

WEB DAY 2026

How Preply tracks and guide the Design System adoption in React and React Native

Stefano Magni

Tech Lead
Preply



WEB DAY 2026

Kudos



These slides are public!



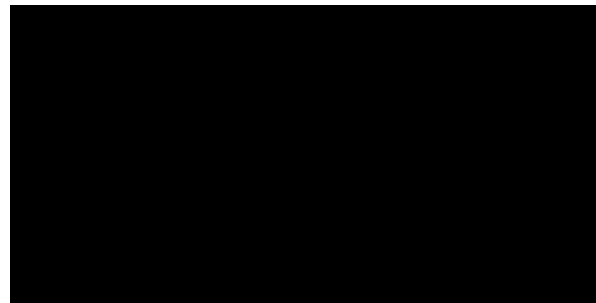
noriste.github.io/design-system-coverage-webday-2026

Why

To measure the DS impact from a Product perspective

What

**We measure how much
% of a page is built
with DS components.**



How

**We calculate the coverage in users' devices
by using idle callbacks, web workers, and typed arrays
without affecting the UX.**

Stefano Magni

Senior Front-end Engineer II
Design System Tech Lead
@Preply

- [linkedin/noriste](#)
- [x.com/noriste](#)
- [bsky.app/noriste.bsky.social](#)
- [github/noriste](#)



Matteo Ronchi

Frontend Architect
Design System Tech Lead
@WorkWave

- [linkedin/matteoronchi](https://www.linkedin.com/in/matteoronchi)
- x.com/cef62
- bsky.app/cef62.bsky.social
- [github/cef62](https://github.com/cef62)



WHY

are we doing it?

Measuring Impact



Measuring Impact

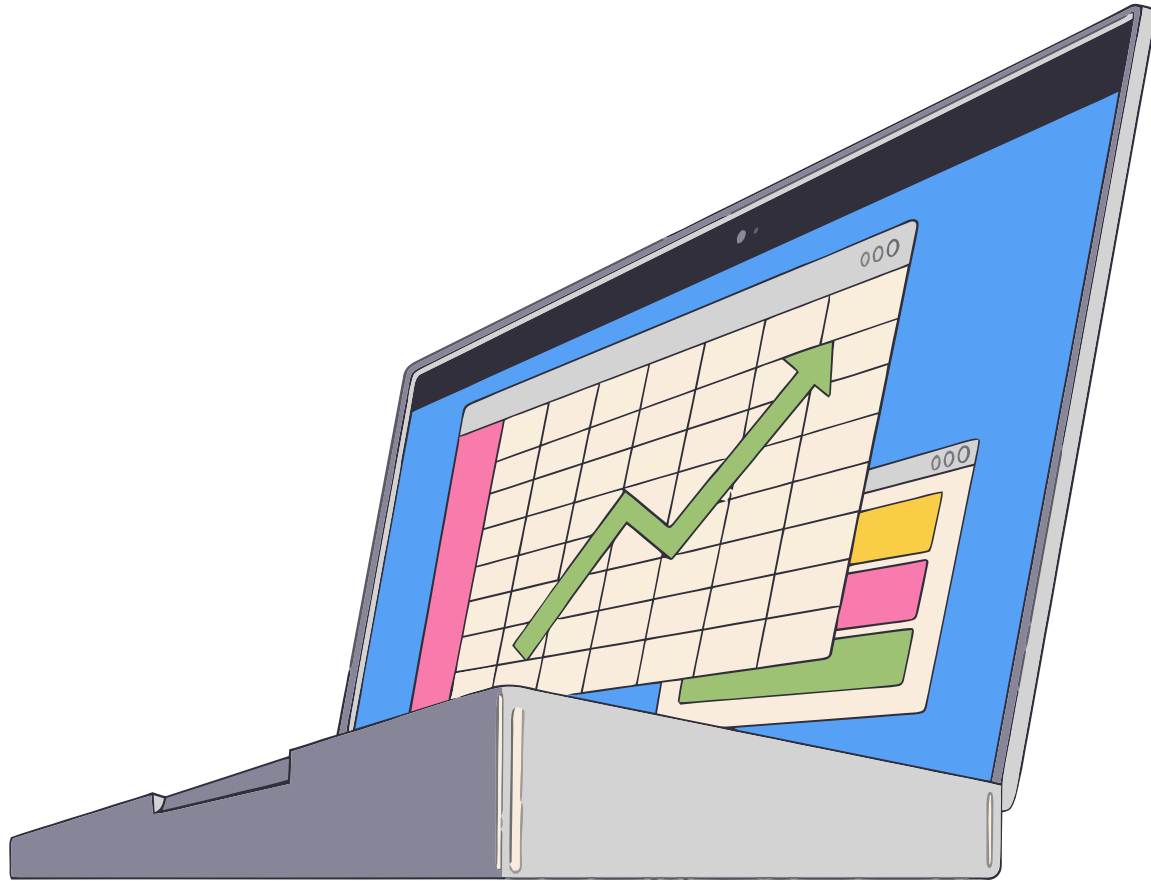
Product teams:



Measuring Impact

Product teams:

- We increased conversions by 10% 🥰
- We saved 100K \$ 🥰
- Lifetime value increased by 1% 🥰



Measuring Impact

Product teams:

- We increased conversions by 10% 🥰
- We saved 100K \$ 🥰
- Lifetime value increased by 1% 🥰

DS team:



Measuring Impact

Product teams:

- We increased conversions by 10% 🥰
- We saved 100K \$ 🥰
- Lifetime value increased by 1% 🥰

DS team:

- Our shiny button is used 176 times 😞🤔



Measuring Impact

Product teams:

- We increased conversions by 10% 🥰
- We saved 100K \$ 🥰
- Lifetime value increased by 1% 🥰

DS team:

- Our shiny button is used 176 times 😞😞





Measuring Impact

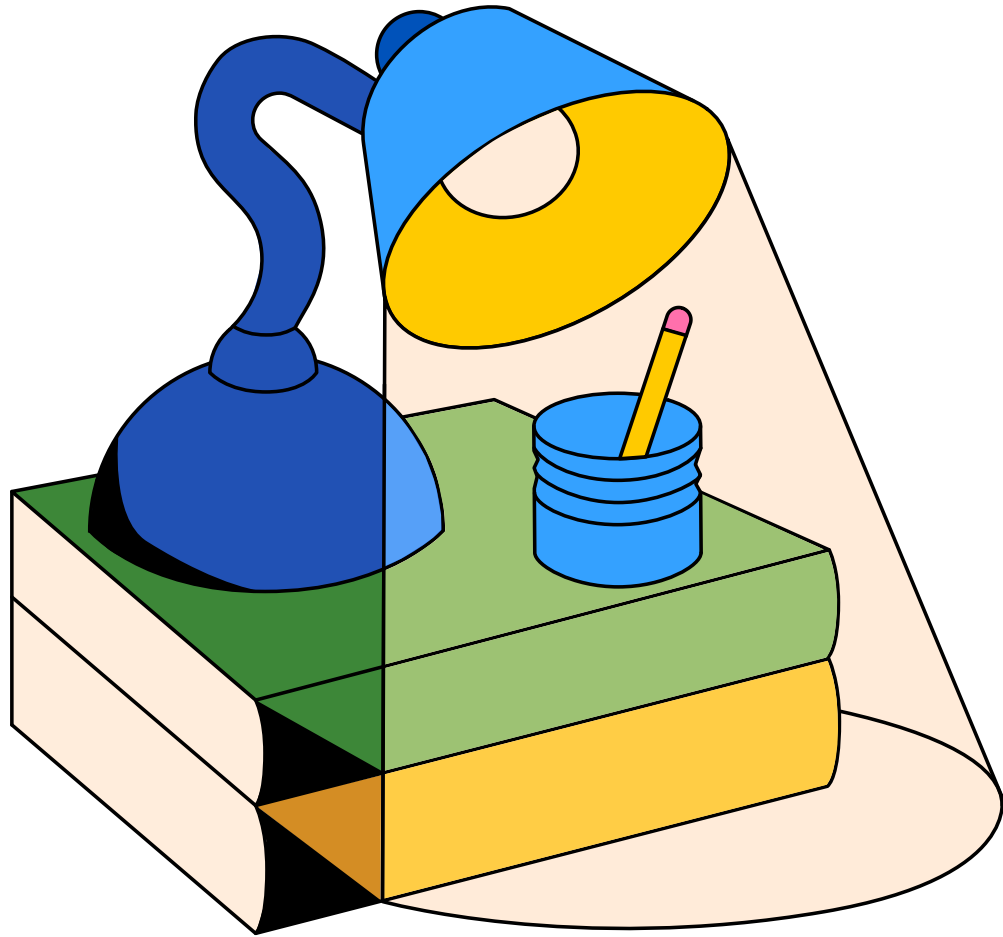
Product teams:

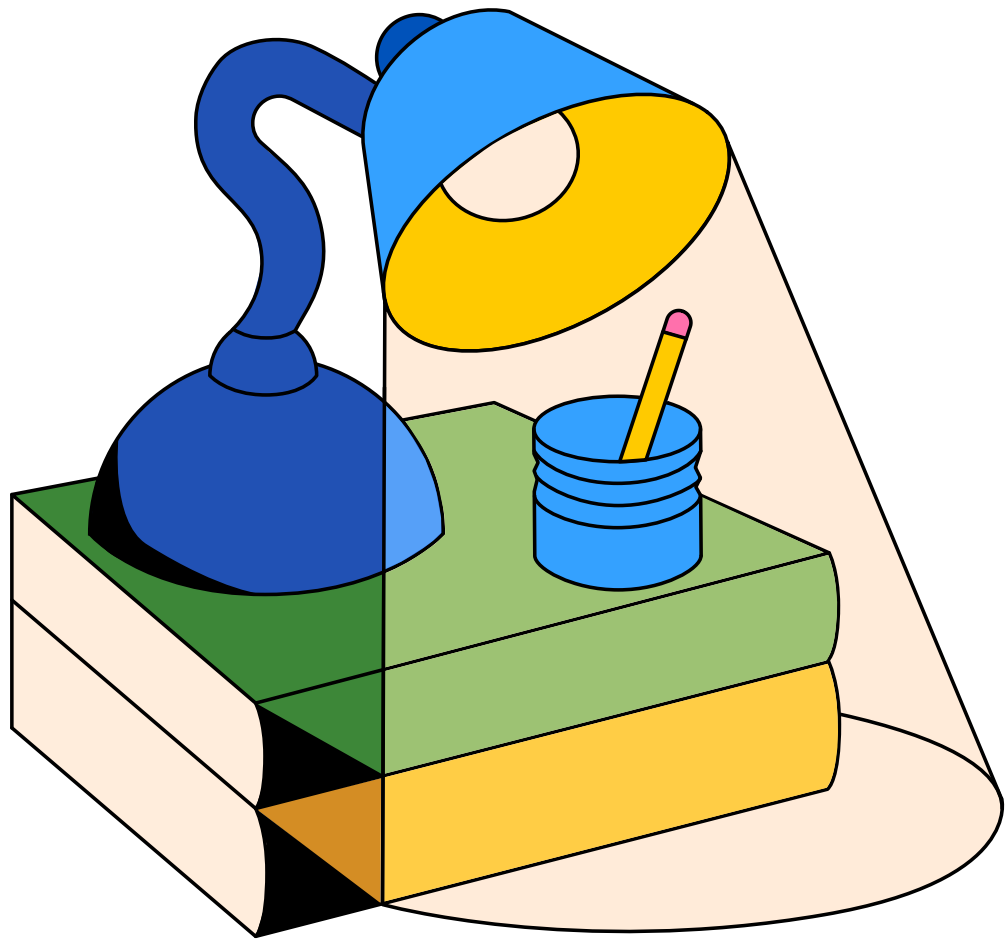
- We increased conversions by 10% 🥰
- We saved 100K \$ 🥰
- Lifetime value increased by 1% 🥰

DS team:

- Our shiny button is used 176 times 🤔
- 40% of the UI is built with the DS 🥳👏

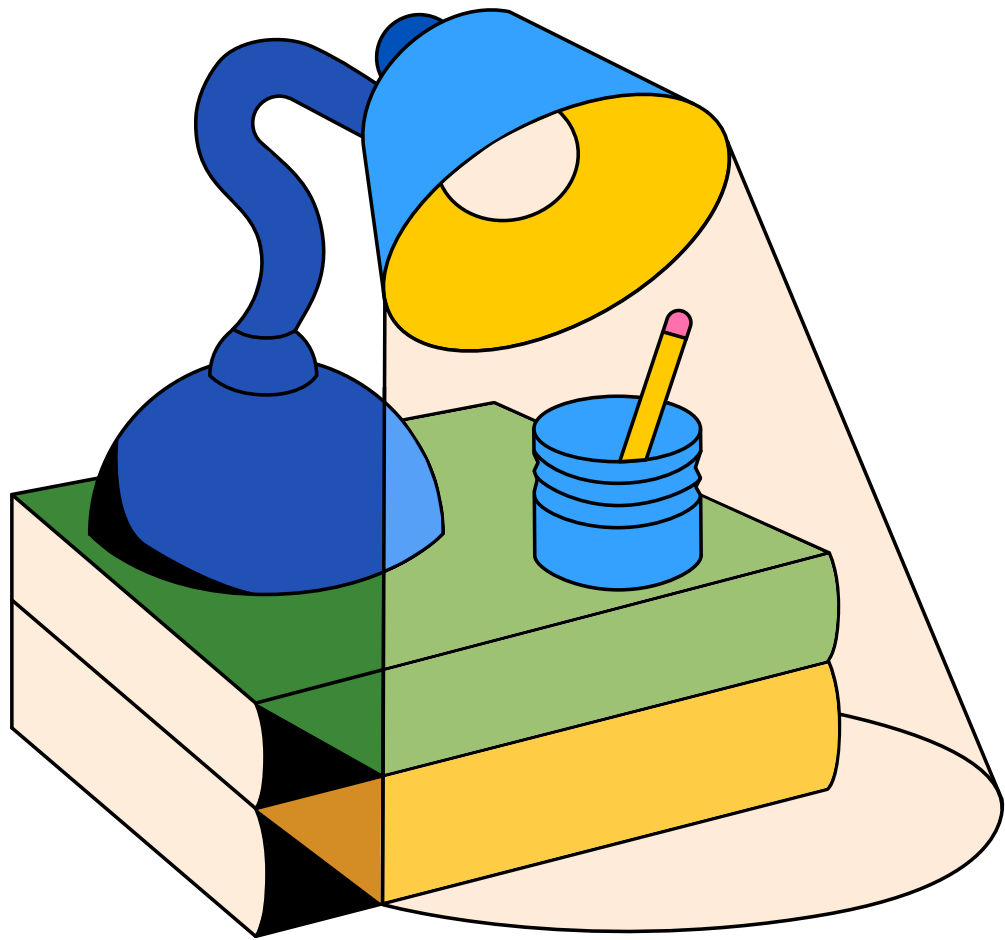
Static Analysis





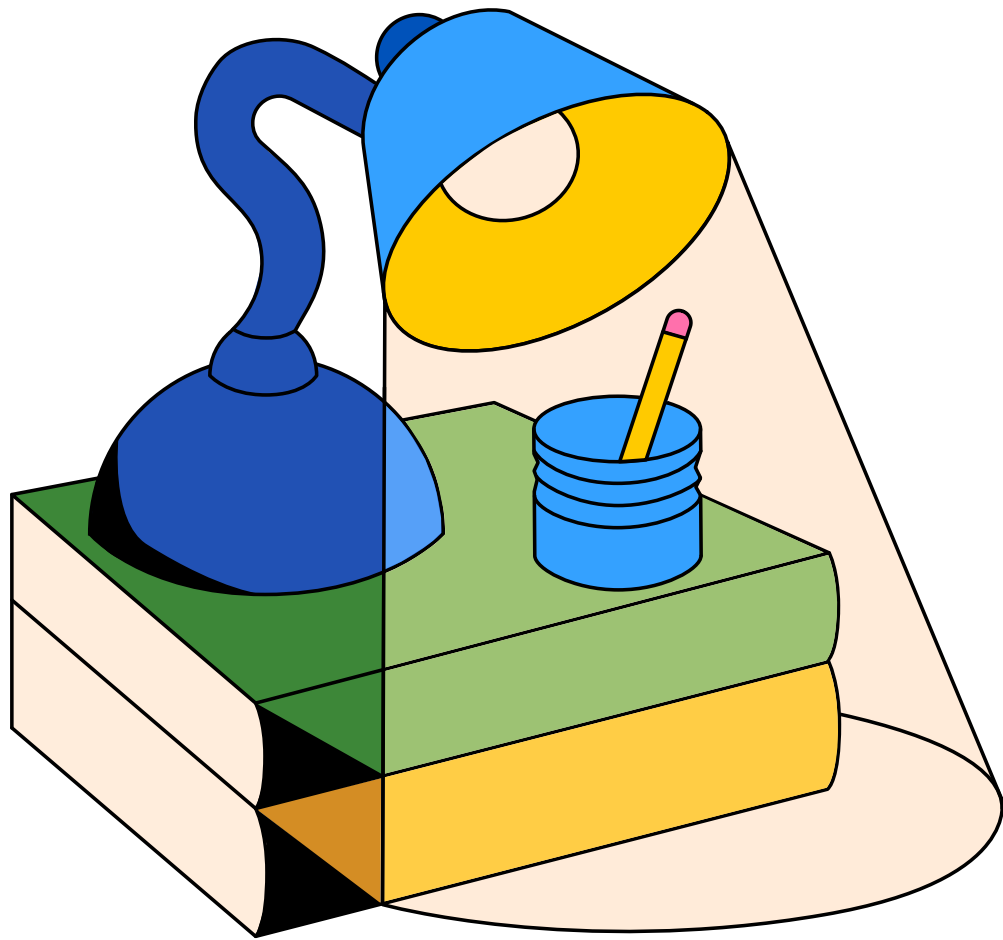
Static Analysis

- Measuring the **impact** of a change in the codebase



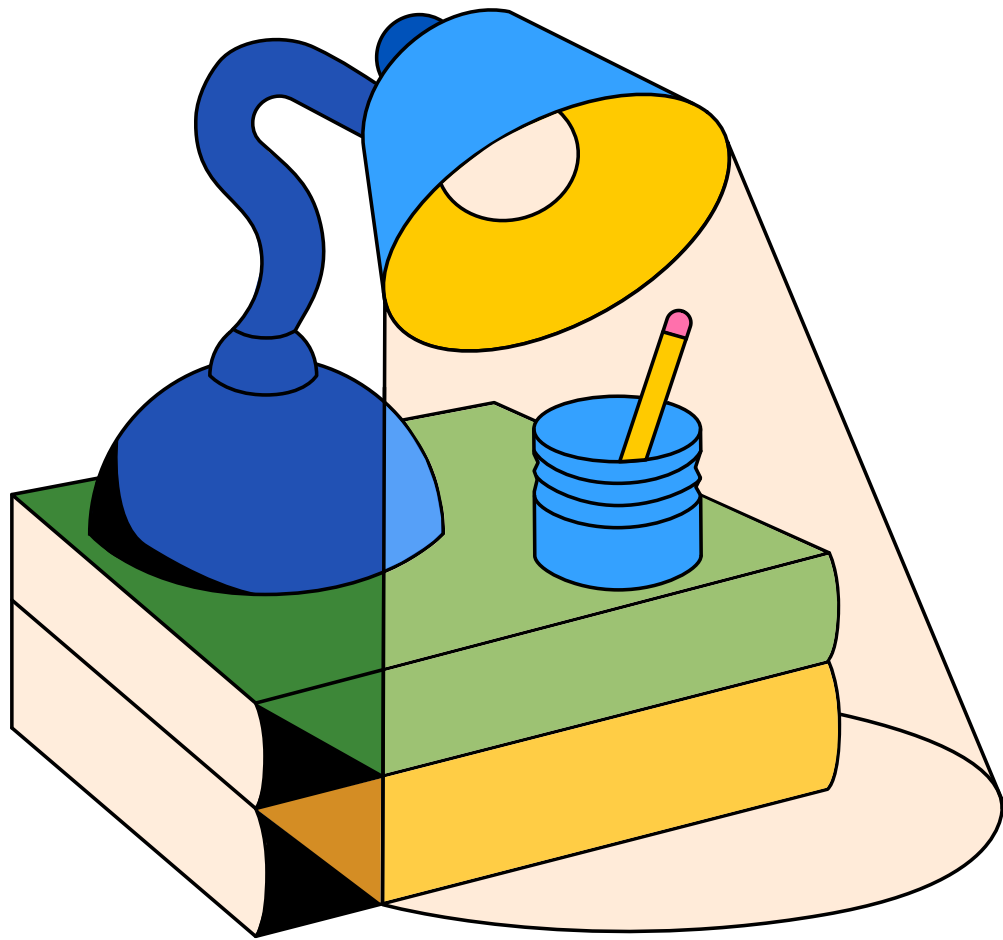
Static Analysis

- Measuring the **impact** of a change in the codebase
- Identifying the components most used



Static Analysis

- Measuring the **impact** of a change in the codebase
- Identifying the components most used
- Identifying the components requiring the **most maintenance**



Static Analysis

- Measuring the **impact** of a change in the codebase
- Identifying the components most used
- Identifying the components requiring the **most maintenance**
- **Those Metrics Are Not helping the leadership.**

Visual Coverage



Visual Coverage

- Measuring the impact on real pages



Visual Coverage



- Measuring the impact on real pages
- Identifying the pages most impacted by the DS



Visual Coverage

- Measuring the impact on real pages
- Identifying the pages most impacted by the DS
- Provides metrics that can be included in a Company's goals.

WHAT

are we doing

Requirements



Requirements

- It must be visual



Requirements

- It must be visual
- It must be **easy** to understand



Requirements

- It must be visual
- It must be **easy** to understand
- It must not be technical



Requirements

- It must be visual
- It must be **easy** to understand
- It must not be technical
- It must be **independent** from the Product teams.



We could color DOM elements...

...on the actual web page!



- Should we measure the usage of the DS tokens?



- Should we measure the usage of the DS tokens?
- Should we measure the usage of the DS components?



- Should we measure the usage of the DS tokens?
- Should we measure the usage of the DS components?
- What do we want to measure?
 - Areas
 - Borders
 - Or what?



We decided to draw lines around the components

Preply English, EUR

Home Messages My lessons For business

How's it going, Stefano?

Your lesson starts in 2 days

March 31
Tuesday · 17:00 – 17:50
English with Leila

Before your lesson

- Review Lesson Insights
- Complete your daily exercise · 3 mins
- Ask your tutor for homework
- Choose your course

Up next

March 31
Tuesday · 17:00 – 17:50

The visual coverage formula is:

$$\text{green pixels} / (\text{green pixels} + \text{red pixels})$$

The screenshot shows the Preply website interface with several elements highlighted by red and green boxes:

- Header:** Preply logo, Search tutors, Subscribe, 2 lessons to schedule, English, EUR, and user profile icons.
- Navigation:** Home, Messages, My lessons, and For business buttons.
- Message:** "How's it going, Stefano?"
- Lesson Card:** "Your lesson starts in 2 days", tutor profile (Leila), date (March 31), time (Tuesday 17:00 - 17:50), and "Join in 2 days" button.
- Pre-lesson tasks:** "Before your lesson" section with tasks: "Review Lesson Insights", "Complete your daily exercise • 3 mins", "Ask your tutor for homework", and "Choose your course".
- Up next:** "Up next" section with a lesson card for March 31, Tuesday 17:00 - 17:50.

The visual coverage formula is:

$$\text{green pixels} / (\text{green pixels} + \text{red pixels})$$

- It's a percentage value

The screenshot shows the Preply website interface with several elements highlighted by red and green boxes:

- Header:** Preply logo, Search tutors, Subscribe, 2 lessons to schedule, English, EUR, and user profile icons.
- Navigation:** Home, Messages, My lessons, and For business buttons.
- Message:** "How's it going, Stefano?"
- Lesson Card:** "Your lesson starts in 2 days", tutor profile (Leila), date (March 31), time (Tuesday 17:00 - 17:50), and "Join in 2 days" button.
- Before your lesson:** A list of tasks: "Review Lesson Insights", "Complete your daily exercise • 3 mins", "Ask your tutor for homework", and "Choose your course".
- Up next:** A section for the next lesson, showing the same date and time as the current lesson.

The visual coverage formula is:

$$\text{green pixels} / (\text{green pixels} + \text{red pixels})$$

- It's a percentage value
- It directly measures what the users see.

The leadership team liked the idea!

But as we discovered...

Not all the components' weights are the same

How's it going, Stefano?

Your lesson starts in 2 days



March 31

Tuesday · 17:00 – 17:50

English with Leila

Join in 2 days

Before your lesson



Review Lesson Insights



Complete your daily exercise · 3 mins



Ask your tutor for homework



Choose your course



Up next

March 31

Tuesday · 17:00 – 17:50



Preply English, EUR

Home Messages My lessons For business

How's it going, Stefano?

Your lesson starts in 2 days

March 31
Tuesday · 17:00 – 17:50
English with Leila

Before your lesson

- Review Lesson Insights
- Complete your daily exercise · 3 mins
- Ask your tutor for homework
- Choose your course

Up next

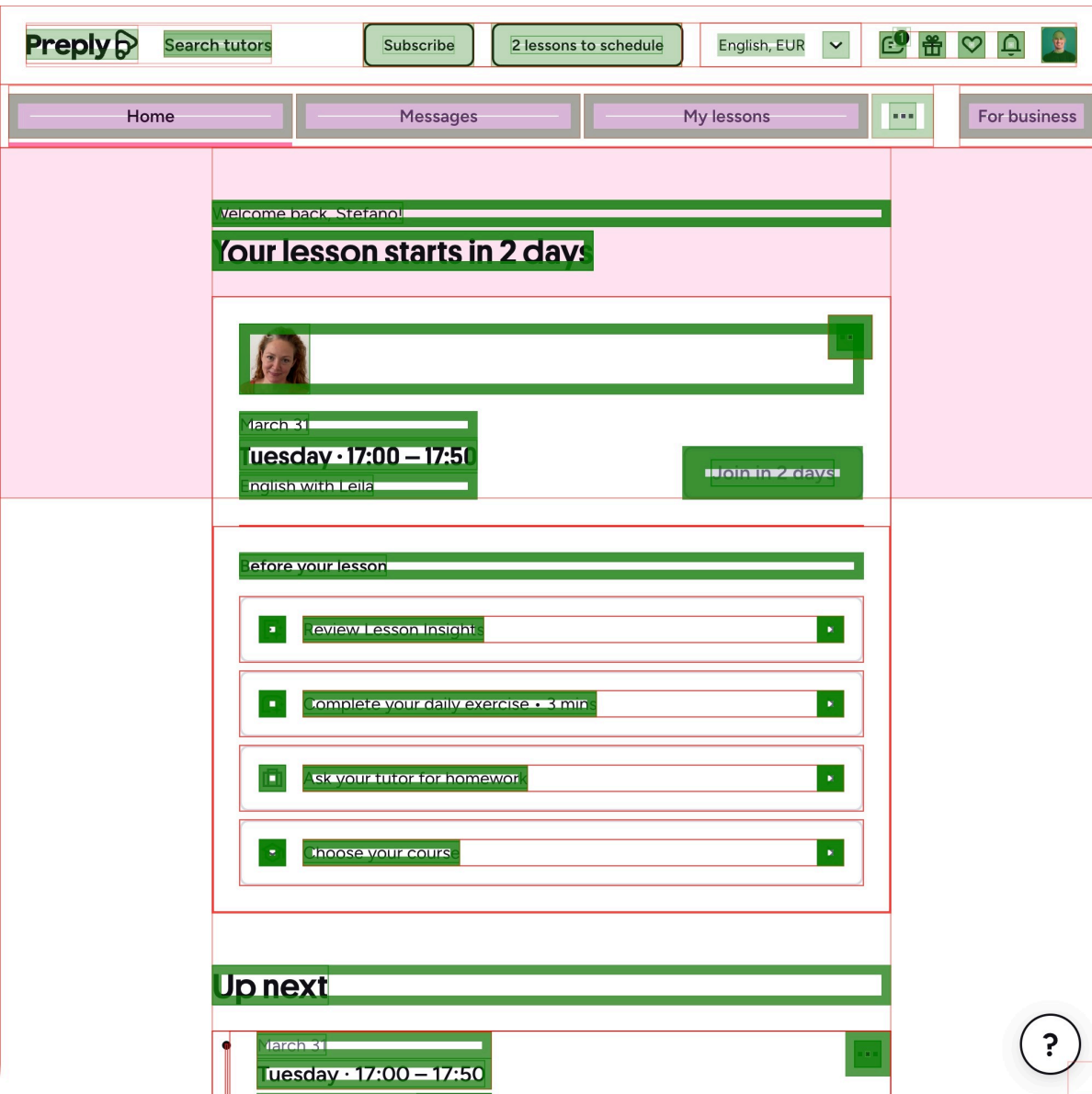
March 31

Tuesday · 17:00 – 17:50

- The coverage must express the impact on the user (UI/UX/A11y)

The screenshot displays the Preply user interface. At the top, there is a navigation bar with the Preply logo, a search bar for tutors, a 'Subscribe' button, a '2 lessons to schedule' indicator, a language/currency selector set to 'English, EUR', and icons for notifications, gifts, and a profile. Below this is a secondary navigation bar with 'Home', 'Messages', 'My lessons', and 'For business' options. The main content area features a welcome message 'Welcome back, Stefano!', followed by a prominent reminder: 'Your lesson starts in 2 days'. Below the reminder is a card for a lesson with a tutor's profile picture, the date 'March 31', the time 'Tuesday • 17:00 – 17:50', and the subject 'English with Leila'. A 'Join in 2 days' button is also present. Underneath is a section titled 'Before your lesson' containing a list of tasks: 'Review Lesson Insights', 'Complete your daily exercise • 3 min', 'Ask your tutor for homework', and 'Choose your course'. At the bottom, there is an 'Up next' section with a lesson card for 'Tuesday • 17:00 – 17:50'. A help icon (question mark in a circle) is located in the bottom right corner of the interface.

- The coverage must express the impact on the user (UI/UX/A11y)



- The coverage must express the impact on the user (UI/UX/A11y)
- The weight of a component is subjective and unique per Product.

Better, but not enough!

My lessons

+ Schedule lesson

Refer your company
Let your employer pay for your lessons

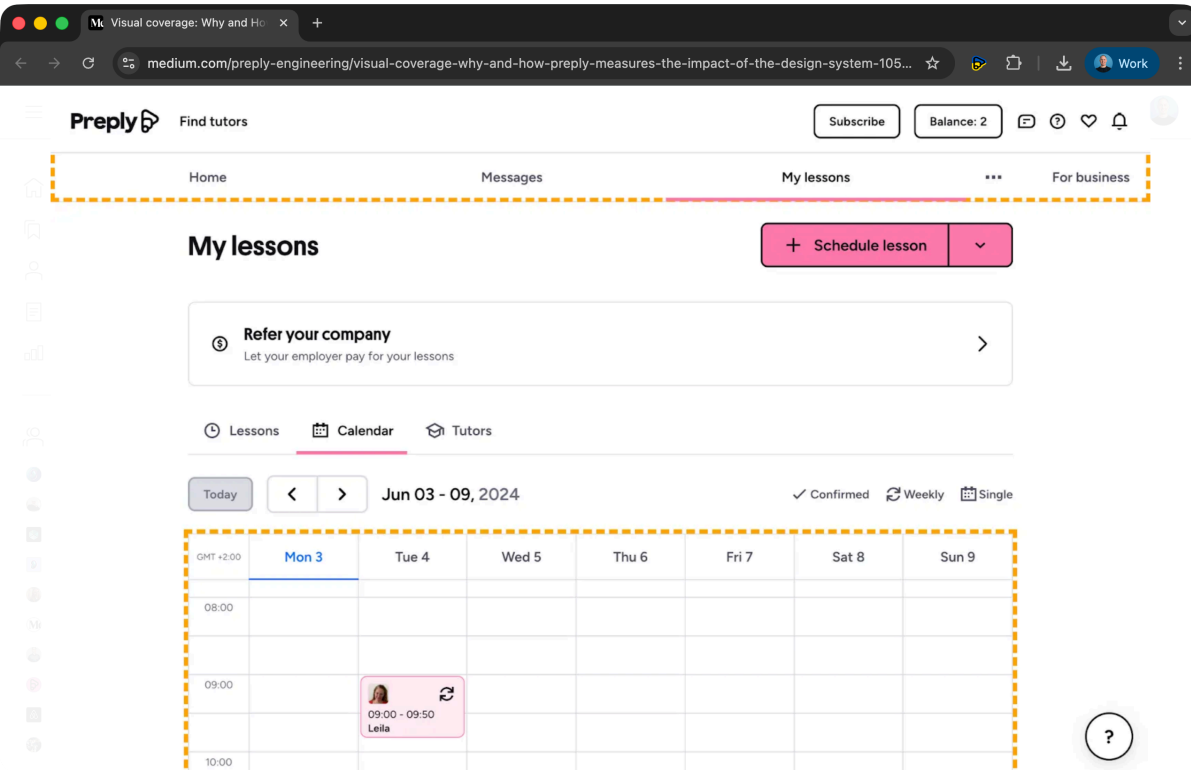
Lessons **Calendar** Tutors

Today < > Jun 03 - 09, 2024 ✓ Confirmed ↻ Weekly 📅 Single

GMT +2:00	Mon 3	Tue 4	Wed 5	Thu 6	Fri 7	Sat 8	Sun 9
08:00							
09:00							
10:00							



- Multiple teams can work on the same pages



The screenshot shows the Preply website interface. At the top, there's a navigation bar with 'Home', 'Messages', 'My lessons', and 'For business'. Below this, the 'My lessons' section features a '+ Schedule lesson' button and a 'Refer your company' card. A navigation bar below that includes 'Lessons', 'Calendar', and 'Tutors'. The main content is a calendar for 'Jun 03 - 09, 2024'. The calendar grid shows a lesson scheduled for Monday, June 3rd, from 09:00 to 09:50, with the tutor 'Lella'. The calendar is framed by a dashed orange border. A circular help icon with a question mark is visible at the bottom right of the calendar area.

- Multiple teams can work on the same pages
- A page coverage must be calculated per team.

HOW

are we doing it

First attempt

First attempt

- Color the borders via CSS

First attempt

- Color the borders via CSS
- Transform DOM into a PNG

First attempt

- Color the borders via CSS
- Transform DOM into a PNG
- Count the pixels.

Did it work?

Did it work?

- Yes!

Did it work?

- Yes!
- But it was slow

Did it work?

- Yes!
- But it was slow
- It took up to 5 seconds on a high-end device...

Did it work?

- Yes!
- But it was slow
- It took up to 5 seconds on a high-end device...
- ...for a single page

Did it work?

- Yes!
- But it was slow
- It took up to 5 seconds on a high-end device...
- ...for a single page
- and it was blocking the main thread

Did it work?

- Yes!
- But it was slow
- It took up to 5 seconds on a high-end device...
- ...for a single page
- and it was blocking the main thread
- No way to run it on any CI pipeline.

What If...

What If...

...we change strategy?

It may sound wild, but...



It may sound wild, but...

- If we make the process fast enough



It may sound wild, but...

- If we make the process **fast enough**
- Without blocking the main thread



It may sound wild, but...

- If we make the process fast enough
- Without blocking the main thread
- We could run it in production



It may sound wild, but...

- If we make the process fast enough
- Without blocking the main thread
- We could run it in production
- Measure the coverage in real time



It may sound wild, but...

- If we make the process fast enough
- Without blocking the main thread
- We could run it in production
- Measure the coverage in real time
- On the users' devices.



A few benefits

A few benefits

- Users navigate to all the pages (the E2E tests cover mostly happy paths)

A few benefits

- Users navigate to all the pages (the E2E tests cover mostly happy paths)
- Measuring what the users see aligns with the business goals

A few benefits

- Users navigate to all the pages (the E2E tests cover mostly happy paths)
- Measuring what the users see aligns with the business goals
- More significant data set to collect (like page popularity)

A few benefits

- Users navigate to all the pages (the E2E tests cover mostly happy paths)
- Measuring what the users see aligns with the business goals
- More significant data set to collect (like page popularity)
- Independence from the Product teams.

Step by step

Hello world

Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum id odio metus. Integer tempus lectus leo, eget scelerisque mi finibus ac. Ultrices est. In scelerisque, sapien vitae vehicula posuere, nulla ante facilisis risus.

button

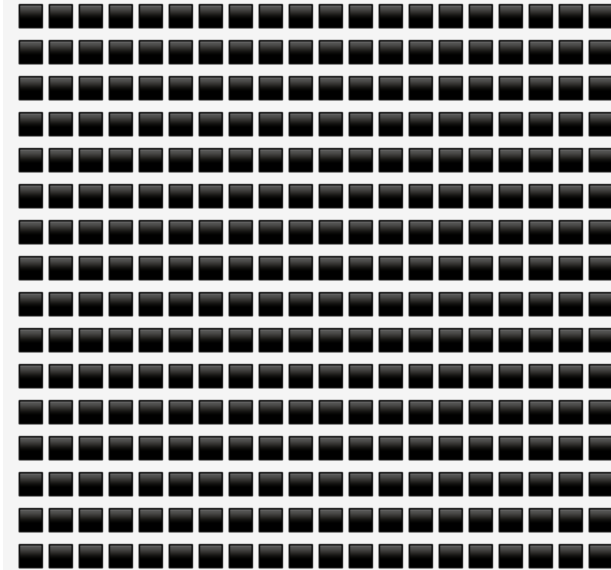
DS button

DOM parse result

```
[
```

```
]
```

Bitmap



Step by step

Hello world

Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum id odio metus. Integer tempus lectus leo, eget scelerisque mi finibus ac. Ultrices est. In scelerisque, sapien vitae vehicula posuere, nulla ante facilisis risus.

button

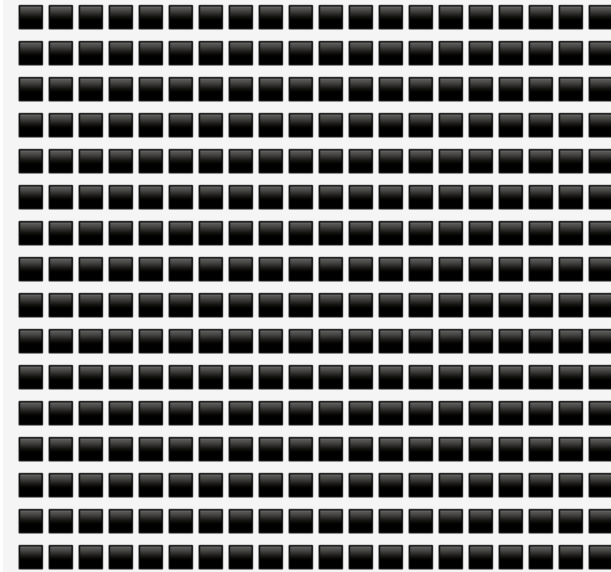
DS button

DOM parse result

```
[
```

```
]
```

Bitmap



Step by step

Hello world

Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum id odio metus. Integer tempus lectus leo, eget scelerisque mi finibus ac. Ultrices est. In scelerisque, sapien vitae vehicula posuere, nulla ante facilisis risus.

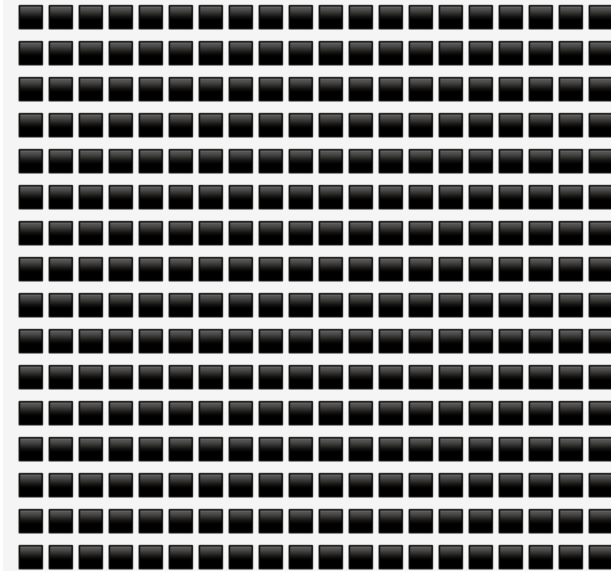
button

DS button

DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] }  
]
```

Bitmap



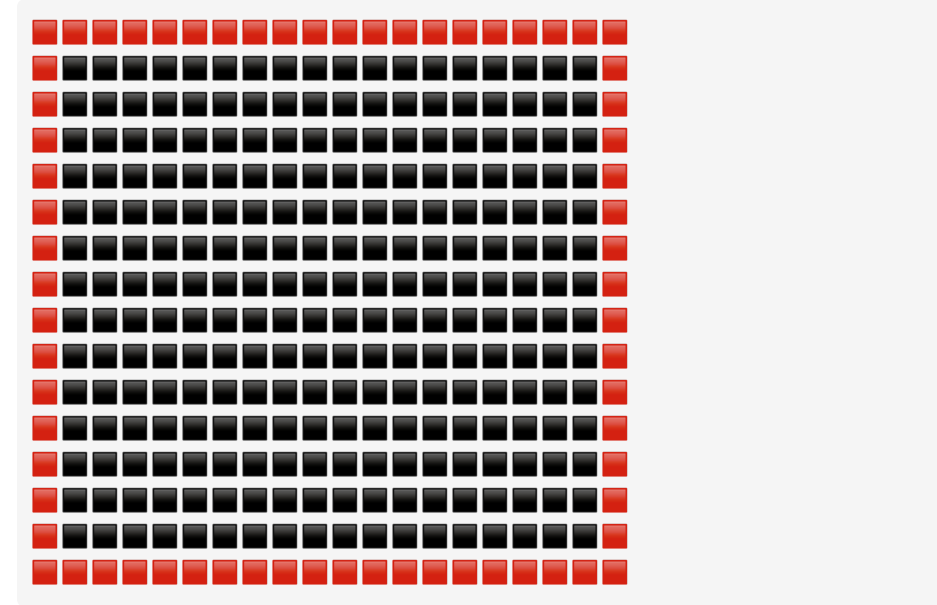
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] }  
]
```

Bitmap



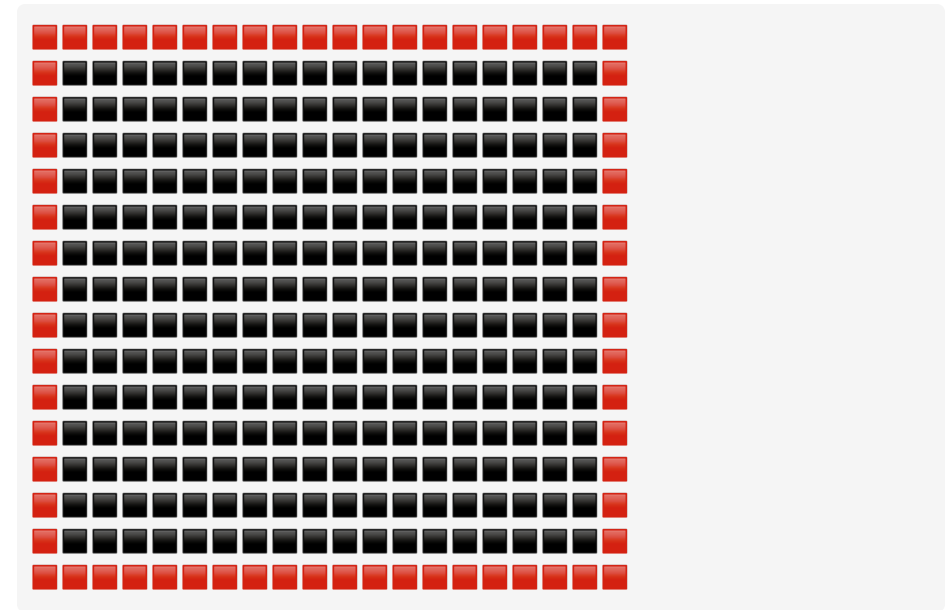
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  
]
```

Bitmap



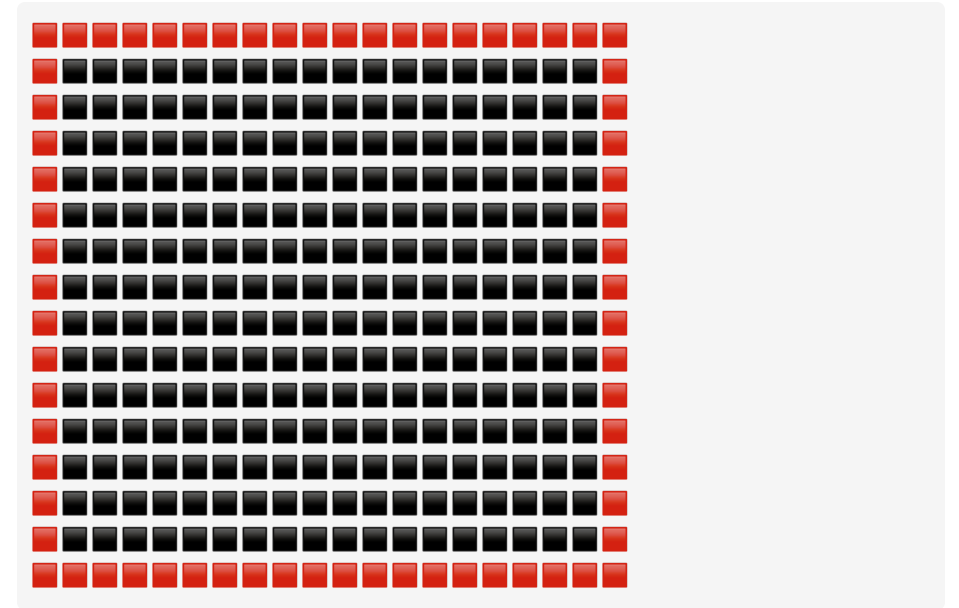
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" }  
]
```

Bitmap



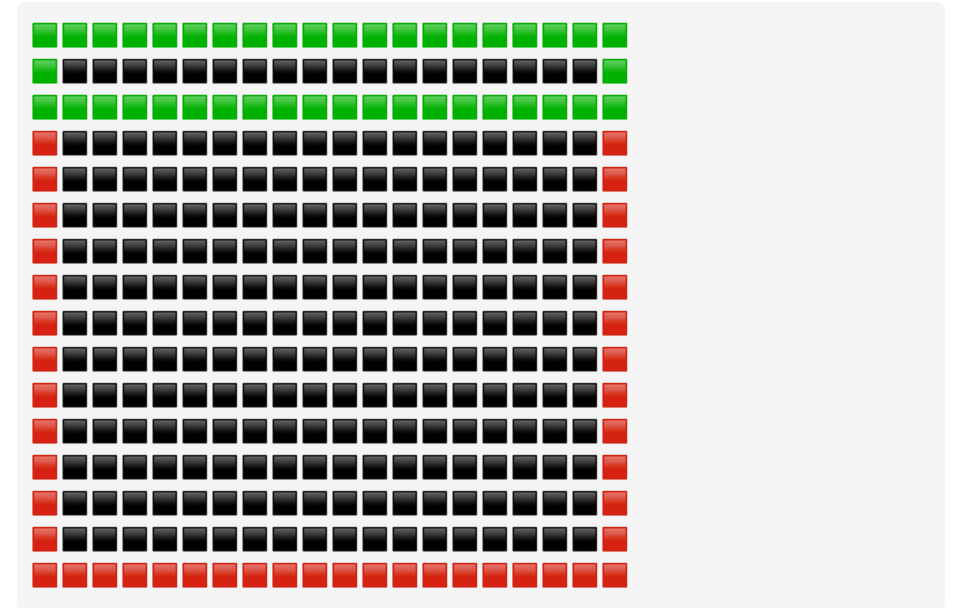
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" }  
]
```

Bitmap



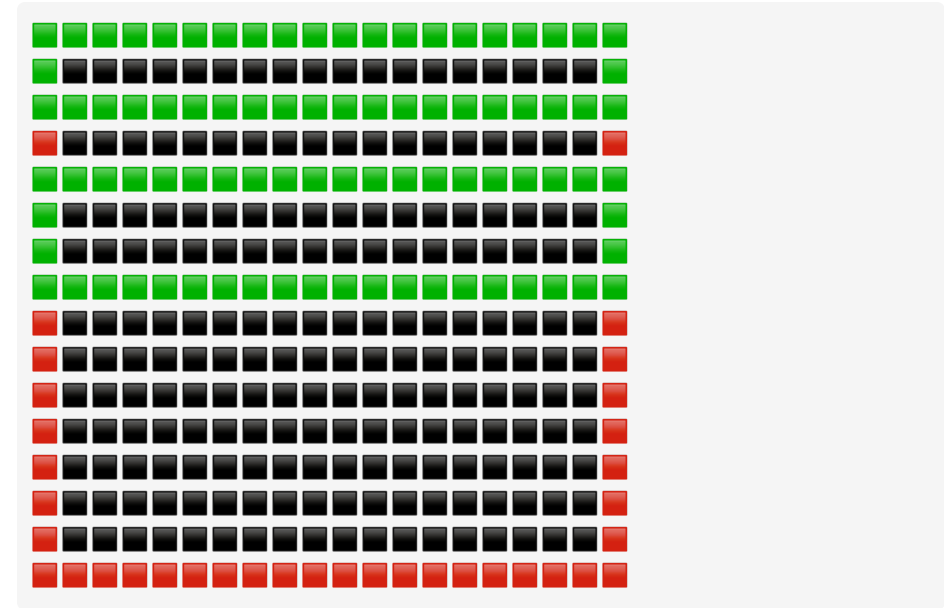
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
]
```

Bitmap



Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] }  
]
```

Bitmap



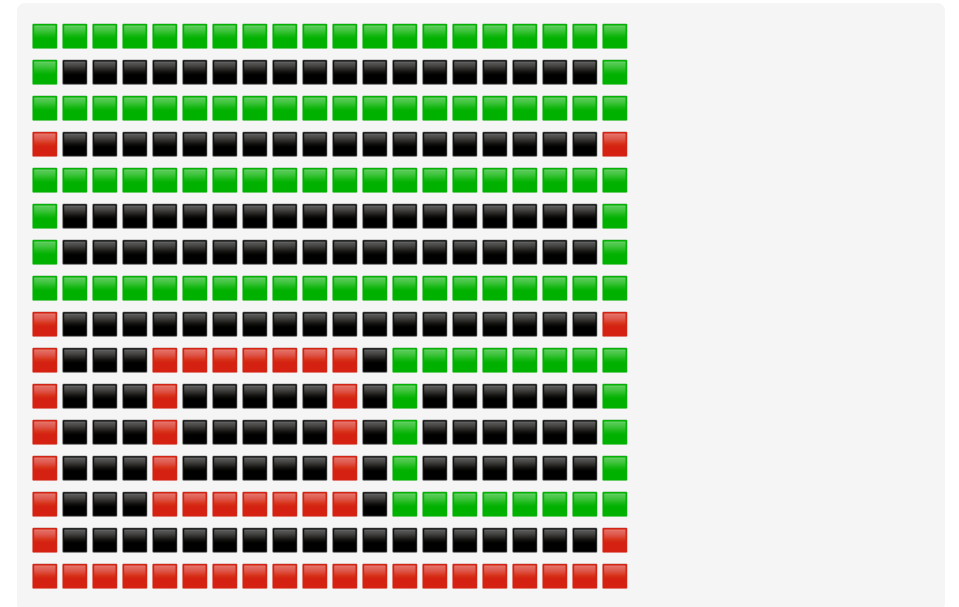
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] },  
  { "rect": [16, 12, 8, 4], "id": "Button" },  
]
```

Bitmap



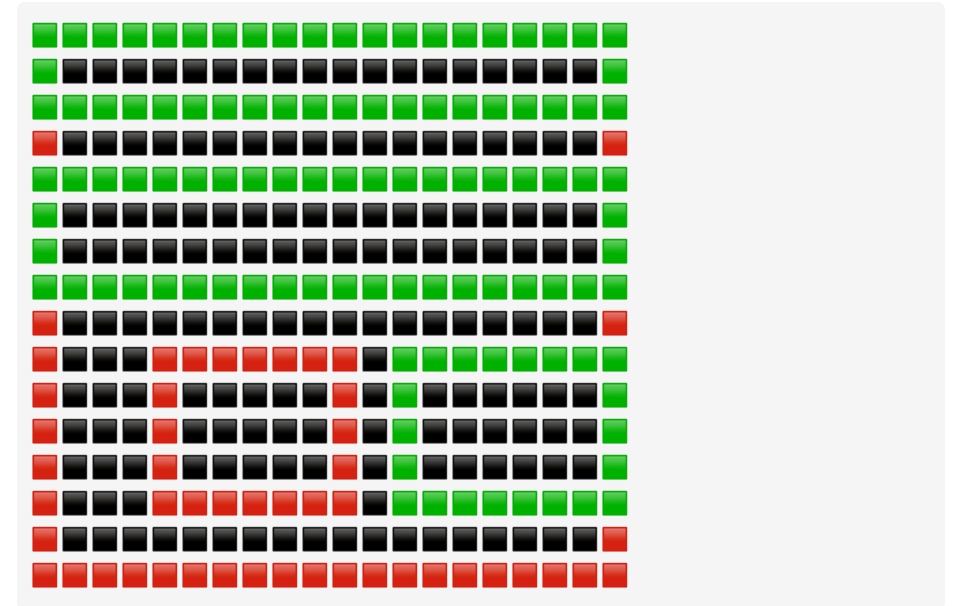
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] },  
  { "rect": [16, 12, 8, 4], "id": "Button" },  
]
```

Bitmap



Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] },  
  { "rect": [16, 12, 8, 4], "id": "Button", "weight": 2 },  
]
```

Bitmap



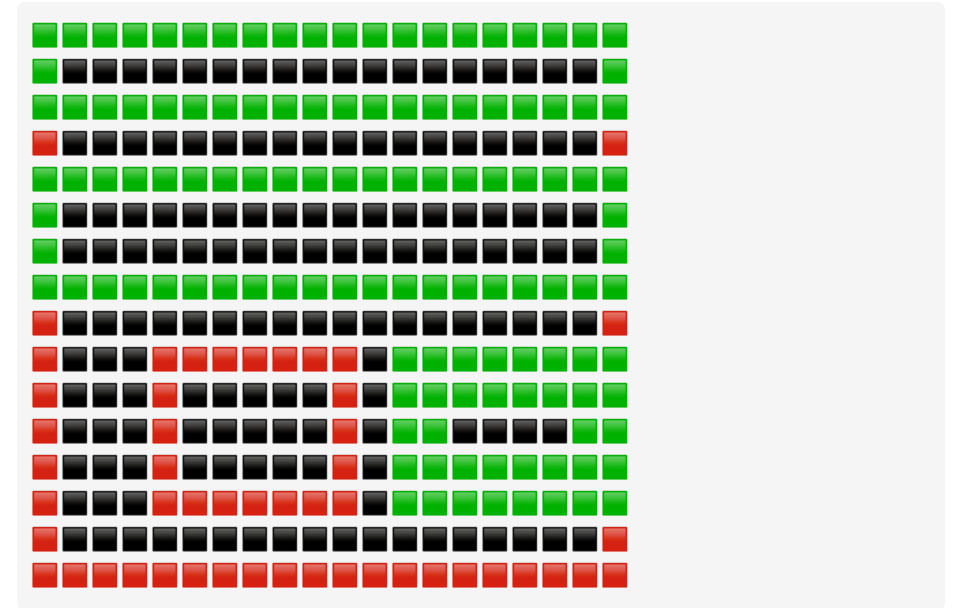
Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] },  
  { "rect": [16, 12, 8, 4], "id": "Button", "weight": 2 },  
]
```

Bitmap



Step by step



DOM parse result

```
[  
  { "rect": [0, 0, 20, 16] },  
  { "rect": [0, 0, 20, 3], "id": "Heading" },  
  { "rect": [8, 0, 20, 4], "id": "Text" },  
  { "rect": [16, 3, 8, 4] },  
  { "rect": [16, 12, 8, 4], "id": "Button", "weight": 2 },  
]
```

Bitmap

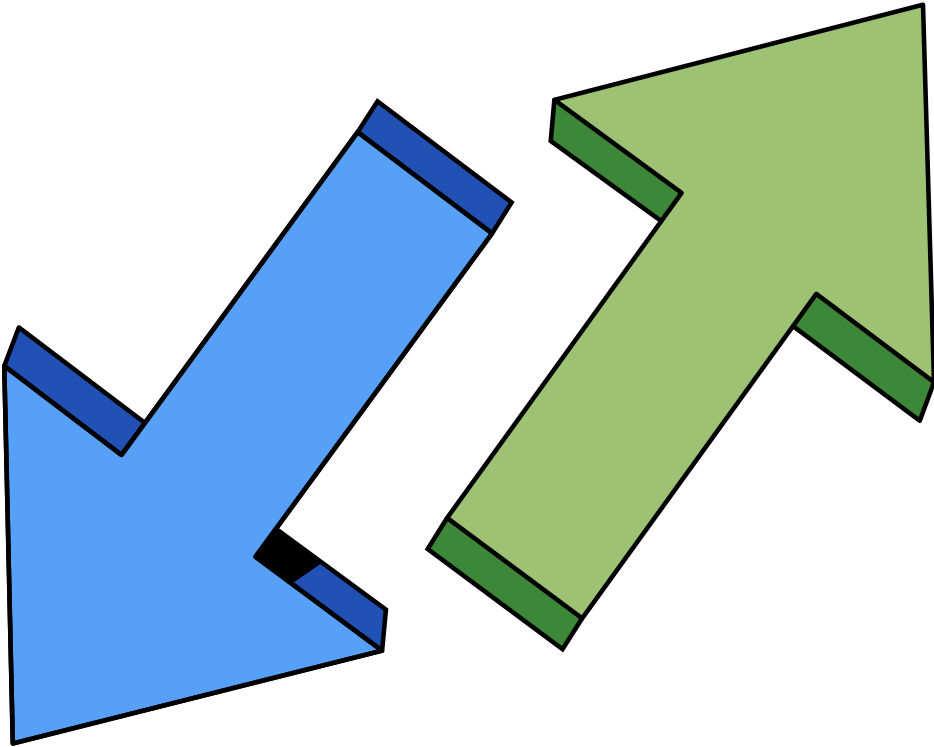


■ = 122
■ = 51

DS visual coverage = 70.5 %

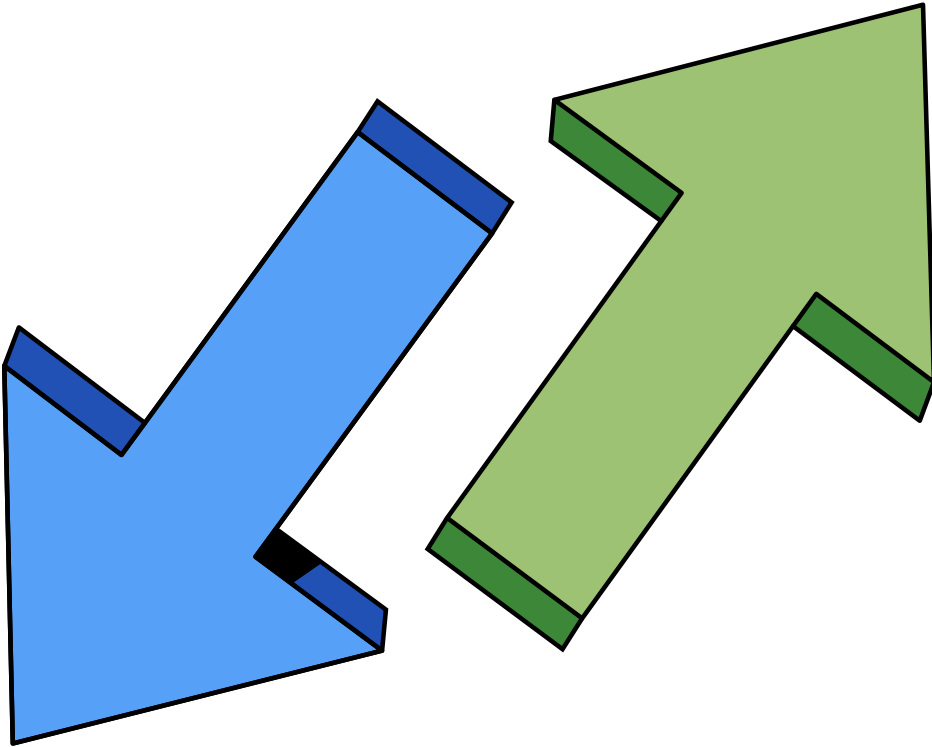
It's a two-step process

Analyze



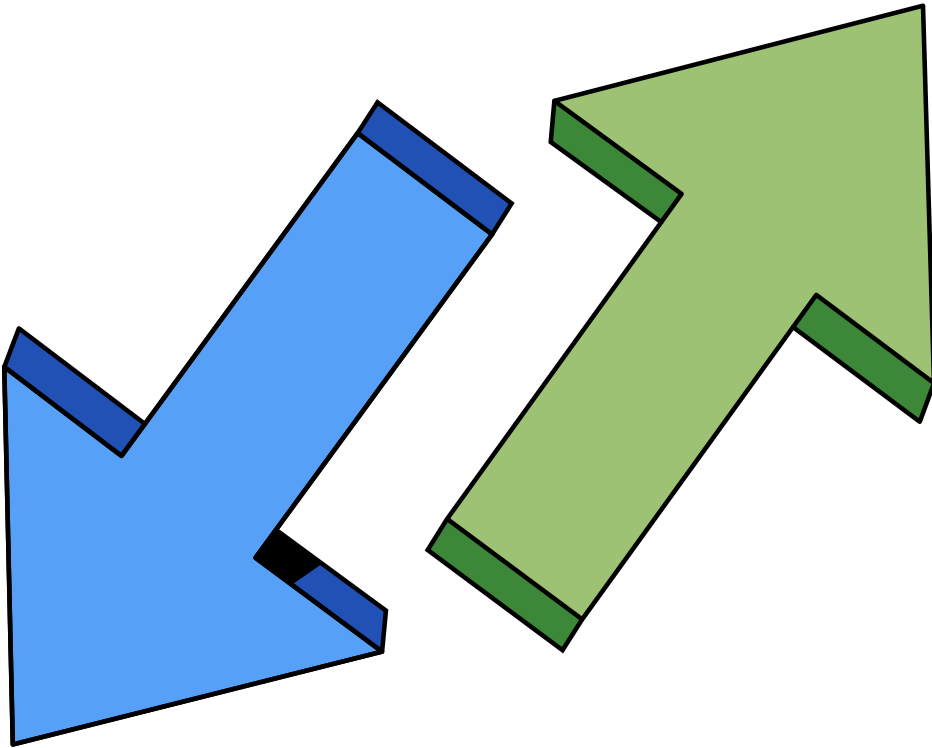
Analyze

- Traverse the DOM



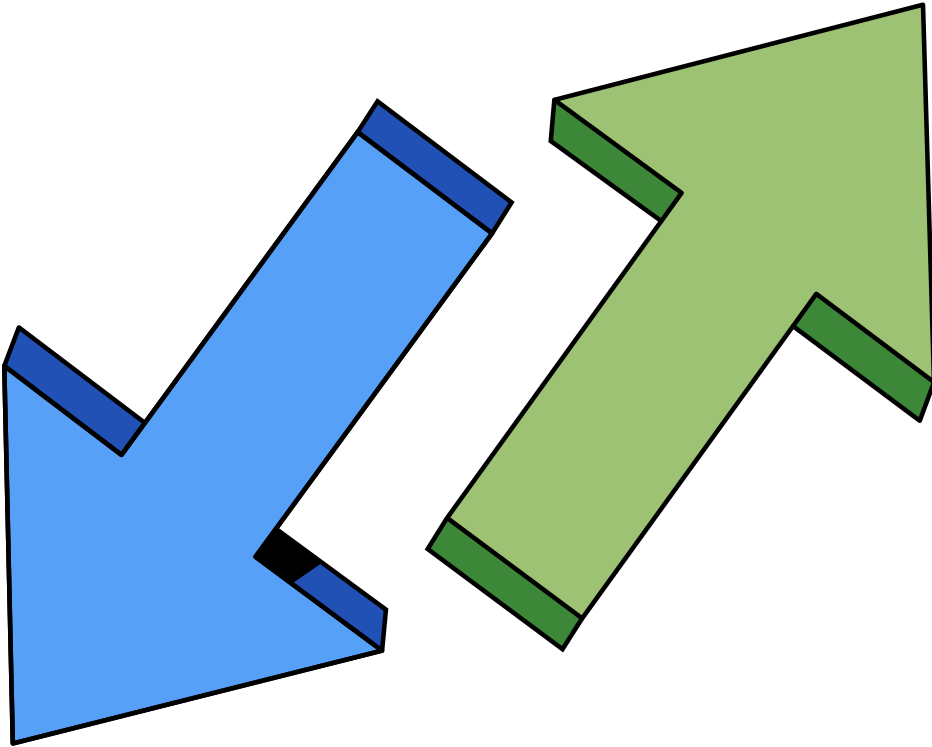
Analyze

- Traverse the DOM
- Collect elements' size and position

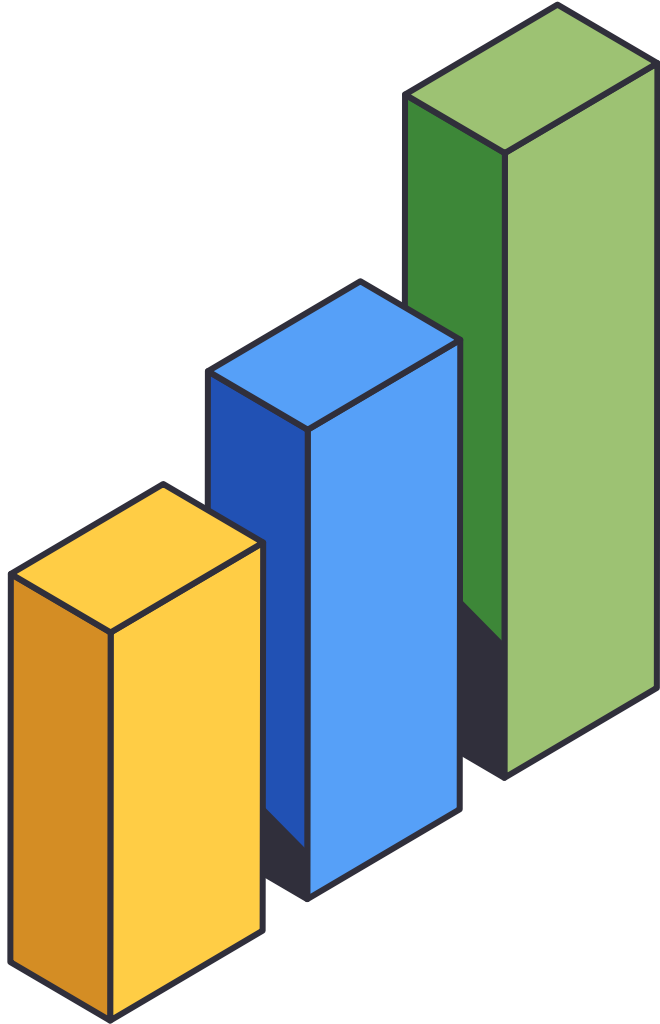


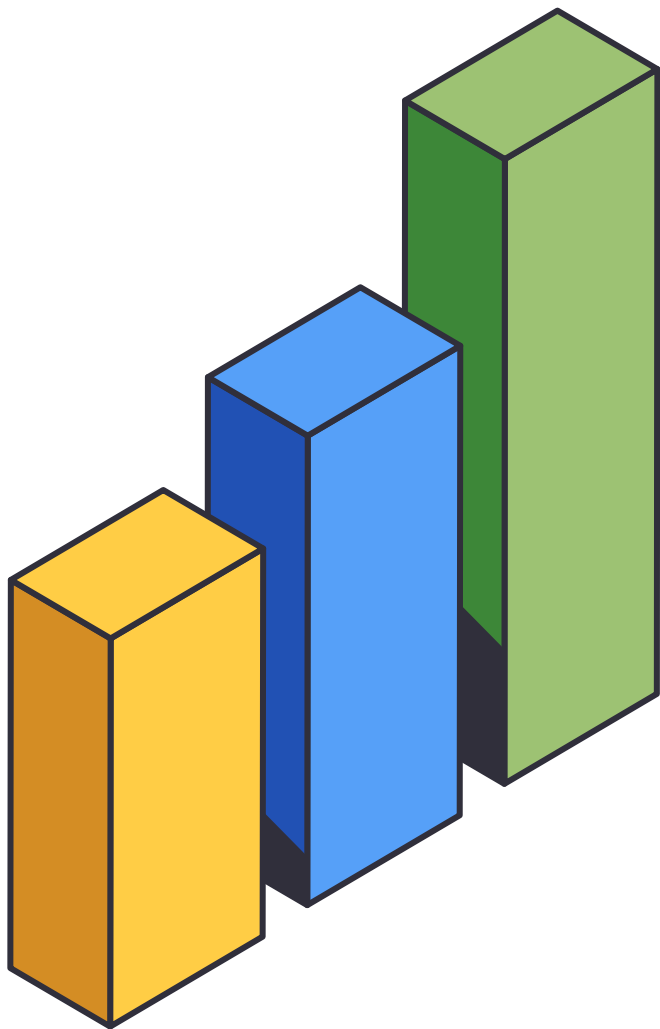
Analyze

- Traverse the DOM
- Collect elements' size and position
- Detect which elements are part of the DS.



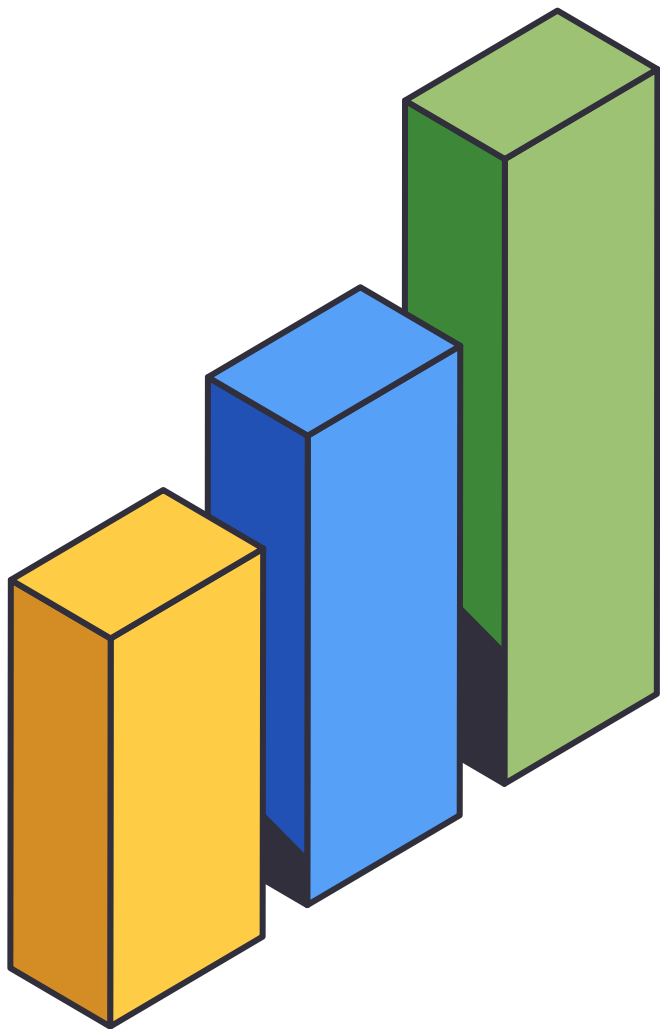
Optimize the parser





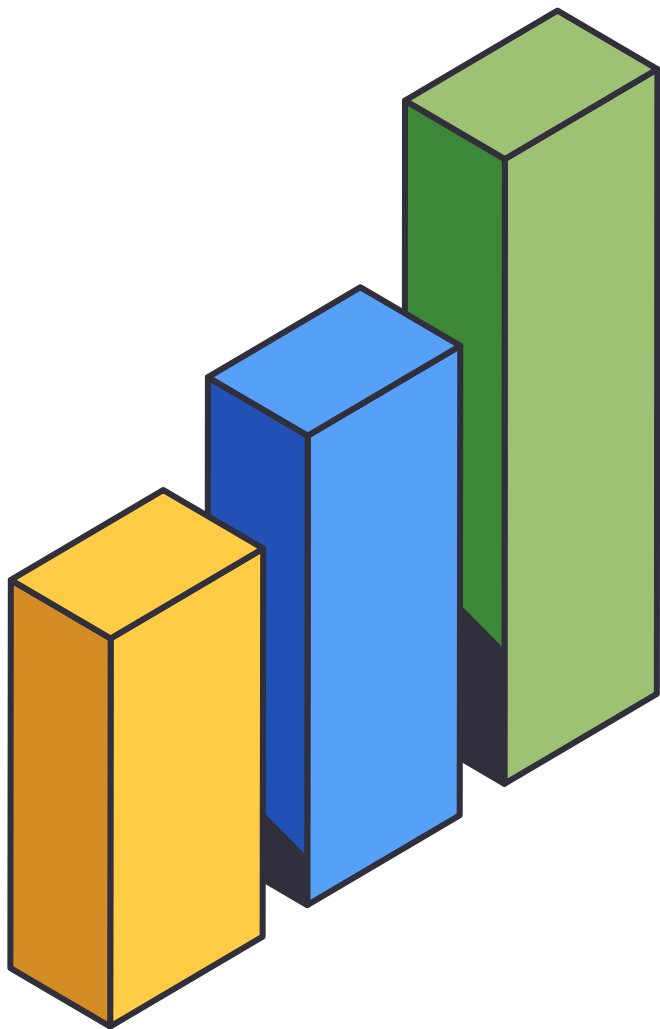
Optimize the parser

- Traverse using Idle Callbacks



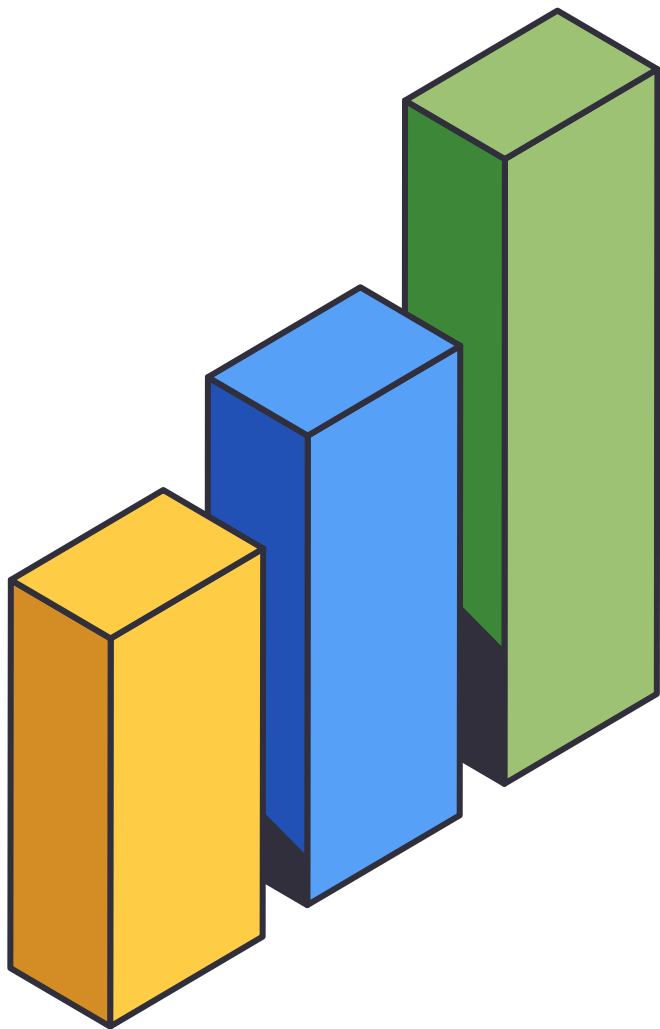
Optimize the parser

- Traverse using Idle Callbacks
- The browser provides a time window to act



Optimize the parser

- Traverse using Idle Callbacks
- The browser provides a time window to act
- The parser **pauses and resumes** as needed



Optimize the parser

- Traverse using Idle Callbacks
- The browser provides a time window to act
- The parser **pauses and resumes** as needed
- The parser **accepts** incomplete DOM trees.

Draw the bitmap



Draw the bitmap

- Transform the parsed DOM into a bi-dimensional array



Draw the bitmap

- Transform the parsed DOM into a bi-dimensional array
- The bitmap has the same size of the viewport



Draw the bitmap

- Transform the parsed DOM into a **bi-dimensional array**
- The bitmap has the same size of the viewport
- Pixels are **color-coded**



Draw the bitmap

- Transform the parsed DOM into a **bi-dimensional array**
- The bitmap has the same size of the viewport
- Pixels are **color-coded**
- The closest-to-the-user elements are drawn on top

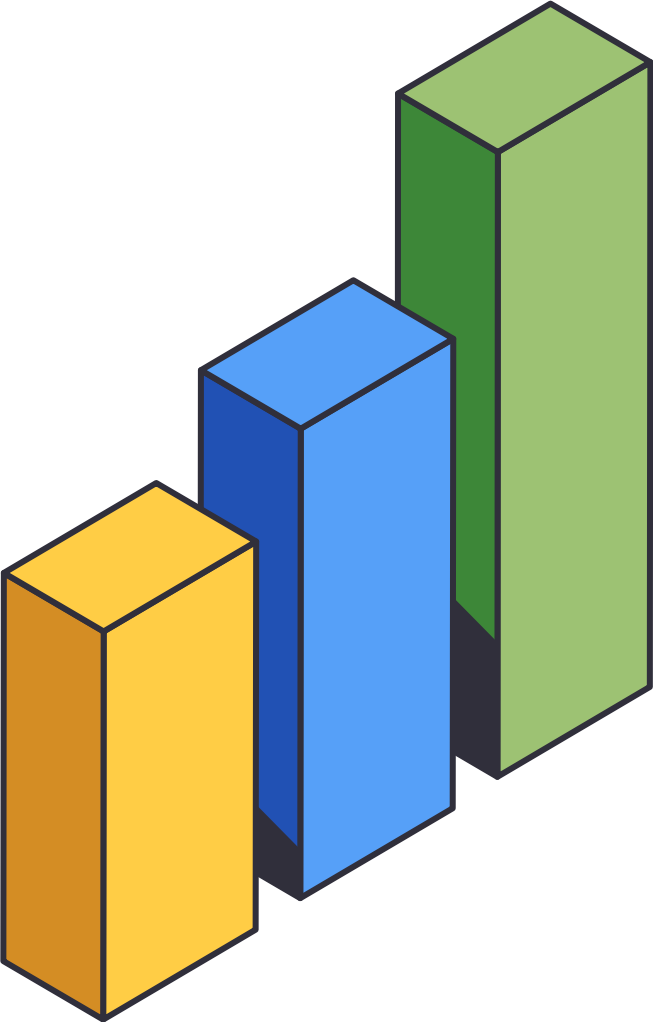


Draw the bitmap

- Transform the parsed DOM into a **bi-dimensional array**
- The bitmap has the same size of the viewport
- Pixels are **color-coded**
- The closest-to-the-user elements are drawn on top
- A component's weight affects the **border thickness.**

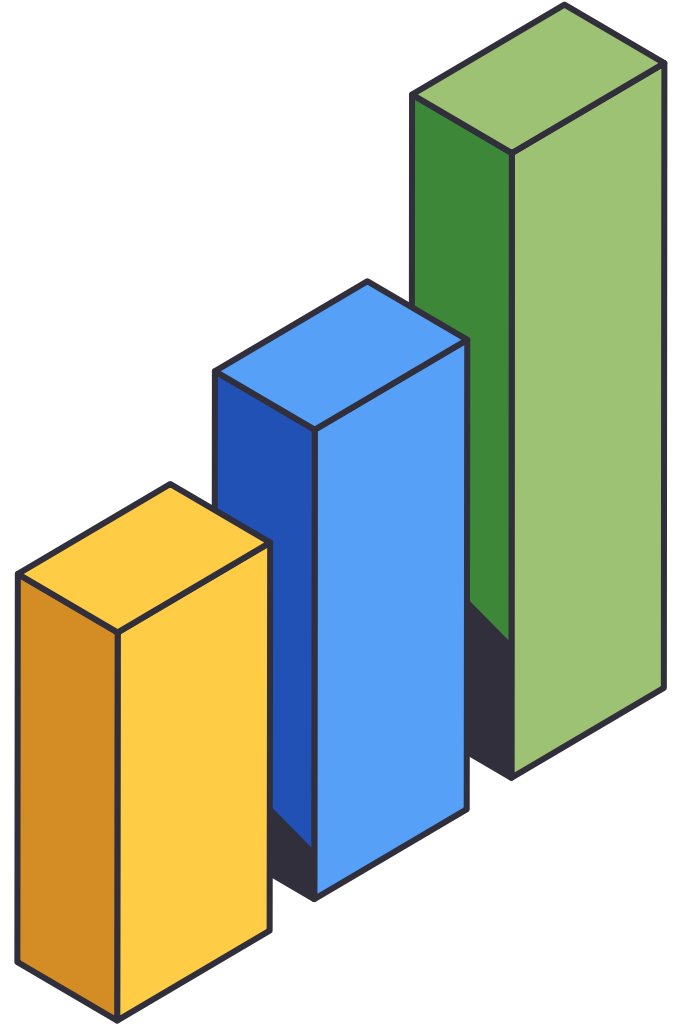


Bitmap Optimization



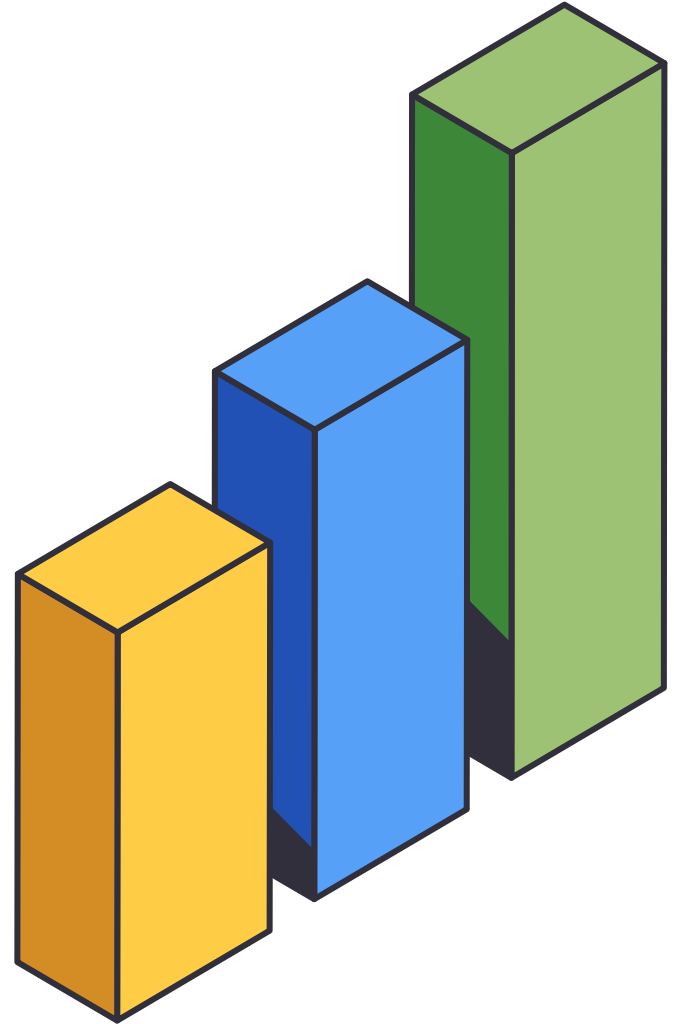
Bitmap Optimization

- We got help from an expert:
Massimiliano Mantione



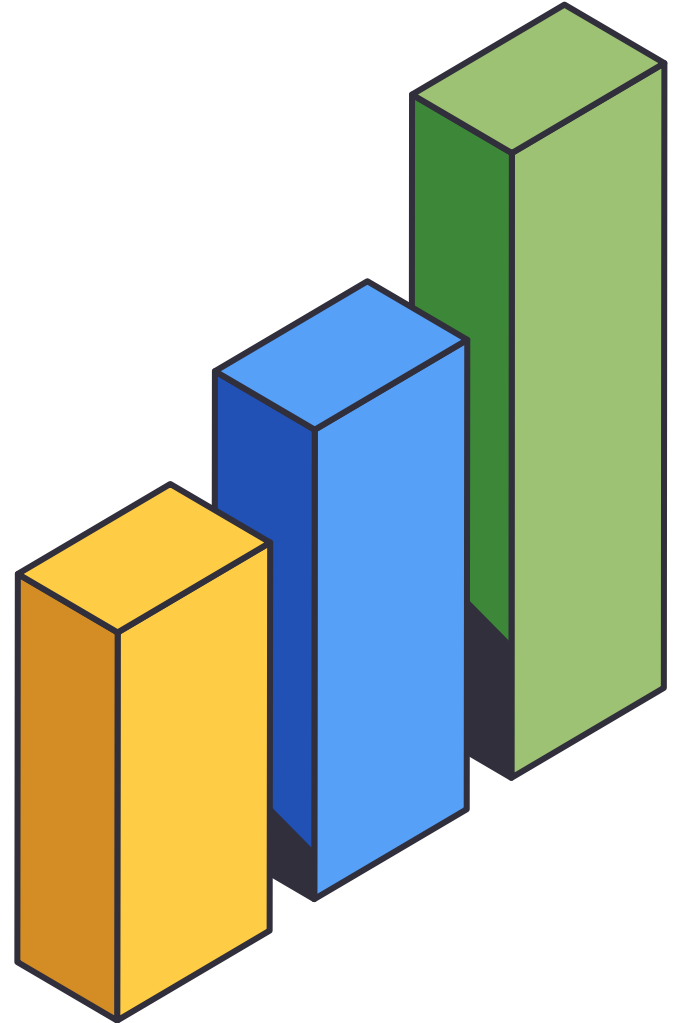
Bitmap Optimization

- We got help from an expert: Massimiliano Mantione
- Transform the array to be mono-dimensional



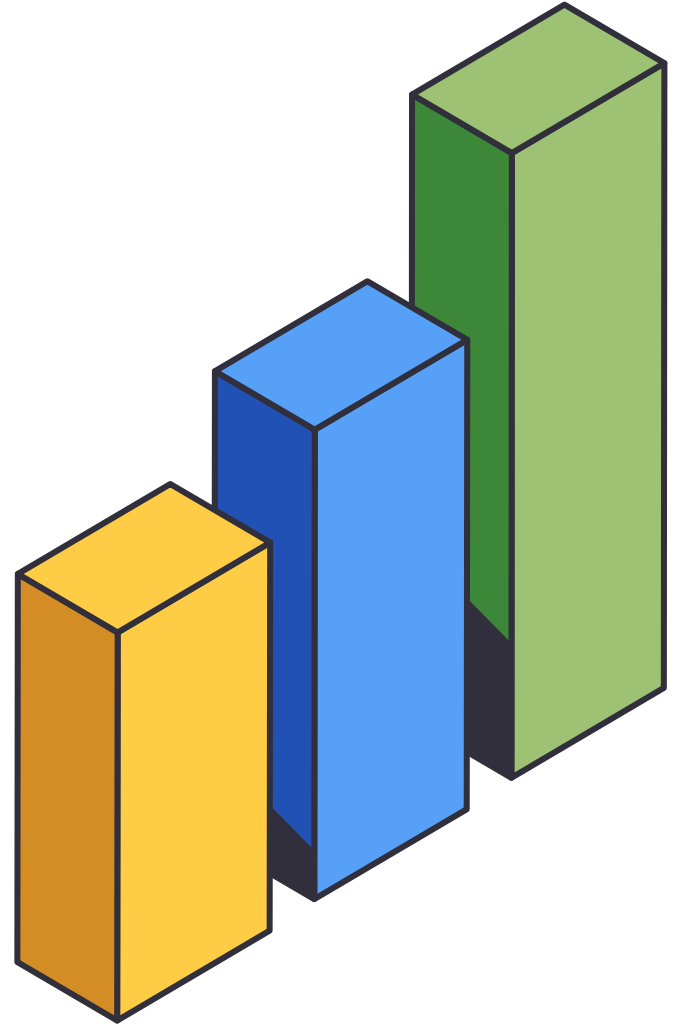
Bitmap Optimization

- We got help from an expert: Massimiliano Mantione
- Transform the array to be mono-dimensional
- Switch to Typed Arrays



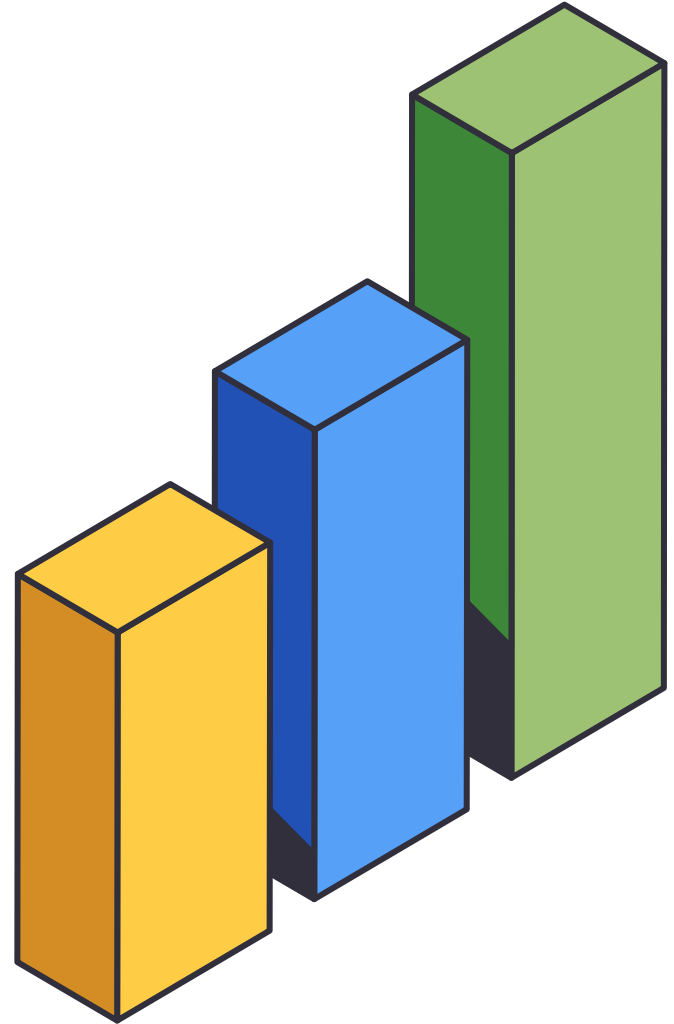
Bitmap Optimization

- We got help from an expert: Massimiliano Mantione
- Transform the array to be mono-dimensional
- Switch to Typed Arrays
- Make the code branch-free, removing most conditional statements



Bitmap Optimization

- We got help from an expert: **Massimiliano Mantione**
- Transform the array to be **mono-dimensional**
- Switch to **Typed Arrays**
- Make the code branch-free, removing most conditional statements
- Run the bitmap drawing in a **Web Worker**.



This is all great, but...

This is all great, but...
...show me the numbers!

- **X** The initial implementation took 5 blocking seconds

- ❌ The initial implementation took 5 blocking seconds
- 🚧 The first bitmap-based version took 50 blocking ms, and 30MB of memory

- ❌ The initial implementation took 5 blocking seconds
- 🚧 The first bitmap-based version took 50 blocking ms, and 30MB of memory
- ✅ The final version took 35 non-blocking ms, and 0.3 MB of memory

- ❌ The initial implementation took 5 blocking seconds
- 🚧 The first bitmap-based version took 50 blocking ms, and 30MB of memory
- ✅ The final version took 35 non-blocking ms, and 0.3 MB of memory
- ⓘ *The Huawei P9 takes 900 non-blocking ms on a big page.*

Was it worth it?

Yes!

Yes!

- It can run on any web application (100% plain JS)

Yes!

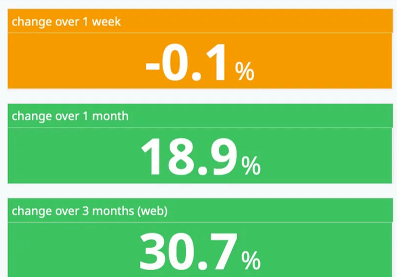
- It can run on any web application (100% plain JS)
- It's configurable
 - You are responsible to identify the DS components
 - You are responsible to set the components' weight.



The DS coverage on preply.com is weighted on page popularity, and it's impacted by the device size and the UGC.

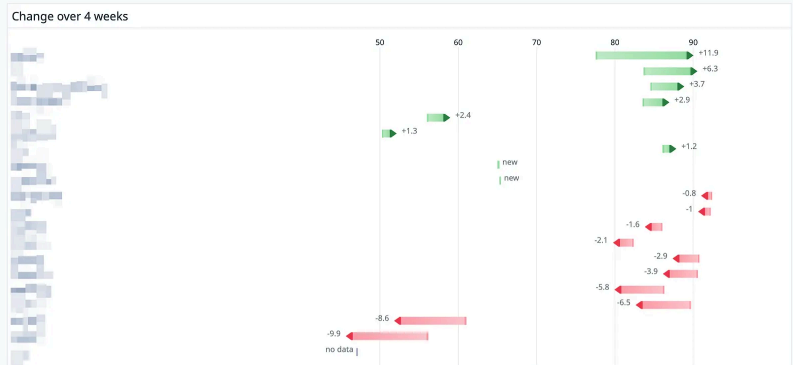
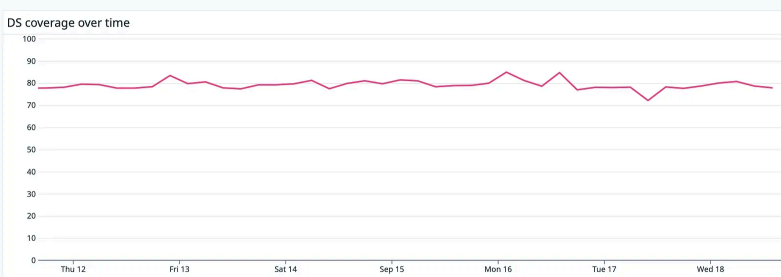
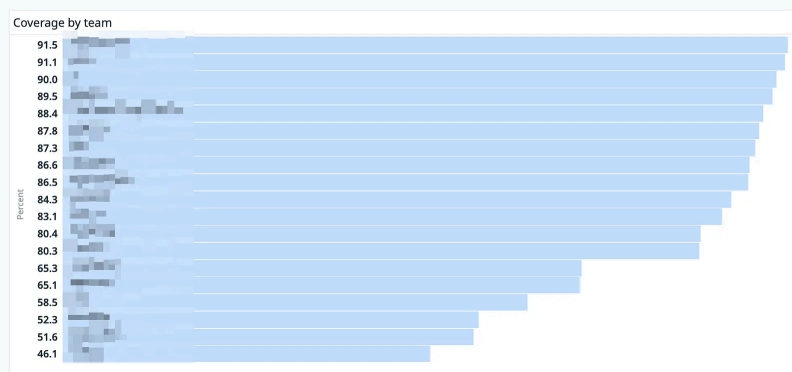


The DS coverage on app is counted inside the iOS EZE tests.

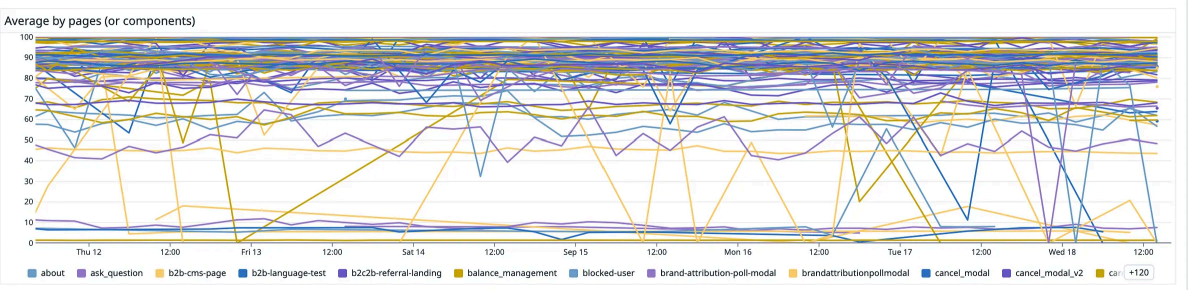
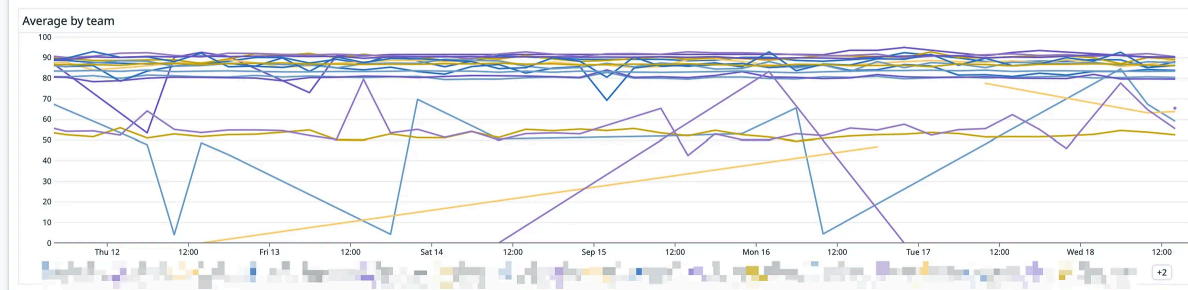


Official Path Design System website: [preply.com](#)

- If you need to count, analyze or increase the DS coverage in your page, look at [Design System visual coverage tools](#)
- If you want to know how the DS coverage is calculated, look at [How is the DS visual coverage calculated?](#)



Coverage by slices over time

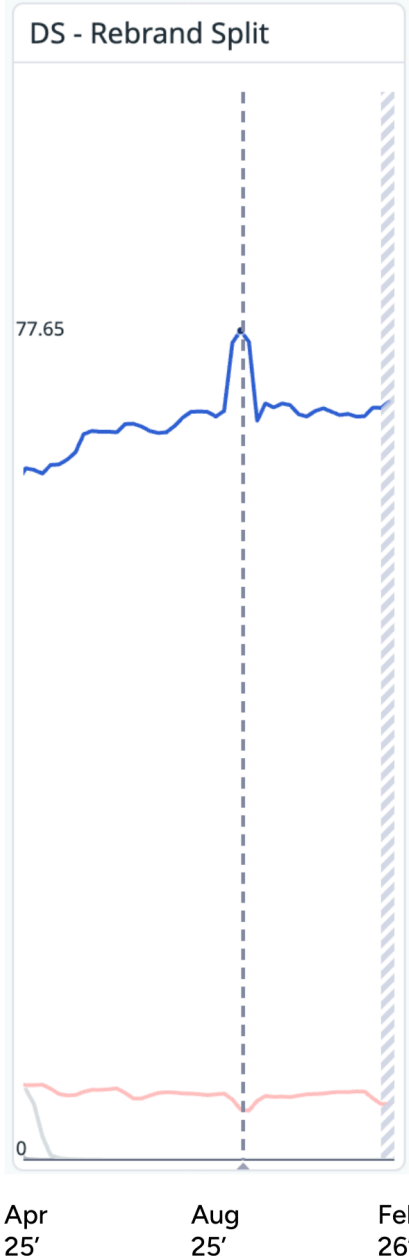


There is no good or bad coverage, every page is unique and Path Design System does not support all the pages the same way.

Coverage containers (pages or components) allow to count the coverage separately in case multiple teams own the same page.

For instance, team owns all the calendars, and Path does not include a calendar, which is why the team's coverage is limited.

See the [How is the DS visual coverage calculated?](#) page for more info.



Apr 25'

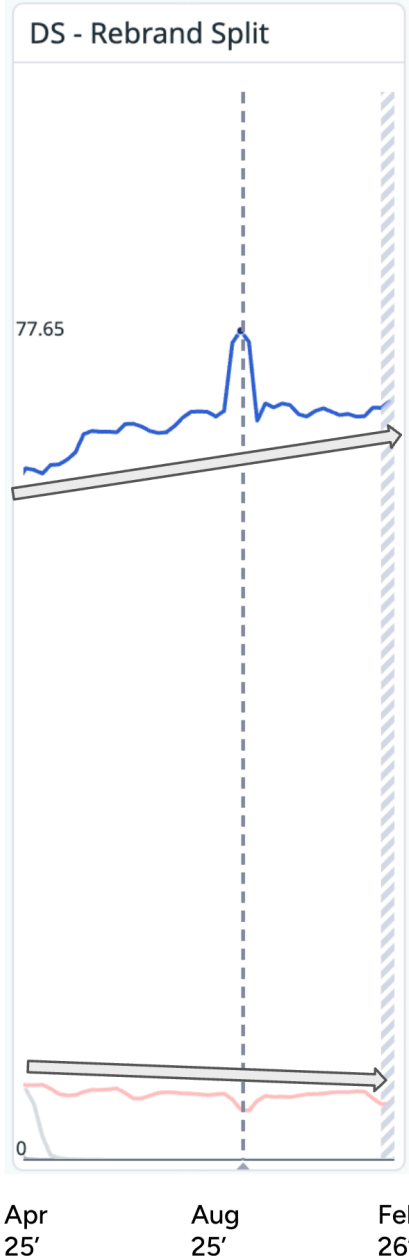
Feb 26'

77.65

Apr 25'

Aug 25'

Feb 26'



Apr 25'

63.5% DS

77.65

Feb 26'

70.4% DS

7.1% Rebrand

5.2% Rebrand

Apr 25'

Aug 25'



Feb 26'

The progress so far




The progress so far

-  Preply battle tested in production (2+ years)





The progress so far

-  Preply battle tested in production (2+ years)
-  Detailed articles
 - Visual coverage part 1
 - Visual coverage part 2
 - For more articles go to medium.com/preply-engineering






The progress so far

-  Preply battle tested in production (2+ years)
-  Detailed articles
 - Visual coverage part 1
 - Visual coverage part 2
 - For more articles go to medium.com/preply-engineering
-  Open Source the project

The progress so far

-  Preply battle tested in production (2+ years)
-  Detailed articles
 - Visual coverage part 1
 - Visual coverage part 2
 - For more articles go to medium.com/preply-engineering
-  Open Source the project
-  Scaling it at WorkWave

The progress so far

-  Preply battle tested in production (2+ years)
-  Detailed articles
 - Visual coverage part 1
 - Visual coverage part 2
 - For more articles go to medium.com/preply-engineering
-  Open Source the project
-  Scaling it at WorkWave
-  Helping docplanner.com and monday.com to adopt it.

What about React Native?

(2024)

What about React Native?

[2024]

- It didn't offer the same convenient worker APIs (now it does)

What about React Native?

[2024]

- It didn't offer the same convenient worker APIs (now it does)
- It didn't offer a way to query the tree yet without blocking the UI

What about React Native?

(2024)

- It didn't offer the same convenient worker APIs (now it does)
- It didn't offer a way to query the tree yet without blocking the UI
- We run the coverage in E2E tests

What about React Native?

(2024)

- It didn't offer the same convenient **worker APIs** (now it does)
- It didn't offer a way to query the tree yet without blocking the UI
- We run the coverage in **E2E tests**
- A small **Swift** script extract the available view coordinates

What about React Native?

[2024]

- It didn't offer the same convenient **worker APIs** (now it does)
- It didn't offer a way to query the tree yet without blocking the UI
- We run the coverage in **E2E tests**
- A small **Swift** script extract the available view coordinates
- Given it's riskier than on Web...
 - Is static analysis a valuable proxy metric for visual coverage?

Slides - noriste.github.io/design-system-coverage-webday-2026

OSS repo - github.com/preply/design-system-visual-coverage

We hire in Barcelona, London, Kyiv

preply.com/en/careers

- [linkedin/noriste](https://www.linkedin.com/company/noriste)
- x.com/noriste
- bsky.app/noriste.bsky.social
- [github/noriste](https://github.com/noriste)



WEB  DAY
2026

Thanks!