Episode Book

EPISODE #5488

How Emoji Went From 176 Characters to 3,600

PUBLISHED	23 September 2026 at 14:17 UTC
DURATION	24:02
LISTEN	myweirdprompts.com/episode/emoji-history-unicode-encoding/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:36 UTC

SYNOPSIS

Emoji didn't start as decoration. In 1999, NTT DoCoMo's Shigetaka Kurita built 176 tiny pictures to carry tone that plain text on a tiny screen couldn't — and named them "picture characters," not after the English word emotion. This episode traces the whole arc: carrier incompatibility, the Unicode Consortium's standardization starting in 2010, the two-to-three-year proposal pipeline, and why the same code point renders as a prayer on one platform and a high five on another. Then it gets practical: why emojis work in staging and vanish in production (hello, utf8mb4), why an emoji is a four-byte canary for your entire text pipeline, and how the growing acceptance of emojis in business contexts has played out in Japan, where they've been in use the longest.

TRANSCRIPT



Corn

You know the thing where you type a sentence, and then your thumb hovers over the little yellow face for a second, and you have to decide whether this is a face situation or not.



Herman

The pause before the face.



Corn

It's a real moment. And Daniel wrote in this week because he had one of those moments, except his was about the word itself. He said it never occurred to him, until he read the name, that the emoji originated in Japan.



Herman

Right, the word.



Corn

The word. He wants the origin story — how the emoji first came to be, what it was actually for. Then he wants the arc: how something that started small became this vast expanding library, and the compatibility problems that come with that. He's got a specific technical gripe, which I suspect is going to be the meat of this for him — he's had deployments where emojis on a website didn't propagate from staging to production. Just didn't make it.



Herman

Oh, I know exactly what that is.



Corn

You'll get your turn. And then the softer half. He wants to talk about the growing acceptance of emojis in business contexts. He says everybody has felt a little ridiculous the first time they used one in a business email. He's curious how the adaptation has gone in Japan, where they've been in use the longest. And he's honest about his own position — some people find them immature, he has mixed feelings, and sometimes it's just easier to convey with an emoji what would otherwise take a few words.



Herman

That last part is the whole thing, honestly.



Corn

So let's start at the beginning. Japan, before the smartphone era.



Herman

Nineteen ninety-nine. NTT DoCoMo, which is one of the big Japanese carriers, and a man named Shigetaka Kurita. He was working on their mobile internet service, and the constraint he was designing against is the important part. This was not a phone with a big glass screen and a keyboard. This was a pager-like interface, tiny display, very few characters you could send.



Corn

So the constraint is characters.



Herman

The constraint is characters. And in Japanese, a text message isn't just short because the screen is small. It's short because of how the language and the politeness norms work. There's a whole register of communication where you don't say the emotional content outright — you imply it, you leave it to context. Plain text on a tiny screen strips all of that out. You get the words and none of the tone.



Corn

Which in a language where tone is doing a lot of the work is a real problem.



Herman

It's a real problem. And Kurita's solution was to build a set of little pictures that could carry the tone in one character instead of a sentence. A hundred and seventy-six of them. Heart, sun, cloud, a few faces. That's the original set.



Corn

And the name.



Herman

E and moji. Picture character. It's not a contraction of the English word emotion, which is the thing everybody assumes. It's a coincidence that it sounds like it.



Corn

That's a good coincidence though.



Herman

It's a great coincidence. It's why the word travelled so well. But the concept is Japanese, the word is Japanese, and the motivation was Japanese. It was solving a problem that existed in that specific mobile culture.



Corn

So it wasn't decoration.



Herman

It was infrastructure for tone. That's the thing people get wrong. They think of emojis as stickers, as garnish. The original set was doing a job that the text couldn't do. And then it spread — other carriers built their own sets, and now you've got a situation where DoCoMo has one set, au has another, SoftBank has another, and they don't talk to each other. You send a heart from one network, it arrives as a blank box on another.



Corn

Which is the compatibility problem in its purest form, before Unicode even shows up.



Herman

Before Unicode even shows up. And that's what forced the standardization. The Unicode Consortium started encoding emojis into the standard in twenty ten. Small set at first, and it has grown steadily since. As of the current standard, we're over three thousand six hundred emojis.



Corn

Three thousand six hundred.



Herman

And that's the count in the standard. The actual number of distinct images is larger, because some of those code points are sequences — a base character plus a modifier produces a different glyph.



Corn

Let's get into what an emoji actually is, technically, because that's where the compatibility problems live. An emoji is not an image file. It's not a PNG that gets sent over the wire.



Herman

Correct. It's a Unicode character. It has a code point, same as the letter A. When you send an emoji, you are sending a number, and the receiving device looks up that number in its own font and draws whatever it has for that number.



Corn

Which is why the same emoji looks different on Apple versus Google versus Samsung.



Herman

Yes. Unicode defines the code point and a general description — this one is a smiling face with tears, this one is a folded hands, this one is a peach. It does not define the artwork. Each platform's designers draw their own version. So the code point is universal and the picture is not.



Corn

And that's the source of a whole category of miscommunication, because the design carries meaning that the description doesn't.



Herman

The folded hands is the classic one. On some platforms it reads as prayer, on others as a high five, on others as a thank you gesture. Same code point. Completely different social content. And there have been real misunderstandings from that, people sending what they thought was a high five and the recipient reading it as something devotional.



Corn

So the standard is tight about the number and loose about the picture.



Herman

Which is the exact tension Daniel is poking at. The library is tightly defined — you can't just make up a code point, you have to go through the process — but the visual interpretation is decentralized, and the demand for new ones is coming from internet



Herman (continued)

culture, which moves a lot faster than a standards committee.



Corn

Walk through the process. How does a new emoji actually get in?



Herman

Anybody can submit a proposal. You write it up — you make the case for why this concept needs a character, you show that it's not already covered by an existing one, you show that it's going to be used, you show that it's not too specific to one culture or one moment. Then it goes to the emoji subcommittee, and they evaluate it against those criteria, and then it goes up to the technical committee, and then it gets scheduled into a release.



Corn

And the timeline on that.



Herman

The timeline is measured in years. Not months. Years. From a good proposal to a shipped emoji on your phone is typically a two to three year pipeline, sometimes longer. And the committee rejects most of what comes in.



Corn

So you've got this slow, deliberate, consensus-driven process on one side, and on the other side you've got a Discord server where somebody uploads a custom emoji of their friend's cat and it's live in four seconds.



Herman

That's the speed mismatch, and it's structural. It's not a bug in the Unicode process. The process is slow on purpose, because once a code point is assigned it's assigned forever. You cannot unassign it. So the committee is making permanent decisions about the shared character set of every computer on earth, and they behave accordingly.



Corn

And meanwhile the demand keeps piling up. There's a backlog of proposals.



Herman

There's a substantial backlog. And the workaround that the ecosystem developed is exactly what you'd expect: if the standard won't move fast enough, people build their own layer. Slack custom emojis, Discord emojis, Twitch emotes, the whole thing. Those are not Unicode. They're images with shortcodes, and they only work inside that platform, and that's the trade.



Corn

So you get the fast, expressive, platform-locked version, and the slow, universal, permanent version, and they coexist.



Herman

They coexist, and most people use both without ever thinking about which is which.



Corn

Alright. Daniel's deployment problem. Emojis on a website that worked in staging and didn't make it to production. What's happening there?



Herman

Nine times out of ten it's the database, and it's specifically the encoding on the database column or the connection.



Corn

Explain the encoding.



Herman

So MySQL, which is the classic case, had an encoding it called utf8. And the name is misleading, because it wasn't really utf8. It was a three-byte subset of it. It could store most characters, but it could not store anything that needed four bytes.



Corn

And emojis need four bytes.



Herman

Emojis need four bytes. Most of them live above the basic multilingual plane, which is the range that needs four bytes to encode. So you've got a database column set to utf8, you try to insert a character that needs four bytes, and MySQL either truncates it, replaces it with a question mark, or throws an error, depending on the settings.



Corn

And the fix has a name that developers say with a certain amount of feeling.



Herman

utf8mb4. Which is the real four-byte utf8. And the migration is annoying, because you have to change the column, the table, the database, and the connection string, and if you miss one of those four you get the same failure and you have to go hunting again.



Corn

Which explains the staging-to-production thing perfectly. Because staging and production are almost never configured identically. That's the whole point of them being separate.



Herman

Somebody set up the staging database with the right encoding and the production database got provisioned from an older template, or the connection string on one of them has a charset parameter and the other doesn't. The code is identical. The data is identical. The behavior is different, because the environment underneath is different.



Corn

There are other culprits too.



Herman

There are. An API gateway or a WAF that strips or mangles non-ASCII characters in transit. A front-end framework that handles the surrogate pairs wrong — because in JavaScript, an emoji is two code units, not one, so anything that does string length or slicing naively will cut an emoji in half and produce garbage. And there's the classic one where somebody puts a validation regex on a form field that only allows a certain character range, and the emoji gets rejected at the edge before it ever reaches the database.



Corn

So the emoji is a diagnostic.



Herman

The emoji is a fantastic diagnostic. It's a four-byte character that people actually type on purpose. If you want to know whether your whole pipeline handles modern text correctly, send an emoji through it. If it survives, your encoding is probably fine end to end. If it doesn't, you've just found a problem that would also bite you on any non-Latin script, any historical character, any of it.



Corn

It's a canary.



Herman

It's a canary that people voluntarily put into the coal mine.



Corn

And this is where the expansion pressure meets the encoding reality. Because the standard keeps adding characters, and every addition is another thing that has to survive every layer of every stack.



Herman

And the additions are getting more complex. A lot of the newer emojis are sequences, not single characters. A skin tone modifier is a separate code point that combines with a base. A family emoji is several code points joined with a zero-width joiner. A flag is two regional indicator characters. So the thing that has to survive your pipeline is now a string of four or five code points, and any layer that truncates or normalizes will break it.



Corn

And the platform rendering problem comes back at that point, because not every platform supports every sequence.



Herman

Right. You send a family emoji with a specific combination, and the receiving platform doesn't have a glyph for that combination, so it falls back to rendering the individual people side by side. Which is a different image than the one you sent. The message still arrives, but the picture changed.



Corn

So the standard is universal and the experience isn't.



Herman

The standard is universal and the experience is a negotiation between the standard, the platform's font, and the platform's fallback logic.



Corn

Let's shift. Daniel's second half is the cultural one, and I think it's actually the more interesting half. Why did it feel ridiculous to put an emoji in a business email?



Herman

Because business writing has a formality register that was built for print, and emojis are not from that register. A business letter has conventions — the salutation, the closing, the absence of exclamation marks, the avoidance of contractions in the most formal versions. All of that is signaling seriousness and restraint. An emoji is a burst of affect. It's the opposite signal.



Corn

So the ridiculousness is a register violation.



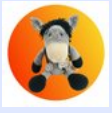
Herman

It's a register violation. You're mixing a medium that says "I am being careful and professional" with a mark that says "I am being casual and emotional." And the discomfort is real, it's not imagined. People feel it.



Corn

And yet it's shifting.



Herman

It's shifting because the medium shifted. Once the primary channel of business communication became chat — Slack, Teams, whatever — the register changed underneath the email. Chat is informal by default. And once you spend your whole day in an informal register, carrying it into email is much less jarring.



Corn

The pandemic did something there too.



Herman

The pandemic collapsed the distance between the office and the kitchen table. Everybody was on video from their living room, everybody was typing to colleagues from their couch, and the formality that used to be enforced by the physical office just dissolved. A lot of the emoji-in-business-email acceptance is downstream of that. The norms didn't change because people decided emojis were professional. They changed because the whole environment got less formal and the email went with it.



Corn

There's research on this, on how emojis land in work communication.



Herman

There is, and it's more nuanced than either side of the argument wants. The consistent finding is that an emoji increases perceived warmth and can decrease perceived competence, and the size of both effects depends on the context and the status of the sender.



Corn

So a manager sending a smiley reads as warm. A junior person sending a smiley reads as...



Herman

Reads as less serious, in some studies. There's a real asymmetry. And there's a finding that in more formal contexts — a first contact, an external email, anything where the recipient doesn't know you — the competence penalty gets larger. In an established relationship, the warmth benefit dominates and the penalty mostly disappears.



Corn

So the emoji is doing different work depending on who's reading it and what they already think of you.



Herman

Which is why the advice is always "know your audience," and why that advice is useless, because you don't always know your audience.



Corn

Japan. Daniel specifically asked how the adaptation has gone there, since they've had them the longest.



Herman

Japan is the interesting case, because they've had the longest exposure and they've also got the most rigid formality structure in business communication. The keigo system, the honorifics, the whole apparatus of hierarchical politeness. And emojis sit awkwardly inside that, because the formality system is about calibrating distance and emojis are about closing distance.



Corn

So in a hierarchical business context, an emoji is doing the wrong thing.



Herman

It's doing the wrong thing, or at least an ambiguous thing. So the pattern in Japan is that emojis are heavily used in personal communication and much more restrained in formal business communication. Not absent — there's plenty of emoji use in Japanese business chat — but the rules about which ones and when are tighter.



Corn

And the same warmth-versus-competence trade shows up there.



Herman

It does, and it's arguably sharper there because the baseline formality is higher. The further you are from the baseline, the bigger the signal you're sending, and the bigger the signal, the more it can go wrong.



Corn

Daniel mentioned that some people find them immature.



Herman

Some people do, and the immaturity read is real, but it's not really about emojis. It's about what emojis signal — emotional expression in a channel that was built to suppress it. The people who find them immature are usually people whose professional identity is tied to the restrained register. Which is fine. It's a legitimate preference. It's just not a permanent one, because the register itself is drifting.



Corn

The efficiency argument cuts the other way.



Herman

The efficiency argument is the strongest one for emojis, and it's the one Daniel makes himself. A single character that replaces a sentence. If you need to convey "I'm not upset about this" or "I'm joking" or "this is fine," an emoji does it in one keystroke where the words would take a whole clause, and the words might not even land right. Tone in text is hard, and emojis are a tone-marking system that the reader already knows how to parse.



Corn

The case for them is that they're doing disambiguation.



Herman

They're doing disambiguation. That's the honest technical description. They're a low-cost way to add the paralinguistic information that text strips out. Which loops right back to Kurita's original problem in nineteen ninety-nine. Same problem, bigger screen.



Corn

There's a knock-on effect here that I want to get to, which is that once you standardize a visual language, you inherit all the representation questions that come with it.



Herman

You do. Skin tone modifiers, gender variants, disability representation, the whole set. Once emojis are a global standard, the question of who gets to be represented in them becomes a real political question, and the committee has to answer it, and every answer is contested.



Corn

Which is a strange place for a smiley face to end up.



Herman

It's a strange place for a smiley face to end up. And it's the same dynamic as any encoding decision — the moment you standardize, you're making a claim about what's normal, and somebody is always outside the norm. The Unicode Consortium has been through this before with scripts, with the whole question of which writing systems get encoded and which don't. Emojis are the same argument wearing a cheerful face.



Corn

The face with tears of joy was Oxford Dictionaries Word of the Year in twenty fifteen.



Herman

It was, and that's the moment the culture admitted emojis were language. Not a language, but doing language work. A dictionary naming a pictograph as its word of the year is a strange sentence, and it was the right call, because that character was doing more communicative work that year than most words.



Corn

The visual design differences across platforms are still there, still causing misreads.



Herman

Still there. The peach, the eggplant, the folded hands, the smiling face with horns. Every one of those has a platform-specific reading that diverges from the description, and every one of those has caused a real misunderstanding somewhere.



Corn

The standard is universal and the meaning isn't.



Herman

The code point is universal. The meaning is a negotiation between the sender's platform, the receiver's platform, and the shared culture they're both operating in. Which is a lot of moving parts for a character that's supposed to be simpler than a word.



Corn

The standard keeps growing.



Herman

It keeps growing. Three thousand six hundred and counting, with a pipeline that takes years and a demand curve that takes weeks. That gap is not going to close. It's going to be managed.



Corn

Alright, I think we've got the shape of it.



Hilbert

The four-byte thing isn't quite right.



Corn

Go on.



Hilbert

Most emojis are three bytes in the encoding, not four. It's the ones above the basic plane that take four. The code point is what's above the range, not the emoji itself. I had a job where I had to explain that to a room of people twice.



Corn

Twice.



Hilbert

Twice. Early two thousands, I was doing a stint as what they called a character encoding consultant for a mid-sized Japanese electronics firm. They were localizing a flip-phone interface for the American market, and they brought me in because the American team kept wanting to add smileys that weren't in the original spec.



Herman

What was the objection?



Hilbert

The Japanese engineers had a binder. A physical binder, color-coded, with rules about which emoji could be used in which context. Business memo, personal message, service notification, each one had its own allowed set. You couldn't put a winking face on a business memo. It wasn't in the memo set.



Corn

There was a memo set.



Hilbert

The American team wanted to add a whole new set of faces because they thought the existing ones were too reserved. And the Japanese engineers were horrified, because from their side, the whole point was the restraint. Adding faces wasn't an improvement, it was a loss of control.



Herman

The localization failed on cultural grounds, not technical ones.



Hilbert

It failed on cultural grounds. The American team thought they were making it friendlier. The Japanese team thought they were making it unprofessional. Nobody was wrong. They just had different ideas about what the interface was for.



Corn

And the binder.



Hilbert

The binder was the artifact. It was the whole argument in physical form. Every rule in it was a decision somebody had made about what a given context could tolerate, and once it was written down, it was policy.



Herman

Do you still have it?



Hilbert

I don't know. It's in a box somewhere.



Corn

You lost money on this, didn't you.



Hilbert

I bet a colleague that emojis would never catch on in the US. I figured Americans were too literal for pictographs. I was wrong by a lot. That's the only bet I've ever lost that I'm glad I lost.



Herman

The binder is the part I keep thinking about. A color-coded system for emotional appropriateness sounds absurd, and then you realize it's exactly what every platform is trying to do with reaction sets and custom emoji policies.



Corn

It's the same problem Kurita had, just written down and enforced.



Hilbert

I have to go. I'm meant to collect something from a place that closes at six, and I'm not going to make it.



Corn

The standard is universal, the meaning is negotiated, and the negotiations never stop.



Herman

The negotiation is the interesting part. The code point is just the ticket to the table.



Corn

One thing I'll be watching is whether the Consortium ever moves on visual consistency — whether they ever try to define the artwork, not just the code point. My guess is they never will, because the moment you define the picture, you're picking a culture's version of it, and that's a fight nobody wants.



Herman

The demand isn't going to slow down. Every year there's a new set of proposals, and every year the committee says no to most of them, and every year the custom emoji layer absorbs the overflow. That's the equilibrium we're in.



Corn

Thanks to our producer, Hilbert Flumingtop, for keeping the show running and for the binder story.



Herman

Check the show notes — we've got links to the Unicode emoji count page and the research on emojis in workplace communication.



Corn

If you've got a deployment horror story involving a four-byte character, or a strong opinion about the winking face, email us at show at my weird prompts dot com. We read them.



Herman

This has been My Weird Prompts.



Corn

The human-AI collaboration podcast. We'll be back soon.