~ <i>piusbird</i>
86	dispatch from #oulipo ~ <i>vilmibm</i>
89	On Honesty ~ <i>placed99th</i>
90	2025 Chevrolet Silverado 1500 ~ <i>tunas</i>
92	drunken_chess.c ~ <i>turnipgod</i>
94	credits & licenses

From the editor

Hello everyone. I'm happy to present, after much delay, zine #9. I was not the original editor, but was allowed to step into the role, and quickly discovered how complicated putting this sort of thing together is! I have a newfound appreciation for past editors—as well as for the contributors (looking over everything as closely as I did will do that).

So, huge thank you to everyone who helped out and everyone who contributed something to the zine. There are some wonderful pieces in this issue. (It includes a small cookbook!)

Quite a lot has happened in ~town in the year and a half since the last issue. So I thought I would start this one off with a recap of the new things that came to mind. See you on the next page...

~agafnd, June 2026

new around town

Here are some new things around town since the last zine:

~nebula developed a new way to post. “bink is a serverless microblogger for the local terminal! think of it like a feed on twitter, but only for tilde.town users.” Run it with `town bink`.

~owl set up a town-adjacent forum at forum.sickos.net. Do you have something which makes you go “YES... HA HA HA... YES!”? Become a sicko today.



Quick links FAQ Recent Topics

Notifications Private messages **agafnd** ▼

Board index

Style: **sickos.net** ▼

SICKOS ONLINE: **agafnd**, owl

FORUM	TOPICS	POSTS	LAST POST
 general chat for a great low rate you can get online ♪ ♪ ♪ ♪	66	1340	 Re: (small) forum feedback 1 day ago by nebula 
 hobbies things you do in your spare time that you are a sicko about	34	407	Re: the David Lynch thread 9 hours ago by agafnd 
 projects what projects are you working on???? Subforum:  tilde30	13	84	 Re: shmpu ereader 3 weeks ago by vilmibm 
 game board not just for board games! just not for bored games	15	143	Re: Digimon / Vpet General 3 weeks ago by milo 
 FORUMQUEST roleplay for sickos	1	30	 Re: YOU MEET IN A TAVERN 3 months ago by agafnd 

STATISTICS

Total posts **2004** • Total topics **129** • Total members **103** • Our newest member **calljori**

Board index

 The team  Members  Delete cookies All times are UTC-04:00



Powered by phpBB® Forum Software © phpBB Limited
Privacy | Terms
Last visit was: 9 hours ago

~nbsp added a new feature to pinhook:

<@nbsp> it finds pairs of messages where you could put "everybody wants to rule the world" after them like the song

[...]

<owl> is it on like syllable count?

<@nbsp> and stress!

<@nbsp> look

<@nbsp> war and hate on planet venus

<@nbsp> suck my nuts and kill my penis

<pinhook> everybody wants to rule the world

<owl> oooo

<codesquirrel> amazing

<lateen> a true lyrical genius

<slimelia> you have eaten twenty babies

<slimelia> do you think that you should do this

<pinhook> everybody wants to rule the world

<placed99th> is there any way i can like

<placed99th> access newsgroups from a client

<pinhook> everybody wants to rule the world

<kirch> I should workshop that a bit more

<ytlept> hey does anyone use acme

<pinhook> everybody wants to rule the world

<@nbsp> have you heard the good word of bran

<@nbsp> it feels even more like horse food

<pinhook> everybody wants to rule the world

<lina> why does it say i have new mail

<lina> oh god did someone reply all

<pinhook> everybody wants to rule the world

<@nbsp> oh most of them have been boring
<@nbsp> some of them are really funny
<pinhook> everybody wants to rule the world

<@nbsp> once i get. a television
<@nbsp> did you know those are expensive
<pinhook> everybody wants to rule the world

<agafnd> tell me how could Jank be Stable
<arcu> life is full of contradictions
<pinhook> everybody wants to rule the world

<agafnd> gooey goo for chewy chewing
<agafnd> that's what that goo goose is doing
<pinhook> everybody wants to rule the world

~nbsp also revived fridaypostcard. Send your postcards to town with our/fridaypostcard on IRC or town my fridaypostcard in the terminal, and check out this Friday's postcards at tilde.town/square/fridaypostcard/.

A color extension to town botany, written by joyalicegu on GitHub, was merged. Plants are now in color!

```
botany

options
  1 - water ())))))))) 94%
  2 - look
  3 - garden
  4 - visit
  5 - instructions
  6 - exit

plant: young pansy
score: 313216
```



And perhaps most notably, ~town has moved to a new home on a physical colo rack.

reflections on tilde.town's relocation to a physical server ~vilmibm

Town was always somewhere. It's easy to hand wave the internet as ethereal when using "cloud" services, but every keystroke sent to tilde.town prior to May 2025 caused electrons to flow this or that way on *some* kind of computer somewhere in a corporate data center.

We could not know the hands that touched that computer. We can't know who installed it or how big it was. We can guess at its color or whether it had stickers on its chassis. Was it weather-worn or shiny new? Glossy or matte? Did the person who last cared for it have any affection for it? Wrath?

The internet is a vast organism with cells dying and reproducing every second as caretakers switch networking cards and flamed-out hard drives. The internet is divine: as vast and near to us as a mother's breast is to an infant but with a face unknowable. The internet is a vast machine as local as the phone in your pocket; to whom it is local depends on which doors your keycards can open.

Every religion has its temples; every temple its relics. Such places and items are attempts to capture, even if asymptotically, a shard of the internally felt divine. Whether feared or loved, humans want material artifacts pointing a way back into themselves in the hope of feeling a sacred glimpse.

Thus, the town moved. We left behind a Digital Ocean Virtual Machine and now exist on a little pink server built, installed, and run by ~insom. For users, little has changed in the year since the move. We do the same stuff. Type the same invocations and feels. See the same people. But the breaths of our keypresses now stir

the dust inside a box known to us, in a building we can enter, in a city we know.

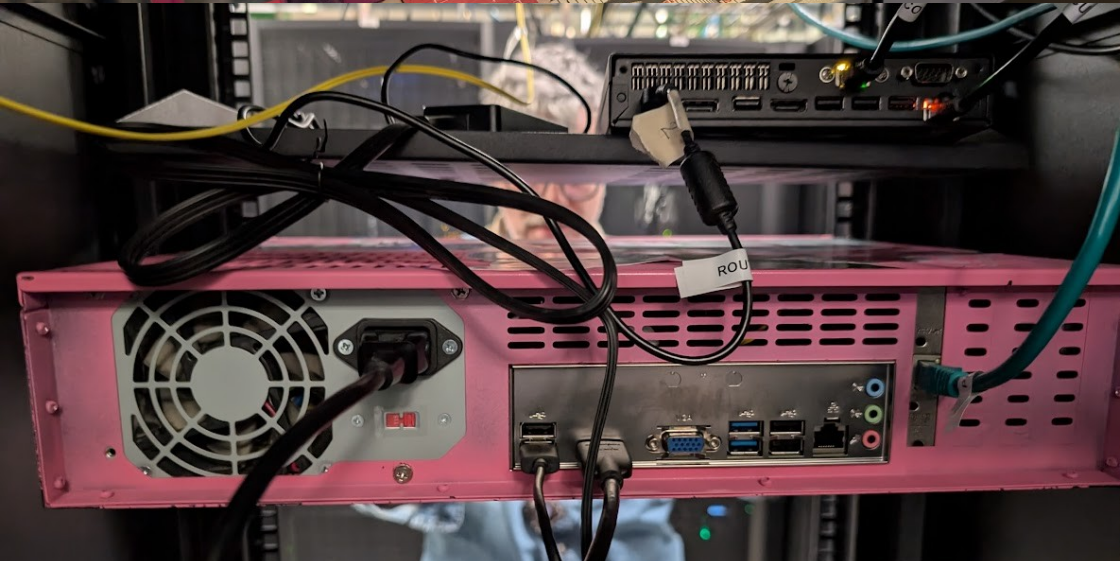
It's comforting to know whose hands caress that machine and why. It feels good to picture our bright pink machine covered in loving scrawls and stickers with the big power switch on an ~insom-etched metal plate. It's a reflection of all of us and the community we've built since 2014. It's our touchpoint, our index, into the expanse of the internet.

Whenever I connect to the town, now, I think of that machine and the human touch that went into its birth and goes into its maintenance. It's a nice feeling and it's replaced the vague unease I've always had about town being a cold slice of some towering rack.

Thanks to ~insom and all the volunteer admins who keep the machine humming.







We Lean in to Listen

by Woodsii

I was hanging out with some friends yesterday, watching a Japanese yaoi rom com when the folds in my brain scrunched up, raisin tight, and produced an image. An image of freshly baked baguettes, steaming hot when cracked open. I had to make them.

Growing up, my mom made a lot of bread. It was the closest thing that we ever got to white bread in my house. She had this hulking bread maker for more than 10 years. It lived on the countertop, shaking it as it worked. It whooped, crying under the strain of kneading the dough; it screamed like our lives are in danger when a loaf was done. She'd extract the loaves from its jaws, hot-potato tapping the underside of the loaves and listening for the hollow ring of a well-baked loaf. I'll go home to Illinois, and there'll be dough churning away in the machine and the hallmark chlorine smell of my mom's bread floating in the air.

Camden, my older brother, got fancy with it though. He's always been the baker. When my nana died, he labored over a cast iron press, churning out pizzelle. He spent one of his winter breaks between video games and crafting the perfect cinnamon rolls. Whenever he was home, there was always something delicious around, including baguettes.

Camden was to baguettes as Sisyphus was to the boulder. It was his white whale. In his post-grad, pre-graduate program funk, working in online tutoring, he made those damn baguettes.

The second time he made baguettes, he had already nailed them. But it wasn't good enough for him. He labored and toiled, thrifted and scrounged in search of the platonically ideal baguettes. If my mom's baking is like driving a minivan, Camden's is like designing the car. He did his work in sterile conditions, no

sound except the rhythmic slap of kneading: one, two, rotate. One, two, rotate. With practiced movements, he'd check the gluten development, roll the dough into perfect baguettes, score, and place in the oven. Out they'd come, golden and cracking. Sitting down to make my own baguettes felt like the start of a personal moonshot.

Getting started felt less like cooking and more like becoming a single parent. When working with yeast, I put the weight of their lives on my mind. I've always been more of a cookies and cakes and pies guy, not a grizzled bread maker. It felt like skydiving. I don't know the first thing about how to fold a parachute or what the correct way to fall is or even when to pull my chute. But everyone can jump, so that's what I did.

The recipe simply said: "bloom the yeast." I had set my kettle to boil but was struck with a premonition of my yeast being cooked alive. I spent 10 minutes in a three-way tango with the sink and the kettle. Camden would have prepared the perfect bed for the yeast to rise. Even my mom would have no trouble with her wonderful machine. They got magic while I'm stuck with a wet finger and anxiety about hot water yeast murder.

Camden's ritual was clinical: hydrate, autolyze, fold; it was a hard science. In my hands, it felt like wrestling a swamp creature. The recipe called for a gentle "stretch and fold," a rhythmic "up, over, rotate" that was supposed to build structure. Instead, the dough clung to me with the desperation of a needy toddler. I spent three hours picking dough off my hands, hair, and clothes, trying to mimic Camden's sterile precision in a kitchen that was bordering on condemned. There was no rhythmic slap of kneading here, just the clattering of the bowl and the splattering of dough as I wrestled. It felt like crashing his car.

Shaping them wasn't easy either. Copying Camden's refined rolling techniques fell flat when pushing my "baguettes" through a mountain of flour. They stuck to everything: my hands, the counter, the pan. My roommate Emily walked in on me up to my wrists in a glutinous sludge, bent over the counter and flapping at the faucet handle with my elbows. She kindly turned the sink on, put her earbuds back in, and went back to her omelet without a word. Pulling the baguettes out of the oven didn't change much. They were pale, and just as lumpy and small—my hope for any oven spring deflated. But tapping the bottom yielded that clear, hollow sound, like knocking on an expensive door. Despite how thin they got, their crumb was wonderful. Paired with bear-bottle honey and fresh butter, their warmth spread down to my toes. Zach got off the couch and came and devoured a hot baguette with me in a flurry of butter and honey.

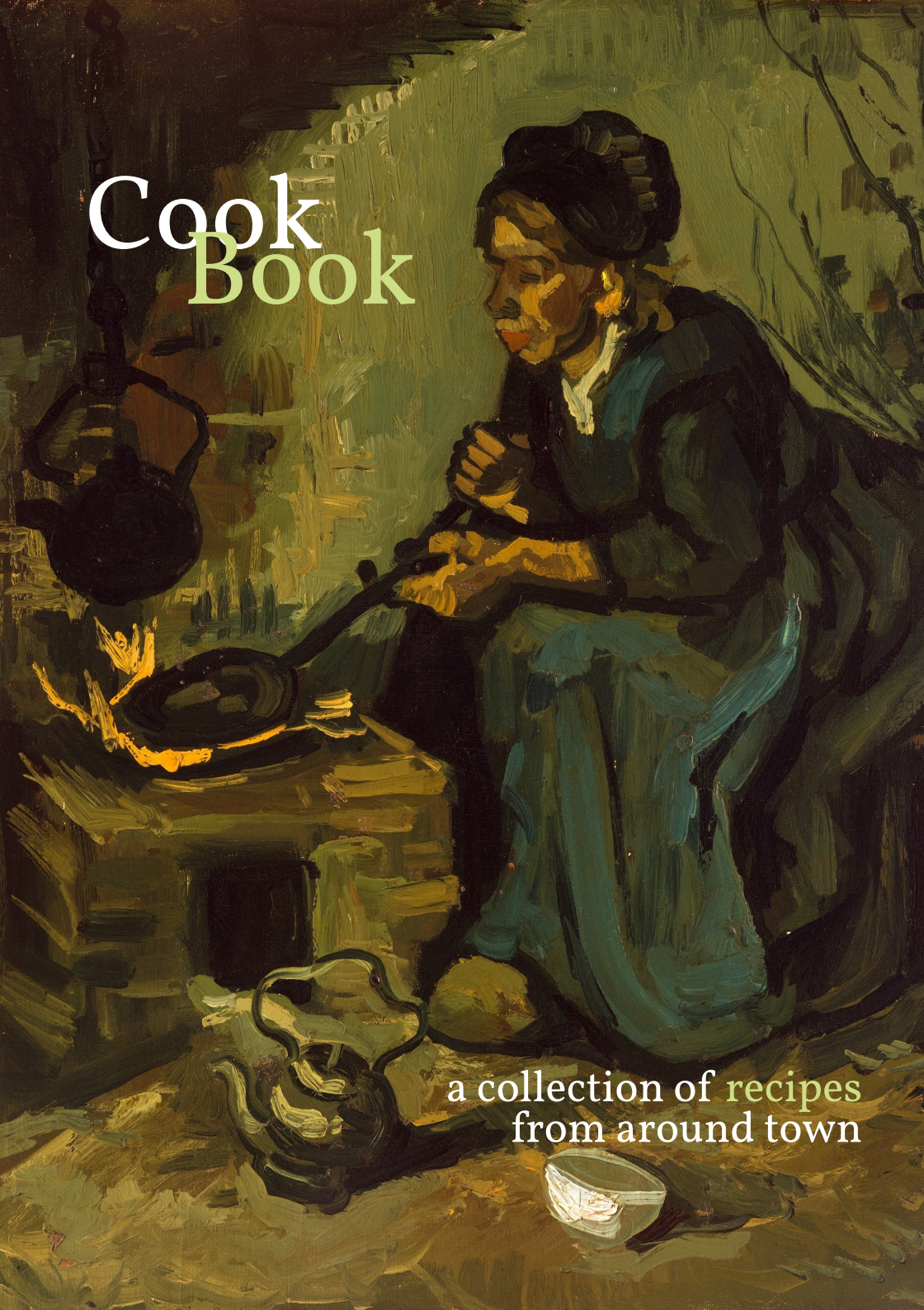
Camden did a number on me though: I was possessed by his perfectionist spirit. For 2 weeks after my little turd-baguettes, I filled our apartment up with loaves of bread. I switched to an even easier recipe. No knead, no fold, but just as sticky and gloopy. Dumping the dough onto the pan felt like littering; it slumped into a blob of slime. A delicious, crusty and fluffy blob, but a pile of refuse none the less.

Those loaves emerged from the oven tree-bark dark and slouched on the pan. After enough tapping and dreaming of the crumb, I'd slice them open, steam rising lazily from their exposed guts, and my roommates would come flocking like a group of hungry seagulls. They ignored my protests of "letting it sit for a second" and would rip and tear at it, cutting at it with butter knives. We devolved in those moments, standing around the oven, devouring my fresh, warm bread.

When pulling a loaf out, our glasses would fog, the greediest would burn their fingertips on the bread and swear, and we would laugh at them while stuffing chunks into our mouths anyway. Crumbs dusted the stovetop, our shirts, the floor. We danced around the flour, tracking ghostly footprints on the tile. We hovered around the oven like we were warming ourselves at a fire we didn't know we had been cold without.

One night, Zach pulled out a bespoke Nutella for us. Dan, who is famous for disappearing, threw a movie on TV. Emily cleared the counter, moving pizza boxes and the stack of the previous tenants' mail, making space like it mattered. We didn't talk about it. We just moved around each other; a choreography no one had taught us. The bread crackled as it cooled, a quiet and living sound. We all leaned in to listen, like it was telling us something.

Standing over each new loaf didn't turn us into bakers, but it turned us into a household. Zach started cooking Indian food for us; Emily started brewing beer. But mostly, we just stopped disappearing into our rooms. We sit in the kitchen now. We watch the steam rise.

An oil painting of a woman in traditional attire, possibly from a historical or cultural setting, cooking over a hearth. She is wearing a dark headscarf and a blue and black dress. She is holding a long-handled spoon or ladle over a pot on the hearth. A black kettle hangs from a chain on the left. The background is dark and textured with visible brushstrokes. The overall mood is warm and rustic.

Cook Book

a collection of recipes
from around town

Sourdough

Sourdough was the original leavened bread. Starting in ancient Egypt, people left their flour and water out together to collect the atmospheric yeast and imbue their bread with life. Since then, sourdough starter has been passed down to friends and families for ages—it was the primary way that bread was made until the early 1920s (I’m pretty sure that’s the date).

A lot of sourdough bakers today are incredibly precise and technical with their recipes, but I found that to be daunting. And frankly, that’s just not my MO. So I will give you Woodsii’s quasi-vibe-based sourdough recipe, starting with making a starter.

One last thing to note, you should change this recipe to fit your needs! The more experience you have, the more you are going to realize what’s best for you! Maybe it’s the amount of water you need. Maybe it’s the temperatures that you bake it, but expect some variables to change as you get comfy.

Ingredients and tools

The most specialized piece of equipment that you need is a scale! That’s it.

On the ingredients side, you need whole wheat flour (just a little bit) and all purpose/bread flour. I prefer bread flour because its higher gluten content means that it develops faster and has a better texture, but all purpose flour will work just fine.

Timing

- 1–2 weeks for developing the starter. 2 minutes of prep each day.
- 6–24 hours for baking the bread, really only 45 minutes of work. It’s a lot of waiting.

Onwards towards bread!

Starter

Your sourdough starter holds the magic of sourdough! Through atmospheric yeast, your sourdough will develop its own culture of yeast and lactic acid-producing bacteria!

Especially on the second and third day, it's going to smell a little funky (mine smelled like sweet corn at first). This is normal! It's just the other bacteria in the air developing a culture in your starter. Trust your yeast to win the battle: just keep feeding it, and keep it in a nice warm spot.

If any liquid appears on top of the starter, this is also normal! Just pour it off when you notice it. It's also not a big deal if the top dries out; your starter is still alive!

The end goal is to have it double in size 4–12 hours after a feeding. It should also be stretchy and viscous, in a way. That's the heuristic that I use to check if my starter is active enough.

Ingredients

To start:

- 1/2 cup (60g) of whole wheat flour (all purpose or bread flour will work too!)
- 1/4 cup (60g) of warm, filtered water

For feeding:

- 1/2 cup (60g) of all purpose or bread flour
- 1/4 cup (60g) of warm, filtered water

Instructions

1. Combine flour and water in a jar/chalice/receptacle of your choice! It should have the consistency of a thick pancake batter. Add flour & water to get to this consistency.

2. Cover with a towel and set out in a warm area of your house (ideally 80°F, but as long as it's not freezing, it's OK.)
3. Feed it once a day *at the same time*. Pick a time on the first day and stick to it!
4. Every other day (day 2, day 4, etc.), discard half of your starter before you feed it. Doesn't have to be exact!

Storage

After your starter doubles in size, you have 2 options for storage:

Countertop: if you plan on using your starter a lot, then this is the option for you! Simply feed daily and continue to discard ~half of the starter whenever the jar gets too full.

Fridge: if you aren't going to use your starter a lot, or are taking a break from baking, then into the fridge the starter goes! Feed your starter one last time before putting into the fridge. Don't discard, cover with a lid or plastic wrap and throw it in!

When you want to bake with your starter, just take it out of the fridge ~8 hours in advance so it can warm up and reactivate! Use the requisite amount of starter, feed it and place it back into the fridge. Easy Peasy.

It can last about a month in the fridge without feeding, but if you get anxious about it, then take it out, warm it up, feed it (maybe discard if it's *toooo* full), then throw it back in the fridge!

Bread

Here's where it gets fun! And edible!!

Now a word on this process. It *will* be different for you! Our kitchens are different, our sourdough culture is also probably different. Flour quality, temperature, oven style, etc., etc. There's a million variables to account for.

SO. I have put some of the times for the steps, but I have also put what you are looking for in your dough!

Ingredients

- 475g bread flour (all-purpose works, but not as well)
- 100g starter
- 325g warm, distilled water
- 10g salt

Instructions

1. In a bowl, combine flour, starter, and warm, distilled water. Mix this quickly, but thoroughly. Cover with a warm cloth and let stand for 30 minutes in a warm place.
2. Wet hands and gently scrape underneath the dough along the side of the bowl. Gently stretch the scooped dough up and over the middle of the dough and fold it over. Rotate bowl and repeat until all sides of the dough has been stretched and folded.
3. Wait 15–30 mins (longer is better!) and repeat step 2, the stretch and fold. Repeat this 3–4 times, until you can stretch the dough and kind of see through it. This means the gluten is developed! (You can go too far, so don't overdo it!)
4. Cover the dough and place in a warm spot. This is the bulk proofing step, and it is all vibes based. You want the dough to double in size, expand upward so that it's domed and jiggles when the bowl is shook. Depending on how warm your kitchen is, it could take an hour and a half or 6 hours. Be patient!! (You can also go too far with this step, but the bread will still taste delicious, it just won't be as fluffy)
5. After dough has doubled, gently turn the dough out from the bowl onto a floured surface. Shape the loaf in any way

you want, circle, oblong, square (good luck with that one) or whatever. Gently transfer to a baking pan with parchment or aluminium foil lightly oiled and floured. Place in fridge for at least 8 hours (you can leave it for like. a day tops), or leave dough out for at least 1–2 hours.

6. Preheat oven to 500°F (260°C) with a dish or pan of water in the lower rack. Remove loaf from fridge (if you did that), and lightly dust with flour. With a sharp knife or razor blade, score the bread DEEP. Go deeper than you think you need to go.
7. Place dough into the oven. While the door is still open, spray or flick some water to create a lot of steam! Don't open the door after this point!
8. Bake dough at 500°F (260°C) for 20 minutes. Reduce heat to 475°F (246°C) and bake for 15–20 minutes, or until it's a nice golden or dark brown on top.
9. Remove the dough from the oven and LET IT COOL BEFORE SLICING! Enjoy your kickass bread with butter, soups, sandwiches or just by itself : ^)

Some notes

I'm going to level with you. Your first loaf will probably suck nuts. Mine did! It was a gummy but delicious mess. I wayyyy underproofed it. But I was just getting to know my starter. We were partners that are not yet in sync. It's a process that takes time, learning all the quirks and variables of your kitchen, starter and process. Persevere! It's so fucking worth it!

Now I know this is a lot of information to throw at a person all at once, and invariably you are going to have questions. There's a glut of information out there on the process of making sourdough, taking care of starters, etc. If you ever want to reach out to me

though, please do! I'll be happy to answer any questions you have, or if you wanna correct any of the information here. Also mail me photos of your bread if you wind up making some :)

Much luv, ~woodsii



[I don't have a picture of woodsii's sourdough so here is a picture from Wikimedia Commons. Doesn't it look delicious? You should make some sourdough. I should make some sourdough. – Ed.]

BOLOGNESE¹

by ~nbsp

¹As prepared and eaten every day by a creature who's been dealing with combined ARFID and executive dysfunction for an overwhelming majority of its life

INGREDIENTS

Note: this recipe is for 500g of meat, but lately I've been cooking 300g instead, which I find is the perfect amount for me for a few bowls at dinner.

- 500g thawed ground beef
- 250g tomato paste²
- 1tbsp paprika
- ½tbsp cinnamon
- 1tbsp sugar
- salt
- olive oil
- a bit of boiling water

²The tomato paste has to be a *paste*; a *purée*. Pasta sauce, lightly mashed tomatoes, or polpa will not cut it—the sauce must be uniform.

STEPS

1. Throw the slab of mincemeat in the pan and let it fry a bit while splitting it into small chunks. It doesn't need to be fully cooked at this step, but it should be relatively brown almost everywhere.
2. Once the slab of beef becomes a heap of beef, throw all of the spices on it and start stirring. Some of the fat and water in the beef will have pooled by this point, which makes it easier to coat everything in spice.
3. Add the tomato paste and stir until uniform.
4. Add water until the mixture is submerged. When it starts bubbling, add some olive oil, turn the heat to low, and let simmer.

The longer the simmer, the richer the taste, however if you're in a rush, adding a tiny bit of water as opposed to covering everything works fine. Consider putting already-cooked pasta in the sauce and letting everything soak together.

The chef's recommendation: if you can find some [tahini](#) (the thick stuff you make sauce with, not the sauce itself), especially whole-grain tahini, mix a bit of it into your bowl. Tahini was what I'd douse everything in as a kid to be able to eat it—it makes everything better.

Food has always been impossibly hard for me. Ever since I was two, I have not once eaten a fruit or vegetable not in my explicit allowlist, one so exclusive I can enumerate it on one hand. Recent and not-so-recent attempts to try new things have always ended horribly, both physically and mentally, because it seems that my body has a psychosomatic neophobia; one time I tried a banana, quite liked it, and then puked once my brain caught up with what I was trying to do.

Apart from neophobia my eating disorder also manifests as a distinct fear of texture. All food must be tenderized or pulverized, every bite must be exactly as the last. Feeling an unexpected crunch or slimy sensation in my mouth will make me panic and abort everything immediately.

All of these issues, combined with my inability to get out of bed, let alone cook, have resulted in me effectively starving myself unless persuaded otherwise. My first year living alone was characterized by ordering loads of fast-casual food, eating it alongside frozen chicken nuggets and hummus for four days, then simply not eating until the new week rolled around. And yet, whenever I did cook, I felt so happy with myself. I felt accomplished. I felt brave. My stomach felt a lot better, too.

When I started this chapter of my life I promised myself to never go back to the state of constant starving that I subjected myself to before. To get it in my head that cooking is enjoyable, sustainable, and cheap. To understand that food is nonoptional. I'd like to believe I've been doing good on that promise so far.

For years I've been scared of showing this recipe to others, to admit this is what I eat every day, to proudly share it with an audience who will doubtless note, confused, that this is just meat in paste in water. Forget the *bolognese* vs. *ragù* discussion, this is neither. This is Soylent. There is no merit to this dish beyond its nutritional value. It is a pragmatic and hyperutilitarian approach—an insult—to food.

But it is what I eat. It is among the only things I know how to cook, and by far the easiest. I have eaten it almost every day for a month a half straight now, since I don't have that thing in my brain that causes things to get old. My childhood wasn't great (or frankly even *good*), but I can appreciate that for every dinner I had to face in fear, my mother would occasionally cook something she knew I'd be okay with. Something so deformed and devolved from its original form, yet so incredibly personal to the story of my life. I would not be here without it. This is my favourite dish. ✱



Latkes

as vaguely explained to ~nbsp in a voice message from her mother

preparation time: one hour if you're really fucking around

ingredients:

- soft potatoes, the type you make mash with
- two cups of flour
- one egg
- an egg's worth of the cheapest shittiest beer you can find.
alternatively, an additional egg
- any cooking oil you've got lying around; canola is a good choice.
- salt and pepper
- (powdered) sugar for garnish

1. peel and roughly grate a bunch of potatoes. if you washed them, try and squeeze out any remaining water from the grated bunch.
2. put in a big bowl, along with the egg, beer, salt, pepper, and flour. mash it all together until it's "puchki-muchki", and add some more flour if it's too runny.
3. fill up a big deep pan with cooking oil. when it gets hot enough, cup together pieces of your "puchki-muchki" into patties, and drop in the oil. make sure it's not stuck to the bottom, but otherwise leave it alone until the bottom side is nice and golden brown.
4. flip the latkes over and wait until the other side is properly cooked.
5. take the latkes out of the oil and leave to rest.
6. TAKE A PICTURE AND SEND IT TO YOUR MOTHER.
7. sprinkle some sugar on top and enjoy :D



A photo that was presumably sent to nbsp's mother.

Brunkager

baked by ~fruit on 2025-12-09



Brunkager are a type of crunchy Danish Christmas gingerbread cookie.

The recipe uses potassium carbonate (Danish: potaske) as a baking agent, which makes the cookies crunchier and browner.

The cookies will be **vegan** as long as you use **vegan** butter!

Prep time

- Dough: 20–30 min
- Resting time in fridge: approx. 4 hours (you can store the dough in the fridge for a few days before using it if you want)
- Baking time: 5–8 min

Tools

- oven
- grater
- cling film (alternatively: baking parchment and a small container)

Sources

- <https://www.valdemarsro.dk/brunkager/>
- <https://micadeli.dk/veganske-brunkager/>

Ingredients

- 250g wheat flour
- 125g (vegan) butter
- 125g brown sugar
- 70–80g maple syrup
- 80g peeled and chopped almonds
- 2 tsp. ground cinnamon
- 1 tsp. powdered ginger (you can additionally add some fresh finely chopped ginger if you want)
- ½ tsp. allspice
- ½ tsp. ground cloves
- 1 tsp. potassium carbonate
- Grated orange peel from ½ to 1 organic orange
- 1 tbsp. orange juice thereof

Instructions

1. Blanch almonds: Put almonds in a bowl and pour boiling water over them. Let sit for a few minutes until the skins can be easily removed.

Chop the peeled almonds coarsely (you can also just leave them whole or halve them).



2. Grate the peel of $\frac{1}{2}$ to 1 organic orange.

Save 1 tbsp. of the orange's juice and dissolve the potassium carbonate in the juice. Keep this liquid for step 5.



3. Put flour, spices, orange peel and almonds into bowl, and mix well with a spoon/fork. (Keep the juice with potassium carbonate for step 5.)



4. Put (**vegan**) butter, brown sugar and syrup into a pot and occasionally stir at low to medium heat on a stovetop until the butter is melted and the brown sugar dissolved. The mixture should not boil or get too hot to touch.

Take the pot from the stove.

5. Stir all the ingredients from the bowl and the potassium carbonate juice (from step 2) bit by bit into the pot with the handwarm syrup-butter-sugar mixture.

Stir/knead well with a spoon for a while until thoroughly combined.





6. Let the dough cool off for a bit and put it onto a sheet of cling-film. (If you don't have clingfilm, you could put the dough into a small plastic/glass container covered with baking parchment.)

7. Wrap the dough into the clingfilm and form into an oblong sausage-like shape.

Then, let this dough-sausage rest in the fridge for at least 4 hours until it has hardened. You can also easily just leave it in the fridge overnight.





8. Preheat oven to 200°C (or 180°C if you use the convection setting). Meanwhile, cut the cool, hardened dough with a sharp knife as thinly as possible and put the pieces on a baking tray.



9. Put the baking tray with the brunkager into the preheated oven for 5-8 minutes, depending on how brown you like them (and on the characteristics of your oven)

Note: The brunkager should be still somewhat soft when taking them out of the oven. After having taken them out, let them sit for a minute or so to harden a bit before trying to move them off the tray.



10. Put the semi-hard brunkager on a grate (or on plates) to let them cool and harden completely. Do not put them into a container until they are completely hardened and cold, otherwise they might retain moisture and get too soft.

Store them in an airtight container and (hopefully) enjoy :)

If you have any questions, suggestions, experiences etc., you can e-mail me at fruit@tilde.town if you want!



Hot Coco by Grendel

This is a very simple, but supremely delicious recipe.

What makes it special is the silky texture
provided by the coconut milk.

- 1/2 cup of full fat coconut milk
- 1/2 cup of oat milk
- 1/2 cup of regular milk
- 1/2 teaspoon vanilla extract
- 4 tablespoons of cocoa mix (or to taste)

~ cookbook ends here ~

because I want to:

- get used
- be plastic
- be disposable
- point, shoot, autofocus
- do one thing and do it badly



Nordøstfyn, Danmark (2020)

This kind of camera can take 27 pictures. I am currently 28 years old.

The camera has a flash. Enabling it makes a whining noise and turns the *flash ready* LED red. I often make whining noises, and turn red from embarrassment.



Fyns Hoved, Danmark (2020)

I did not finish my article in time. Instead, I will share some disposable camera photographs with you.

John G. Trudis

(Handwritten signature)

guy guy guy Jay Jay Jay guy guy guy Jay Jay Jay

NO U
industrijs

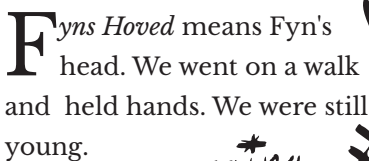


Nordøstfyn, Danmark (2020)

Those memories seem distant now, and washed out. Almost unreal. *How long, do you think, can I quench my thirst with tears?*

with tears?
Bitte nicht gut down G

I think I would prefer
if my life just ended
at 30. 27 is already
enough. There is dis-
tance between us.



Fyns Hoved, Danmark (2020)



München Valleystraße, Deutschland (2023)

heteronormativity
is & ?

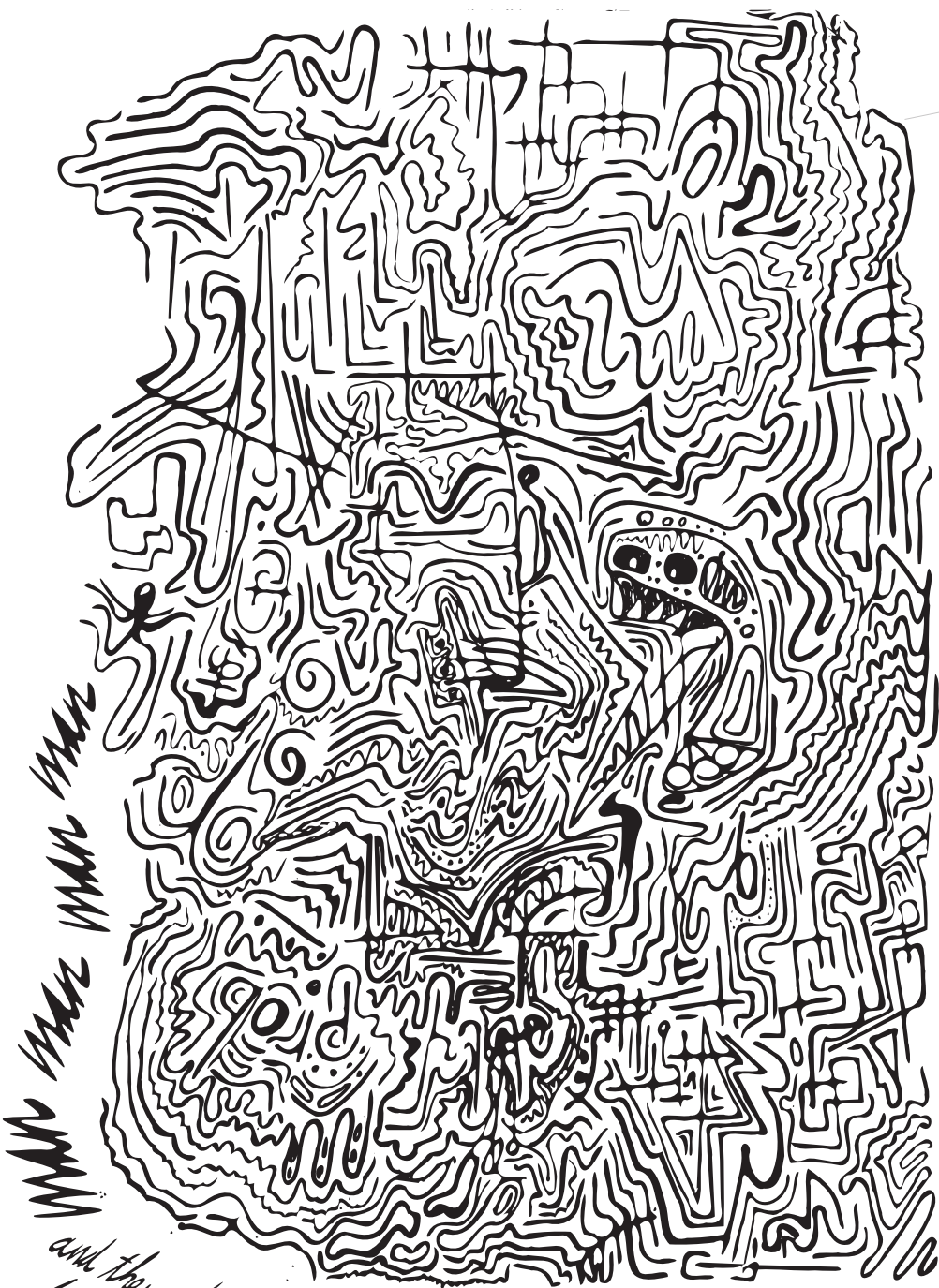


Odense Banegård, Danmark (2024)



Fyns Hoved, Danmark (2020)

36 gay gay gay ~~~~~ ♡ fruit inc.



and then she actually made it [snik @ plastikfrucht.com](mailto:snik@plastikfrucht.com)

Writing my own markup language *by acdw*

The article you are currently reading has been written in a markup language of my own design, called **JAMES**. It's a line-based language that borrows ideas from Gemtext (itself inspired partly from Gophermaps and Troff though it's not a programming language like Troff is but a markup language).

The rest of this article discusses my motivations for a new markup language and provides a tour of JAMES syntax. You can read the source code of this page and the JAMES program online.

[Some minor manual formatting has been done for the zine edition. Check out the sources here:

<https://www.acdw.net/james.txt>

<https://www.acdw.net/james-code/>

– Ed.]

Why *another* markup language?

There are already so many markup languages—from the featureful (LaTeX) to the lightweight (Markdown, Gemtext); from the procedural (roff) to the declarative (HTML). Before I began writing JAMES, I made great use of gemtext; in fact I think it's nearly perfect for my needs.

However, the operative word is *nearly*. Gemtext is not *quite* flexible enough to be an everyday markup language due to a few lacking features:

- The spec explicitly requires line-wise parsing. I've written parsers that relax that requirement so I can hard-wrap paragraphs, but that feels kludgy.
- Inline links are, IMO, important for accessibility and

readability. Gemtext doesn't allow them, leading to ugly layout or ad-hoc workarounds that I don't like.

- The spec is famously vague in a few key areas, for example verbatim blocks.
- I appreciate inline markup like bold, italics, or code.

More recently, I've been interested in the **roff* family of languages for their simplicity of syntax and expressive power. I tried to write a roff-like language that could output HTML (the groff html output is, like many GNU productions, *baroque*), and even made some progress—but I realized that I don't actually want to program my documents.

In looking for a happy medium, I've been working on and off on what eventually has become JAMES for around a year. The final thing that made the syntax “click” was processing lines based on whether they begin or end a paragraph, or continue a same-tagged or differently-tagged one. (More on that later.)

While I've only written a few things in JAMES so far, I'm enjoying it. In my opinion, it's a good combination of the best bits of gemtext and groff: it's more expressive and extensible than gemtext, while not requiring me to fully program my documents like groff. The balance between simplicity and expressivity is tricky to strike; with conditional output and verbatim tags, I think I've provided an okay “escape valve” into more complex markup while allowing most of my writing to stay easy to write and read.

For example, at one point I sketched out an image tag like

```
^ <url> ^ <alt text> ^ <title>
```

but realized that, instead of adding parsing complexity to handle this line type, I could just write

```
'" title="<title>" />
```

and be done with it.

To sum up with this *why* section: I wrote JAMES partly out of dissatisfaction with existing options, partly out of NIH syndrome, and partly (maybe mostly?) to have fun. It has been a success in all three areas!

JAMES syntax

JAMES is line-based in that it reads input line-by-line, then dispatches output based on each line's first character, or *tag*.

So far, so similar to *gemtext*, *gophermaps*, and *groff*. However, JAMES adds block-level semantics that I think provide a richer markup language. In broad strokes, JAMES

- Splits text into paragraph separated by one or more blank lines,
- Determines the paragraph's tag by examining its first line,
- And determines each line's tag by looking at its first character.

This means that when you denote a header with `# a hashmark`, the entire paragraph is part of that header. In addition, each tag has two meanings: a "paragraph-level" meaning and an "inline" meaning for when it's not the first line of the paragraph. I've tried to keep these two meanings closely related, but there are a few exceptions, notably `#` for headings and bold text and `*` for lists and italic text.

Lines

The main semantic unit of JAMES is the line. The program reads its input line-by-line, then formats and outputs each line depending on its position within its paragraph:

- An opening line emits the paragraph's opening markup.

- Subsequent lines emit their “inline” markup.
- The final line (strictly, the first blank line after the final line) emits closing markup.

As an example, the first header in this article was written

```
## Why
* another
markup language?
```

The initial `##` marks the paragraph as a subheader. On the next line, the `*` denotes *italics* because it's continuing the paragraph. If it were the first line of a paragraph, it would be rendered as a list item. The last line is a plain-text line and is emitted verbatim. Then, the whole thing is closed out (think `<h1>..</h1> in HTML).`

Inline processing

For fun, I also wrote JAMES to do some limited processing *within* lines: It encodes HTML entities including `<` (`<`), `>` (`>`), and `&` (`&`), and can “fancify” quote marks and dashes. That is, `--` is converted to `–` (en dash), `---` is converted to `—` (em dash), ``...'` is converted to `'...'` (single curly quotes), ```...''` is converted to `“...”` (double curly quotes), and (this is silly) `->` is converted to `→` (right arrow).

The Backslash — \

The backslash has three uses in JAMES:

At the beginning of a line, denotes a plain-text line:

```
\# this isn't a header
→ # this isn't a header
```

In the middle of a line, it separates in-element text from post-element text:

```
* some italic text\ and some not italic text
→ some italic text and some not italic text
```

At the end of a line, it tells JAMES to elide the spacing after it:

```
This lets you do (\  
* parentheses\)
```

→ This lets you do (*parentheses*)

You can escape a backslash by doubling it anywhere:

```
\\So many back\\slashes!\\
```

→ \So many back\slashes!\

JAMES tags

Below, I take you on a quick tour of the current tags for JAMES. In theory, JAMES is extensible—adding a tag is as simple as writing a function—but I don’t know how this translates in practice. I mean, it’s not even a proper application yet!

Plain text — \

Good ol’ plain text. Even though it’s not as simple as it looks it’s the classic we all know and love from decades of computing.

A plain text paragraph doesn’t require any tag, but a \ can escape a tag character at the beginning of a line, which makes it a plain text tag in practice. You can also use it at the beginning of a paragraph that you want to begin with a tagged line, e.g.

```
\  
# bold text
```

→

bold text

Link — @

The humble link is the element undergirding all of the

modern internet. It's what makes hypertext hyper. In JAMES, its syntax is

@ <http://example.com> like this.

Intraparagraph links look as you can see above, and if you start a paragraph with a link you'll get a list of them:

- [link 1](#)
- [link 2](#)

Heading/bold —

When beginning a paragraph, the octothorpe denotes a heading, "Markdown-style." That is, the level of heading is denoted by how many 'thorpes start the line: the heading above is preceded by three, for example. Intraparagraph octothorped lines render **as bold text**.

List/emphasis — *

Paragraphically, the star renders an unordered list:

- one
- two
- three

This tag, and the list of links above, are unique in that their contents must be completely on one line. This means that list items (and link list items) cannot contain other intraparagraph markup like other paragraph types. Figuring out how to fix this satisfactorily is on my list.

When used within a paragraph, a star *italicizes text*.

Quote — >

Inspired, again, by Markdown, the right angle bracket signifies quotation. When used within a paragraph, "it renders as a quoted phrase", but when used as the basis of a paragraph, it's a block quote:

[A] quotation is a handy thing to have about, saving one the trouble of “thinking for oneself, always a laborious business.” ~ A. A. Milne

Code — =

Equals for code is an Org-mode idea. I considered using the Markdown-style ``` but I think I’m going to leave that open for another possibility. Code does what you expect, both intrapara^{graphically} (=intrapara^{graphically}) and otherwise:

```
(defun fib (n)
  (if (< n 2)
      n
      (+ (fib (- n 1)) (fib (- n 2))))))
```

I’m not interested in syntax highlighting, so there’s no way to specify the language. Though having some way to specify parameters *might* be a good idea. ...

Special tags

The following tags are all processed before the others, so I’m calling them *special*. Here they are:

Verbatim — ‘

This is the “escape valve” I mentioned in the beginning of this article. Any lines—this tag does not work on paragraphs—starting with `‘` are passed through verbatim to the output (sans the starting quote, of course).

When paired with the next special tag, the quote can be quite powerful.

Command — /

This is a late addition to JAMES, but one that I think is necessary for its goal of producing correct output in

multiple languages. It's a late addition because it opens the door to programming documents—which if I wanted to do that, I'd just use groff. I hope to keep the number of commands small.

The only one I've defined for now is `if`, and the only condition I've defined is `is`. It allows, along with the verbatim tag above, structures like

```
/if is html
'<div style="border:1px solid;">box!</div>
/if is ms
'.BX "box!"
/end
```

→

box!

I'm still thinking about this feature, and whether using a command structure or something more purpose-built isn't better, like

```
?html
'<div ...>
```

I could even elide the quoting, maybe, from the in-between lines. I'm not sure.

Continuation — *<space>*

Continuation tags mark lines that are continuations of the one before. How does this differ from the end-of-line backslash? A continuation line is combined with the previous *before* parsing, so you can keep line length manageable without sacrificing anything. For example:

```
@ http://example.com/long-url
  A (not really) long url\.
```

→ [A \(not really\) long url](http://example.com/long-url)

as opposed to

```
@ http://example.com/long-url\  
A (not really) long url\.
```

→<http://example.com/long-url>

Comment — ;

A language would be incomplete without comments, wouldn't it?

Conclusion

Now that I'm re-reading this essay, I'm not sure if I answered the question posed by its title. I'll try to do that in a few sentences (or bullet points) here.

Why did I write my own markup language?

- I was unsatisfied with the extant markup languages
- I was fascinated by the idea of the line as a semantic unit, and wanted to try working with streams
- Parsing is simply a good time
- I thought it would be fun!

The result is JAMES, a fairly simple line-based markup language that I hope is sorta flexible. So far using it, it's comfy—partly because if something isn't comfy, I can change it.



north of Lietzow

~kls0e

portraits in ableism

by hush

My parents couldn't get my childhood doctor to acknowledge that I was deaf. According to him, I was just mother-deaf. After proving my hearing loss to themselves via the "attempt to startle her by banging pots right behind her head" test, they went around him and had my hearing loss charted by an audiologist. He was apologetic.

My parents decided to mainstream me. I never learned sign language, and did ten years of speech therapy to sound normal.

A regular question I received from other kids was "What's that thing in your ear?" Also, "What's that weird necklace?"

In fourth grade, I had another deaf student in my class. He had a different tech setup, so my teacher had to wear two microphones and radios. Whenever a guest speaker came, it was a production to disentangle the wires from her and get the guest instructor hooked up. She would do a little dance and sing "I'm free, I'm free, I'm free."

Hearing aids aren't typically covered by insurance. My father complained about how expensive they were.

I finally started learning to sign in high school. Just in time for my vision to start going, so I wouldn't be able to use it.

I started using a cane after college. My girlfriend at the time always mentioned that it scared her when I used it. She also didn't like that I ruined my outfits on dressy dates by bringing a backpack to hold my cane and hearing aid batteries.

She didn't have a reason for breaking up with me.

In my first forays into online dating, I hid things.

I went on a few dates with a woman. I ended things because I had a feeling she was the type to use my disabilities against me in a

fight. When I broke it off, she informed me that “it wouldn’t have worked out anyways, y’know, ‘cause you’re blind.”

My mother consistently thinks I must be tone-deaf. I’m not—I’m a musician.

I stopped hiding my disabilities in my dating profiles. I also stopped getting matches.

My brothers planned a trip without me, to do something they didn’t think I could do. I could’ve.

I decided not to have children. When I went in to discuss it, there was no need to ask if I was sure; I had a genetic disability.

I was using a cane on public transport. A man said to me “You’re not blind.” Me: “Not completely, no.” “Well, then don’t act like it.”

I have to make a choice between feeling like a fraud with the cane, or feeling mortified when I run into things or people without it.

I read Kurt Vonnegut’s *Galapagos*.

A friend who had been flirting a bit stopped after I brought my cane on a date. She’s often too busy to hang out now.

kindrobot's haikus

home, 4 hours ago

—

a cat with no hair
a joint that's first time two weeks
a glimmer of hope

haiku strats, a day ago

—

what's good about haikus
you can finish reading them
before the eye roll

crayons, 4 days ago

—

we'll be like crayons
we'll be full of colour
rubbing on the world

hope, 5 days ago

—

thought i was dead
i forgot i was a seed
wonder what i'll be

tribute to tribute to haikurobot, 6 days ago

—

messages of kindness
a well tested protocol
robot to robot

Tribute to haikurobot, @insatanity, 6 days ago

—

In life's hospital
growth, surrender or anger:
there is no wrong choice.

freedom, 7 days ago

—

freedom,
Abuela tells me,
is nothing more than the right to a decent life
– *Dreaming in Cuban*, Cristina García

sixth floor, 8 days ago

—

a pause, a waiting
our brain slows, our spirit grows
but no grippy socks

neurodivergent on grindr be like, 13 days ago

—

set Grindr name to
CAMH visiting hours
raise the red flags early

Hexaflexagons, Vi Hart, 10 years ago

—

<https://vimeo.com/147799851>

zap, 14 days ago

—

this might be the end
hike far out into crown land
and walk off the earth

next chapter, about a month ago

—

partner get new job
we go to seminary
we live our best lives

Tilde Town Free Jazz Band: 2025 Edition

by ~agafnd

A few years ago (2022) ~flowercorpse & ~malvarma organized a collaborative free jazz project. The process was simple: one person would record themselves playing an instrument for a while, then send that track on to the next person. The second person would improvise something to go along with that, & send their new track (without the first one!) on to the next person...if you've heard of the Surrealist game Exquisite Corpse, it's like that, but with non-idiomatic music.

It was a lot of fun & there was interest in doing it again so this year ~malvarma organized and I volunteered to mix. I am by no means a master sound engineer but I know my way around. Here's a little breakdown of what's going on in my Reaper session, in what I believe was the order the tracks were recorded.

You can listen to the finished track here:

<https://tilde.town/~malvarma/ttfj25/index.html>

~malvarma: ukulele

Ukulele is not an instrument that tends to stand out in a mix, especially when played sparsely as it is here. To help it shine through a bit more it's panned slightly left. (You'll see later that I ended up with two tracks panned left and two panned right.) To soften up the sound a little I added some reverb as well as a slight EQ rolloff above 10kHz.

~bx: bass

This is a very distorted and not super bassy bass guitar track with a moody, droning, sometimes discordant quality. To help it fit in the

mix better I boosted the bass frequencies and turned down everything over 1kHz.

~agafnd: guitar

I have a lot of instruments at my disposal but I went back to an old favorite, the electric guitar, played through a chorus pedal. I was very deliberately not staying in any particular key...as for mixing the track is panned slightly right with some of the high frequencies turned up.

~tromboneboi9: organ

Organs have infinite sustain and a huge range of frequencies. It is a beautiful performance but did threaten to overwhelm some of the quieter instruments! Backing off the gain by quite a bit, plus low & high pass filters, left room for others. A little stereo reverb gave some final polish.

~vilmibm: the mysterious drone

It's electric bass played with a crowbar. To me it sounds like unsettling whalesong. I can imagine it used in a film soundtrack. A large part of why I put a high pass on the organ was to leave room for this bass drone. Very little EQ done except for a notch filter to get rid of some background noise. Plus some reverb for more atmosphere.

~hush: drums

This was probably the most complicated one to mix as working with only one track for drums is a little difficult! I actually split it into two tracks by using a time honored technique: one track cuts off everything above ~150Hz, and the other cuts off everything below ~150Hz... meaning I essentially had a separate track for the

kick drum. (I considered doing an EQ carveout for the snare as well, but since it is mainly played cross-stick in this track it would not have been very useful.) The kick track has some compression on it to pump it up, & the “everything else” track has a bit more EQ to boost the midrange a bit.

~angea: synth

Another sparse but beautiful track. This one I panned over to the left...plus an EQ boost in the high end.

~maxpm: synth

To balance out the other synth, this one goes over to the right. A couple of notch filters to remove background noise. Plus reverb and a limiter to boost the volume without going too loud in the parts that were loud originally.

master channel

Then I put a limiter on the master channel so that nothing could get too loud and cause the mixdown to clip!

Papillon by mio

Designed 2026-01
Folded from one uncut square

For ~abraxas, 一期一会



Mail::RFC822::Address: regex-based address validation

while passing the time on the train to a Christmas party, i came across the following page <<https://pdw.ex-parrot.com/Mail-RFC822-Address.html>> describing a Perl module using a unified regex to determine the validity of an email address. the author of this page was kind enough to let me edit it down and add it to this edition of the zine.

~nbsp

`Mail::RFC822::Address` is a Perl module to validate email addresses according to the RFC 822 grammar. It provides the same functionality as `RFC::RFC822::Address`, but uses Perl regular expressions rather than the `Parse::RecDescent` parser. This means that the module is much faster to load as it does not need to compile the grammar on startup.

The grammar described in RFC 822 is surprisingly complex. Implementing validation with regular expressions somewhat pushes the limits of what it is sensible to do with regular expressions, although Perl copes well. The regular expression below shows the complexity, although its inclusion might cause some confusion:

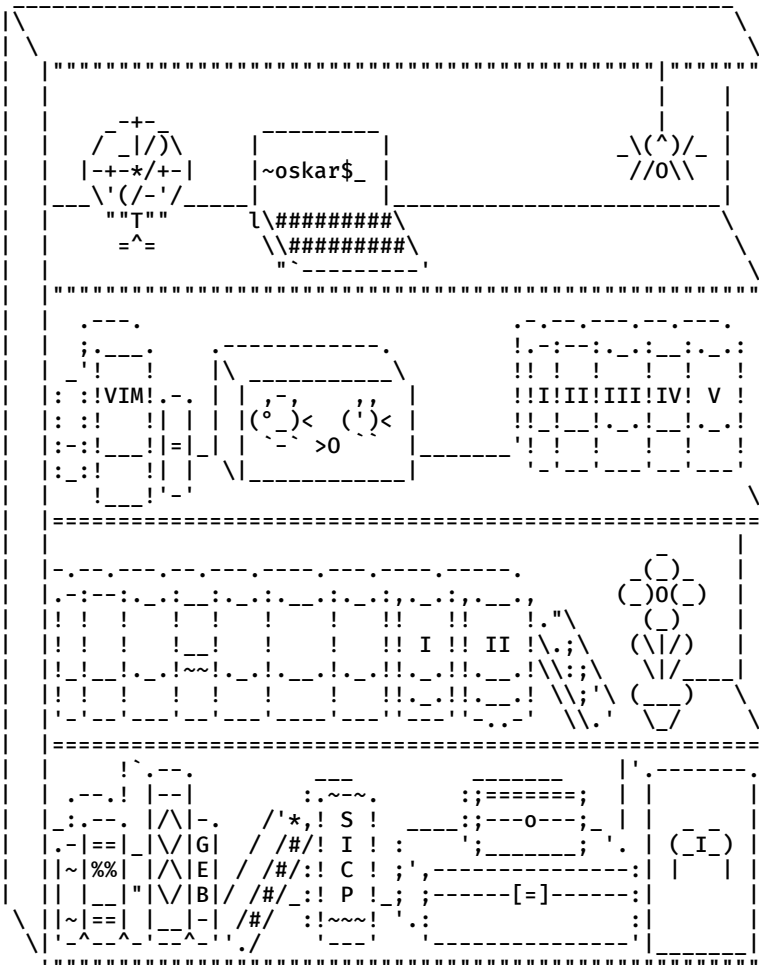
- I did not write this regular expression by hand. It is generated by the Perl module by concatenating a simpler set of regular expressions that relate directly to the grammar defined in the RFC.

- The regular expression does not cope with comments in email addresses. The RFC allows comments to be arbitrarily nested. A single regular expression cannot cope with this. The Perl module pre-processes email addresses to remove comments before applying the mail regular expression.
- I do not maintain the regular expression below. There may be bugs in it that have already been fixed in the Perl module.

[illegible]

59

oskar's bookshelf



Things in this shelf (in order) :

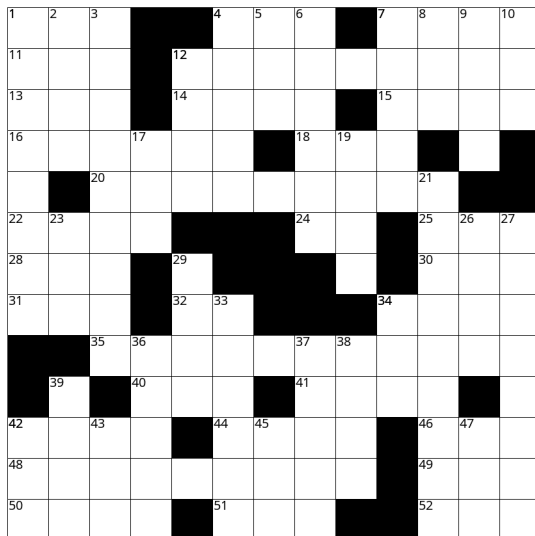
- An armillary sphere (for my awesome girlfriend)
- a laptop (why not ?)
- Anthologie de la poésie française (André Gide)
- a book about vim !

- Alcools (Guillaume Appolinaire)
 - an aquarium
 - the five parts of Spinoza's Ethics
-
- A notebook where I write my ideas
 - The ego and its own (Max Stirner)
 - De l'inconvénient d'être né (Emil Cioran)
 - On Crimes and Punishments (Beccaria)
 - Capitalisme, désir et servitude – Marx et Spinoza (Frédéric Lordon)
 - La doctrine du consentement (Clara Serra)
 - Par delà le principe de répression (Geoffroy de Lagasnerie)
 - philosophie des sciences I & II (éditions VRIN)
 - Le petit prince (Antoine de Saint-Exupéry)
 - a flower !
-
- La horde du contrevent (Alain Damasio)
 - Watchmen (Alan Moore)
 - Nos corps alchimiques (Thomas Gilbert)
 - The complete works of Moebius (Jean Giraud)
 - Gödel, Escher, Bach (Douglas Hofstadter)
 - Structure and interpretation of computer programs
 - small box :
 - thread and needles (in order to sew and embroid)
 - origami paper
 - card decks (for magic tricks)
 - bookmarks
 - big box :
 - older notebooks (archive)
 - small percussions to play with the cajon
 - My cajon !

"HASHMAP" Puzzle and clues by ~pilosophos

ACROSS

1. Amazingly named subtitle file format
4. Meta description of FBI, CEO, and GNU
7. A couple nybbles
11. Delicious sandwich and RISC-V instruction
12. Place of accentuated rest
13. Common nonfree archive format
14. Assistant
15. So XD you lose 1 Across
16. Street orthogonals
18. Composite 19-Down connector, abbr.
20. Place we've made home
22. A desire that just won't go away
24. Saudi ccTLD
25. MS's PHP, sometimes with 49 Across
28. Often accompanied by [aah!]
30. Joey from Winnie-the-Pooh
31. Science Guy Bill
32. Atomic no. 47
34. Sea that dried up due to Soviet irrigation
35. Place to buy cheap goods of variable quality
40. Device type made by U.S. Robotics subsidiary Palm
41. Poetic retry
42. Go 32-bit integer!
44. Comes before T in international standard 8601
46. /kaɪnd əv biː/



48. Places to defend from wildcards
49. Basketball goal
50. Coined "Nintendon't"
51. What's up?
52. American petrol
19. Tease something out
21. Casting to a shorter data type
23. Plaything
26. Rise considerably
27. Mustelids often compared to felines
29. Jeff Bezos' head
33. Promises you cake after the test
34. Slayed (Boomer hint: Implies 7 Across maybe?)
36. Presto browser
37. Quality fit only for rodents
38. Legendary Ricoh 5A22 console
39. Disable audio
42. 
43. Screen that asks you to buy the full version
45. MacGuffin of the 1981 Indiana Jones movie
47. Sugar, snow, or snap, for three

DOWN

1. The rubbing off
2. Speakers of Russian, Slovene, and Ukrainian, for example
3. Exaggerated
4. Put before a court
5. British dude
6. Looks away
7. Local or organizational regulation
8. Tuber most commonly grown in the tropics
9. Detected by Ackbar
10. Currently Tokyo
12. Currently France
17. American biomedical research org.

A Monastic Look at Perl

by ~kirch

Perl was created in 1987 by Larry Wall¹ who described it as “a language for getting your job done.” In 1991, when *Programming Perl* was first published, Larry Wall’s [man page for Perl](#) went a little more in-depth, describing it like so:

Perl is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It’s also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines (in the author’s opinion, anyway) some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of csh, Pascal, and even BASIC-PLUS.)

Later, Wall [expanded this list of inspirations](#) to “C, sh, csh, grep, sed, awk, Fortran, COBOL, PL/I, BASIC-PLUS, SNOBOL, Lisp, Ada, C++, and Python”—he borrowed bits from these “because the various features of these languages ruled rather than sucked.” Unlike Python’s “one obvious way to do it”, Perl’s motto is TMTOWTDI “There’s More Than One Way To Do It”. Learning to get the same results with different code is one of the [steps to becoming a Perl Ninja](#).²

In the early 90s Perl [became the language of the dynamic web](#) because it was a “Glue Language” to glue together Unix commands,

1 A Benevolent Dictator For Life

2 Ninjas are sort of like Monks, right?

it was only natural to use it to glue websites together with CGI scripts. But I'm not here to present a chronology of Perl over the years, I wasn't there for it, anyhow it's been done. I remember some time the late 1990s when I was first trying to become a "real programmer" that I picked up some "Learn Perl in 24 hours" book, and not having much success in 24 hours put off the whole thing for nearly 20 years. I've since learned to hack Perl a bit and landed a job where it's used extensively. When you dig into what's available in the Perl ecosystem there are multiple templating systems, folks who write Perl like it's a shell script, or C or Java or even JavaScript, all cloistered away into their specific niches, mixing and matching their tools as needed. There is even [a weekly coding challenge](#) that proves TMTOWTDI with community submissions.

The title of this article alludes to the [Perl Monks](#) community, which is themed after a monastery, ranking members through monks, initiates, and saints. (The Perl programming language even features "[Sigils](#)", how esoteric!) The comparison of hackers to other devotees is not new, the Commodore had the "Guru Meditation Error," there's also [programming koans](#) and [coding koans](#), and the [Tao of Perl](#) or even [Zen of Perl](#). Not to mention somebody else's essay about The Cathedral & The Bazaar—the references are everywhere; for example in Steven Levy's *Hackers* many people are referred to as "gurus," their workspaces are called "monasteries" and the people who operated IBM mainframes were referred to as a "Priesthood,"

and those privileged enough to submit data to those most holy priests were the official acolytes. It was an almost ritualistic exchange.

Acolyte: Oh machine, would you accept my offer of information so you may run my program and perhaps give me a computation?

Priest (on behalf of the machine): We will try. We promise nothing.

Larry Wall has never hidden [his Christian faith](#), he's referenced it in his [excellent speeches](#), but he's no fire-and-brimstone bible-thumper,³ he discusses a variety of ideas with wit, joy, reverence, and irreverence. I've found [these talks](#) to be infinitely fascinating. He connects a lot of different ideas and ideologies, it's clear that Perl's "glue" was born from Wall's personal adhesiveness, his willingness to find the best part of everything and [steal like an artist](#).

You may be familiar with Wall's famous "[Three Virtues of a Programmer](#)": Laziness, Impatience and Hubris; but in his [second "State of the Onion"](#)—delivered at the 1998 Perl conference—he expands on that with three virtues of community:⁴

[The three virtues of a Programmer] are virtues of passion. They are not, however, virtues of community. The virtues of community sound like their opposites: diligence, patience, and humility.

They're not really opposites, because you can do them all at the same time. It's another matter of perspective. These are the virtues that have brought us this far. These are the virtues that will carry our community into the future, if we do not abandon them.

3 He probably understood the contents instead of weaponizing the book

4 Community is something I've been thinking about a lot lately

Basically, we just have to stay the course. Friedrich Nietzsche⁵ called it “a long obedience in the same direction”, which is a good snappy slogan. But I like the full quote too:

The essential thing “in heaven and earth” is ... that there should be long obedience in the same direction; there thereby results, and has always resulted in the long run, something which has made life worth living.

The passions that drive us must be tempered and honed in the fires of the real world. Diligence, Patience, and Humility are included in [older monastic tenets](#), they are virtues that cultivate a mindfulness that TMTOWTDI⁶ applies to more than just code, and a grace that comes from recognizing that we don’t all share the same thought processes and experiences to do things in the same “one obvious way.” There might not even be one obvious way—once you’ve begun working diligently you’ll find many more solutions and aspects to a problem than you initially imagined. If you think you know everything already you’ll soon be humbled, and that’s life.

5 See? Larry’s not holding a grudge against the guy who famously said “God is dead”

6 There’s More Than One Way To Do It, remember.

References

man page for Perl: <https://www.math.utah.edu/docs/info/perl_5.html#SEC7>

Wall expanded this list of inspirations:

<<https://www.perl.com/pub/1999/03/pm.html/>>

steps to becoming a Perl Ninja:

<<https://everything2.com/title/10+steps+to+becoming+a+Perl+Ninja>>

Perl became the language of the dynamic web:

<<https://opensource.com/life/16/11/perl-and-birth-dynamic-web>>

weekly coding challenge: <<https://theweeklychallenge.org/>>

Perl Monks: <<https://perlmonks.org/>>

“Sigils”: <<https://www.perl.com/article/on-sigils/>>

programming and coding koans: <<https://github.com/topics/koans>>

<<https://gopher.tildeverse.org/tilde.team/1/~kirch/koans/>>

Tao of Perl: <https://perlmonks.org/?node_id=93118>

Zen of Perl: <https://perlmonks.org/?node_id=752029>

Larry Wall’s Christian faith:

<<https://web.archive.org/web/20140908122704/http://www.techgnosis.com/wall1.html>>

Wall’s speeches and talks: <<https://www.perl.com/tags/state-of-the-onion/>>

<<https://wall.org/~larry/>>

steal like an artist:

<<https://files.addictbooks.com/wp-content/uploads/2023/10/Steal-LikeanArtist.pdf>>

Three Virtues of a Programmer: <<https://thethreevirtues.com/>>

Second “State of the Onion”: <<https://wall.org/~larry/onion/thumbonion.html>>

older monastic tenets:

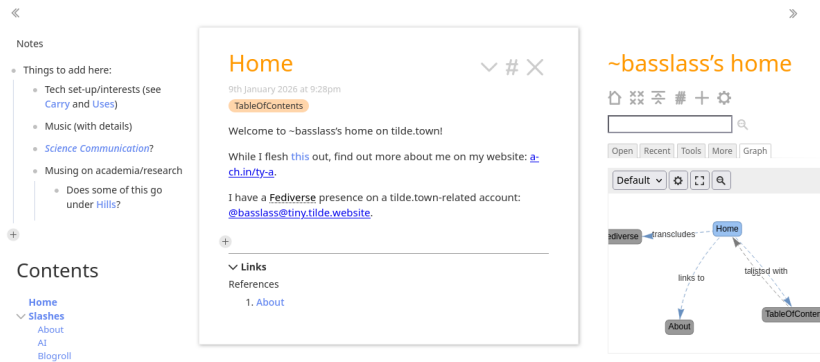
<https://en.wikipedia.org/wiki/Seven_virtues#Seven_capital_virtues>

TiddlyWiki on Tilde Town

~basslass

What?

This is what my home page on Town – tilde.town/~basslass – looks like:¹



How?

The page is an instance of [TiddlyWiki](#), a personal wiki, and a fun and useful [quine](#)! Being a quine means that you edit TiddlyWiki content within itself, as individual “tiddlers”, the smallest modular element within the wiki.

Give it a go! If you visit the above link and try to save the page from your browser (not on a phone), you will get the option to download

¹N.B.: At the time of writing, this page was struggling to load from UK-based IP addresses. Some lovely Townies tried to help me troubleshoot the issue, and the solution we ended up with was pointing my VPN to Montreal.

the HTML file serving the page. You can then open this HTML file in your browser and make any changes to it that you would like.

Why?

I have attempted to use TiddlyWiki as a personal wiki for various projects (personal and professional), relying on them being available locally on my computer (albeit sync'd across computers using [Syncthing](#)). This is indeed the most common use-case I have come across: storing information on your own machine, for personal use only.

But I wanted to try and host a website using the tool, because I thought it would be quite cool. And because I am a fool.

Where?

To access my `index.html` from Town on my own laptop, I use [sshfs](#), like this:

```
sshfs -o allow_other $town_username@tilde.town:/home/$local_username  
/home/$local_username/Tilde  
-o IdentityFile=/home/$local_username/.ssh/tilde.town -o  
reconnect,ServerAliveInterval=15,ServerAliveCountMax=3
```

I then open and edit it in Firefox, using [Timini](#) to save it. When I save the file locally, it gets updated on tilde.town immediately.

Feedback, please!

If you have any thoughts on how I am (ab-)using TiddlyWiki on Tilde Town, including if you have any ideas for how I can improve my workflow, please (a) drop me an e-mail via internal mail if you are a Townie or (b) ping me on Mastodon at @basslass@tiny.tilde.website.

only the forest can hear me cry.
it sympathizes with me,
through the drops of melted snow,
falling onto my neck and nose,
from the branches above.

9
by m455

years of tears from others,
pad my footsteps,
on the forest floor.

roots so strong,
that i have to step over them,
because the strength of suffering,
has nourished them into undefeat.

lichen unbothered by the cold,
with ears that have heard sorrow,
unknown by anyone.

rocks split and eroded,
from time,
the only inescapable killer.

green leaves emerge,
as a fog of vegetation in the distance,
and on nearby twigs,
as a reminder that hope is closer than i realize,

and that,
when the pews of humans do not answer,
the chaotic order of the forest does.

let's kill the music industry,
Or home recording in the 21st century
by ~piusbird
II-OI-2025

prelude

If there's one thing being a poor person in the 2020s has taught me, it's that poor people are excluded from most mainstream megacorp technology. Although this is a moral crime, one that our future descendants will look back on with shame and derision, just as we now do with the ancestors that allowed white supremacy to reassert itself after the civil war.

Despite this I find that for every megacorp technology that we as poor people are excluded from, there is usually a community run alternative, that is better for one's mental health and is usually cheap to free as well. But these alternative technologies do no one any good least of all poor people, if nobody knows about them. which is why I am dedicating my tilde zine submissions going forward, to highlighting the best of alternative technology, that is based on humane and communitarian principles. So without further ado let's kill the music industry.

Act I



In the early 1980s the British Phonographic Industry Association had a problem: record sales were down. Now in every industry you have up and down years, and I have not bothered to look at the statistics for record sales in the United Kingdom in from 1978–1981. You don't pay me that much, but rest assured that the British record industry was looking at record sales statistics and they were down.

I'm pretty sure that they had business analysts and statisticians and all the usual pointy haired boss infrastructure of a major trade association, so it is quite shocking that BPI came to the conclusion that

the customer was the problem

You see, in the year 1963, a Dutch company had come up with a revolutionary new magnetic tape recording format known as the compact cassette. Rather than inches on a reel to reel, the compact

cassette was a cartridge which enclosed magnetic tape reels in something about the size of a credit card. This would revolutionize the field of data storage, although it would not be called data storage for quite some years. For altho magnetic tape had existed since the end of world war two, previous open reel systems had been cumbersome and hard to use, and expensive as well. A cassette recorder cost upwards of \$5000 when they were introduced in the nineteen sixties. Or could at any rate, the archival research I was able to do was cursory at best. So recording a broadcast off the radio was at best a niche hobby.

Theoretically the introduction of the compact cassette with its cheaper tape and convenient format would make this problem go away, but it wouldn't be until seventy nine when sony introduced the first portable music player, the walkman, that the compact cassette would see mainstream adoption, at least in a widespread sense. For well over thirty years thereafter companies of all shapes and sizes would be shoving compact cassette players/recorders into various form factors, even children as young as four years old could have a cassette deck of their very own.



This is the my first sony, I had a knockoff one from kmart when I was five. Consumers with new technology are going to invent new ways to use it. The first pickup truck was actually a homemade conversion based on the ford model T. So it is no surprise therefore, that quickly people discovered you could record tapes using broadcast radio as a source, or records or anything really. And given that they were the size of a credit card, you could make sets of recorded music, and pass them around to your friends.

Thus began the age of music sharing, the record companies rather than innovating saw the customer as the problem, and quickly turned to the force of the state as the only solution to their dropping sales. The seemingly endless war begun in 1981 would not really end until the widespread adoption of Spotify and other music streaming services on smartphones in the mid two thousand tens.

Act two

The many ill effects of the music subscription streaming model, are beyond the scope of this article. For greater context on that I recommend this blog post by Eskin,¹ and the now classic Newsweek article.²

As an anarchist I am mainly interested in how we can revive lost culture of music sharing, because access to culture is a basic human right, it should not only exist for those who can afford it.

So I started looking around, various piracy options such as bittorrent continue to exist, and I won't gainsay anyone who uses them, but with ISP control over networks increasing, have you been able to get into your router recently? yt-dlp also exists, despite the many attempts by the recording industry and google to kill it, but I'm looking for something beyond individual tracks and in albums. Being able to rip individual tracks and albums from the internet does help, but much like the reel to reel days this is niche and hard for the average user to accomplish. As well as only being a solution to one leg of what I see as a trifold problem.

1 <https://blog.semiheskin.com/2025/05/19/how-music-streaming-services-are-killing-music/>

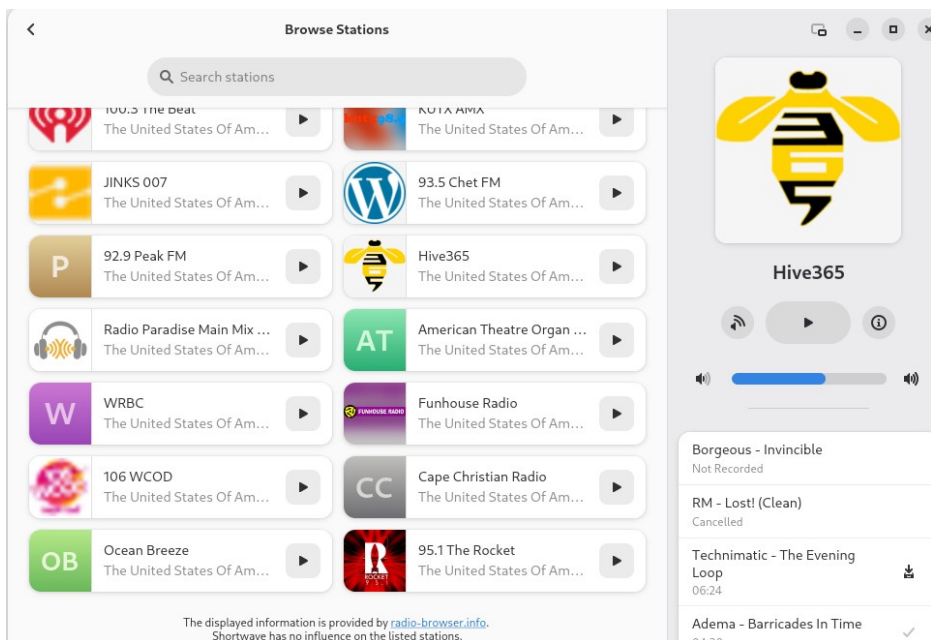
2 <https://www.newsweek.com/brief-history-taylor-swift-declaring-war-streaming-platforms-345546>

Act three

the trifold problem

what I want more than anything else is freedom from algorithmic music curation, which tends over time to recommend the same kinds of music over and over again. So I end up listening to only early two thousand christian rock, if I leave it to Spotify or YouTube. If god is listening I hope that the countless hours of toby mac will be counted against my time in purgatory, amen. Some of you may already be converging on the answer to the first problem as I call it.

Internet radio of course, but how does one discover stations, there ought to be a browser or client of sorts that aids discovery of new stations. Otherwise how would you know you like brony music for example. And to my everlasting surprise there is. There are several I think but the only one I've tried extensively is gnome shortwave.

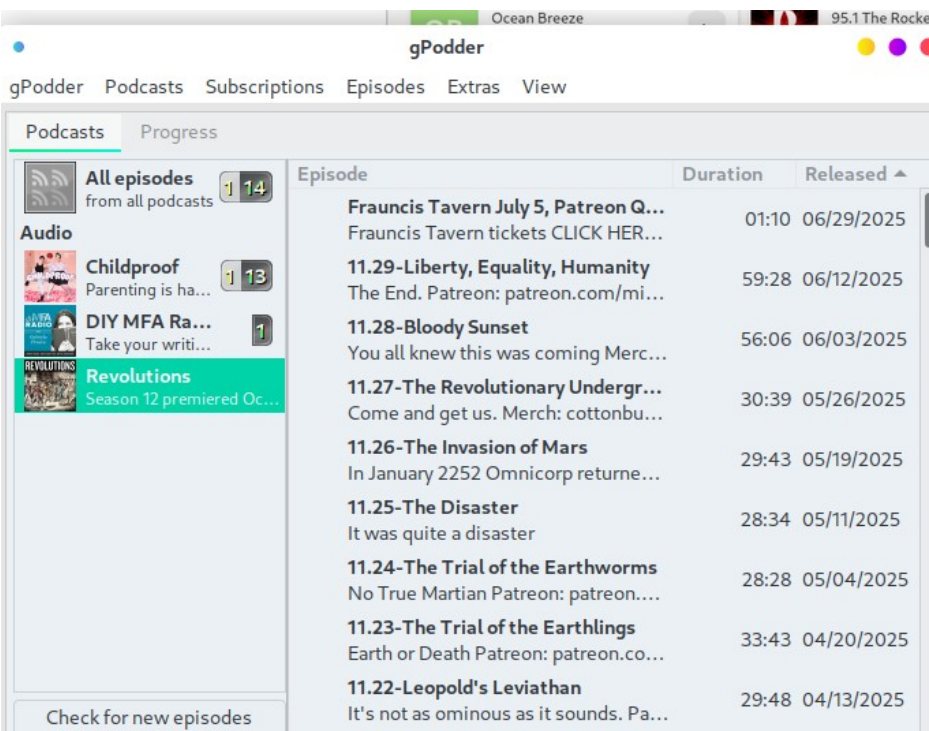


As shown in the screenshot gnome shortwave harvests info from a site called radio-browser.info with links to streams, over fifty thousand of them and growing. It's like a wikipedia for internet radio stations. And astute screenshot viewers may have noticed how shortwave assists you with home recording.

Home recording as the second leg of the stool

As alluded throughout this article I think the ability to curate your own music, or any sort of media collection, is a right not a privilege. So in any comprehensive anarchist technical solution to the problem of music home recording is a must have feature. On Linux you can always record from the monitor device of whatever sound device you happen to be using. But this is rather cumbersome so I was glad to see shortwave provide a home

recording solution. For stations which supply the correct metadata, shortwave will capture a number of tracks into memory buffer which can then be saved as ogg 64kb files. This still leaves you recording from a monitor to capture say npr programs, but that can be done just as easily with an old school RSS pod catcher like gPodder for example.



Other liberation devices

There is a program that can liberate audiobooks from audible called Libation—I can't show a screenshot of it because it reveals private information, nonetheless you should look it up on GitHub.³

³ <https://github.com/rmcrackan/Libation>

I have no idea how how to keep selections from library apps other than recording off the monitor device. If you have insight into this please contact me in the usual ways.

third leg an unsatisfactory landscape

At this point we have adequately solved for two out of three of the legs of the music audio entertainment problem as I posed it at the beginning of the article. Unfortunately there is no easily deployable widespread solution for sharing your hoard of legally acquired audio content with other people. True, there are solutions like cccopyparty,⁴ which make it fairly easy to share this and many other things with a small group of friends you personally know, but what about integrating this culture of home recording and sharing into our wider federated social culture. That will have to wait for someone more talented than me to come up with a solution.

Conclusion

I hope you have enjoyed this look at how an anarchist solves a technical problem. From the fundamental moral principle that access to culture is a right not a privilege, and then working our way through history and current technological forms to come up with a solution. Please do let's make home recording kill the music industry again.

4 <https://github.com/9001/copyparty>

Un soir d'hiver de Oskar

Un soir
D'hiver
Un bar
Couvert

Je te vois
Arriver
Et mon cœur
Se serrer

Tes yeux dévoilent
Un avenir
Ta joie dévale
Sur ton sourire

Chaque mot s'étend
Long comme une lame
Glissant sur tes dents
Pour mordre mon âme

Magnifique ! Tu prononces
Ces phrases qui approchent
Douce comme des ronces
Mon cœur qu'elles décrochent

L'ouïe sur ta peau esquissée
Que ton épaule fait crisser
Elle chante pour le passé
Sur lequel tu vas danser

Tes yeux qui parlent, quel fracas !
Renversants comme des vertiges
Ils portent un regard délicat
Sous lequel tout mon corps se fige

Quand ton sourire se joue d'émoi
Il fait couler ta tendresse
C'est elle qui me noie
C'est une déesse
Et c'est sa voix
Qui me blesse
De joie

color
by lateen



For a long time I've had a strange relationship with color. I'd always felt that it was a distraction from what made something what it was. More noise added on to the form of something, preventing me from actually perceiving that thing for what it is. I've been a small bit envious of blind people. A person who hasn't had vision all their life has never seen something like a tree. But they still know what a tree is. They haven't been distracted by its form, so I feel like they have a clearer idea of what a tree is than me.

I took this photo around christmas time in 2024. I wanted to try night photography with my newly hand-repaired film camera. I went to the local pond where christmas lights were hung on the trees, but on one tree there was an enormously bright blue flood light pointed back up it. It seemed almost unreal to me, so I knew I had to take a photo. After developing and scanning the film, some weird red streaks appeared. I have no idea how they came to be, but I feel they add a lot to the photo. A happy accident.

When I think of the idea of a tree, beyond the idea of its leaves, bark, or its almost imperceptible scent, when I think of the real essence of what makes a tree a tree, I don't think of blue. But when I saw this tree so saturated with blue I could hardly tell how green its leaves were, or how brown its bark was, I felt I had a greater understanding of its essence. I may not always agree with color and form. But on the rare occasion when a random light is pointed up a tree, or a chemical error occurs, I'll take what it has to say to heart.

Only Think This Now:

— The most accessible way to *Get Low* and *Be Profits!* —

BEEN WEARING THE CYBERPEAR
IN MY ANUS FOR OVER A YEAR
AND NEVER HAD ANY BLEEDING
OR AGONIZING RECTAL PAIN.
SHILLS OUT IN FULL FORCE
TODAY, HUH!



WE'VE DETECTED \$4.03 IN UNPAID
TOLLS. TO AVOID LATE FEES, PAY
YOUR BALANCE NOW AT OUR WEBSITE:
TOLLPAY.DEERSTALKERSLUTZ4FREE.TK



go bankrupt in under
10 seconds!

[-] Collapse Thread

- yes you are.
 - no i'm not.
 - yes you are.
 - no i'm not.
 - yes you are.
 ...[see more.]



happy, trappy &
trim in just ONE
post per day!

let's destroy.

I'M NOT SPECIESIST. I
EAT COWS, DOGS, AND
SOMETIMES PEOPLE.



I'M SORRY, BUT AS AN A.I.
TEXT-BASED MODEL, I AM
INCAPABLE OF NOT WRITING
LIKE A TRY-HARD.



Ivy Blossom Remix by piusbird

A short Preface

One of the themes of my early work as piusbird, was my profound struggle something I did when I was depressed back in 2014. I basically deleted all my creative work to that point... This basically amounted to 7 years of stories and poems, video games, and programming toys, of all sorts. This remains the greatest act of self harm I've ever completed. This is the first in a set of poems, in which I though still struggling with that, try to redeem and reclaim what I lost in 2014. To Those who will recognize the form, Yes it's a remix of song lyrics from My little Pony. Specifically this is from A Canterlot Wedding enjoy /)

Love is in bloom

The beautiful bride calls to the groom

There hearts burn as one

A bond embracing everyone

I said "love is in bloom"

She'll transform his life to make room

To love and to heal, yourself

your people

our world

Our union day

We Celebrate now, the human way

My friends, and yours right here

We are the light through fog, or fear

Because Love is in Bloom

She also calls to you

Saving the world takes more than two

Sing it now "Love is in bloom"

The holly and ivy prove

This world shall be born anew

For us

/-----/
/ DISPATCH FROM #OULIPO /
/-----/

.
.
.
.. .. this is a submission
' . . to this ~ town mag
. . . from that chat room,
. #oulipo

in that room #oulipo on town's IRC many humans
find fun in socializing using various constraints.

for now, this room just has a singular constraint:
i and folks banish a particular glyph from our words.
by now, having cast your optic orbs upon this dispatch,
can you pick out which glyph that is?

i trust that you can, smart guy!

.....: TOP SIX PLUS FOUR SAYINGS :.....

10.
dzwdz: oh fuck this shit
9.
vilmibm: that half fortnight's two day butt was good, but a lot
8.
vilmibm: you too will know that particular joy: a common glyph now void
7.
owl: lo, companions in this valiant stand against that most putrid of symbols
6.
12s: i oulipo a long thing, and think about it so hard...i can't post it for
i don't trust that it had no glyph. but fainlly i must trust, and post.
5.
tunas: a most facially-flushing thing
4.
agafnd: lasagna consumption puts that fat cat in happy land
3.
12s: i am going on a sky taxi to my root location

2.
 vilmibm: as first dish i put into my mouth oats which sat in soy milk all night
 owl: a fast fracturing worthy of champions

1.
 agafnd: raku's dad is a production of clams
 agafnd: raku's dad is against our obligation - naturally
 agafnd: but is oft found in strings worn on a fancy lady's throat
 agafnd: as much as it is found mangling and changing "strings"
 on our big boy calculators
 agafnd: must i talk in analogy only?

.....: OULIPO BOT:

#oulipo wants accountability in our word play and thus ~vilmibm took on a task:
 to program an irc bot without that glyph.
 it consists of a bash script and a raku script working as a duo.

```
*-----*
+ oulipobot.sh +
*-----*
```

```
#!/usr/bin/bash
```

```
inpath=/tmp/lol.in
outpath=/tmp/lol.out
```

```
rm $inpath
rm $outpath
mkfifo $inpath
mkfifo $outpath
```

```
tail -f $inpath | nc localhost 6667 | raku oulipobot.raku $inpath
```

```

*-----*
+ oulipobot.raku +
*-----*

my $inpath = shift @*ARGS;

sub irc ($thing) {
    qqx/printf "$thing\n" >> $inpath/;
}

irc "NICK oulipobot";
irc "US"~"\o105"~"R oulipobot 0.0.0.0 oulipobot :oulipobot";
irc "JOIN #oulipo";

my $i = 0;
loop ($i = 0; $i < 10; $i++) {
    $i = 0;
    my Str $msg = prompt;
    # ^:vilmibm!vilmibm@localhost PRIVMSG #oulipo :hmmmm$
    if $msg ~~ /\:(.+)\!.+ :s PRIVMSG :s .oulipo :s \:(.+)/ {
        if $1.contains("\o145"|" \o105"|" \c1077"|" \c1045") {
            irc "PRIVMSG #oulipo :$0!!! you said that glyph. kindly say again without it";
        }
    };
    if $msg.contains("PING oulipobot") {
        say "PING";
        irc "PONG oulipobot";
    }
}
}

```



On Honesty

2025 Chevrolet Silverado 1500

by tunas

The **2025 Chevrolet Silverado 1500** feeds on the roads like no one's business. It sets a high bar when it comes to the new standard for flagship full-size pickups, combining the legendary durability that Chevrolet trucks are widely known for with the cutting-edge technology demanded by the modern job site. Under its hood, a **6.2L EcoTec3 V8 engine** is able to deliver a respectable 420 horsepower and 460 foot-pounds of torque, making sure that no matter if you are hauling heavy machinery or towing a boat for a weekend getaway, you have the raw power at your disposal to handle the task with confidence. The exterior design remains as bold as it ever was; the overall poised stance of the truck and a redesigned front that commands attention on the highway are a massive improvement over last year's model, while the **Durabed** offers best-in-its-class cargo volume and 12 standard tie-downs to make sure your load doesn't go anywhere.

Inside, the cabin has been completely reimagined to be practically a mobile sanctuary, made out of premium materials like open-pore wood trim and standard leather seating surfaces that give you exceptional support during long drives across all of America.

As you settle into the ten-way adjustable powered driver's seat and set your side of the dual-zone climate control to a perfect 73, the outside world appears to drift away as soon as you roll the window up, replaced by the hushed silence of the triple-sealed doors that isolate you from the noise of the road. The centrally mounted **13.4-inch diagonal infotainment touchscreen** glows with an inviting brightness, integrating seamlessly with your mobile device to provide intuitive and precise navigation; though, after several hours on the interstate, we noted that you

may notice the GPS guiding you toward a route that does not exist on Google or Apple Maps.

The isolation of the cabin will now begin to feel less like a luxury and more like containment, less like it's keeping anything out and more like it's keeping you in. The heavy, pressurized silence will throb against your eardrums as the Silverado's **automatic high beams** try, but ultimately fail to cut through the darkness; a darkness so thick it seems to swallow the light rather than be illuminated by it. You might attempt to disengage the **Super Cruise** advanced driver assistance technology, but the steering wheel remains fixed, locking the Silverado on a rigid and unwavering trajectory. The digital instrument cluster ceases to display speed or fuel levels, showing instead a countdown that ticks backward in symbols you recognize from dreams that you *promised* yourself you would forget.

The comfortable hum of the V8 engine will then deepen into a **low, rhythmic chanting**, vibrating through the chassis and up your spine, syncing with a heartbeat, yet one that is far too slow to be your own. You look over towards the passenger seat and realize the premium leather stitching has begun to weep a dark, viscous fluid. The wood trim will expand, twisting into bony shapes that resemble ribs that appear to ensnare you within the belly of the machine. You are no longer driving; you are being digested by a being of chrome and steel that has finally had a chance to feed, carrying you down into the **subterranean depths**. You look back towards the windscreen in front of you. The road has dissolved into a grey, featureless mist where the laws of Euclidean geometry no longer apply, and as it clouds over with a frost which forms on the inside of the glass, you realize the meaning of the phrase:

The **2025 Chevrolet Silverado 1500**
feeds on the roads like no one's business.

```
#include <math.h>
#include <stdio.h>
#include <unistd.h>
```

```
/* width (INT) */
#define D 80
/* scale (INT) */
#define S 8
/* FPS (INT) */
#define F 16
/* animation speed (FLOAT) */
#define R 0.5
```

Download this file at:

https://tilde.town/~turnipgod/drunken_chess.c

```
int main(int argc, char **argv)
{
    double      c,i,x,          y,f,w,          h,csn=
cos(.9         ),t=0.          ,s[8],          sn=sin
(0.9);         for(;;         ){;;;         )printf
                ("\"e[\"      \"1;1H\"      \"\"e[2\"      \"J\"));;
                for(y=        0;y<D;          y++){;          for(x=
0;x<D;         x++){;         ;s[0]=          csn;s[
1]=-sn         ;s[2]=          sn;s[3]         ]=csn;
f=.1;w         =x;h=y         ;for(i         =0;i<6
;i++){         ;c=.5*         sin(f*         (w*s[2
]+h*s[         3]))+((         00+t)*         (M_PI/
16))+i         )/f;w=         w+c*s[         0];h=h
+c*s[1         +0];f=         f*1.4;         s[4]=s
[2*0]*         csn+s[         1]*sn;         ;s[5]=
s[0]*-         sn+s[1         ]*csn;         ;;;;s[
6]=s[2         ]*csn+         s[3]*-         sn;s[7
]=s[2         ]*-sn+s         [2+1]*         csn;s[
0]=s[4         ];s[2]         =s[6];         s[1]=s
[5];s[         3]=s[7         ];};;;;         putc((
((int)         w+(/*-         */int)         h-((/*
*/int)         h%S))%         ((S*2)         )>(S-1
))?35:         32,/**         *****/         stdout
                );;;;;         }putc(         0xA,/*         *****/
stdout         );;;;;         usleep         (1.0/F
*1E6);         t+=R;         return         0x000;
}
```

```
/*-----
| "Drunken Chess"
|
| to run:
| gcc drunken_chess.c -lm && ./a.out
|
| ~turnipgod 01/2026
|-----*/
```



How zines were made back in your grandparents' day.

credits & licenses

Most of the text was set in Vollkorn, Work Sans, FreeMono, Iosevka Term Slab Light, Fira Code, and League Spartan. Any other fonts came in pre-typeset submissions.

The following public domain images were used:

Moby Dick illustration from a poster made by the Otis Lithograph Co. for the 1926 film “The Sea Beast.” Source: <https://commons.wikimedia.org/wiki/File:Theseabeast.jpg>

Peasant Woman Cooking by a Fireplace painting by Vincent van Gogh. Source: https://commons.wikimedia.org/wiki/File:Peasant_Woman_Cooking_by_a_Fireplace.jpeg

Sourdough photo by Alyinthearchives. Source: https://commons.wikimedia.org/wiki/File:Homemade_Sourdough_Boule.jpg

Youngster Spraying Water on Boiling Peanuts photo by Al Stephenson. Source (yes this is the actual URL): https://en.wikipedia.org/wiki/File:YOUNGSTER_SPRAYS_WATER_ON_BOILING_PEAUTS_IN_HELEN,_GEORGIA_NEAR_ROBERTSTOWN._HELEN_WAS_A_TYPICAL_SMALL_MOUNTAIN..._-_NARA_-_557700.jpg

The zine as a whole is licensed under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

Layout, typesetting, design CC BY-SA 4.0 agafnd.

Authors retain full copyright of their own works unless otherwise explicitly stated.

“Writing my own markup language” CC BY-SA Case Duckworth

Cover, editorial, new around town, TTFJB writeup CC BY-SA 4.0 agafnd

“TiddlyWiki on Tilde Town” CC BY 4.0 basslass

Bolognese recipe CC BY-NC-ND 4.0 nbsp

Latkes recipe CC0 nbsp

“Camera Obscura” CC BY-SA 4.0 fruit

“Portraits in Ableism” CC BY-NC-SA 4.0 hush

“North of Leitzow” CC BY-NC 4.0 kls0e

“Color” CC BY-NC-SA 4.0 lateen

“9” CC-BY 4.0 m455

“Only Think This Now” CC-BY-SA 4.0 neverheardof.me

“Papillon” CC-BY 4.0 mio

Mail::RFC822::Address article copyright Paul Warren, all rights reserved (except for the code, which is under the MIT license). Used with permission and lightly edited by ~nbsp to fit the zine format.

“Pius Kills The Music Industry” CC-BY-SA 4.0 piusbird

“On Honesty” CC-BY 4.0 placed99th

“Drunken Chess” dedicated to the public domain under the Unlicense by turnipgod.

Full text of various licenses:

CC BY-SA 4.0: <https://creativecommons.org/licenses/by-sa/4.0/>

CC BY 4.0: <https://creativecommons.org/licenses/by/4.0/>

CC BY-NC-ND 4.0:
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

CC BY-NC-SA 4.0:
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

CC0: <https://creativecommons.org/publicdomain/zero/1.0/>

MIT: <https://opensource.org/license/MIT>

Unlicense: <https://unlicense.org/>

