

# Blitz

A lightweight, modular, extensible web renderer



**Dioxus Labs**

# What is Dioxus

- “React” but in Rust
- Renders to
  - Web (WASM)
  - Webview (IPC)
- Until now there has been no “native” rendering

```
pub fn App() -> Element {  
    // count will be initialized to 0 the first time the component is rendered  
    let mut count = use_signal(|| 0);  
  
    rsx! {  
        h1 { "High-Five counter: {count}" }  
        button { onclick: move |_| count += 1, "Up high!" }  
        button { onclick: move |_| count -= 1, "Down low!" }  
    }  
}
```

Copy

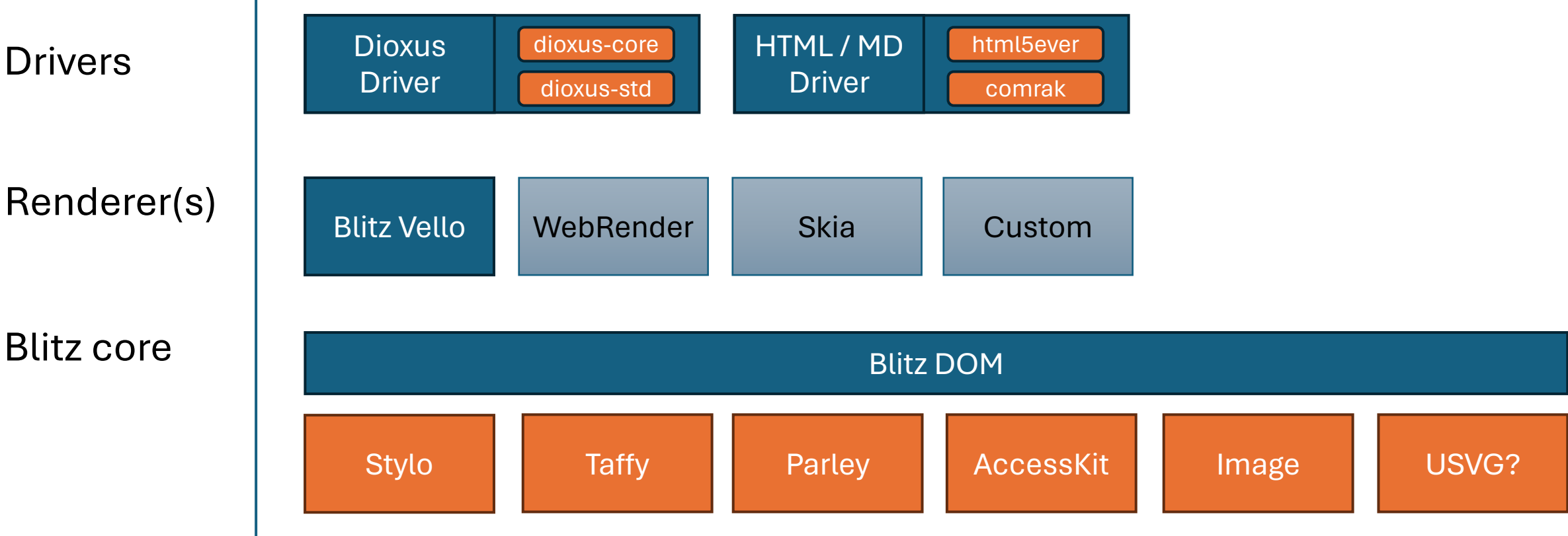
```
rsx! {  
    div {  
        padding: "0.5rem",  
        position: "relative",  
        onmouseenter: move |_| {},  
        div { font_size: "1.5rem",  
            a { href: url, onfocus: move |_event| {}, "{title}" }  
            a {  
                color: "gray",  
                href: "https://news.ycombinator.com/from?site={hostname}",  
                text_decoration: "none",  
                "{hostname}"  
            }  
        }  
    }  
    div { display: "flex", flex_direction: "row", color: "gray",  
        div { "{score}" }  
        div { padding_left: "0.5rem", "by {by}" }  
        div { padding_left: "0.5rem", "{time}" }  
        div { padding_left: "0.5rem", "{comments}" }  
    }  
}
```

Copy

# What is Blitz

- An HTML/CSS renderer
  - Think: webview without JavaScript
- Similar: Electron, Tauri, Sciter, Ultralight, Litehtml
- Built on Servo and Rust UI ecosystem components
  - stylo, html5ever, taffy, parley, accesskit, vello, wgpu
- Targeting Application UI (+ Embedding) use cases
- Lightweight (~12mb binary size)
- Modular

# Architecture



# Ecosystem-first Modularity

- Using existing solutions from the ecosystem
- Contributing back to those solutions
- Making first-party code usable standalone
- Ecosystem compatible licensing (MIT/Apache2.0)
- Build only what you need:
  - Video support, Image formats (incl. SVG), Layout algorithms, CSS property support
- Extend with own implementations
- Bring your own renderer

# Binary size

(stripped release build on macos)

## Servo (98 MB)

File	.text	Size	Crate
8.3%	20.3%	10.2MiB	script
6.6%	16.2%	8.1MiB	mozjs_sys
3.0%	7.3%	3.6MiB	std
2.5%	6.0%	3.0MiB	webrender
2.2%	5.3%	2.6MiB	style
1.7%	4.2%	2.1MiB	[Unknown]
1.0%	2.4%	1.2MiB	net
0.8%	2.0%	1.0MiB	webgpu
0.8%	2.0%	1019.1KiB	naga
0.8%	1.9%	980.1KiB	webdriver_server
0.6%	1.5%	792.0KiB	servoshell
0.6%	1.5%	781.0KiB	layout_thread_2013
0.6%	1.5%	758.7KiB	canvas
0.6%	1.5%	755.8KiB	mozangle
0.6%	1.4%	709.1KiB	layout_thread_2020
0.5%	1.3%	655.8KiB	layout_2013
0.5%	1.3%	648.3KiB	servo
0.5%	1.2%	593.5KiB	image
0.5%	1.1%	569.0KiB	layout_2020
0.4%	1.1%	539.1KiB	wgpu_core
0.3%	0.8%	388.7KiB	servo_config
0.3%	0.7%	369.4KiB	devtools

## Blitz (12 MB)

File	.text	Size	Crate
12.0%	24.5%	1.9MiB	style
5.7%	11.7%	917.4KiB	std
4.5%	9.2%	718.1KiB	wgpu
4.3%	8.8%	686.7KiB	blitz_dom
3.9%	7.9%	622.0KiB	naga
1.7%	3.6%	278.3KiB	rustls
1.2%	2.5%	195.6KiB	wgpu_core
1.0%	2.0%	160.5KiB	vello
0.9%	1.8%	144.4KiB	ureq
0.9%	1.8%	140.8KiB	dioxus_blitz
0.9%	1.8%	138.6KiB	tao
0.8%	1.6%	123.4KiB	ring
0.7%	1.4%	110.3KiB	image
0.6%	1.3%	101.9KiB	tokio
0.6%	1.2%	97.3KiB	taffy
0.5%	1.1%	86.4KiB	wgpu_hal
0.5%	1.1%	83.7KiB	exr
0.5%	1.0%	78.0KiB	image_webp
0.5%	1.0%	77.1KiB	html5ever
0.5%	0.9%	73.5KiB	vello_encoding
0.4%	0.8%	61.7KiB	cssparser
0.4%	0.8%	60.9KiB	jpeg_decoder

Electron: ~130mb   Ultralight: ~22mb   Sciter: ~34mb

# CSS Support (Stylo)

- Huge feature set:
  - Parsing, selector resolution, inheritance, media queries
- Very fast + reliable (but heavy)
- Neat trait-based API

BUT:

- Poor documentation
- Features not in servo disabled

# Rendering (Vello + WGPU)

- Built on WGPU
- Innovative: compute shaders
- High performance (120fps)
- Immature (crashes, features)

Renderers are pluggable

-> Webrender, Skia, BYO (e.g. Bevy)







# Text Support (Parley)

- Pure rust stack
  - Skrifra, Fontique, Swash (RustyBuzz?), Parley
- System font enumeration
- Shaping, bidi aware layout
- Selection/Editing (Soon!)
- Inline-block

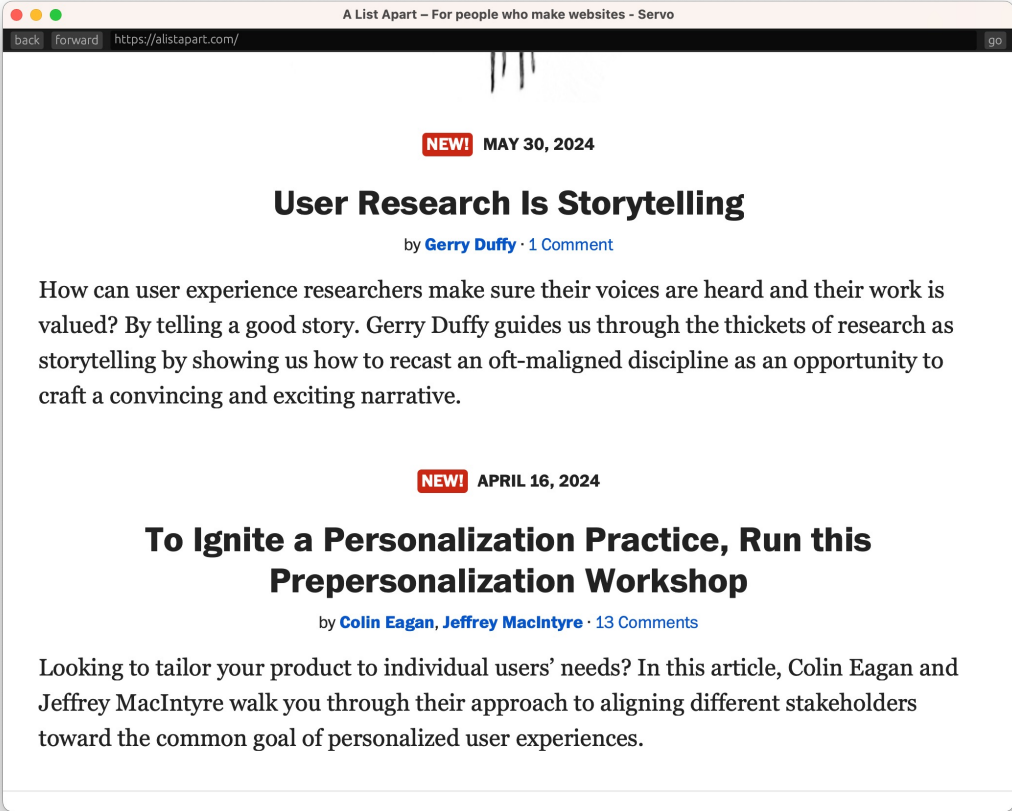
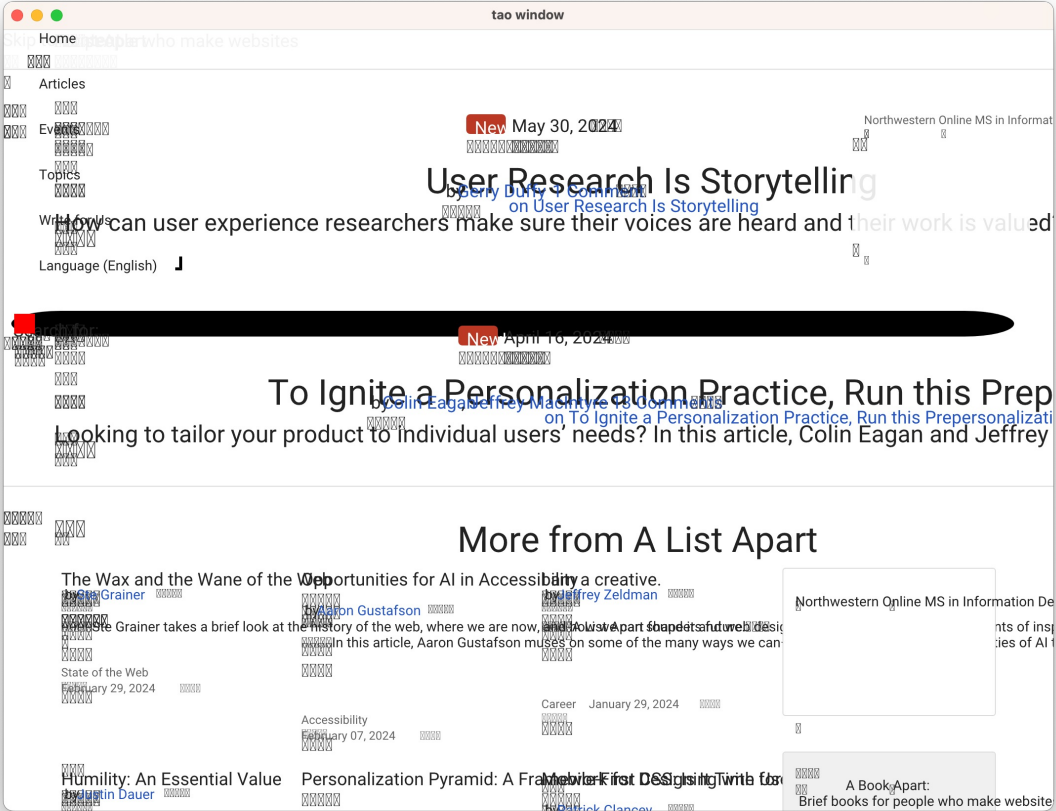
Not yet integrated into Blitz.

(inline-block, tree lowering)

Lots to add: wrapping modes, padding/border/margin support, floats.

Some text here. Let's make it  
a bit long  er so that  
 line wrapping kicks in  
😊. And also some اللغة العربية  
arabic text.

# AListApart.com (Blitz vs Servo)



# Layout (Taffy)

- Flexbox + CSS Grid (+ Block)
- Trait based API
- Coded “to the spec”
- 1000+ tests (but no WPT)

The sky is the limit:

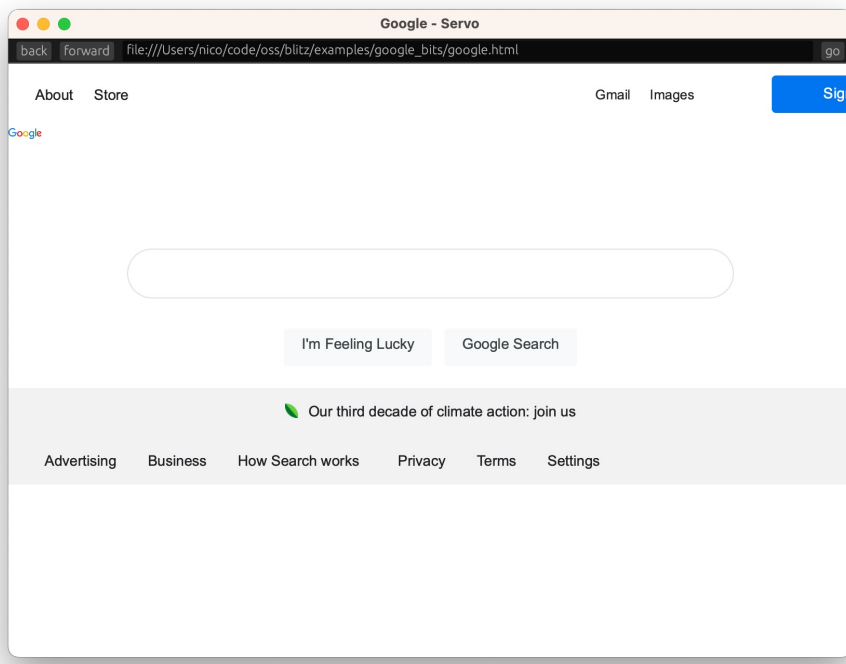
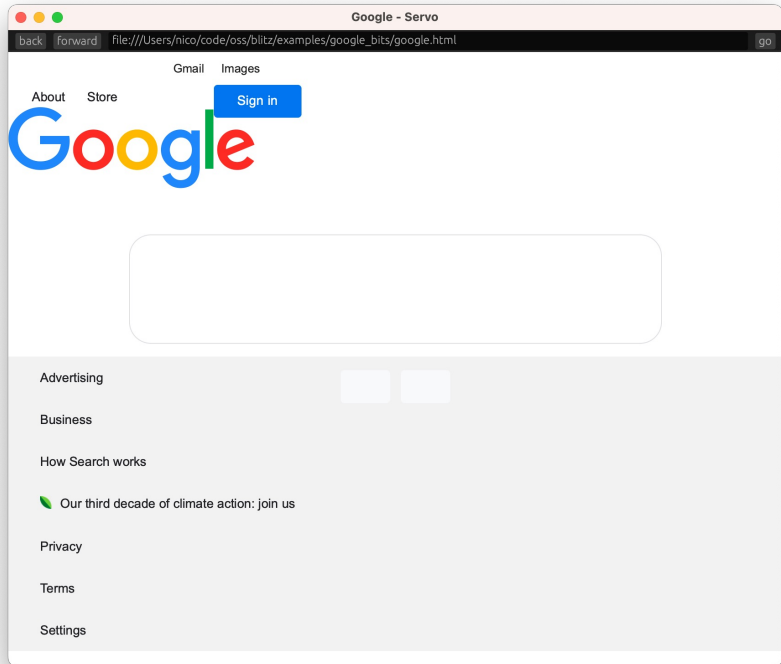
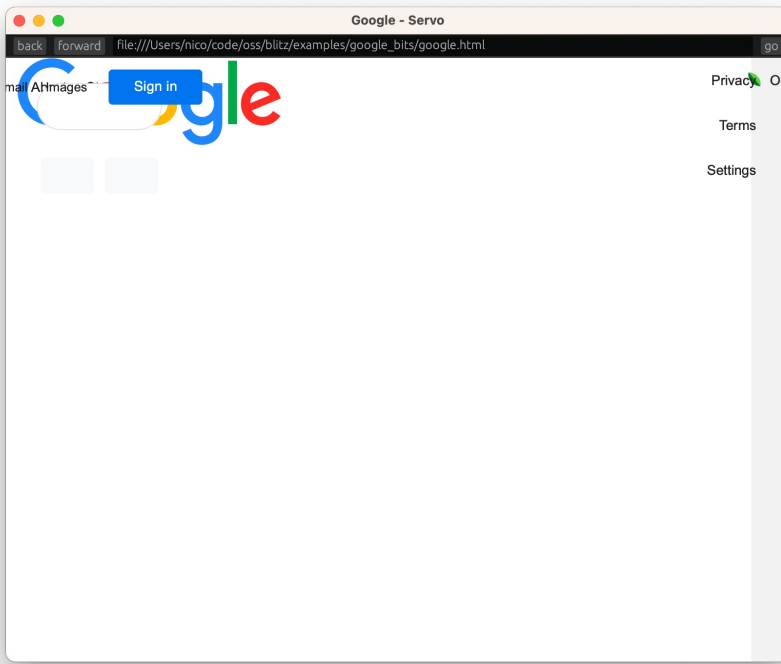
- Tables, Subgrid, Masonry, “Stretch”
- Parallel layout

## WPT:

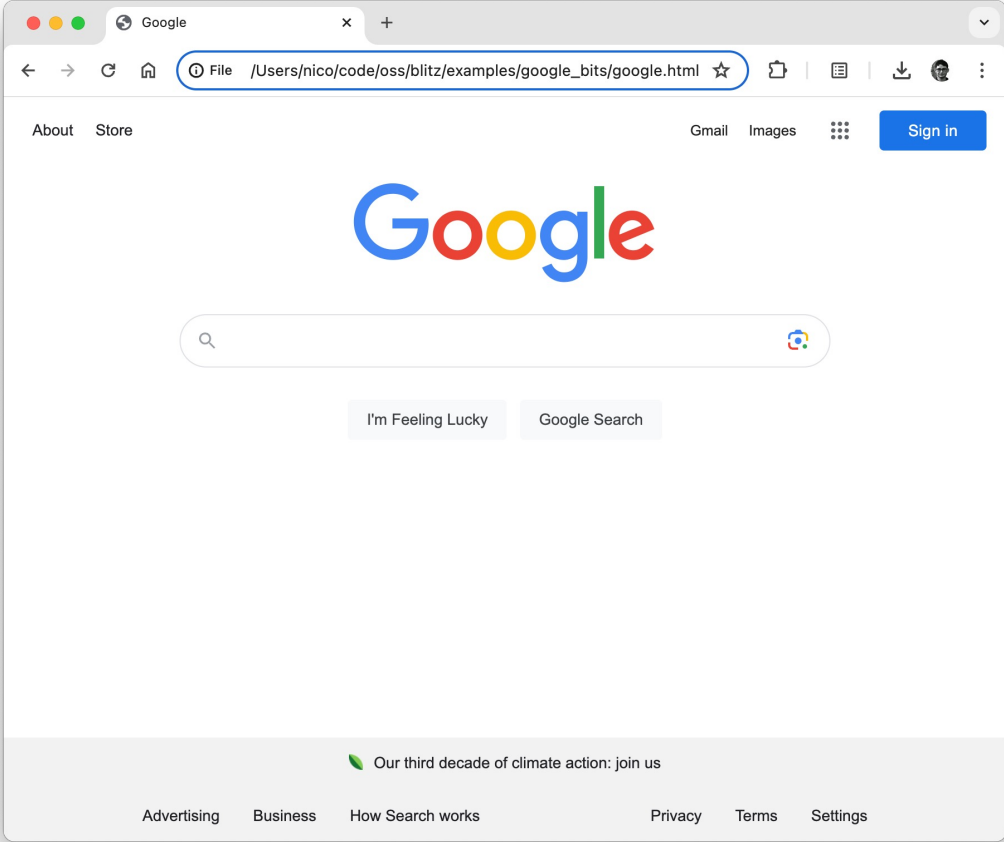
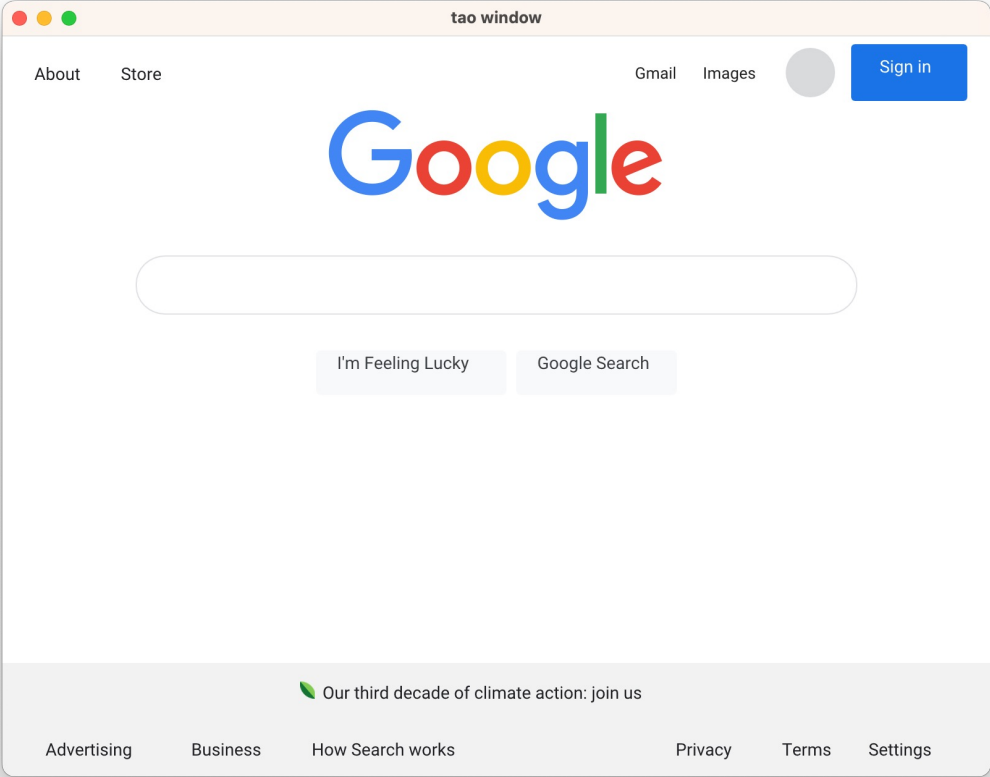
- Cannot run WPT due to lack of scripting support
- Want to introduce new “layout ref” test type (+ make layout tests script-free)

```
<div id="test-root" style="flex-direction: row; width: 100px;">
  <div style="flex-basis: 60px; flex-grow: 1; flex-direction: column;">
    <div style="width: 70px; height: 100px;"></div>
  </div>
  <div style="flex-basis: 0px; flex-grow: 1; flex-direction: column;">
    <div style="width: 20px; height: 100px;"></div>
  </div>
</div>
```

# Google.com (Servo)



# Google.com (Blitz vs Chrome)



**Integration with Servo?  
Gecko?**

**Let's chat**

# Roadmap

- Text / Inline layout / text input (parley)
- Font loading (system & web)
- Event handling & Interactivity
- SVG support
- calc(), content-box, static position
- Custom widgets
- Performance and Robustness
- Devtools

# Get in touch

GitHub:

[DioxusLabs/blitz](https://github.com/DioxusLabs/blitz)

Dioxus Discord:

<https://discord.gg/U4TYsKdN>

Servo Zulip:

<https://servo.zulipchat.com>

Linebender Zulip:

<https://xi.zulipchat.com>

Social media:

**@nicoburns** (Github, Reddit, Twitter)

Email:

**hackfest@nicoburns.com**

# Blitz



**Dioxus Labs**