

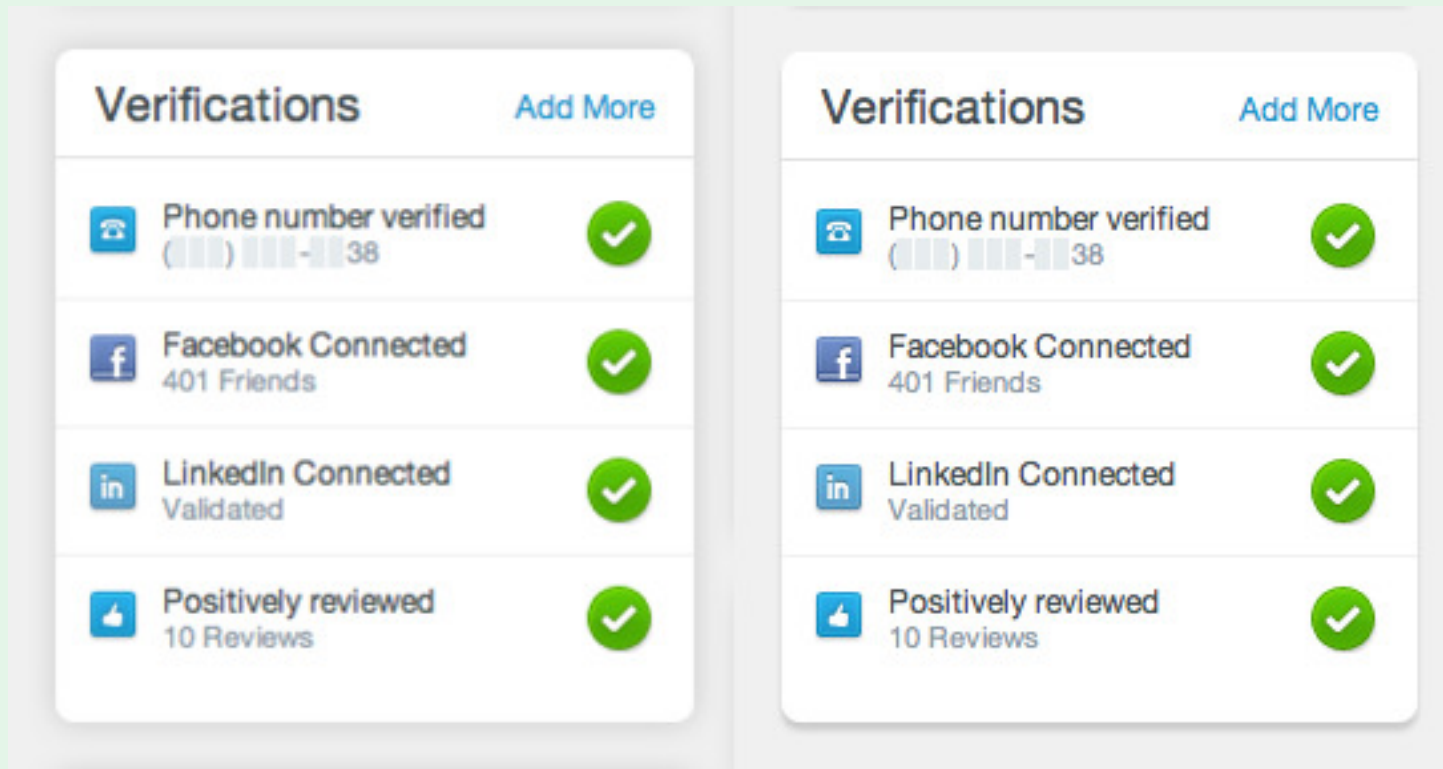
Performances de rendu CSS

D'accord, mais pourquoi ?

Pour avoir une interface fluide :

- Lors du défilement
- Lors des transitions et animations CSS
- Lors de la manipulation de contenu

Le saviez-vous ?



Airbnb a réduit ses ombres de boîte pour améliorer les performances de défilement

Qui suis-je ?

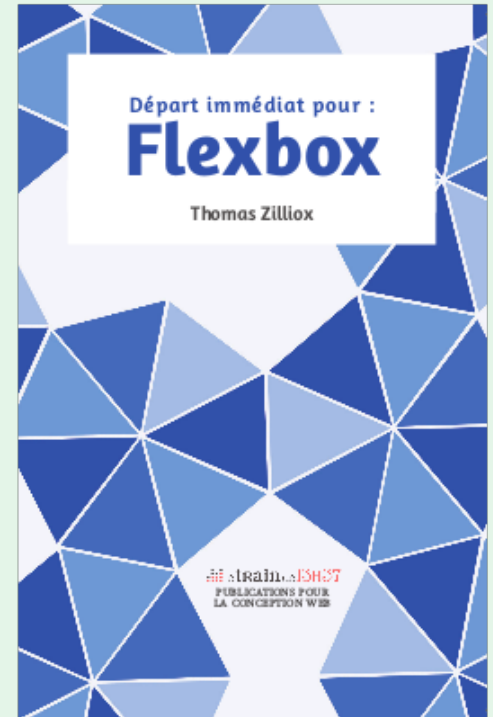


Thomas Zilliox

Expert CSS freelance

Auteur de

Départ immédiat pour Flexbox



Le saviez-vous ?

Jake Archibald wrote...

Don't use flexbox for overall page layout

Posted 05 February

When I was building this blog I tried to use `flexbox` for the overall page layout because I wanted to look cool and modern in front of my peers. However, like all of my other attempts to look cool and modern, it didn't really work.



Hello, I'm Jake and that is my face. I'm a developer advocate for Google Chrome.

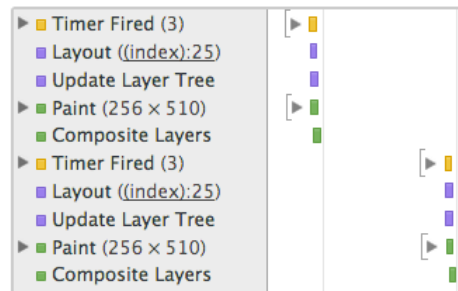
Vous ne devriez certainement pas utiliser Flexbox pour votre gabarit, pour des raisons de performances

Le savez-vous ?

The Problem with SVG

To achieve silky-smooth animation, then, you should obviously stick to these properties. And at Khan Academy, we generally did (with a few exceptions).

However, **SVG doesn't provide these same performance guarantees**. In other words, applying linear transformations to SVG elements *does* trigger re-layout and re-painting. Just take a look at this [Fiddle](#) and the timeline it produces in Chrome—see the purple layout events followed by green paint events?



Il ne faut pas faire de transformations sur des images SVG !

Le saviez-vous ?

Fix scrolling performance with CSS will-change property



Chris Ruppel

Ajouter `will-change: transform` sur vos éléments en `position: fixed` et ce sera plus rapide!

Des astuces fiables

ou des recettes de grand-mères ?

Des astuces fiables ?

- Ombres de boîte
- Flexbox
- SVG
- will-change

Des astuces fiables ?

- Ombres de boîte : 2011
- Flexbox : 2014
- SVG: 2014
- will-change : 2015

Des astuces fiables ?

- Ombres de boîte : Firefox 7, Chrome 15
- Flexbox : Firefox 26, Chrome 32
- SVG : Firefox 33, Chrome 39
- will-change : Firefox 39, Chrome 43

Comment évaluer ?

Apprendre :

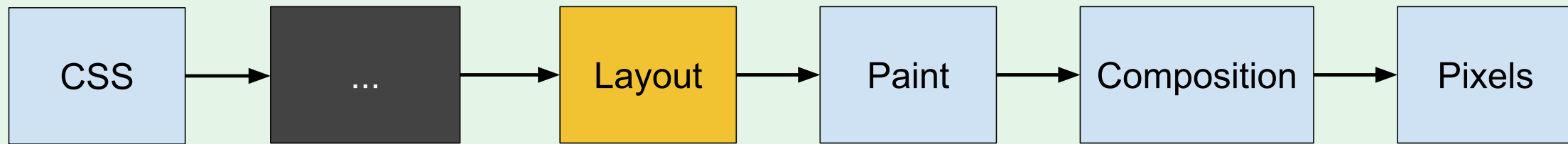
- 1- Comment fonctionne un navigateur
- 2- Comment tester par soi-même

Disséquons un navigateur

Disséquons un navigateur



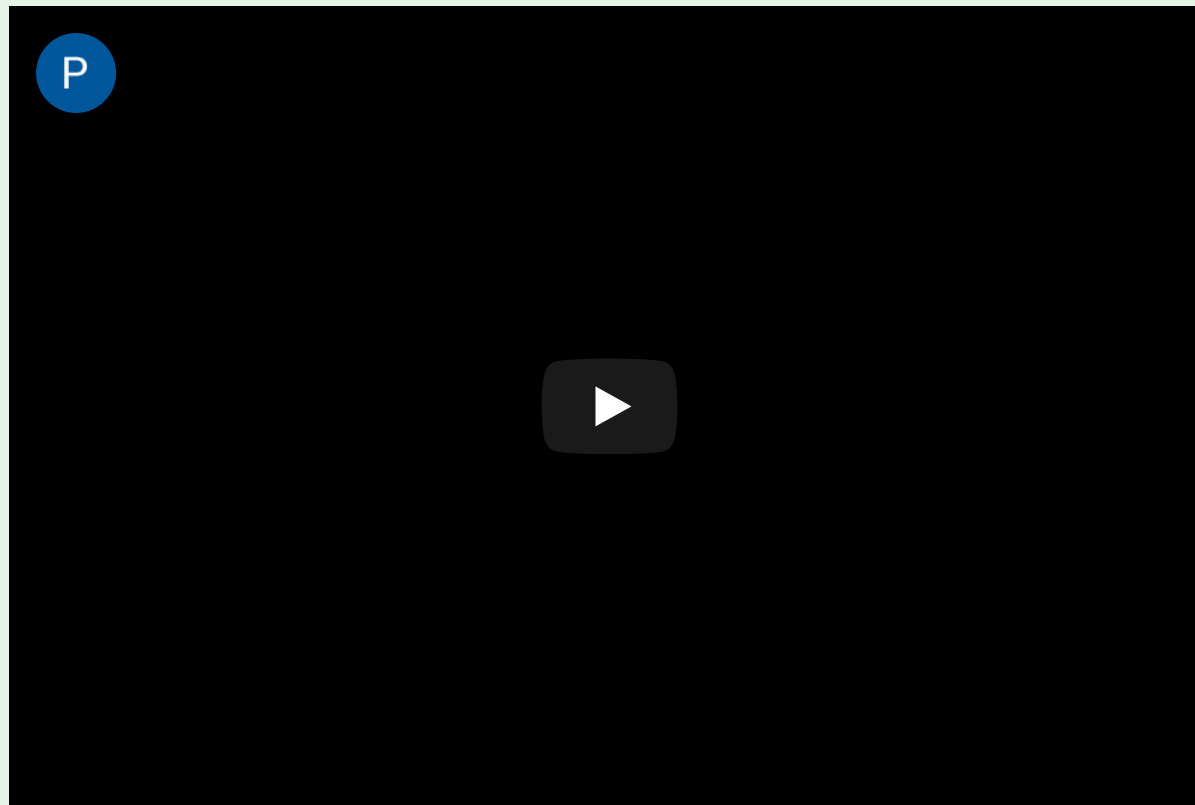
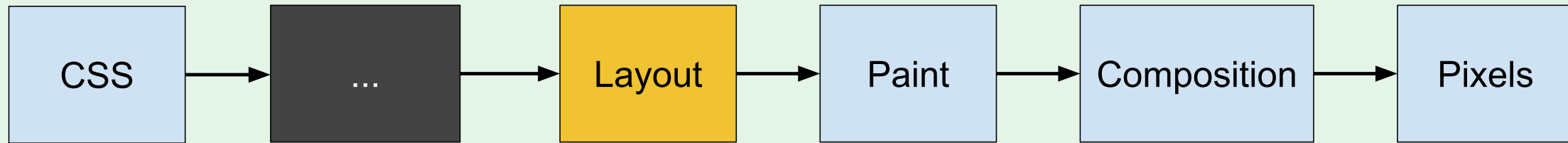
Layout



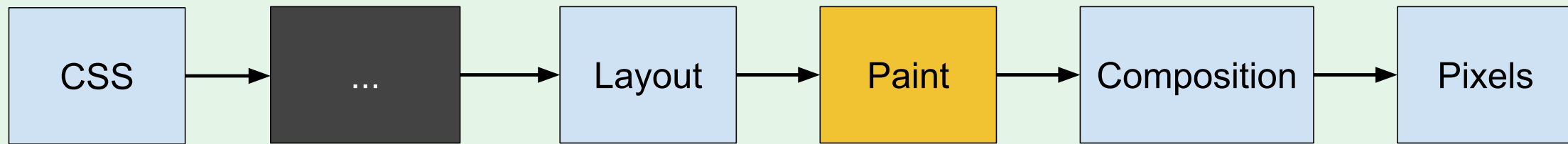
1- **layout** : calcule la taille et la position de chaque noeud

Le *layout* est une étape bloquante.

Layout



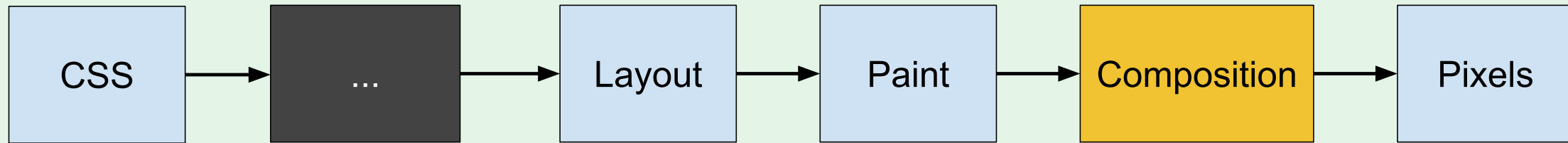
Paint



2- **paint** : calcule chaque pixel pour chaque noeud

Les données sont alors chargées dans la carte graphique.

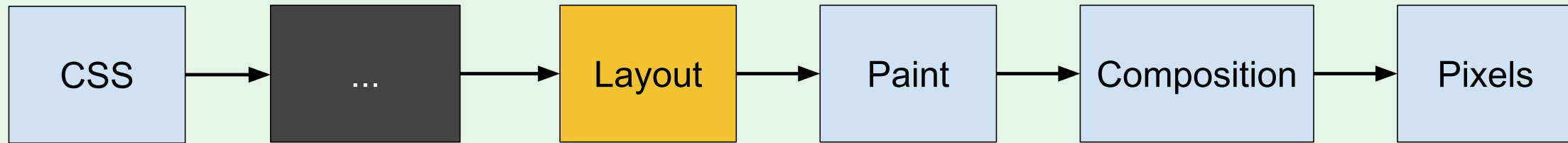
Composition



3- **composition** : affichage des pixels à l'écran

Une étape plutôt rapide !

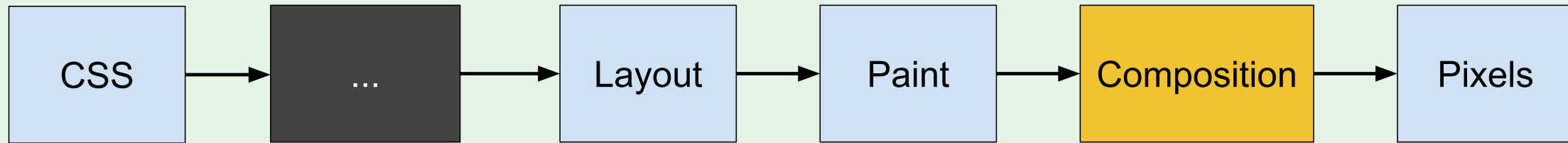
Exemple



Lors d'une transition sur `width`, on déclenche un *layout*.

Ce sera difficile pour le navigateur !

Exemple



Lors d'une transition sur `transform`, c'est une *composition*.

Ce sera facile pour le navigateur !

Transitions

La *composition*, c'est seulement 3 propriétés possibles :

1- opacity

2- transform

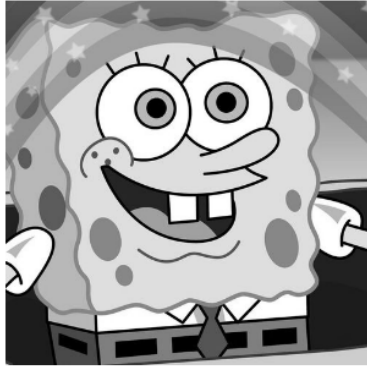
3- filter

CSS filters

No Filter

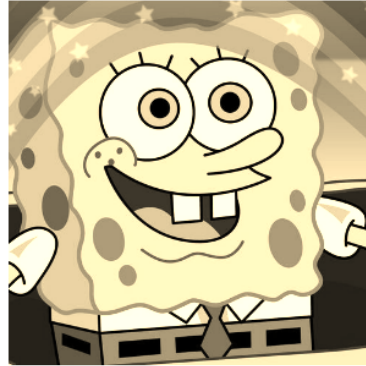


Grayscale



```
.grayscale {  
  -webkit-filter: grayscale(1);  
  filter: grayscale(1);  
}
```

Sepia



```
.sepia {  
  -webkit-filter: sepia(1);  
  filter: sepia(1);  
}
```

Saturation



```
.saturate {  
  -webkit-filter: saturate(4);  
  filter: saturate(4);  
}
```

Hue-rotate



```
.hue-rotate {  
  -webkit-filter: hue-rotate(90deg);  
  filter: hue-rotate(90deg);  
}
```

Invert



Opacity



Brightness



Contrast



Blur

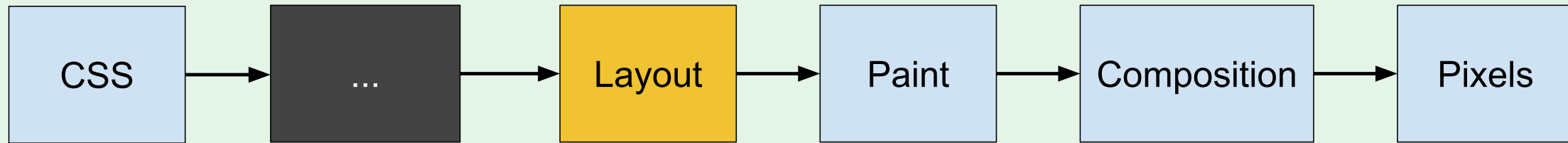


filter est supporté par tous les navigateurs récents !

Attraper les repaints

Attraper les tous...

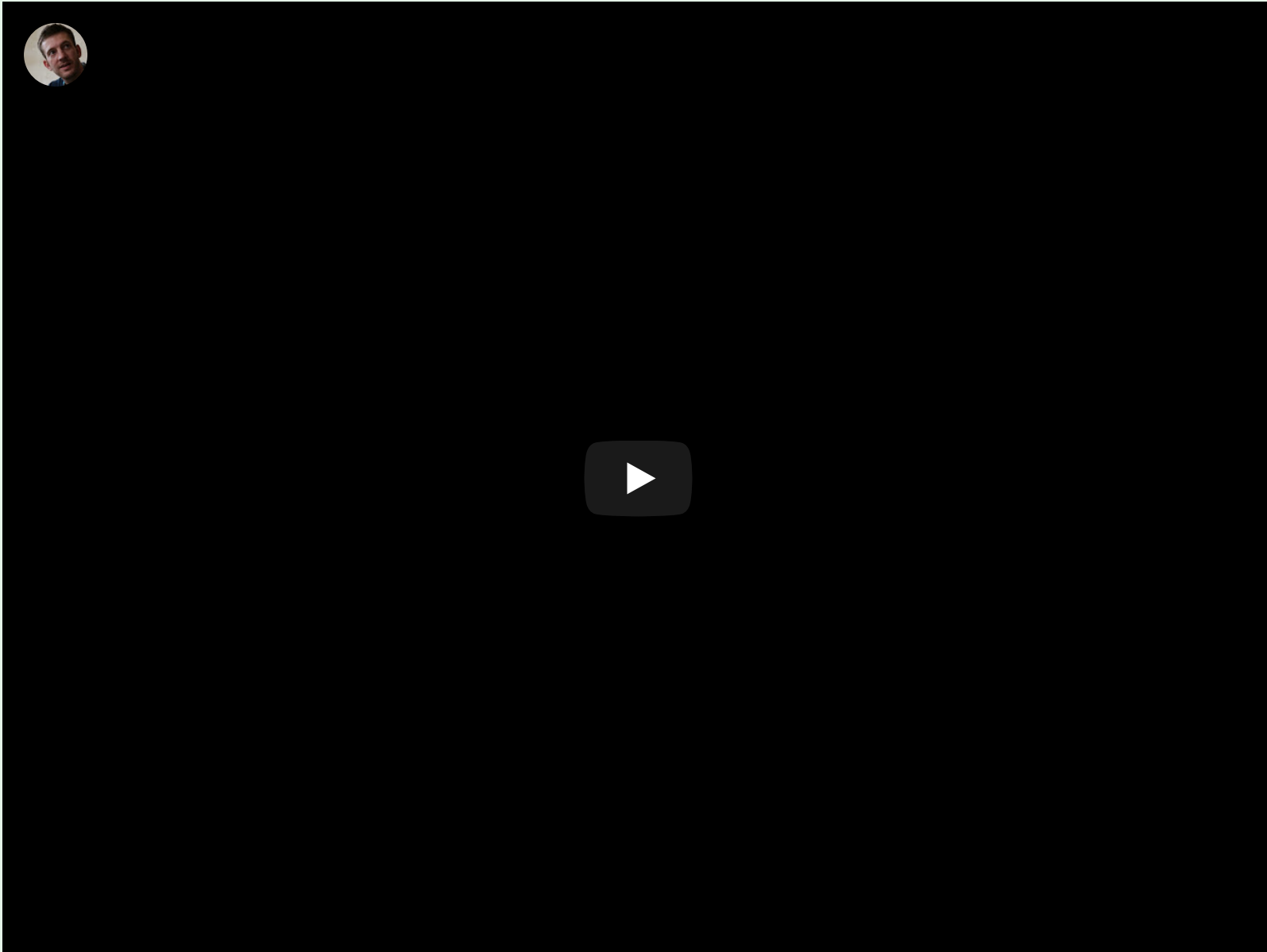
Pourquoi les repaints ?



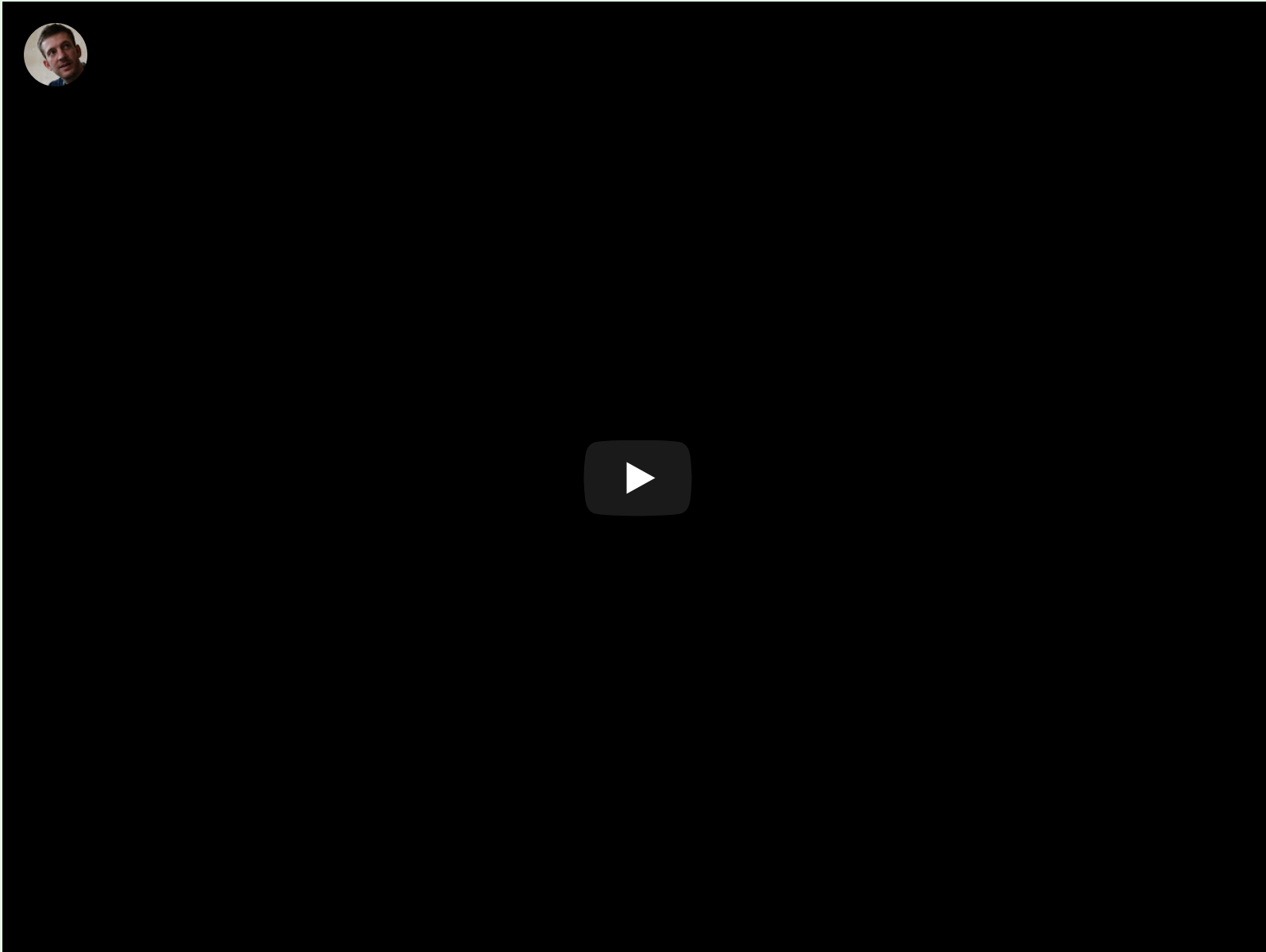
On peut identifier les *layouts* et les *paints* grâce aux Dev Tools.

Les outils pour les *paints* sont plus visuels !

Dev Tools Chrome



Dev Tools Firefox



Recettes de grand-mères

ou non ?

will-change

C'est encore utile pour Chrome,
mais plus pour Firefox.



On peut faire des transformations sur les SVG,
mais pas sur un sous-élément.

Flexbox

Flexbox n'est pas lent,
mais c'était vrai de la première syntaxe.

Ombre de boîte

Les ombres portées ne sont pas redessinées au défilement,
mais cette histoire était à propos des Chromebook.

Conclusion

Il faut tester et ne croire personne !

(même pas moi)

Merci !

Des questions ?

@iamtzi

<https://tzi.fr/slides/speed2018>

<https://tzi.fr/slides/speed2018.pdf>

