

What's new in Chrome [YuMdsHIXatY]_原文

2026年06月08日 15:14

00:05

Hello, hello everyone, how are you doing? We good? Will so at the start, did you hear the Scouse accent or the Liverpool accent in the keynote when they were doing? Yeah, 45 minutes of that now it's going to be brilliant and I'm Paul, I lead the Chrome Developer Relations team, it's great to see everyone here. I'm thrilled to be back. I was here a year ago talking about what's new in Chrome, but before we start, I actually going to show you a new demo that we've got that highlights two of the Apis that we're really excited about web MP and HTML in Canvas, so can we just switch to the demo, please?

00:39

So this is my childhood bedroom 30 years ago when I was starting to learn how to program, I had all these books.

00:46

Actually, this is a lie. I didn't actually have these books because only HTML was the thing that was actually around at the time. I was a pearl programmer when I was actually starting things. Anyway, brilliant, hello. So anyway, but I did have this browser, this thing that I could browse the web and find new things. I could maybe play some games. Now, this is the HTML and Canvas bit, right? We've got a 3D experience. I'm terrible at games at the moment, you know, I could then browse the web, know selected text, and do all these really fancy things around the web.

01:18

I found a lot of new friends when I was actually browsing and on the internet at the very start of things. And then I also had this slide deck. You, I wanted to be a web developer and I wanted to be a developer advocate, even though the role didn't exist at the time. Anyway, the thing I want to show you here actually, is I developed these slides. I'm not going to present. These aren't my actual slides. You're lucky today. But the thing I like about HTML in Canvas is the fact that we have these very related experiences like this.

01:45

This is a full 3D experience, but we have the browser and the content just embedded in, and so we can do things like translate to Welsh. Does anyone know any Welsh? Now? I do a little bit. I live in Wales and I'm learning at the moment, and then we can even do things like find webgpu, like things that we've never been able to do inside a canvas based experience before.

02:08

We have access to the DOM, I think this is just incredible anyway, so you jump forwards 30 years, right? And thankfully I'm no longer living with my parents. We have these agents roaming the web and doing things. And in Chrome, one of the things that we really want to do is make sure that the web can participate in these agentic experiences. So I'm just going to show web MP I don't know whether you can see that properly at the moment. But this experience here has exposed a number of tools to essentially agents at this point.

02:40

And so what I can do is I can do a single one. I can just say, oh, command a turn off the light. It might ask me which lights, but we'll see which one it does. So there we go, turned off the lights. That's really cool.

02:53

But you know, when I was a kid, my mom hated me on the computer in the evening. And so what I'd do is I'd maybe turn off the computer and then started again once she had gone. So let's see if it works. And you say, there we are, the computer. I can maybe zoom in and we rebooted. And so we've had this like multi-step flow of turning the machine off and turning it back on. But anyway, this is just a simple demo. We can just go back to the slides if possible.

03:20

I just wanted to kind of show you some of the things that were kind of possible with Htl and Canvas and also web MP. I'm very excited about them.

03:28

But this is the early days, right? When I started my development journey, this is what my environment looked like. This any Q basic people, brilliant if you're not old enough to remember. This was before modern ldes, before we had language servers and syntax highlighting, like in line debugging. We just had these experiences. And there's a fun fact, right? I actually nearly quit programming before I even started because I didn't realize American English was a thing and that you spelt color incorrectly without a U. But anyway, I learned what to do in the end. It literally took me a week debug in to work out what a syntax error was for it, but I got over it at least.

04:03

Anyway, you jumped forwards 30 years is, and now my development environment is it's pretty damn similar. It's a text based experience, but the thing here is the actual strides that we've made in development environments and development processes. We've just taken massive leaps, and especially within the last six months.

04:22

So our primary goal in Chrome and why I do what I do is to get everyone ready for this next generation of web development. And the way I look at it, at least anyway, there's like three different components. One, we want to make sure that your sites can participate in this agentic future. And the web is a first class experience. The second is that we want to make sure that your tools can also kind of understand modern web capabilities and how to build for a modern web baseline, even if the models and the tools that you use don't have that knowledge built in. And the third part is we also want to kind of get you ready for the years ahead as well, right?

04:58

We are going to talk about some of the things that Chrome is introducing as experiments in a lot of cases where we would like a lot more feedback. But I think it's kind of nice to kind of see where the platform is going so that you can kind of plan ahead as well.

05:11

Now, you're going to hear agents quite a lot. I know one person is counting how many agents, how many times say, hey, how many have we got? Two agents, 3, brilliant. Anyway, so we live in a world where people can start to get things done just by using their voice, the natural language, and this to me marks one of the most significant shifts in computing since the invention of the web.

05:33

Now with the web, most of the platform shifts actually come quite slowly. So since the invention of the iPhone, like it took us nearly a decade to get from building desktop first experiences to mobile first experiences, and it we had this technology to turn the browser into a mobile based experience, but the industry had to move and it moved very, very slowly. And it took a huge amount of effort for developers to redesign their experiences. But when I look at kind of where we are today, over the last six months, we've seen a massive, enormous shift in our role as developers. And I think it's going to happen at a pace that is much faster than the mobile transition.

06:12

Mobile, the web, is transforming from a medium where users like browse and then take action to one where it also has agents to kind of let them Dele people delegate their work for them to be able to complete their tasks. So just like when the web worked well on iPhone or it worked on iPhone, but it wasn't optimized for mobile, today's web works for agent flows, but it's not yet optimized. But the great thing is many agents can just analyze the site either by looking at screenshots or understand in the DOM. And one of the things that we like about this is it really just requires us to recommit to semantic HTML, ensuring accessibility and just recommitting to the kind of the basic principles of modern web development, things that arguably we should already all be doing as developers. But user intent has transformed. It's not single interactions, its actions across multiple phases of a site, maybe across sites as well.

07:08

People want to find the hotel for the upcoming tech conference, maybe integrate it with their colleagues, restaurants maybe. I was told about this. People like to find kombucha bars. I literally learned out what kombucha was this week, still not quite sure. Anyway, north Wales, we don't do kombucha anyway, so the bit here is agents are being able to create these plans that are way more complex, that are way more powerful than we've thought about before. But it really requires an integrated approach, like both in the browser sense, like how the browser is, how we're going to reason about changing the browser, but then how do we help you get your sites evolved to get ready for these multi-step agentic approaches? So obviously the browser is evolving.

07:50

Earlier this year, we introduced Gemini in Chrome and it enables users to chat and multitask without switching tabs. We've created this browser, a system that can navigate sites, summarize content, it can create quizzes, books, help you find the right sleeping bag, and those types of things. But it can do this because web pages describe the function of what is on the site and the function of the site. And so our tools can understand.

08:15

And this gets back to the point, is if we can kind of actually say to the tools that we're using, this is a well designed side. This is the structure. We can help the agents and the tools achieve the users' goals.

08:28

But if you go one little step further, just like how spreadsheets turned everyone into programmers, I think skills like things like skills in Chrome, where everyone can instruct the browser has the same ability as well. So with skills in Chrome, you can browse the library of ready-made skills already. You can get it to kind of like pull out product specs. I'm told protein macros are a thing. I actually don't know what protein macros are as well, extract them from the pages. You can summarize long documents to find what you need, but then you can take those charts and then turn them into essentially skills which are repeatable, remixable, and then you can run them across a number of different websites.

09:10

And I think this is a new level of experience where we're going to enable people to be more like a developer, just at a different level of abstraction. But the good thing is, and this is where I get onto the very developer centric things, is llds can infer the intent of a page by scraping, like scraping the images, scraping the DOM, guessing the clicks. But behind the scenes, it's kind of a little bit hard and opaque. And as developers, we really do like a lot more control. Like we like to be able to guide our tools around our sites, right? And this is where web MP comes in.

09:42

Now, obviously, broadly before web MP, there's a lot of difficulties that emerge. Our Uis that we build often reflect many different human constraints, brand guidelines, UI mandates, regulation. In a lot of cases, our tools don't need to deal with the same kind of experiences that people, people who will browse your site too. And I do think that this is web MC p can help. Now Webm CP is a proposed specification at the moment, and the goal is to transform HTML forms and JavaScript functions into tools that your agents can use and that can make it a lot easier for you to participate in this agentic ecosystem.

10:20

Now, web MP, actually, I have a question. You all know WebMD. Yeah, you're I know you don't know, I speak, so web MTP, our goal here is to integrate many of the learnings with the model context protocol, but however, we're only focusing on the tools part of MCP and instead of having to expose an MCP server or a Cli and then trying to work out how to get that to integrate with your website, you're just exposing your current browser tab and then that is able to share a list of tools that the agent can then interact with on your site and so in the demo that I showed earlier, I exposed the light switches the actual machine, the browser, so you can change pages. They were all the types of tools that I exposed to the agent now we have in a real world example though that was a fun demo, imagine you're running a store and the user wants to kind of buy some shoes, like size 10 shoes that are brown maybe or or whatever In this case with Webm CP, you would expose a number of tools like search, add to cart, maybe check out. It depends on where you want the user to actually interact with, and you would expose these via either a form or HTML JavaScript at this point and then simply the agent should be able to work out which tools to use, which ones to execute, and then get to the point where hopefully that the user's task is complete. Now, the great thing

is about this is that we're able to use the existing semantics of modern web development. Everyone kind of pretty much here knows how to build a HTML form well, we can just add a number of extra attributes onto the form like a tool description, which describes to the agent what this form does and how to interact with it, and then some extra tool parameter descriptions about how the agent should fill the form out. And so, for instance, if you have things like maybe a star rating, you would say this is a star rating or there's a description or some comments, you get the ability to kind of explain to the agent through a normal form with HTML attributes how they should interact with it, which I think is incredible.

12:20

If you want the imperative API, we have navigator, domo, context-transcending, and again, it's a similar type of shape. You give it a name so that the agent knows what tool to call. You give it a description which describes the intent of the actual tool, and then an input schema, which describes the shape of the data that will go into the function. And then when the agent chooses to match the user's intent with this tool, it will then execute the function with the data kind of passed in as well.

12:49

I think this is just a very kind of powerful primitive that we have, where we have both HTML and also JavaScript ways of actually engaging in this agentic ecosystem. But the one thing I really like about this one, obviously, it's actually, I think it's actually quite, I don't want to say simple, but it should be relatively.

13:05

I'm going to say it, it's relatively simple to start. Think about how to integrate into your experience at a technical level, maybe at a wider level about your site architecture. It's a little bit harder. But the real powerful thing about this is because it's running inside the browser, it's running inside the context of where the user is. So you have things like sign in state and other cookies and local storage just available to these tools, something that you don't really get from other kind of tool based MCP based systems where they're not actually running inside the browser. Now, obviously, this all runs within the security boundaries and the security context of the browser. But I think this is a very powerful way of starting to kind of integrate within the ecosystem.

13:48

So I just want to show you a quick demo of one of the people that we've been working with. Now, this is an experiment. It's not a production level system, but Expedia have been doing it some integrations to try and integrate web MP into a hypothetical flow for them. So I'm just going to talk through the demo.

14:04

Now. We start off, it's hard to see a searcher that says find a hotel close to grammar on 4 July. Gemini will then create the task, the plan, the user accepts it, and then they have a number of different tools on the system like search, login, update, login filters, and then navigate to property. And the browser is working out which tool to call. And so in this case, actually, it's called some tools to update the search, but it's also do an actuation on the page so they don't have an update date filter, but it's done the actuation. It will then work out. Well. Actually, we think we want to send the user to this hotel, then we go, it's kind of worked, and then the user has the ability to either confirm or in some cases, the agent might choose to check out.

14:47

Now this is just an experiment. This isn't in production, but it gives you an idea about how these things might work in the future. And I think it's quite exciting to see, especially once we think figure out kind of performance and we work with on the security primitives even more.

15:01

There are a whole set of other people in the industry who have also been experimenting with web MP and given us a lot more feedback around this as well. Now, web MP is designed to be an open specification and we are working across the industry standards bodies. But I will say it is still very experimental at the moment. Now Gemini and Chrome will also start to work out how to integrate this so we can experiment even further. And we're running some of these early tests with some of these partners as well. But if you want to actually stay on top of this, you can register to at this Url for the origin trial. Now we're expecting this will land in Chrome in Chrome 149, which if I get my dates correctly, it's about June 2, So it should be available around June 2 for you to start experimenting with as well.

15:48

And the goal here is once you're in the origin trial, we really want your feedback to shape out like how this API should work. Does it work for you? Does it work for your use cases? And then we can work out how we can actually start to help other developers architect if this does become a standard. Now, that's agents kind of transforming how users will browse.

16:11

I just want to spend some time talking about how the tools that we use will change how we build software, at least anyway. So on a personal level I'm a manager in the team prior to maybe the last six months, I could do development in my free time. But now with some of these tools I'm building every single day. And the weird thing is probably like many people, we're having conversations with a tool that kind of works out what we want to try and do. I don't know about everyone else, but I actually have a microphone, I just use my voice to literally talk to my large language model. I ask kind of what I want to try and do, and then I have a conversation about how we're going to achieve it edge cases. And that's before I then get to get it to start to work.

16:49

And now I'm definitely far from the only developer doing this. I actually ran some, this is from my blog, but I run some analysis across GitHub to understand how many people are using all the different tools to create commits. It's going like crazy. Millions and millions of developers are submitting millions of pull requests every day with agentic tools. And so this is one of the trends that we want to make sure that we're able to help developers with.

17:13

Now, the tension is obviously the tools are advancing at a pretty rapid pace. Our workflows are changing.

17:21

It's at conflict with how the web platform works. If you take one example of a browser, so Chrome, it updates once every 4 weeks, I think in a couple of months, it's going to update once every two weeks, but the models are trained on a version of the web from a year ago in some cases. And so how do we deal with that tension? So today, what happens if I want to build some technology, some examples like my HDL and Canvas demo, I would load up the context with the samples, the examples, the spec, the explainer, some unit tests, sometimes submit that in, and then they're actually pretty good at building. But I have to know about all of that context. And if a general developer, you're not staying on top of things, it's my job to staying on top of all these things. Every other developer is just like I've just got a job to do.

18:08

I need to implement some functionality for my business. So the real question for me comes, can Chrome help in this space? Can we take the decades of web development experience and help you solve and reason about kind of the modern platform? I really want us to get to a point where we're able to build for the web of today, not like the web that was. So I think the answer is yes, we can definitely help in this space.

18:31

So one way whether that we're doing this is with the preview that you saw in the keynote, which is with modern web guidance, which is a skills pack. And the goal here is to give your tools a validated, sorry, there we go, a validated evergreen repository of the best ways to kind of improve the compatibility and implement what is and what we think is the modern web platform. You can install it with a pretty simple universal script, `npx modern web guidance install`, which means that it should work in any IDE or CLI that you have that supports skills.

19:07

And we've built up a number of different skills as well. We have high level skills around web performance, identity, and security. And then there's lots more guidance around specialized skills across, I think it's over 100 of the different common use cases that you as developers have to implement in your sites every day. And then we're able to match that with the modern equivalent on the platform.

19:28

We aim for this guidance to also be kind of very kind of understanding of baseline is probably the easiest way to assay it. So even if a feature that you're using isn't baseline widely available, be able to give you, we should be able to make you more confident and give you the guidance with a set of fallbacks and maybe progressive enhancements as well. So the goal here is that you should be able to rely on the guidance that is coming from these skills packs to be able to be usable by your users. But you also need to understand who your users are. And we're also launching a new tool, an integration into Google Analytics, so that you can understand from your actual pages, your baseline target, and work out your baseline target based on how your users are interacting and visiting your site. And so we can go to baseline, I think baseline check of dot chrome dot de dot.

20:17

Now, I think in all broadly, this is going to enable our coding tools to properly use modern features, ideally to follow best practices from industry experts within Chrome and across the industry and to help you use

platform features in a way that solves your problems without you having to reason about the entire shape of the web platform.

20:37

Now for me I'm actually very excited about this because llds are going to give us or are given us a way to experiment at scales that we've never been able to, but I do want to make sure that we're able to build experiences that are high quality, safe, and then don't come at the expense of cross browser compatibility. And I think this project with the high quality guidance kind of built in and maintained by our team, we're definitely on a step and a path into that space that space If you do want to learn more about modern web guidance.

21:04

Phil who is just sitting there, he's going to be the next talk in this room. And if you're on the live stream, the next talk on the live stream. And he's going to do a deep dive into using modern web guidance and how it works.

21:16

So returning to my conversations with my machine, large language models as I build with agents. I'm not just sending them off to work, right? I actually am working with them in a really, really tight loop. And my tools, they do output the code essentially that I requested, and they run it against a battery of test, but they don't really have a deeper understanding of how the page is running, like what is actually on the page, what is rendered, and how it's working inside the browser. This ability to really deeply inspect pages inside the browser, it's been a developer superpower for years, right? Like we all use prom dev tools or Safari dev, all the different dev tools, but now we want to make sure that with dev tools for agents, your tools also have the same ability to actually deeply inspect your pages.

22:02

And so one example about this is we're going to give you direct access to things like console logs, network traffic, memory traces. We give you access to the accessibility tree. And by bringing this runtime data into your tools, you're able to perform very sophisticated diagnostic tasks like performance profiles, heap snapshots, memory debugging, and then automated Lighthouse tests as well.

22:24

Dev tools for agents. It's an MCP server, so you can just install it the same way that you install any other MCP server into your environment. And then really all you have to do is then say, prompt the agent to run, say, a comprehensive performance analysis of developer dot chrome dot com. There are performance issues and it will tell us what we need to fix there as well. And the goal here is really designed to kind of transform the debugging process and aiming to reduce the amount of manual intervention and also the manual fixes that you have to do because you can get the tools to then subsequently fix the issues that it finds when it runs something like a Lighthouse audit as well. Now, when we first shipped this, I was like, great, this is done. We don't need to update it anymore. Well, it turns out the team's like, now we have to update it.

23:10

We've got loads of ideas. And so we've done a number of different features, and I'm going to go through them pretty quickly. So the first one is we've introduced skills into the DevTools Agents project, and that's going to enable the tools to handle more complex tasks. So if it needs to do, say, memory debugging, we can instruct it to do heap snapshots and then understand how to actually reason about what is in the heap snapshots.

23:34

We have a Cli tool. So if you're on the command line or you want your agents to be a little bit more token efficient, you can use the Cli version of the tool. And we also have a TypeScript API if you want to integrate it into something like your continuous integration environment. We also have the ability to connect to a running instance inside Chrome so that the idea here is that your agent can then pick up from where you left off, which I think is pretty cool, and obviously, people are using it for things like Open Claw. We can also support multi-agent workflows. Now, this is great, so you can connect to multiple instances of Chrome, and this is important if you're creating multi user systems or multi role based kind of systems as well, so you can connect to like two or three or four different versions of Chrome at the same time. And one thing as well is that if you're inside Chrome DevTools and you're inside the network panel, you can say select an element in the network panel and then ask your agent to analyze why it is selected as well. So we have this really nice hook-up between DevTools, DevTools for agent, and then obviously within other things as well, And then the final piece is we also now have the ability to automatically install and debug Chrome extensions.

24:40

So if you're a Chrome extension developer, you can now inspect, you can do the install, you can inspect the service worker, the side panels, the pop ups. It's a whole new level of automation for Chrome extensions as well, all powered by DevTools for agents. I'm very excited by this.

24:55

On one hand, we've got modern web guidance. The goal here is to give you access to the best practices to help you build for the modern web so you can build higher sites, and then on the other side Chrome dev tools for agents, the goal is to just give your tools the ability to really deeply understand what's on the page, any issues, and fix them. It's a really nice tool assisted loop for building sites, and obviously I can't stop here loads more to go still as the tools improve just broadly and we're able to build higher quality site, I think it is important to look at what is. Coming to the platform that is also going to enable you to build those higher quality experiences. And so I'm going to give you a very brief overview of what's come to the web platform and also what is going to come to hopefully Chrome soon.

25:42

But one of the things you'll hear me say quite a lot in the team talk about is like baseline. This is a really important component for us because baseline is the cross browser, how all the features that every single browser supports. So it's what you as developers can really rely on. Since last year when I was on the stage, I think 55 new features have become baseline widely available, which are available for you to be able to kind of use and be confident that you can use it with your user base. That's things like the search element or private file system, subgrid off screening Canvas MathML. There's like, oh, 50 more after this as well. But

I think the bit here is that the exciting thing is not just the features, it's the fact that every single browser vendor, it's to push on the platform at the same time to make it easier for every single developer to build for.

26:33

And again, since I was here last year, 44 features as of April 22 have become newly available things that you can start to plan to use in the near future. They are available across all browsers, but they might not be kind of meet your usage requirements in terms of your browser support matrix. Now, I don't have enough time to go through all 44, but I do want to kind of give you a highlight of how some of these things that have come to the platform is going to change how the web feels.

26:58

And actually, before I go to the next slide, I did a little check this morning. We're actually 52 features as of this morning are now baseline newly available.

27:07

So the first step is, you probably heard us talk about this quite a lot, view transitions. We just think this is going to change the way that the web feels. It enables smooth transitions between different view states inside a page and also across pages without the need for complex JavaScript and light libraries and animation libraries. And it's going to help basically improve the perceived performance of the web.

27:28

And to complement view transitions, The Navigation API is now newly available. And the Navigation API, the goal here is to replace like the A in History API inside the browser. And it lets you kind of manage your applications, state all your navigation through one event, essentially, so you can understand whether someone is pressing backwards and forwards. And you can then hook that up with your kind of view transitions API as well. This is going to make web applications just so much easier to build.

27:56

And then the final piece is you hear us talk about Core Web Vitals a lot, largest Contentful Paint and interact interactions and X paint are also newly available. And what this means is that you can just understand how your users are using your site in terms of performance and responsiveness, irrespective of a browser. So if your user is using iOS, you will have these metrics available to you now. And I think this is one of those ones where we're very excited to see what's happening with Core web vitals just for the last couple of years, because it really does help you improve the performance of your site. And now you can understand it irrespective of whatever browser someone's using.

28:35

Now, I mentioned that there's 44 new features or 52 technically a baseline newly available features. It's a lot to keep up with, right? And so with web statuses DoD, you can use this new thing called baseline alerts. And the goal here is to enable you to track when features come into a browser, when they become newly available or also widely baseline widely available, you can get notified via email, RSS, or even webhooks as well.

29:01

And the goal is just to make it easier for you to understand how the platform is evolving so that you can just plan your business and your sites way more effectively into the future. Now, one of the things I do, or a couple of things I love about my job is 1, I get to tinker a lot. I build lots of things at the very kind of for the cutting edge of web development with Chrome. But I also get to tell, talk to developers every single day.

29:24

Now, one of the things or two of the things that people say is, firstly, we just need to be ship better, like we need to be able to ship more easily, understand what's on the platform, make sure the platforms predict more predictable. Don't go breaking the web all the time. The goal there is like that's baseline modern web guidance. They're the tools that hopefully help you ship sites more effectively. And then the second part of this is that you want to understand where the web is going so that you can plan for the next one or two years ahead. And so that's what I'm going to talk about now, all the things that are starting to come into Chrome.

29:56

I kind of want to show you some of the bits, like this is the very tactile part of the web. I don't think you'll hear me say agents one more time, but these are the things that people will experience as they use the web. Now, actually, you will hear me say agents one more time or tell a little bit of a lie.

30:09

So the bit here is like we have two areas or two ways that I think about this. One, how do we put on device intelligence into people's hands? Because the technology is very capable and very good. And then the second is like, how do we actually just help you improve people's experience of the web? So for the first area, obviously, IIDs have changed what is possible with a computer. And obviously, they've evolved from things like text completion and then semantic data extraction. Then people moved on to more summarization, and now people are starting to look at function calling for those kind of like agency use cases.

30:43

Now, the tools I'm going to talk about are designed to kind of help everyone think about these new types of experiences that they want to bring to the platform by enabling more intelligence in the browser.

30:55

Now, the way I look at this is that we do want to make sure that everyone should have access to these tools, irrespective of internet connection, and they should be able to run them privately and then also at no cost in the context of large language models.

31:07

Now, the prompt API, it came to Chrome in 148, and it came with multimodal support as well, which means that you can also analyze, for instance, images and understand what's happening. And this basically means it makes it easy for you to kind of think about these new kinds of experiences that you want to add. But it is only available in Chrome, right? This is only a Chrome based experience. And so one of the things that we encourage people to do is to think about where you can use these features progressively without them being core to your application right now.

31:36

But we do have a number of partners trip dot com com, drupal and temmu who have experimented with this API and also shipped it to production to their users. And so I just want to give you a quick overview of trip dot comma. It's a very short demo, but they integrated the prompt API just to make it easier on the left hand side to navigate flights. If you don't have this prompt API, it doesn't appear. But if you do, it's a nice, simple progressive enhancement, like a nice feature for them. And I believe they like using this technology on that side.

32:06

The prompt API also supports structured output. And the reason why we have this is it just enables you to constrain the output from the large language model. So if you're building an application that needs to understand what's come out and respond to things inside, respond to, yeah, I mean, maybe the caption, for instance, you can basically get JSON out, and then you can just integrate it into your application logic just far more effectively than just like passing strings.

32:31

And we've also expanded language support from English only to include French, German, Japanese, and Spanish, with more to come in the future and kenji's at the back, and I hope Welsh is coming soon. Is Welsh coming, is Welsh come in, maybe we'll see, we'll talk later. Anyway, it's fine. I'm learning Welsh.

32:49

Looking ahead, we're also of expanding what we can do with the prompt API, so we're going to introduce the Gemma family of models. And when we introduce that Gemma, a family of models, it's going to enable us to do native function calling, which is going to enable you to build these autonomous client side agents inside the web. Now, we do want to make sure that these upcoming improvements are rock solid, the performance and the quality and the security are all great. So please do sign up for our early preview program so you'll get access to the latest capabilities. But the ability to help us influence the shape of these Apis into the future as well.

33:25

If you want to learn more about this, Thomas steino has A to build new features using built in AI in Chrome, and I encourage you to watch that because it's way more expansive about what we're doing with built in AI on device. Now, when I think about the web and proven experiences, there's been a huge amount of new capabilities that have come to the platform just in the year last year alone. Now, I get really excited by these because they enable people to experience things that they've never seen before, but they enable to do developers to build things that have been previously impossible to do on the platform. And I think we've taken huge strides in kind of new capabilities on the platform this year.

34:03

Single page applications are now able to correctly measure their core Web vitals performance between routes with the introduction of soft nabs. This is going to come in a slightly later version of Chrome, but it our data indicates that if you have a single page app, your core Web Vital pass rates will go up.

34:19

But I do want to show you something that I think is just going to change the way that we build some Uis in the future that are both performance and also UI wins. And I think it's one of the biggest changes to HTML in a decade. So if you go back to my bedroom and look at the clock on my machine, you can see it updating every single second, kind of what you'd expect from a clock. But the thing is that there's no JavaScript on this page. It's updating every single second. And that's because of the declarative partial updates of that implementation that is coming into Chrome. You can now render updates to HTML out of the order that they were sent on the wire and that they appear in the DOM.

34:54

Now, the clock demo, the clock demo isn't all that useful by itself, but I do think it highlights some compelling potential use cases. So for example, if you have a huge amount of application structure, say a massive menu, you can render it at the end of your page. Let the critical content get rendered first, and then you can patch it back into the DOM just a little bit later. It's available in Chrome 148, and also that you don't need JavaScript for it, which is great.

35:19

I just want to show you this other demo as well. You can think about UI placeholders, like if you have a very slow task for looking up user data frequently, we'd send off a different request. You can keep it on the same request and patch it in at the end of the Http request. And because the way that this API works, it will just update the user interface with the kind of, well, the latest data that's come from there. Again, all without JavaScript. But there's huge amounts of different things that we can do with this API. And this is just the start of some of them.

35:49

Now I want to show you 1 other thing where I think I would like to make HTML the new JSON. And we have a new API that's going to help with that. So if you've been using frameworks, you'll probably know about something called the island architecture. I think it's a beautiful pattern, and it's a beautiful concept, but it's going to be even easier for us as just like plain JavaScript developers to do something very similar.

36:09

So the new streaming API for dynamic markup gives you a revamped way to improve experiences by letting you inject HTML into documents by the stream HTML and stream HTML unsafe methods. Actually, there's a whole family of different API calls in this, but what this really means is you can take the body from a fetch request and then just stream it directly into the DOM. You don't need to do these JSON transforms anymore. I think it's very powerful, very new, powerful, primitive. These Apis have both been available in Chrome since Chrome 148, so pretty recent.

36:42

They're going to change the way, I think personally, they're going to change the way that we build web experience as well, but please do go to this Url, try it out, and give us feedback so we can get it across the platform.

36:53

Now, obviously, over the last year, we've been focusing on a lot on making the web just feel better, giving you way more API if you look at it from an API perspective, just a huge amount of CSS and JavaScript APIs that are letting you build more complex and richer user interfaces. But it really is more than just API and aesthetics. Our goal is to help you build sites that just scroll better, transition between views, better, enable people to interact with their devices in a more physical and intuitive way.

37:24

Now I don't have a huge amount of time to go through most of this, but Yuna and Brahma have identified five core quest principles that are going to change the way I think of how we think about building for the web. And so I would encourage everyone to watch that talk.

37:37

But there is one technology that I think is going to change what we build for the web, and that is HTML in canvas. And I'm going to spend a little minute or two talking about that. So you saw from the demo at the start where this D experience, and we can integrate like a fake web browser into a computer that's HTML and Canvas. And what it letscher is just render standard DOM directly into a Canvas element. And this can be used in a number of different ways. You can imagine D application frameworks, maybe like something like QT or even D engines starting to integrate this so that they can just take the functional benefits of the web's like rich semantic UI layer and have real DOM content that's searchable, translatable, accessible. So there was another inter interactive as well right inside the canvas, but I just want to give you a little preview of like some of the things that you might be able to do with this API.

38:28

On one level, you could take an individual element. This is an input type equals range, and basically make it look very different to what we expect. This is semantically correct, there's the input element of just there, and we're providing a new richer UI layer.

38:44

Now, I don't imagine every single website looking like this. I think that it will be actually pretty over the top, but if you have like a consistent design language that you can't implement right now on a page, you can start to think about, well, maybe I can use something like HTML in Canvas to enhance it, and if the API is not there, it's great. It'll just look like your existing website right now. You can go all the way to those D like experiences by integrating it into something like J, so you can have like 3D car dashboard, interactive books, we have a whole demo station out there, which you can look at which shows you what to do again, all accessible interactable takes advantage of all the browsers features like things like translate and search as well.

39:22

I think this is just a new layer of richness that we're going to enable across the web. And personally, it's one of those reasons why I think people will keep coming back to the web because we have these experiences that you just can't get anywhere else. Now, it is designed to be as simple as possible to integrate, and if you

have play Canvas or JS, you can use it today. And it is just as simple as adding a new material or texture into three. And then the library in the browser just take like handle all of that extra complexity for you as well.

39:50

I just think this is a game changing API and I think it's going to change the way that we experience the web. And if you look at the rest of the ecosystem, this is something that we boss has done. It's hundreds of experiments that are happening right now. If you're on X and you search for HTML in Canvas, you'll just see people tearing web pages, explode in web pages. Again, these are all very fun demos, but it just highlights how expressive this API can be. And there's hundreds more things are happening in the ecosystem right now. I mean, we could, I suppose we could watch, you can go to experience outside. I think it's going to be great.

40:25

I think the bit here though, is if you want to try and see what we can do with this API, we're starting to pull together an awesome HTML in Canvas page where we're trying to aggregate the ecosystems experiments in this place as well. So if you have a Higgs moll in Canvas demo, send us a link. We'll try and get it on there as well, because it's a really kind of beautiful expressive API. Okay, so that was a lot.

40:50

How many times did I say Agent 62? Wow, okay, brilliant.

40:55

You can probably tell I'm quite excited for the future of the web platform. I think the platform is going to evolve for the people who use it. You're going to see Apis like HTML and Canvas, a whole bunch of new UI primitives that just make the web rich and expressive and something that is very fun to use. And then at the other side of things, we do want to evolve the web so that it is a participant in these new platforms, things like web MP and getting into the experiences like the agents and the tools that you're using, I think it's going to make it make the web a first class experience across all these different types of platforms. And I think the real important thing for me and why I'm in developer relations is that we just want to help you do this as well. We know it's hard to navigate of the ecosystem and understand what's going on, and that's where things like modern web guidance comes along where we can give you the tools to help you build for the modern web effectively. So that's it, thank you for coming along. You've been a great audience, brilliant.