



LES MODIFICATIONS CSS : IMPACTS ET BONNES PRATIQUES

Martin Cadoux - 26/02/2015

QUELS SONT LES OBJECTIFS DE CETTE PRÉSENTATION ?

- Évoquer le traitement du CSS effectué par le navigateur
- Comprendre les éventuelles conséquences sur la navigation
- Suggérer quelques bonnes pratiques à adopter

Commençons par le commencement.

LE DOM

Document

html

head

body

title

div

"Eleven L..."

h1

p

"Eleven
L..."

"Sous la
plu..."

LE RENDER TREE

Viewport

Scroll

Block

Block

Block

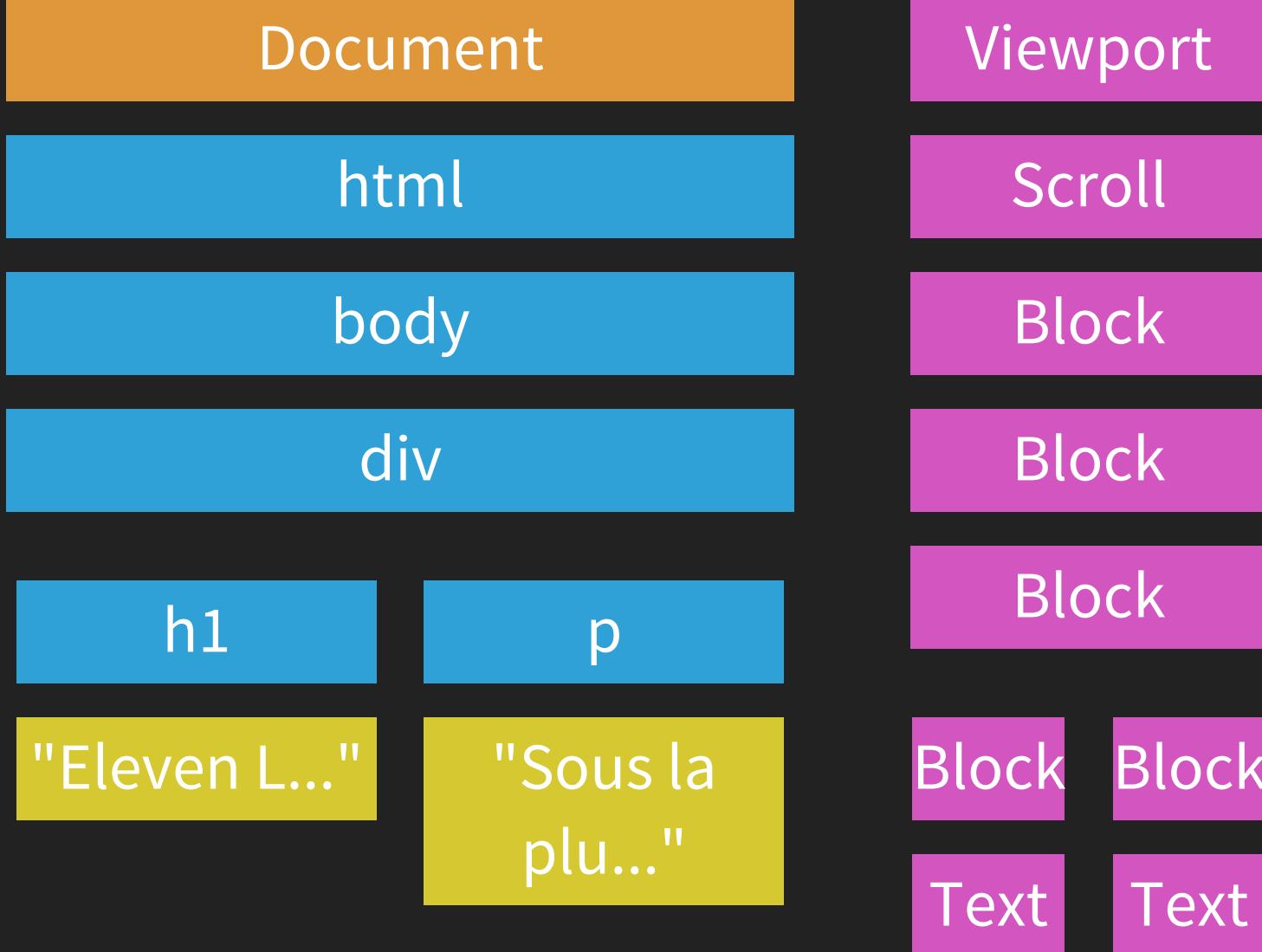
Block

Block

Text

Text

COMPARONS !



Comment le navigateur affiche-t-il tout ça ?

TROIS GRANDES ÉTAPES

1. Layout
2. Paint
3. Composite

LAYOUT

Génère la géométrie et la position de chaque élément

PAINT

Prépare les pixels de chaque élément et les répartit dans des couches

COMPOSITE

Dessine les couches à l'écran

Que se passe-t-il quand je modifie du style à la volée ?

Ça dépend.

Voyons quelques exemples...



"Je sens que ce soir... j'veais conclure."

Padding Color Rotation

Note PDF : Cette diapositive est interactive et a pour but de montrer les différences de comportements entre différents types de propriétés CSS.

LAYOUT

- width / height
- padding / margin
- top / bottom / left / right
- Etc...

PAINT

- color
- background
- box-shadow
- Etc...

COMPOSITE

- transform: translate()
- transform: scale()
- transform: rotate()
- opacity

Quel est le problème ?

Rien de tout ça n'est gratuit.

LAYOUT

- Recalculer le layout d'un élément...
- ...implique de recalculer celui de ses descendants.
- Plus le render tree est long, plus il y a de géométrie à calculer.

PAINT

- Géré par le GPU ou le CPU selon le navigateur et l'élément.
- Très coûteux, particulièrement sur mobile.
- Oblige à repeindre les éléments enfants, voir d'autres éléments.

COMPOSITE

- Peu coûteux.
- N'affecte que l'élément ciblé.

Mais alors qu'est-ce que je peux faire ?

NE PAS COMPROMETTRE LA FLUIDITÉ DE LA PAGE

- Objectif : 60fps
- $1000 / 60 = \sim 16.6$ (ms par frame)
- Utilisez l'outil Timeline de Chrome pour faire vos tests

SE MÉFIER DES ANIMATIONS

- Évitez d'animer ce qui provoque du layout et du paint

```
div {  
    transition-property: width, height, padding, margin;  
    transition-delay: 2s;  
}
```

- Pour les éléments fixed et absolute, ça peut passer.
- Quant aux transformations vues plus haut, faites-vous plaisir !

UTILISER DES SÉLECTEURS COURTS, ABUSER DES CLASSES

- N'oubliez pas : le CSS est lu de droite à gauche.

```
//Pas terrible...
.content > div p a {...}
```

```
//Parfait.
.paragraph-link {...}
```

QUELQUES RESSOURCES

- [CSS Triggers](#) de Paul Lewis
- [How Browsers Work](#) de Tali Garsiel
- [High Performance Animations](#) de Paul Irish et Paul Lewis

Thank you for your attention!