

# Intégration Web

MMI 1 – TP#2 S2

Danielo **JEAN-LOUIS**  
Michele **LINARDI**

# Pratiquons ! - Transitions (Partie 1)

Pré-requis :

- Avoir la ressource ressources/transitions

A télécharger ici :

[https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web\\_travaux-pratiques\\_numero-2.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web_travaux-pratiques_numero-2.ressources.zip)

**Ça fonctionne... mais ça manque de dynamisme**

**Améliorons ceci, rendons ceci plus dynamique**

# Transitions CSS

- Permettent de faire une transition entre deux valeurs d'une propriétés CSS
  - Fonctionne avec toutes les propriétés CSS
  - Le navigateur calcule les interpolations de valeurs
- Représentées par la propriété CSS "transition"

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_animated\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_animated_properties) – Liste de propriétés CSS qui peuvent faire une transition

# Transitions CSS

- Fonctionnent mieux si la propriété "transition" est sur le sélecteur CSS (départ)
  - Et non la pseudo-classe (fin)
- Permettent des transitions simples
- A ne pas confondre avec CSS Animation
  - Non abordé en cours

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

# Transitions CSS

- Calculées grâce à la carte graphique (GPU)
  - Permet de meilleures performances
- Rajoutent du dynamisme à vos sites
- Ne permettent pas une répétition des effets
  - Il faudra utiliser la propriété CSS “animation”

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>



# Point accessibilité – Transitions CSS

- Évitez d'afficher les informations critiques en stade final d'une transition
  - Tous les utilisateurs ne verront pas l'information
- Évitez les durées de transitions trop longues

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/@media/prefers-reduced-motion>
- <https://www.w3.org/WAI/WCAG21/Understanding/animation-from-interactions> - anglais

# Point accessibilité – Transitions CSS

- Proposez une version sans animation (voir sources)
- Essayez de faire une direction artistique de vos transitions
  - N'allez pas changer l'apparence des transitions d'un bouton à l'autre, par exemple

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/@media/prefers-reduced-motion>
- <https://www.w3.org/WAI/WCAG21/Understanding/animation-from-interactions> - anglais

# Transitions CSS

- Définies par un des états grâce aux pseudo-classes comme :hover ou :focus(-within)
- Possibilité d'utiliser d'autres évènements (nécessite javascript)

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

# Pseudo-classe - :hover

- Permet d'appliquer des déclarations CSS sur un élément au survol
- **Applicable sur n'importe quelle balise**

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/:hover>

# Pseudo-classe - :hover

- Peut de cibler plusieurs balises en même temps grâce au système d'imbrications de sélecteurs
  - Séparés par un espace
- **Ne fonctionne pas sur mobile**

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/:hover>

# Pseudo-classe - :hover



```
mon-sélecteur {  
    /* [...] */  
}
```

Ici notre sélecteur CSS aura des propriétés CSS au survol (:hover)

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/:hover>

# Pseudo-classe - :hover - Imbrication



```
mon-sélecteur mon-enfant {  
    /* [...] */  
}
```

Ici notre sélecteur CSS (mon-enfant) aura des propriétés CSS au survol (:hover) de (mon-sélecteur)

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/:hover>

# Pseudo-classe - :hover - Imbrication

- Imbrication utile pour :
  - Modifier plusieurs éléments en même temps
  - Modifier un élément (ou plusieurs) trop difficiles à atteindre à la souris

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/:hover>
- Voir ressources/exemples-transitions pour exemples interactifs



# Transitions CSS – Exemple de code

**transition:** **margin-right** **2s** **linear** **.5s**;

durant 2 secondes

avec un délai de 0.5 secondes

On change la propriété "margin-right"

de façon linéaire

Avec le code ci-dessus, l'élément ciblé verra son margin-right changer

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

# Point performances – transitions

- N'écrivez **jamais** la chose suivante :

**transition:** **all** 200ms linear;

Ne dites **jamais** au navigateur que vous allez faire une transition de toutes les propriétés CSS du sélecteur CSS

Ceci fonctionne mais va provoquer des problèmes de performances

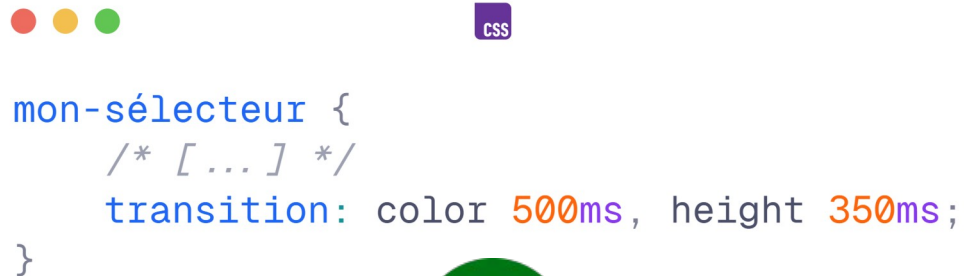
# Point performances – transitions

- Ne mettez **que** les propriétés CSS qui doivent changer dans la valeur de la propriété transition



```
mon-sélecteur {  
  /* [...] */  
  transition: all 500ms;  
}
```

A CSS code editor window with a red prohibition sign (a red circle with a diagonal line) overlaid on the bottom right corner, indicating that this code is incorrect.



```
mon-sélecteur {  
  /* [...] */  
  transition: color 500ms, height 350ms;  
}
```

A CSS code editor window with a green checkmark (a green circle with a white checkmark) overlaid on the bottom right corner, indicating that this code is correct.

# Transitions CSS

- Propriété CSS raccourcie (transition) :
  - transition-property : propriétés CSS qui vont changer
  - transition-duration : durées des transitions
  - transition-timing-function : description de la courbe d'évolution des valeurs intermédiaires
    - Valeur déterminée ou fonction personnalisée
  - transition-delay : délai avant le lancement de la transition

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Transitions CSS – Exemple de code

The diagram illustrates the components of a CSS transition declaration: `transition: margin-right 2s linear .5s;`. The parts are color-coded and labeled as follows:

- transition:** Labeled as the keyword.
- margin-right**: Labeled as the `-property`.
- 2s**: Labeled as `-duration`.
- linear**: Labeled as the `-timing-function`.
- .5s**: Labeled as `-delay`.

Grâce à la propriété "transition", nous gérons quatre propriétés en une

**Source(s) :**

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Transitions CSS

- L'ordre des valeurs ne compte pas toutefois :
  - Le premier nombre rencontré sera la valeur de la propriété "transition-duration"
  - Le second nombre rencontré sera la valeur de la propriété "transition-delay"

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Transitions CSS

- Possibilité de changer plusieurs propriétés CSS en même temps
  - Séparer chaque transition par une virgule (,)

transition: **color 2s linear .5s**, **font-size 300ms**;

transition 1

transition 2

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Transitions CSS

- Durée (temps de transition et délai) exprimée en secondes (s) ou millisecondes (ms)
  - $1s = 1000ms$
  - **La durée nécessite une unité**
- On n'oublie pas la valeur du stade final

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>



# Transitions CSS

- Seule la propriété **"transition-duration"** est obligatoire
  - Le reste le navigateur utilisera les valeurs par défaut
  - Idem pour la propriété "transition" dont seul la durée est obligatoire

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

**Comme pour transition: all, on préférera  
l'indication des propriétés à faire transitionner**

# Transitions CSS



```
mon-sélecteur {  
    /* [...] */  
    font-size: 2rem;  
    transition: font-size 500ms ease-in 0s;  
}  
  
mon-sélecteur:hover {  
    /* [...] */  
    font-size: 1.2rem;  
}
```

La propriété "transition" se trouve dans le sélecteur (et non la pseudo-classe) pour éviter des comportements étranges

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Pratiquons ! - Transitions (Partie 2/3)

Pré-requis :

- Avoir la ressource ressources/transitions

A télécharger ici :

[https://github.com/DanYellow/cours/raw/refs/heads/main/integration-web-s2/travaux-pratiques/numero-2/integration-web-s2\\_travaux-pratiques\\_numero-2.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/integration-web-s2/travaux-pratiques/numero-2/integration-web-s2_travaux-pratiques_numero-2.ressources.zip)

# Transitions CSS

- N'oubliez pas que les sélecteurs CSS s'imbriquent
  - Possibilité d'appliquer une transition sur plusieurs éléments en même temps

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

# Transitions CSS



```
/*  
    Au survol de l'élément qui contient la classe "mon-conteneur",  
    on changera des propriétés CSS de "mon-selecteur-1" et "mon-selecteur-2"  
*/  
  
mon-conteneur:hover mon-selecteur-1 {  
    /* Propriété CSS */  
}  
  
mon-conteneur:hover mon-selecteur-2 {  
    /* Propriété CSS */  
}
```

# Pratiquons ! - Transitions (Partie 4)

Pré-requis :

- Avoir la ressource ressources/transitions

A télécharger ici :

[https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web\\_travaux-pratiques\\_numero-2.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web_travaux-pratiques_numero-2.ressources.zip)

# Transitions CSS - Debug

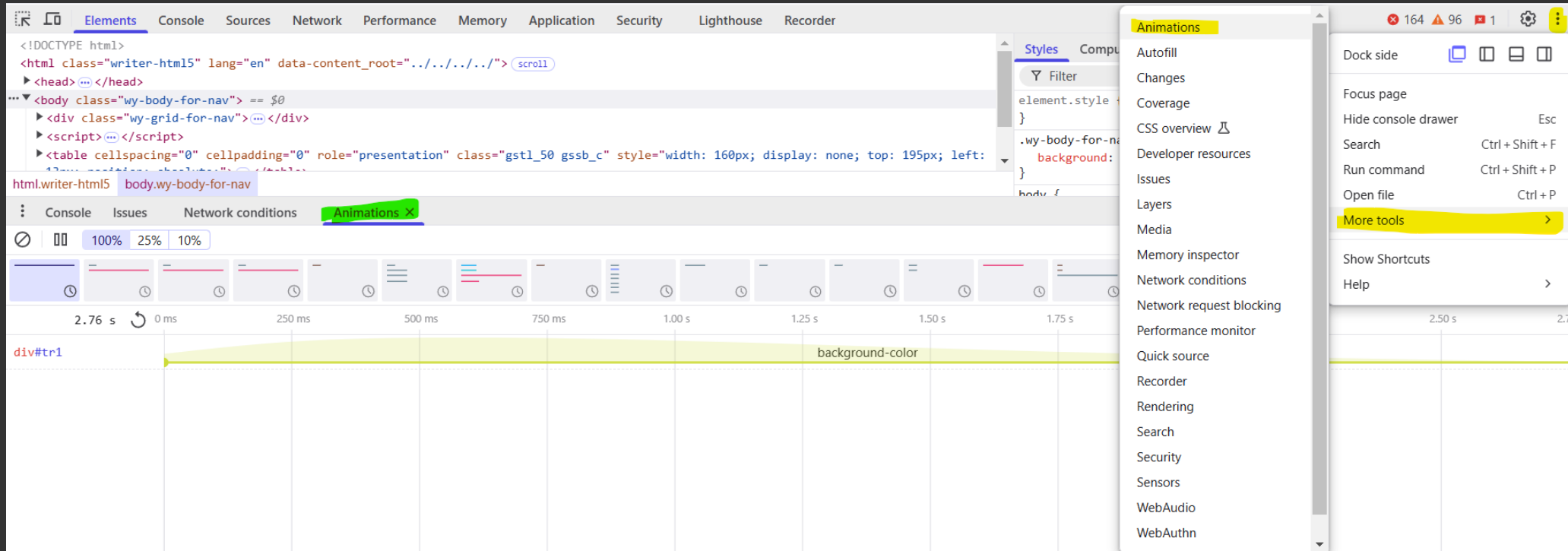
- Utilisation de la console du navigateur
  - Permet de rejouer à l'envie et pas à pas la transition d'un élément sélectionné
  - Fonctionne avec Chrome et Firefox

## Source(s) :

- <https://developer.chrome.com/docs/devtools/css/animations?hl=fr>
- [https://firefox-source-docs.mozilla.org/devtools-user/page\\_inspector/how\\_to/work\\_with\\_animations/index.html](https://firefox-source-docs.mozilla.org/devtools-user/page_inspector/how_to/work_with_animations/index.html)



# Transitions CSS - Debug

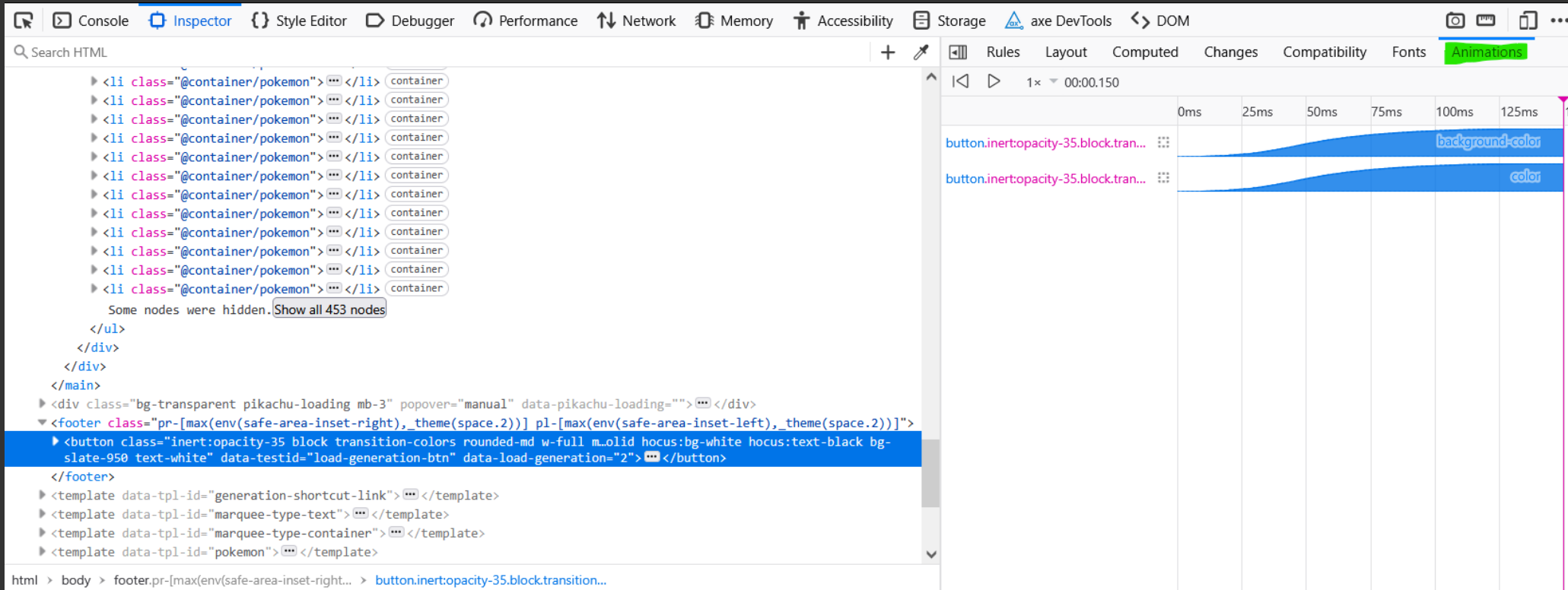


Chrome

Source(s) :

- <https://developer.chrome.com/docs/devtools/css/animations?hl=fr>
- [https://firefox-source-docs.mozilla.org/devtools-user/page\\_inspector/how\\_to/work\\_with\\_animations/index.html](https://firefox-source-docs.mozilla.org/devtools-user/page_inspector/how_to/work_with_animations/index.html)

# Transitions CSS - Debug



The screenshot displays the Firefox DevTools interface. The left pane shows the HTML DOM with a list of `<li class="@container/pokemon">` elements. The right pane shows the Animations panel, which is currently empty. The top toolbar includes various DevTools panels like Console, Inspector, Style Editor, Debugger, Performance, Network, Memory, Accessibility, Storage, axe DevTools, and DOM. The bottom status bar shows the current path: `html > body > footer.pr-[max(env(safe-area-inset-right... > button.inert.opacity-35.block.transition...`.

Firefox

Source(s) :

- <https://developer.chrome.com/docs/devtools/css/animations?hl=fr>
- [https://firefox-source-docs.mozilla.org/devtools-user/page\\_inspector/how\\_to/work\\_with\\_animations/index.html](https://firefox-source-docs.mozilla.org/devtools-user/page_inspector/how_to/work_with_animations/index.html)

# @starting-style

- Règle-at définissant les propriétés CSS de départ
- Permet d'effectuer une transition CSS lors du chargement de la page ou apparition de l'élément

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/@starting-style>
- <https://www.alsacreations.com/tuto/lire/1940-art-des-animations-entree-avec-css-starting-style.html>

# @starting-style



```
.element {  
    /* [...] */  
    transform: scaleY(1);  
    transition: transform 500ms;  
  
    @starting-style {  
        transform: scaleY(0);  
    }  
}
```



```
.element {  
    /* [...] */  
    transform: scaleY(1);  
    transition: transform 500ms;  
}  
  
@starting-style {  
    .element {  
        transform: scaleY(0);  
    }  
}
```

Les éléments avec la classe “element” auront “scaleY(0)” au chargement de la page puis “scaleY(1)” après 350ms  
(Les deux syntaxes sont valides et font la même chose)

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/@starting-style>
- <https://www.alsacreations.com/tuto/lire/1940-art-des-animations-entree-avec-css-starting-style.html>


# @starting-style - Notes

- La règle-at @starting-style a la même spécificité que les déclarations CSS hors de la règle
  - @starting-style doit être mis en dernier

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/@starting-style>
- <https://www.alsacreations.com/tuto/lire/1940-art-des-animations-entree-avec-css-starting-style.html>

# @starting-style - Notes




```
.element {  
  /* [...] */  
  @starting-style {  
    opacity: 0;  
  }  
  
  transition: opacity 0.5s;  
  opacity: 1;  
}
```



## Incorrect

La règle @starting-style sera écrasée par les déclarations CSS qui suivent



```
.element {  
  /* [...] */  
  
  transition: opacity 0.5s;  
  opacity: 1;  
  
  @starting-style {  
    opacity: 0;  
  }  
}
```



## Correct

La règle @starting-style est placée après les déclarations CSS “normales”

### Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/@starting-style>
- <https://www.alsacreations.com/tuto/lire/1940-art-des-animations-entree-avec-css-starting-style.html>

# Pratiquons ! - Transitions (Partie 5)

Pré-requis :

- Avoir la ressource ressources/transitions

A télécharger ici :

[https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web\\_travaux-pratiques\\_numero-2.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s2-integration-web/travaux-pratiques/numero-2/s2-integration-web_travaux-pratiques_numero-2.ressources.zip)

# transition-behavior

- Nouveauté de la propriété transition (2024)
- Permet de faire une transition avec des valeurs discrètes
  - Ex : display: block → display: none
- Deux valeurs possibles :
  - normal : par défaut
  - allow-discrete

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition-behavior>



# transition-behavior



```
element {  
  transition-property: opacity, display;  
  transition-duration: 0.25s;  
  transition-behavior: allow-discrete;  
}  
  
element:hover {  
  opacity: 0;  
  display: none;  
}
```

On peut faire la transition entre “display: block” et “display: none”  
Impossible sans “transition-behavior: allow-discrete”

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition-behavior>

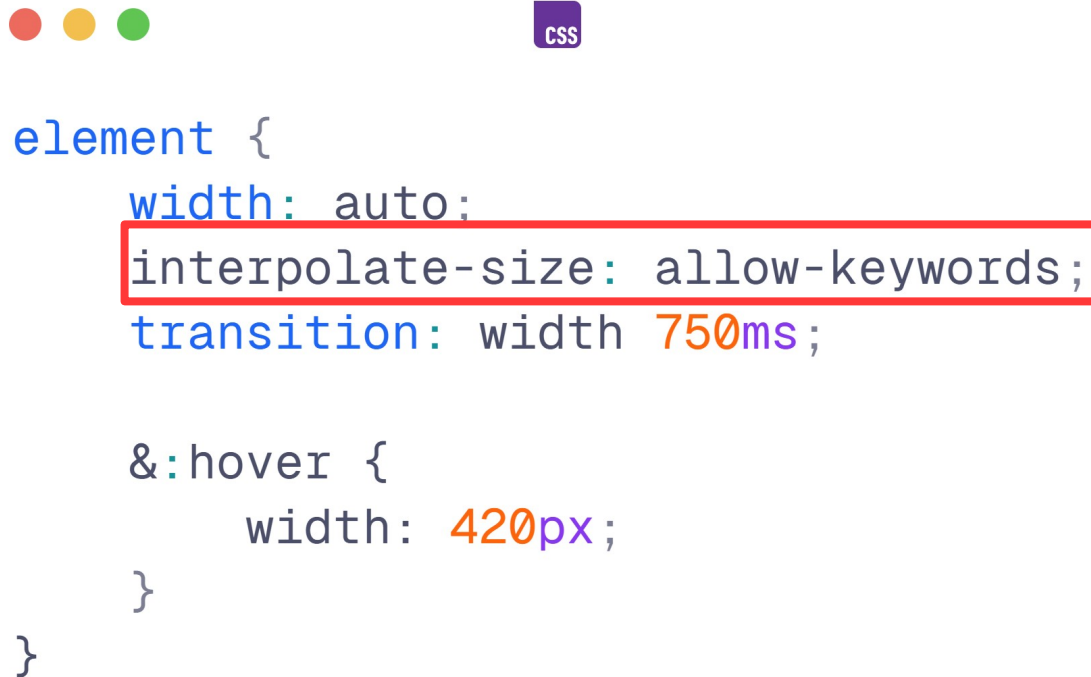
# interpolate-size

- Nouvelle propriété CSS
  - Propriété encore expérimentale (gérée par Chrome)
- Permet de faire une transition de tailles entre une valeur discrète et une valeur finie (et vice-versa)
  - Ex : width: auto → width: 420px
- Deux valeurs possibles :
  - numeric-only (par défaut)
  - allow-keywords

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/interpolate-size>

# interpolate-size



```
element {  
  width: auto;  
  interpolate-size: allow-keywords;  
  transition: width 750ms;  
  
  &:hover {  
    width: 420px;  
  }  
}
```

On peut faire la transition entre “width: auto” et “width: 420px”  
Impossible sans “interpolate-size: allow-keywords”

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/interpolate-size>

# interpolate-size

- Propriété héritée, il est possible de la mettre sur le sélecteur “html” pour l’appliquer sur tous les éléments de la page

A code editor window with a white background and a purple 'CSS' icon in the top right corner. The code is written in a monospaced font with syntax highlighting: 'html' is blue, '{' and '}' are black, and 'interpolate-size: allow-keywords;' is dark blue. The code is as follows:

```
html {  
    interpolate-size: allow-keywords;  
}
```

Dorénavant, tous les éléments de la page ont la déclaration CSS  
"interpolate-size: allow-keywords;"

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/CSS/interpolate-size>

# Performances

- La transition de certaines propriétés CSS font du "repaint"
  - Autrement dit, le navigateur va "redessiner" la page pour appliquer les transitions → Gros problème de performances dépendamment des propriétés CSS
    - Heureusement il y a une solution : **CSS transform**
      - Sujet du prochain TP

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transform>

**Questions ?**

