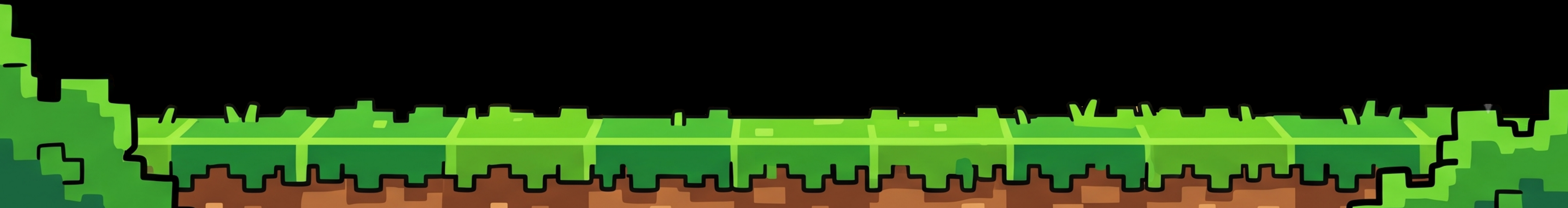


WARP PIPES AND PREFETCHING

Navigating the Web at Mario Speed





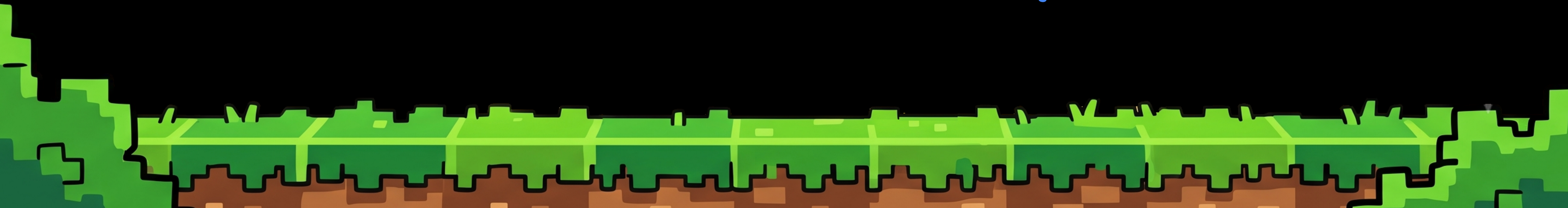


Core Web Vitals

Largest Contentful Paint

Interaction to Next Paint

Cumulative Layout Shift



Largest Contentful Paint

How fast does a website's main image or largest block of text actually appear on the screen?



Interaction to Next Paint

How long does it take for a website to visually react after you click, tap, or type something?



Cumulative Layout Shift

How much does the content of a website unexpectedly shift around while loading?





Speculation Rules API

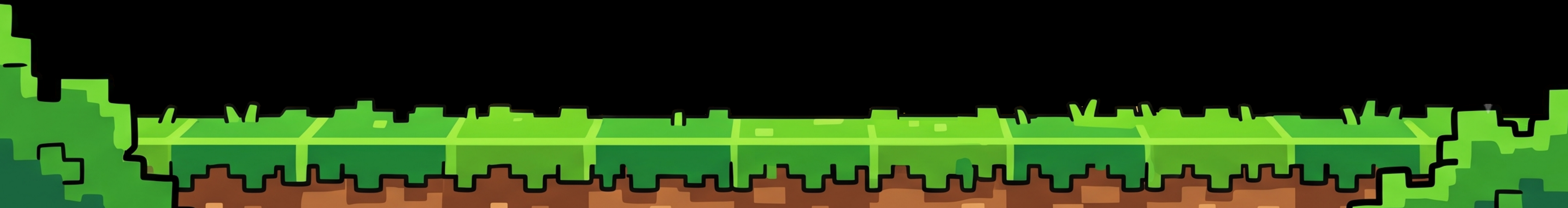
A cheat code to tell a browser to secretly prefetch or preload a page before a user clicks the link, making it appear instantly the moment they do.

VISITOR

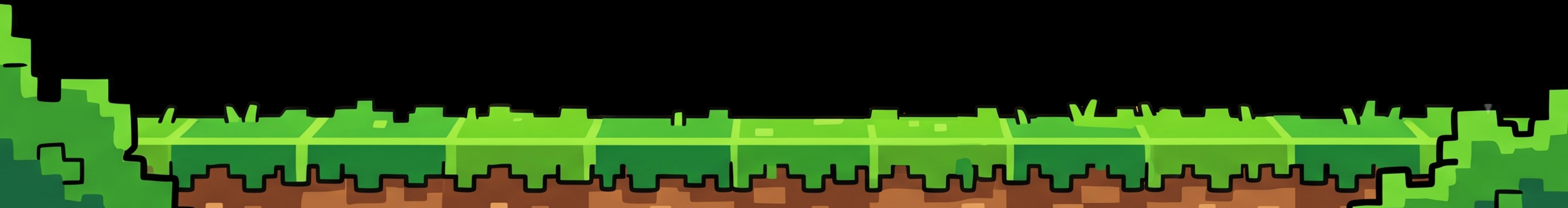
FUTURE PATHS

Prefetch

The browser proactively performs DNS lookups, establishes TCP connections, and downloads and caches specific web resources in memory to significantly reduce network latency for anticipated future requests.

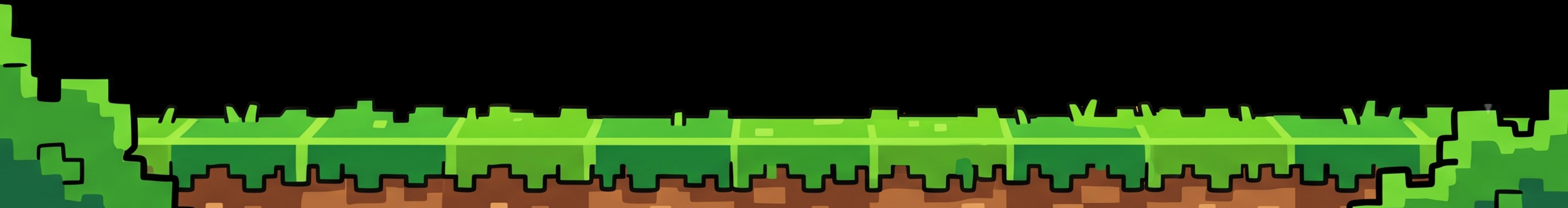



```
<link rel="prefetch" href="/mario.html" />  
<link rel="prefetch" href="/luigi.html" />
```






```
<script type="speculationrules">
{
  "prefetch": [
    {
      "urls": ["mario.html", "luigi.html"]
    }
  ]
}
</script>
```




```
<script type="speculationrules">
{
  "prefetch": [
    {
      "where": { "href_matches": "/*" }
    }
  ]
}
</script>
```



```
<script type="speculationrules">
{
  "prefetch": [{
    "where": {
      "and": [
        { "href_matches": "/*" },
        { "not": { "href_matches": "/logout" } },
        { "not": { "selector_matches": ".no-prefetch" } },
      ]
    }
  ]
}
</script>
```



```
<script type="speculationrules">
{
  "prefetch": [{
    "where": {
      "and": [
        { "href_matches": "/*" },
        { "not": { "href_matches": "/logout" } },
        { "not": { "selector_matches": ".no-prefetch" } },
      ]
    },
    "eagerness": "eager"
  }]
}
</script>
```

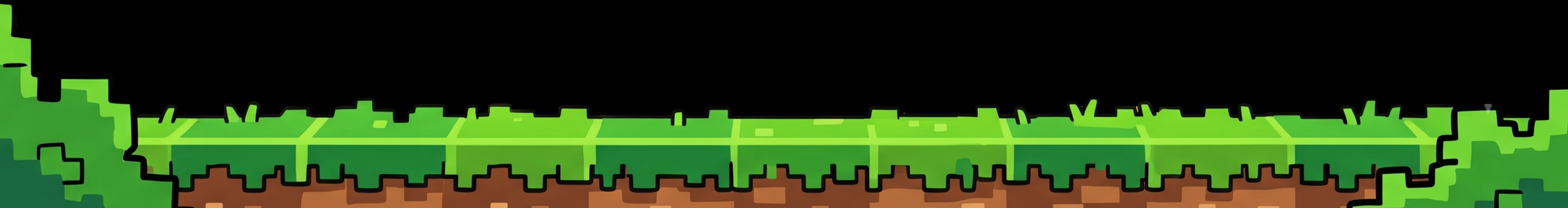

Eagerness

immediate - as soon as the page loads

eager - on hover for 10 ms

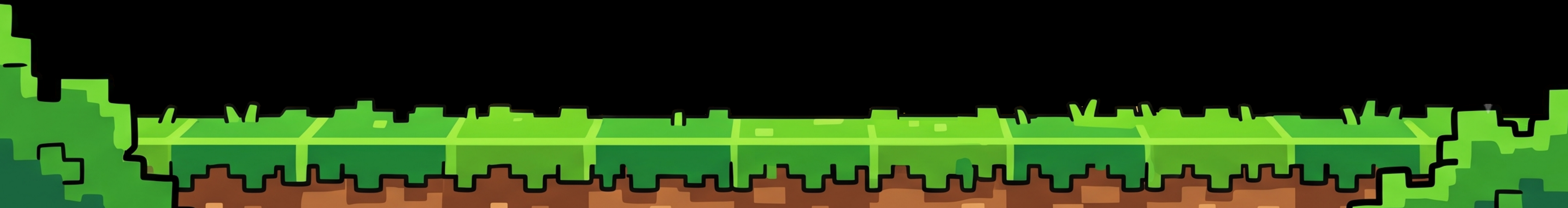
moderate - on hover for 200 ms

conservative - on mousedown



Prerender

The browser proactively fetches a designated URL and executes its entire critical rendering path, including DOM construction, layout, and JavaScript execution, in a hidden background context.




```
<script type="speculationrules">
{
  "prerender": [
    {
      "where": { "href_matches": "/*" },
      "eagerness": "moderate"
    }
  ]
}
</script>
```





```
<script type="speculationrules">
{
  "prefetch": [
    {
      "where": { "href_matches": "/*" },
      "eagerness": "eager"
    }
  ],
  "prerender": [
    {
      "where": { "href_matches": "/*" },
      "eagerness": "moderate"
    }
  ]
}
</script>
```





Mario



Luigi

Elements

Console

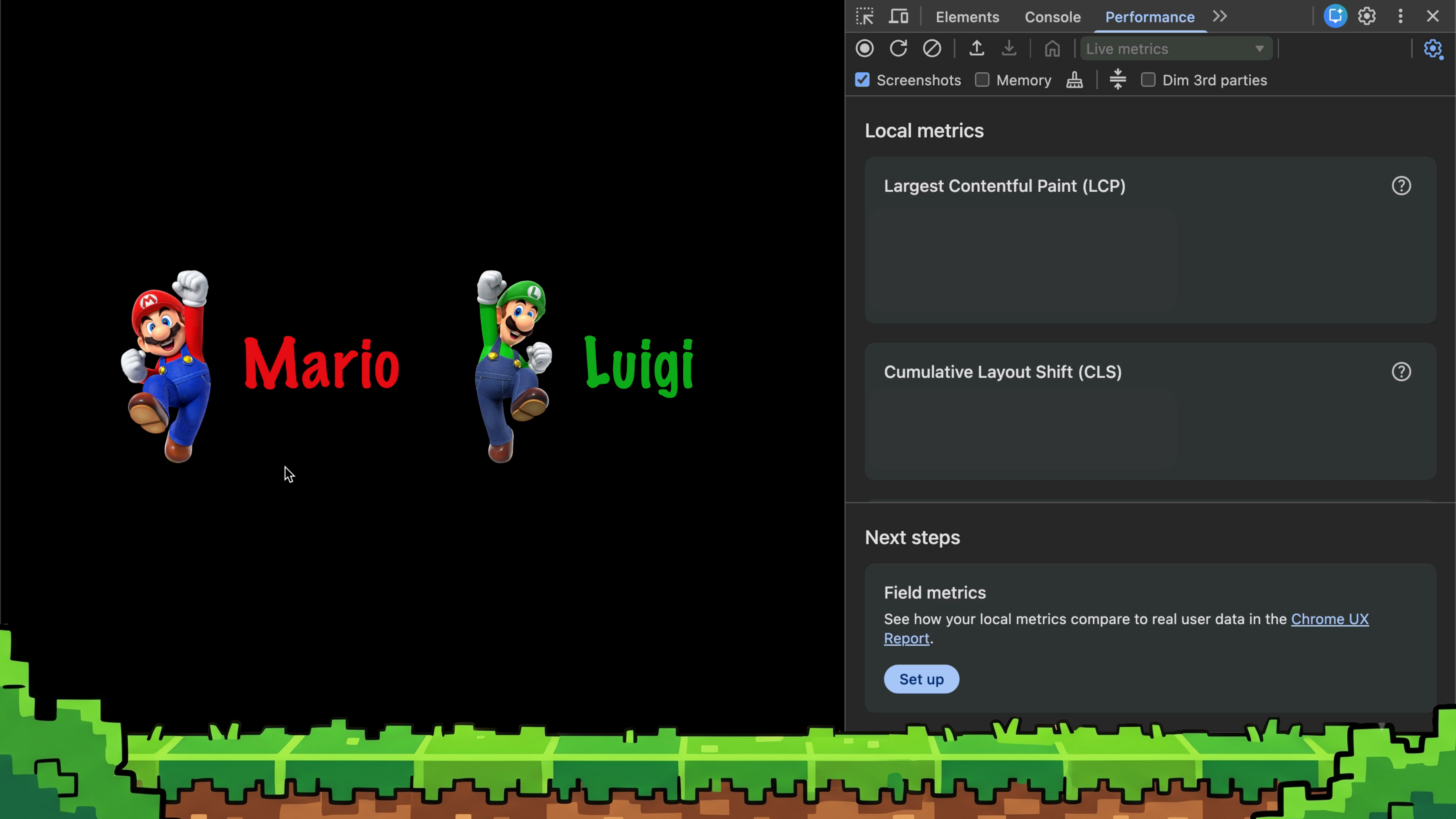
Application >>

Main ▾

	Rule set	Status
Storage - Local storage - Session storage - Extension storage - IndexedDB - Cookies - Private state tokens - Shared storage - Cache storage - Storage buckets	index.html	<u>2 not triggered</u>
Background services - Back/forward cache - Background fetch - Background sync - Bounce tracking mi... - Notifications - Payment handler - Periodic backgroun... - Speculative loads Rules - Speculations - Push messaging - Reporting API - Device bound sessi...		
Frames		



Frames



Mario



Luigi

ElementsConsolePerformance >>

Live metrics

☒ Screenshots

☐ Memory

☐ Dim 3rd parties

Local metrics

Largest Contentful Paint (LCP) ?

Cumulative Layout Shift (CLS) ?

Next steps

Field metrics

See how your local metrics compare to real user data in the [Chrome UX Report](#).

Set up



The screenshot shows the Chrome DevTools interface with the 'Application' tab selected. The left sidebar contains two main sections: 'Storage' and 'Background services'. Under 'Storage', there are icons and labels for Local storage, Session storage, Extension storage, IndexedDB, Cookies, Private state tokens, Shared storage, Cache storage, and Storage buckets. Under 'Background services', there are icons and labels for Back/forward cache, Background fetch, Background sync, Bounce tracking mi..., Notifications, Payment handler, Periodic backgroun..., Speculative loads (expanded), Rules, Speculations (highlighted in blue), Push messaging, Reporting API, and Device bound sessi... At the top right of the sidebar, there are icons for Elements, Console, Sources, and Application, along with a search bar and a close button. The main panel displays a table titled 'All speculative ...' with columns: URL, Action, Rule set, and Status. The table has three rows: /luigi.html Prefetch index.... Not triggered; /luigi.html Prerender index.... Not triggered; and a third row which is mostly obscured by the sidebar. Below the table, there is a message: 'No element selected Select an element for more details'.

URL	Action	Rule set	Status
/luigi.html	Prefetch	index....	Not triggered
/luigi.html	Prerender	index....	Not triggered

Frames

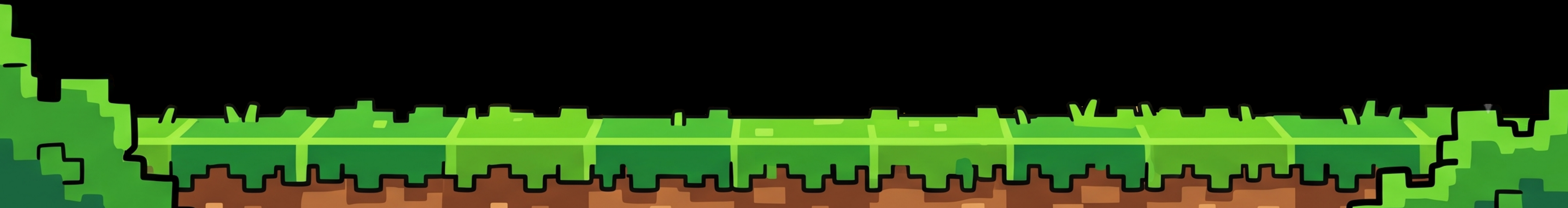


The screenshot displays the Chrome DevTools Application tab. On the left, the 'Storage' section is expanded, showing 'Local storage', 'Session storage', 'Extension storage', 'IndexedDB', and 'Cookies'. Below it, the 'Background services' section is expanded, showing 'Back/forward cache', 'Notifications', 'Payment handler', 'Periodic background sync', 'Speculative loads', 'Rules', 'Speculations', 'Push messaging', 'Reporting API', and 'Device bound sessions'. The 'Speculations' item is highlighted. The right pane shows a table of speculative loads with columns 'URL', 'Action', 'Rule set', and 'Status'. The table contains two rows: one for '/luigi.html' with Action 'Prefetch' and Status 'Ready', and another for '/luigi.html' with Action 'Prerender' and Status 'Not triggered'. The bottom of the image is decorated with a green and brown pixelated border.

URL	Action	Rule set	Status
/luigi.html	Prefetch	index....	Ready
/luigi.html	Prerender	index....	Not triggered

Prerender JavaScript

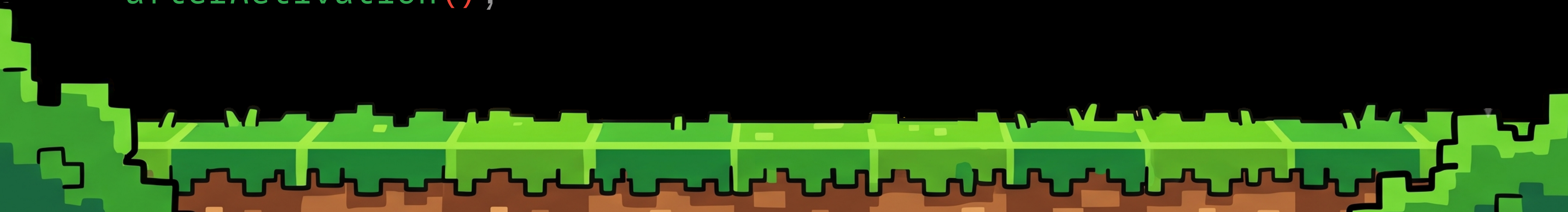
This can completely mess up your analytics by counting fake visits instead of real people, making it impossible to know how many actual humans are visiting your site, where they are from, or what they are really clicking on.



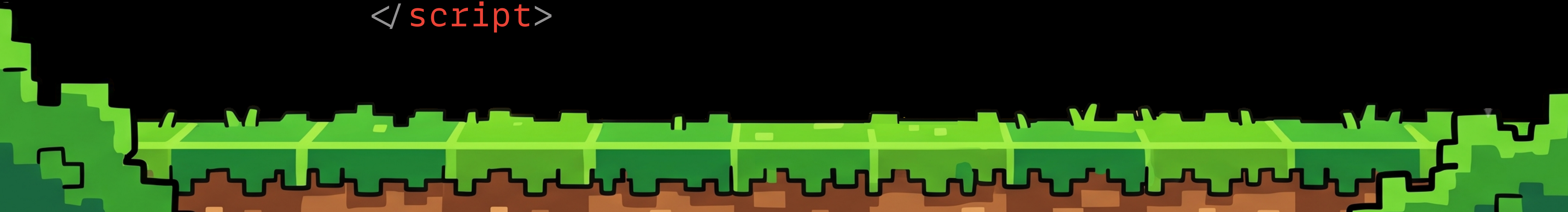

```
const isActivated = new Promise((resolve) => {  
  if (document.prerendering) {  
    document.addEventListener("prerenderingchange",  
                                resolve, { once: true });  
  } else {  
    resolve();  
  }  
});
```

```
async function afterActivation() {  
  await isActivated;  
  // initialise your analytics, ad services, etc.  
}
```

```
afterActivation();
```




```
<script type="speculationrules">
{
  "prefetch": [
    {
      "where": { "href_matches": "/*" },
      "eagerness": "eager"
    }
  ],
  "prerender_until_script": [
    {
      "where": { "href_matches": "/*" },
      "eagerness": "moderate"
    }
  ]
}
</script>
```



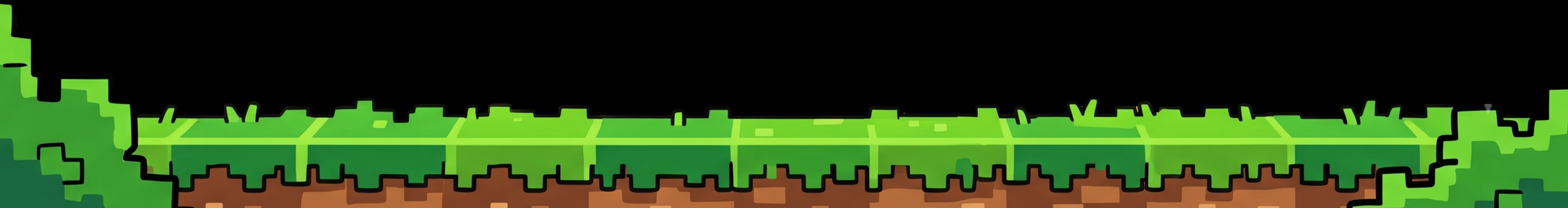
No Speculation

Save-Data

Low memory

Energy saver

Cross-origin iframes





A circular illustration featuring Super Mario Bros. elements. At the top, a Piranha Plant with a red body and white spots is positioned next to a green pipe. Below the pipe is a yellow ? block and a brick wall. A Goomba with a red cap and white body is visible. The center of the circle is a large, glowing yellow star. The bottom half of the circle is composed of blue and red brick walls. At the bottom, there is another green pipe, a Goomba, a Piranha Plant, and a yellow ? block. The entire scene is set against a black background with several white stars.

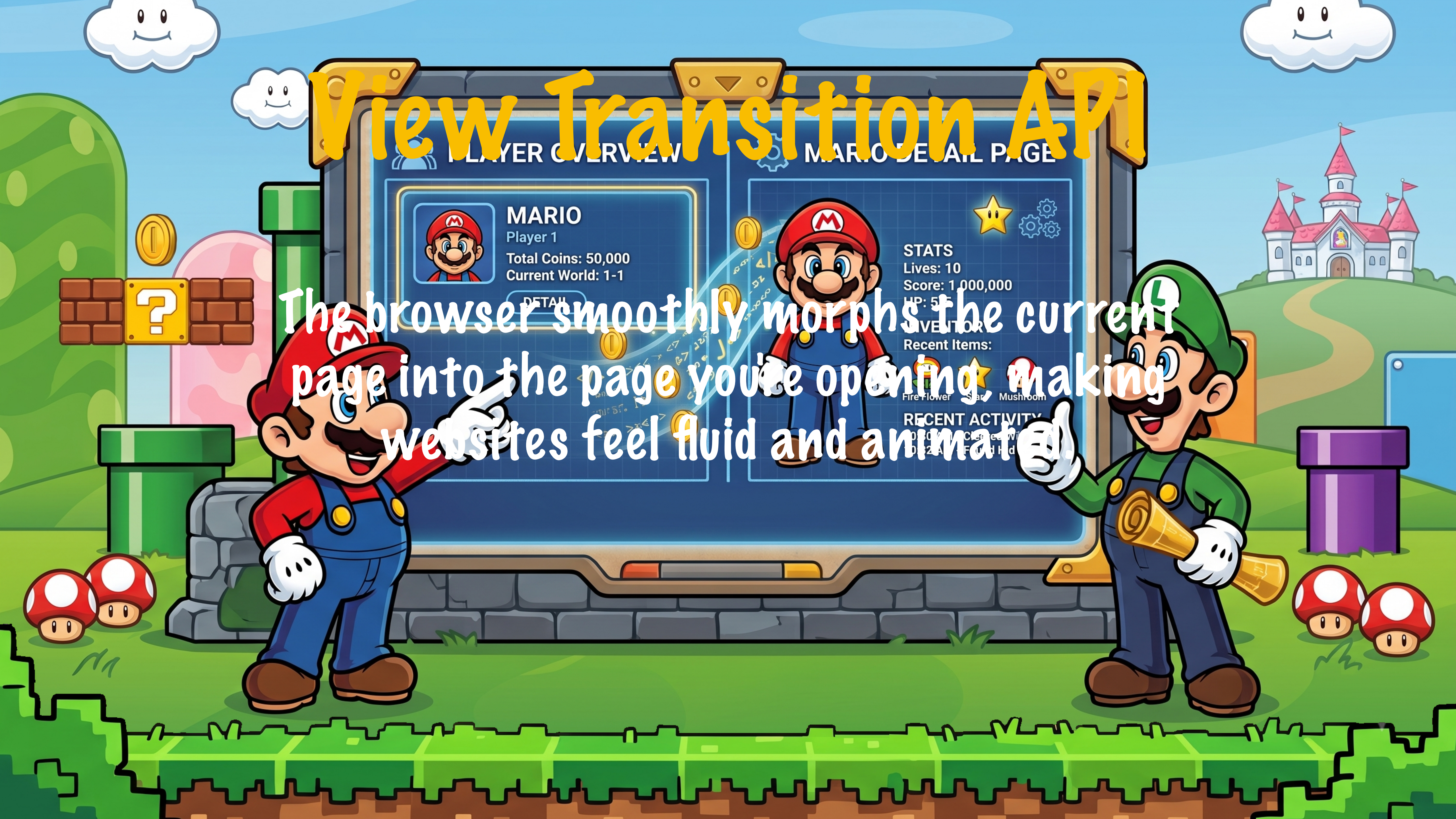


26.2

December 2025

View Transition API

The browser smoothly morphs the current page into the page you're opening, making websites feel fluid and animated.

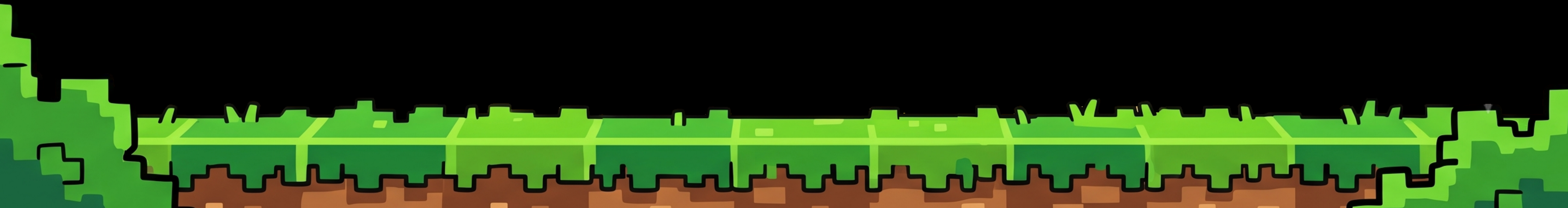


Cross-document MPA

MPA is a stupid word for website

Full page load from server (no SPA)

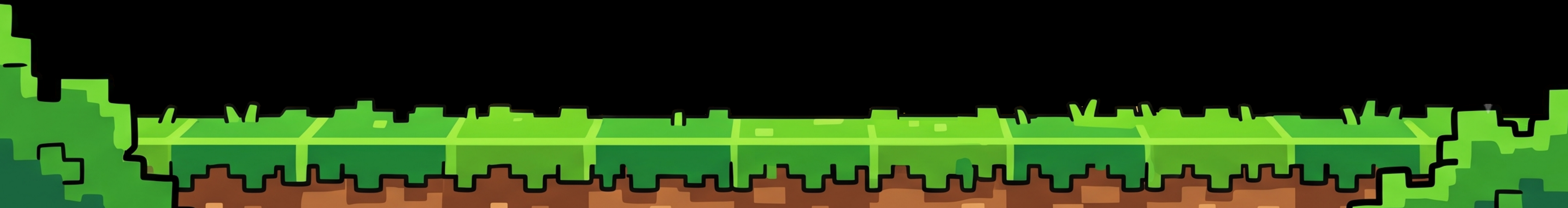
Same-origin navigation only



Same-origin

https://www.example.com:443

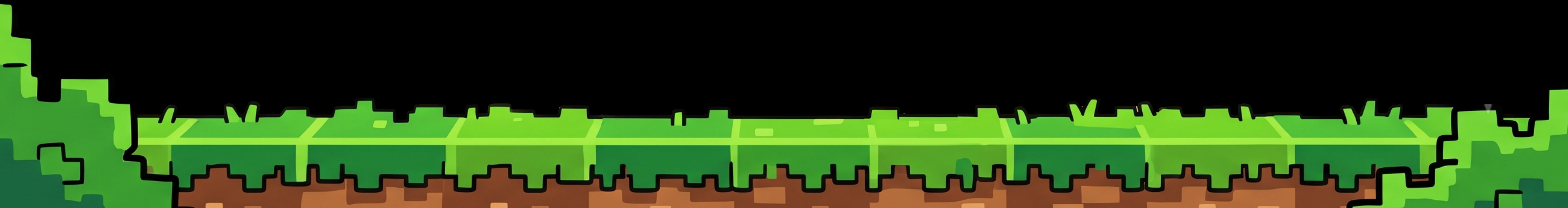
scheme hostname port



@view-transition

Default transition is cross-fade

Both pages need to opt-in



Opt-in

```
@view-transition {  
  navigation: auto;  
}
```





Mario



Luigi

1Main

ElementsConsoleSources

<!DOCTYPE html>
<html lang="en">
 <head>
 </head>
 <body>
 <main>
 <!-- Code injected by live-server -->
 <script type="text/javascript">
 </script>
 </body>
 </html>

htmlbody

StylesComputedLayoutEvent ListenersDOM breakpointsProperties

Filter(*) :hov .cls + -

element.style {
}

body {
 font-size: 1.25rem;
 font-family: "Marker Felt", sans-serif;
 background-color: rgb(0, 0, 0);
 color: rgb(255, 255, 255);
 line-height: 1.8;
}

body {
 display: block;
 margin: 8px;
}

margin 8
border 0
padding 0

A Piranha Plant, a carnivorous enemy from the Super Mario series, with a red, spotted body and a large, open mouth showing sharp teeth and a pink tongue.



Mario



Luigi

1Main

ElementsConsoleSources

<!DOCTYPE html>
<html lang="en">
 <head>
 </head>
 <body>
 <main>
 <!-- Code injected by live-server -->
 <script type="text/javascript">
 </script>
 </body>
 </html>

htmlbody

StylesComputedLayoutEvent ListenersDOM breakpointsProperties

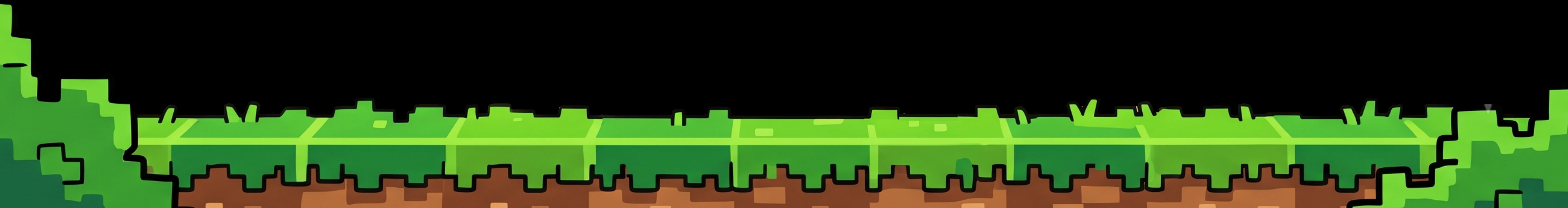
Filter(*) :hov .cls +
element.style {
}
body {
 font-size: 1.25rem;
 font-family: "Marker Felt", sans-serif;
 background-color: rgb(0, 0, 0);
 color: rgb(255, 255, 255);
 line-height: 1.8;
}
body {
 display: block;
 margin: 8px;
}

margin 8
border 0
padding 0

view-transition-name

Animate specific elements

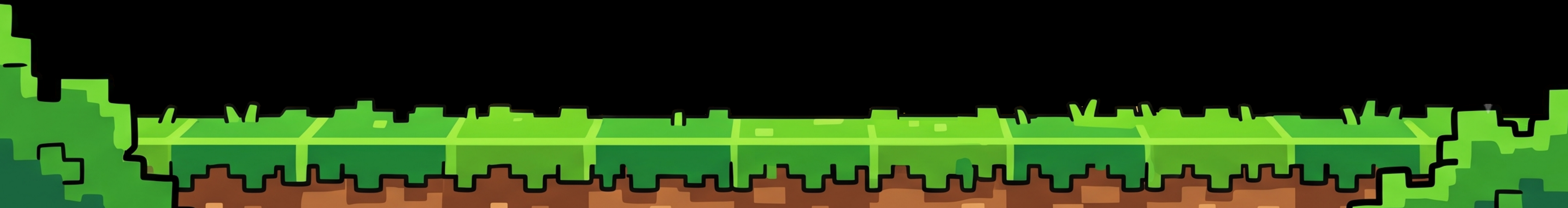
Both pages need to opt-in




```

```

```
<a class="luigi" href="luigi.html">Luigi</a>
```




```

```

```
<a class="luigi" href="luigi.html">Luigi</a>
```

```
@view-transition {  
  navigation: auto;  
}
```

```
a.luigi {  
  view-transition-name: luigi-text;  
}
```

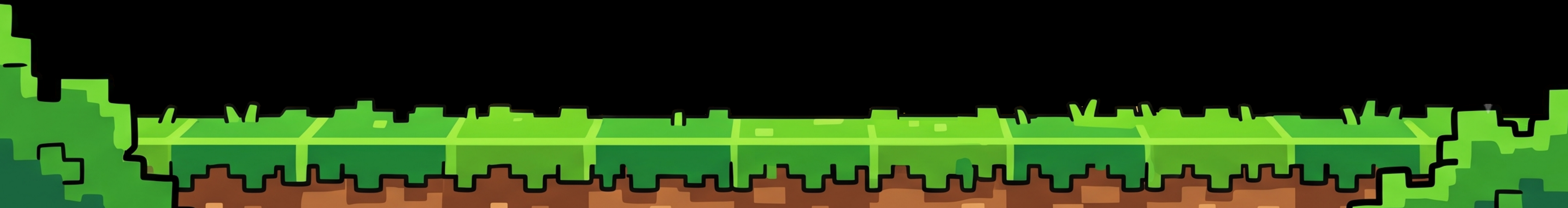
```
img.luigi {  
  view-transition-name: luigi-image;  
}
```



```

```

```
<h1 class="luigi">Luigi</h1>
```




```

```

```
<h1 class="luigi">Luigi</h1>
```

```
@view-transition {  
  navigation: auto;  
}
```

```
h1 {  
  view-transition-name: luigi-text;  
}
```

```
img {  
  view-transition-name: luigi-image;  
}
```





Mario



Luigi



Elements

Console

>>

1

Main

html

body

Styles

Computed

Layout

Event Listeners

DOM breakpoints

>>

Filter

(*)

:hov

.cls

+

element.style {

}

body {

font-size: 1.25rem;

font-family: "Marker Felt", sans-serif;

background-color: rgb(0, 0, 0);

color: rgb(255, 255, 255);

line-height: 1.8;

}

body {

display: block;

margin: 8px;

}

style.css:9

user agent stylesheet

margin 8

border 0

padding 0

8 0 0 1018x720.898 0 0 8

Console

AI assistance

Issues

Animations

100%

25%

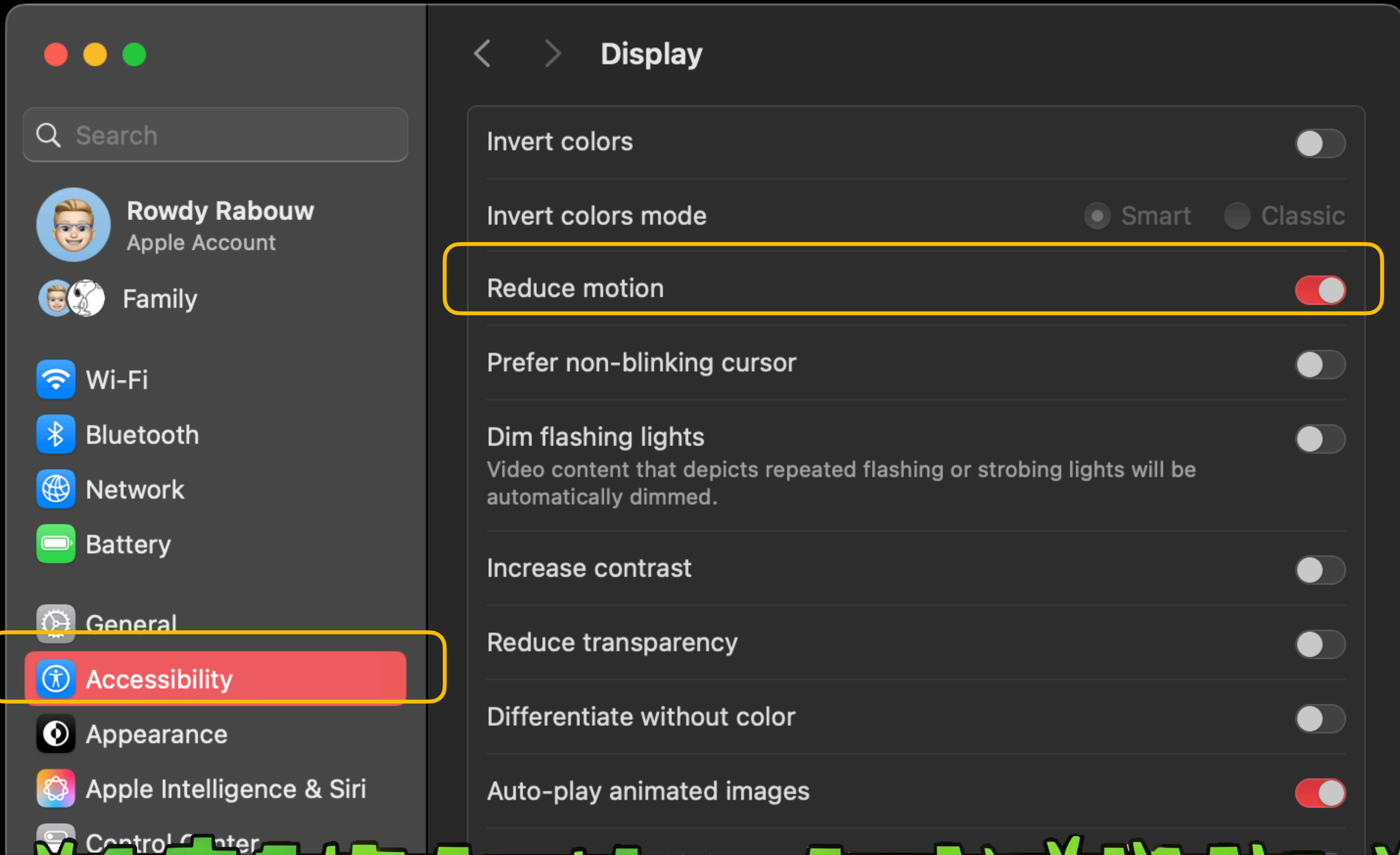
10%

0 ms

No animation effect selected

Select an effect above to inspect and modify







← Settings

Find a setting



Gebruiker

Local Account

 Home

 System

 Bluetooth & devices

 Network & internet

 Personalisation

 Apps

 Accounts

 Time & language

 Gaming

 Accessibility

 Privacy & security

Accessibility > Visual effects

 Always show scrollbars

Off ☐

 Transparency effects

Make some window backgrounds slightly transparent

On ☒

 Animation effects

Off ☐

 Dismiss notifications after this amount of time

5 seconds 

Related settings

 Display

Monitors, brightness, night light, display profile



 Notifications

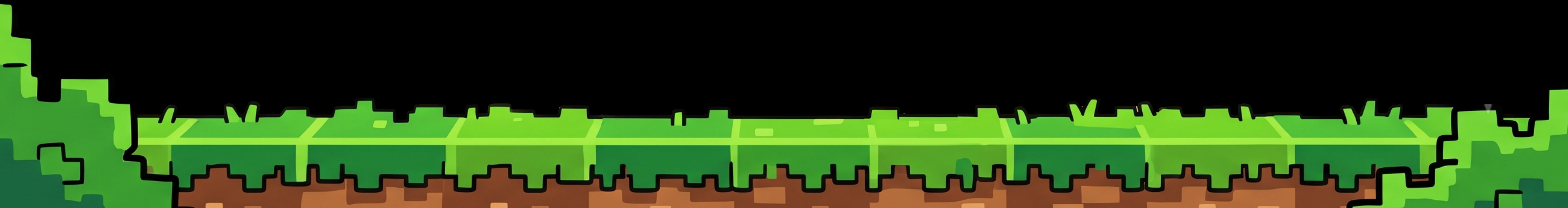
App and system notifications



prefers-reduced-motion

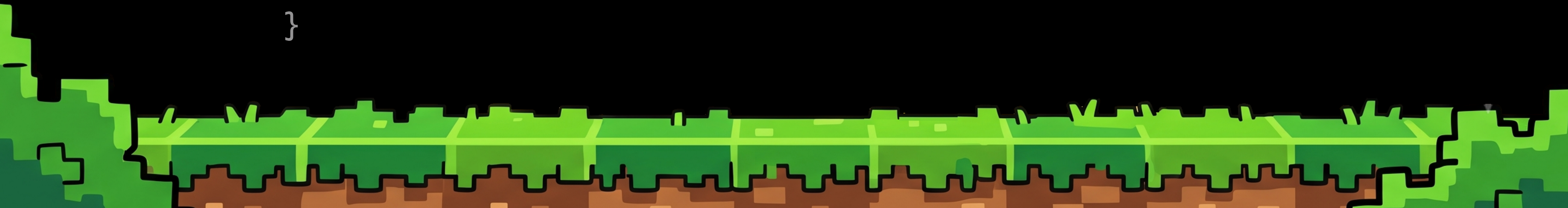
```
@media (prefers-reduced-motion: reduce) {  
  /* less motion for users who prefer reduced motion */  
}
```

```
@media (prefers-reduced-motion: no-preference) {  
  /* normal motion for users who do not prefer reduced motion */  
}
```



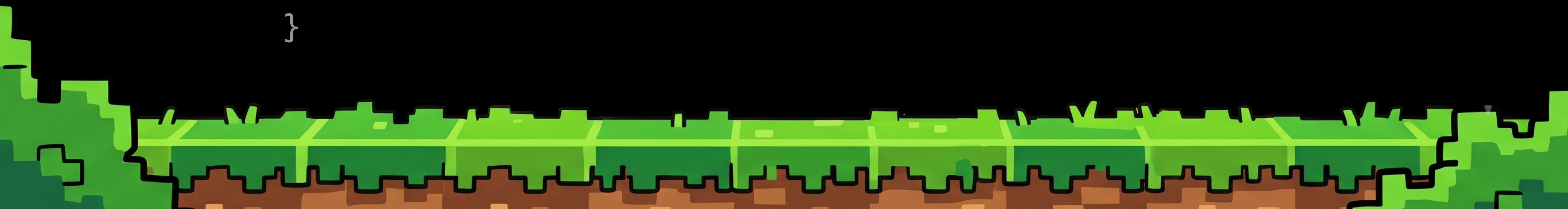

```
@view-transition {  
  navigation: auto;  
}
```

```
@media (prefers-reduced-motion: no-preference) {  
  
  a.luigi {  
    view-transition-name: luigi-text;  
  }  
  
  img.luigi {  
    view-transition-name: luigi-image;  
  }  
  
}
```




```
@view-transition {  
  navigation: auto;  
}
```

```
@media (prefers-reduced-motion: no-preference) {  
  
  h1 {  
    view-transition-name: luigi-text;  
  }  
  
  img {  
    view-transition-name: luigi-image;  
  }  
  
}
```

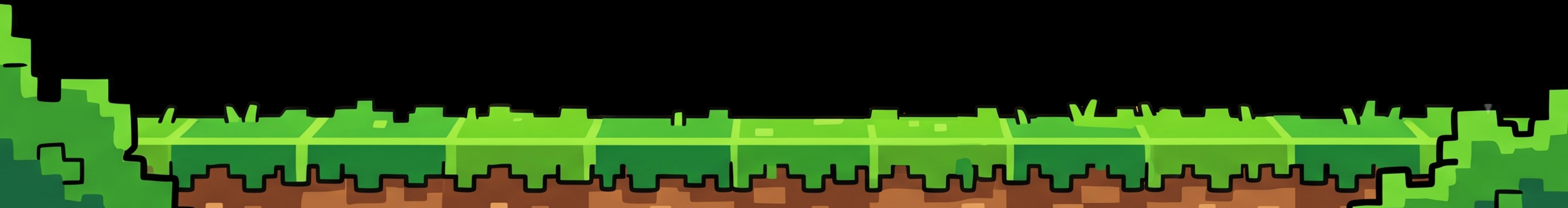


Overwrite default behavior

```
@view-transition {  
  navigation: auto;  
}
```




```
/* the outgoing page snapshot fading out */  
::view-transition-old(root) {  
  animation-name: -ua-view-transition-fade-out;  
  /* animates opacity from 1 to 0 */  
}  
  
/* the incoming page snapshot fading in */  
::view-transition-new(root) {  
  animation-name: -ua-view-transition-fade-in;  
  /* animates opacity from 0 to 1 */  
}
```




```
::view-transition-old(root) {  
  animation: 400ms ease-out both slide-out-to-left;  
}  
  
::view-transition-new(root) {  
  animation: 400ms ease-out both slide-in-from-right;  
}
```




```
@keyframes slide-out-to-left {  
  from {  
    transform: translateX(0);  
  }  
  to {  
    transform: translateX(-100%);  
  }  
}
```

```
@keyframes slide-in-from-right {  
  from {  
    transform: translateX(100%);  
  }  
  to {  
    transform: translateX(0);  
  }  
}
```





Mario

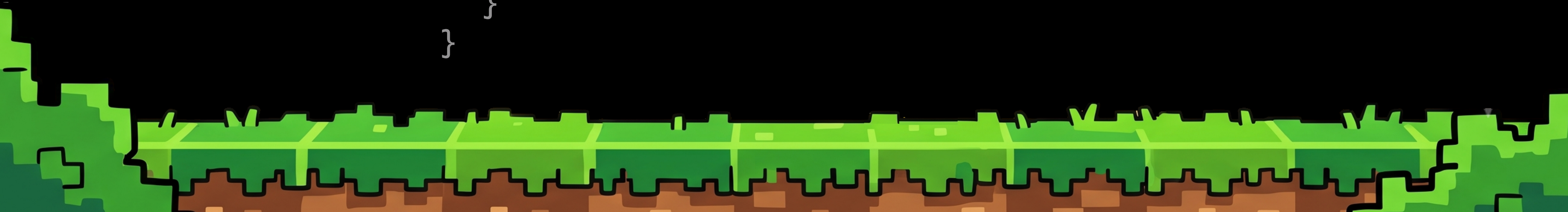


Luigi

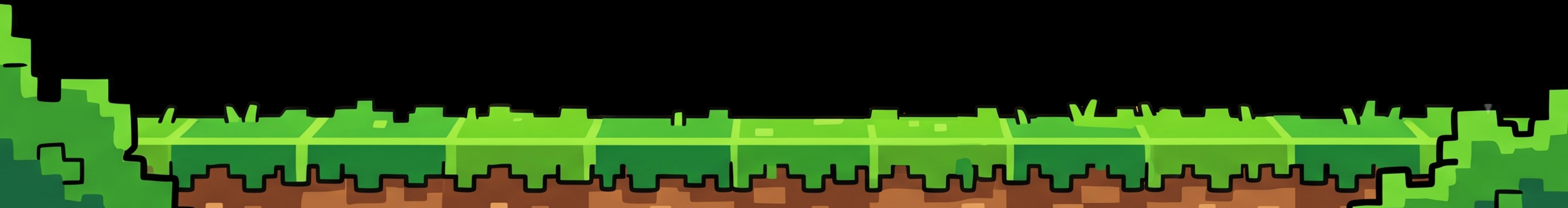



```
@keyframes slide-out-to-right {  
  from {  
    transform: translateX(0);  
  }  
  to {  
    transform: translateX(100%);  
  }  
}
```

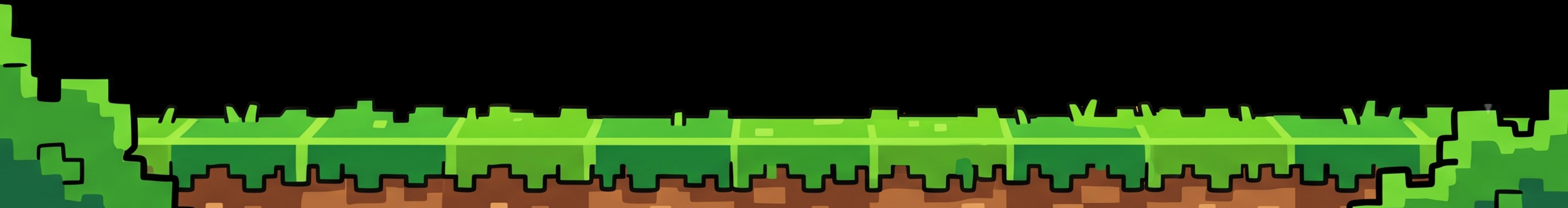
```
@keyframes slide-in-from-left {  
  from {  
    transform: translateX(-100%);  
  }  
  to {  
    transform: translateX(0);  
  }  
}
```




```
html:active-view-transition-type(back) {  
  
  &::view-transition-old(root) {  
    animation-name: slide-out-to-right;  
  }  
  
  &::view-transition-new(root) {  
    animation-name: slide-in-from-left;  
  }  
}
```




```
// fires when the new page is rendering its first frame  
window.addEventListener("pagereveal", applyTransitionType);
```

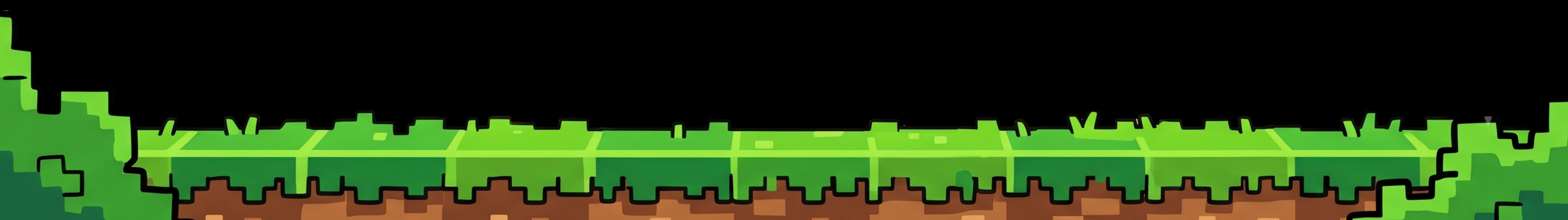



```
// fires when the new page is rendering its first frame
window.addEventListener("pagereveal", applyTransitionType);

const applyTransitionType = (event) => {

  if (navigation.activation.from.index > navigation.currentEntry.index) {
    event.viewTransition.types.add("back");
  }

};
```




```
html:active-view-transition-type(back) {  
  
  &::view-transition-old(root) {  
    animation-name: slide-out-to-right;  
  }  
  
  &::view-transition-new(root) {  
    animation-name: slide-in-from-left;  
  }  
}
```





Mario



Luigi





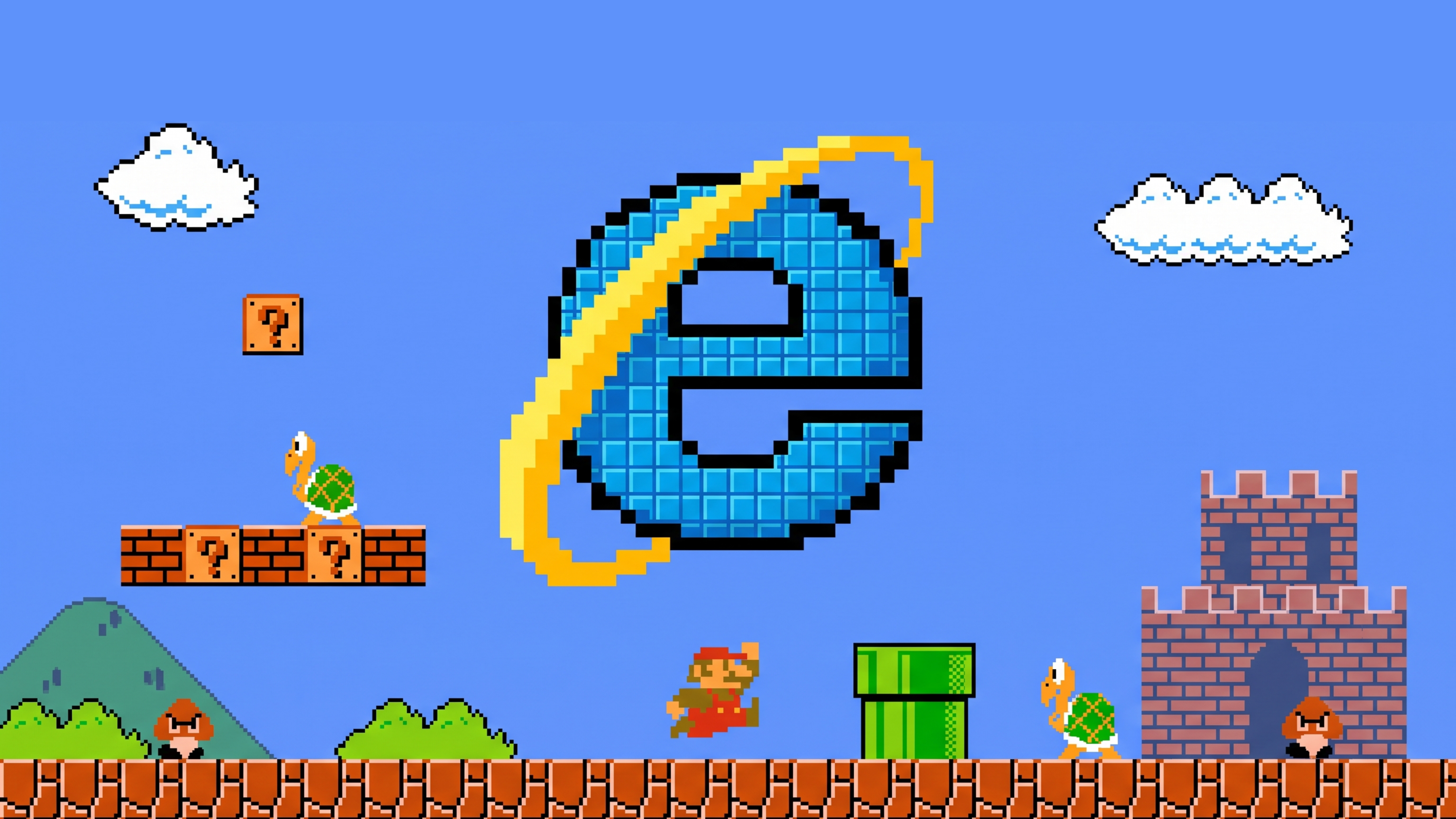
June 2024



June 2024



December 2024





INTERPAGE TRANSITIONS

Transitions were implemented with meta tags placed in the head section.

The meta tag specified the transition type and whether it should occur when the new page loaded or the current page exited.



INTERPAGE TRANSITIONS

0	Box in	A rectangle shrinks towards the center
1	Box out	A rectangle expands from the center outwards
2	Circle in	A circle shrinks towards the center
3	Circle out	A circle expands from the center outwards
4	Wipe up	Horizontal wipe from bottom to top
5	Wipe down	Horizontal wipe from top to bottom
6	Wipe right	Vertical wipe from left to right
7	Wipe left	Vertical wipe from right to left
8	Vertical blinds	Blinds open vertically from left to right
9	Horizontal blinds	Blinds open horizontally from top to bottom
10	Checkerboard across	Box blinds wipe from left to right
11	Checkerboard down	Box blinds wipe from top to bottom



INTERPAGE TRANSITIONS

12	Random dissolve	A pixel fade effect (static-like dissolve)
13	Split vertical in	Vertical window opening from the middle to the sides
14	Split vertical out	Vertical window closing from the sides to the middle
15	Split horizontal in	Horizontal window opening from the middle to top/bottom
16	Split horizontal out	Horizontal window closing from top/bottom to the middle
17	Strips left down	Content revealed from the upper-left corner to the lower-right
18	Strips left up	Content revealed from the lower-left corner to the upper-right
19	Strips right down	Content revealed from the upper-right corner to the lower-left
20	Strips right up	Content revealed from the lower-right corner to the upper-left
21	Random bars horizontal	A vertical line fade effect
22	Random bars vertical	A horizontal line fade effect
23	Random	Randomly selects one of the 23 effects above



INTERPAGE TRANSITIONS

```
<meta http-equiv="Page-Enter" content="revealTrans(Duration=2.0,Transition=23)" />
```

```
<meta http-equiv="Page-Exit" content="revealTrans(Duration=2.0,Transition=23)" />
```



IE Page Transitions (MPA)

Bringing back Internet Explorer's Page Transitions thanks to the View Transition API

[← To MPA Index](#) | [To Random, again →](#)

Hit the back button or click one of the links above.

Built by [Bramus](#). Just for funzies.

INTERPAGE TRANSITIONS

<https://page-transitions.style>

<https://github.com/bramus/ie-page-transitions>



Bramus

Google Chrome Developer Relations



Service Workers

Think of a Service Worker as a mechanic Toad operating in the background pipes while Mario runs the main level.

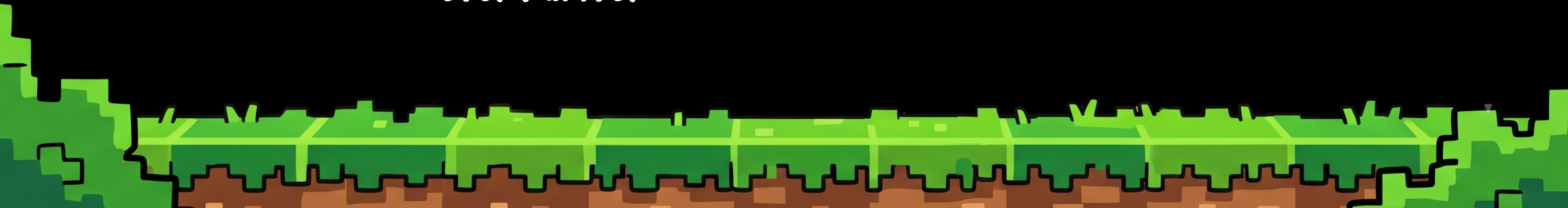


Service Workers

When Mario needs data but the Wi-Fi connection to Peach's Castle is broken, Toad intercepts the request and instantly hands him a cached Fire Flower so the game never stops.

Service Workers

- JavaScript file
- runs in the background in a separate thread
- non-blocking
- scoped to a website
- can't modify page; no direct access to DOM
- asynchronous; promises or async/await
- event driven




```
self.addEventListener("install", async () => {  
  console.info("Service Worker installed");  
  // activate service worker immediately after installation  
  self.skipWaiting();  
});
```

```
self.addEventListener("activate", async () => {  
  console.info("Service Worker activated");  
  // service worker controls the pages immediately after activation  
  await self.clients.claim();  
});
```

service-worker.js




```
self.addEventListener("fetch", async (event) => {  
  event.respondWith(  

```

```
  );  
});
```

service-worker.js


```
self.addEventListener("fetch", async (event) => {  
  event.respondWith(  
    (async () => {  
      const cache = await caches.open("yoshi");
```

```
    })(),  
  );  
});
```

service-worker.js


```
self.addEventListener("fetch", async (event) => {  
  event.respondWith(  
    (async () => {  
      const cache = await caches.open("yoshi");  
      try {  
        const response = await fetch(event.request);  
  
        return response;  
      }  
    })(),  
  );  
});
```

service-worker.js




```
self.addEventListener("fetch", async (event) => {  
  event.respondWith(  
    (async () => {  
      const cache = await caches.open("yoshis");  
      try {  
        const response = await fetch(event.request);  
  
        await cache.put(event.request, response.clone());  
  
        return response;  
      }  
    })(),  
  );  
});
```

service-worker.js


```
self.addEventListener("fetch", async (event) => {  
  event.respondWith(  
    (async () => {  
      const cache = await caches.open("yoshis");  
      try {  
        const response = await fetch(event.request);  
        // never cache the service worker itself  
        if (!event.request.url.includes("sw.js")) {  
          await cache.put(event.request, response.clone());  
        }  
        return response;  
      }  
    })  
  )  
})
```

```
    })(),  
  );  
});
```

service-worker.js


```
self.addEventListener("fetch", async (event) => {
  event.respondWith(
    (async () => {
      const cache = await caches.open("yoshis");
      try {
        const response = await fetch(event.request);
        // never cache the service worker itself
        if (!event.request.url.includes("sw.js")) {
          await cache.put(event.request, response.clone());
        }
        return response;
      } catch {
        return await cache.match(event.request);
      }
    })(),
  );
});
```

service-worker.js


```
document.addEventListener("DOMContentLoaded", async () => {  
  try {  
    // register the service worker if not already registered  
    if (!navigator.serviceWorker.controller) {  
      await navigator.serviceWorker.register("service-worker.js");  
      console.info("Service Worker registered");  
    }  
  } catch (e) {  
    console.error("Error during service worker registration:", e);  
  }  
});
```

script.js




```
<script src="script.js" defer></script>
```

*.html





Mario



Luigi





Mario



Luigi

ElementsConsoleSourcesNetworkApplicationPerformance>>

Application

- Manifest
- Service workers
- Storage
- WebMCP NEW
- Ads

Storage

- Local storage
- Session storage
- Extension storage
- IndexedDB
- Cookies
- Private state tokens
- Cache storage
- Storage buckets

Background services

- Back/forward cache
- Background fetch
- Background sync
- Bounce tracking mitigations
- Notifications
- Payment handler
- Periodic background sync
- Speculative loads
 - Rules
 - Speculations**
- Push messaging
- Reporting API
- Device bound sessions

All speculative ...

Filter

URL	Action	Rul...	Status
/5-service-worker/luigi.ht...	Prerender	5-s...	Not triggered
/5-service-worker/luigi.ht...	Prefetch	5-s...	Not triggered





Mario



Luigi

⌵ ⌵ | Elements | Console | Sources | **Network** ⚠ | Application >> | 11 5 | 🔍 ⚙ ⋮ ✕

🛑 | 🔍 | ☐ Keep log | ☒ Disable cache | Offline | 📶 | ⬆️ ⬇️ | ⚙

⌵ Filter | ☐ Invert | More filters ⌵

All | Fetch/XHR | Doc | CSS | JS | Font | Img | Media | Manifest | Socket | Wasm | Other

20 ms40 ms60 ms80 ms100 ms

Currently recording network activity

Perform a request or reload the page by using the "Reload page" button or by pressing ⌘ R [Learn more](#)

Reload page





✓ 45

September 2015



✓ 17

April 2018



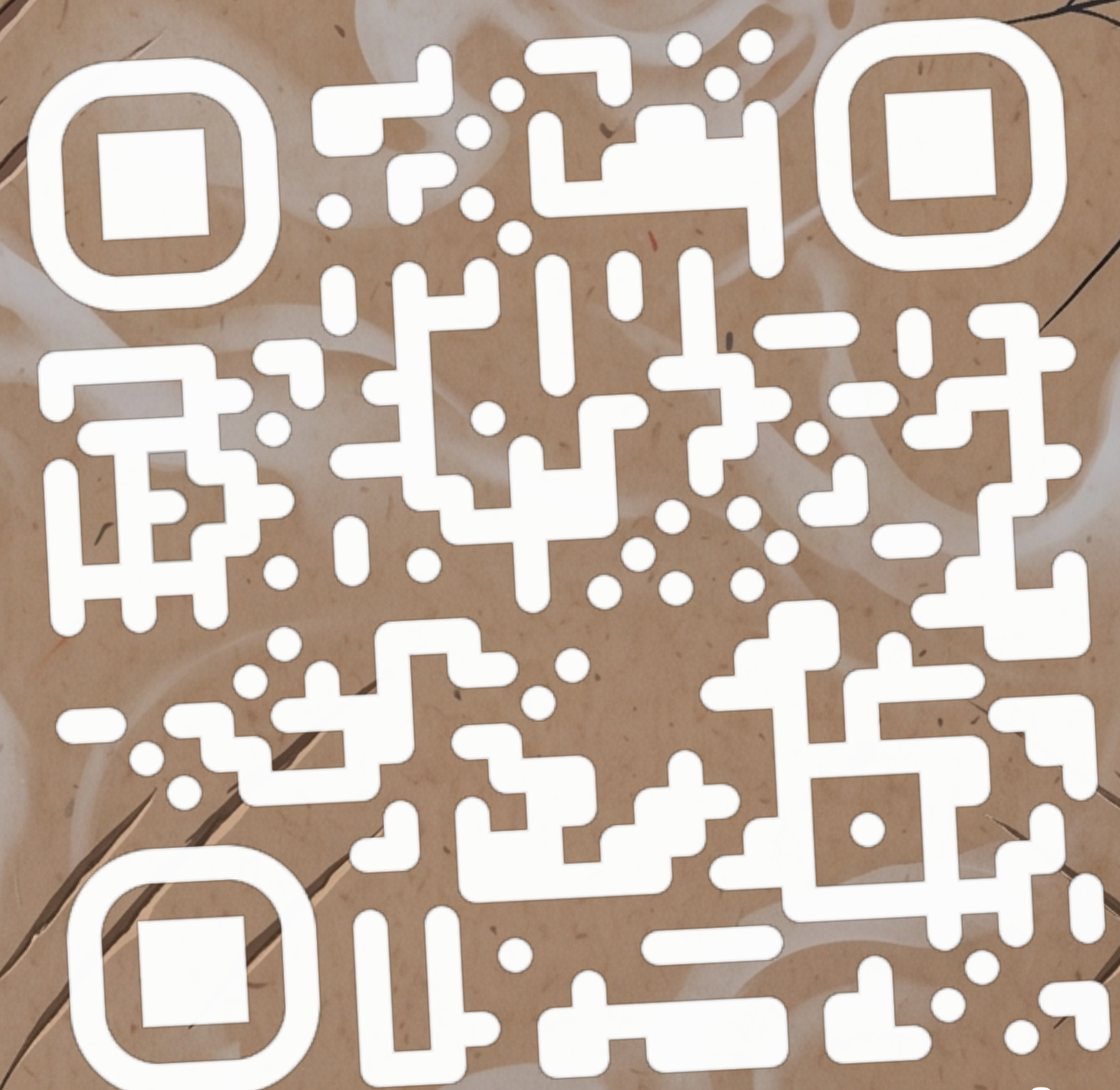
✓ 44

January 2016



✓ 11.1

March 2018



rowdy.codes/spooky