

17/11/2022

Version 4



ESS3

JÉRÉMY PERROUAULT

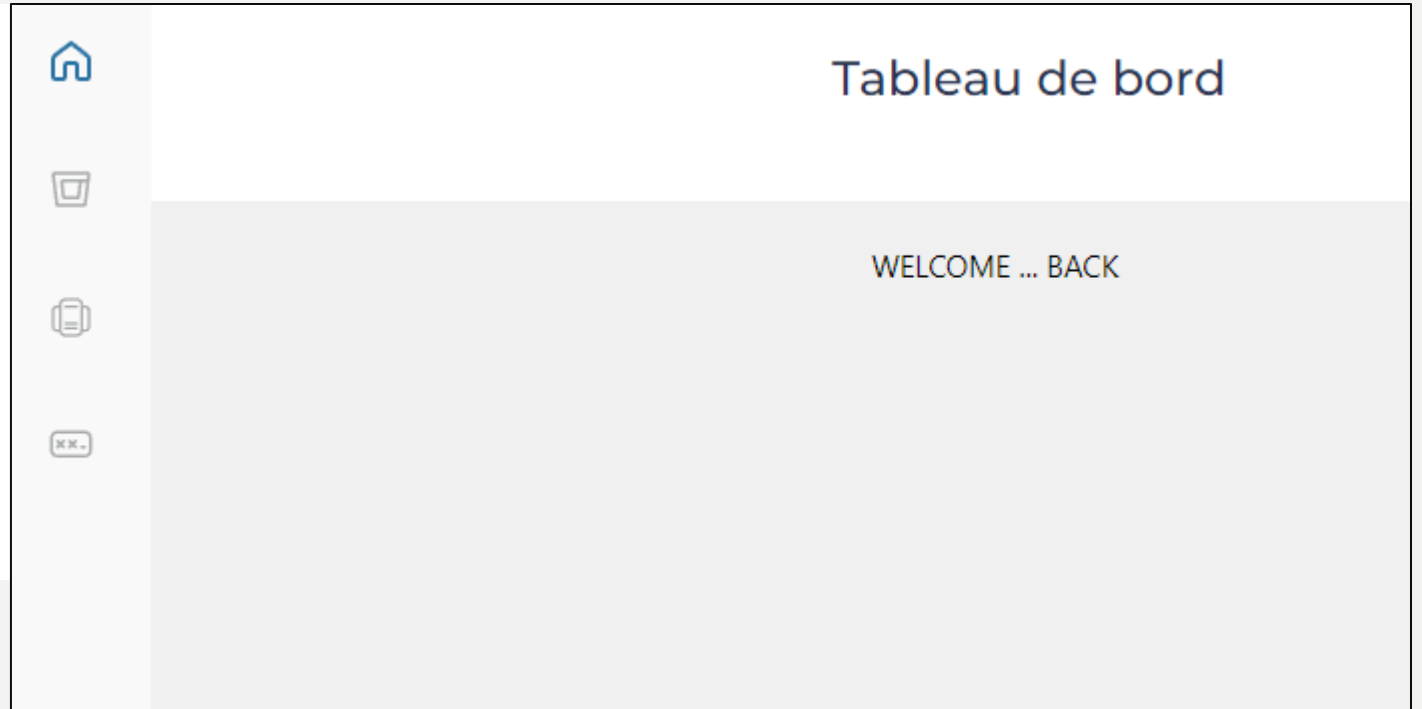


PRÉSENTATION

PRÉSENTATION DE CSS3

PRÉSENTATION

- Cascading Style Sheets
- Met en forme les éléments HTML



PRÉSENTATION

- CSS s'accompagne de sa problématique : la compatibilité entre navigateurs
- Pour s'assurer de la compatibilité : <http://caniuse.com/>

PRÉSENTATION

- Le style CSS peut s'écrire
 - Sur un élément en utilisant l'attribut « style »
 - Dans la page HTML, dans les balises <style>
 - Dans un fichier CSS bien séparé qui sera appelé par la page HTML
 - Un élément pourra faire référence à une ou plusieurs « classe(s) CSS », via l'attribut « class »

```
<link rel="stylesheet" href="assets/css/le-fichier-css.css" />
```

- Le CSS se compose de l'élément suivi de description de propriétés CSS

```
selecteur {  
  propriete-css: valeur;  
  autre-propriete-css: valeur;  
}
```

```
<div class="nom-classe-css"></div>
```

```
<div class="classe-a classe-b"></div>
```

PRÉSENTATION

```
.titanic {  
  float: none;  
}
```

```
#triangle-des-bermudes {  
  display: none;  
}
```

```
#usa + #mexique {  
  border: dashed 1px grey;  
}
```

```
#usa[data-president="Trump"] + #mexique {  
  border: solid 30px white;  
}
```

```
.bruce-banner {  
  height: 200px;  
  color: pink;  
  transition: all 5s ease-in;  
}
```

```
.bruce-banner.hulk {  
  height: 400px;  
  color: green;  
}
```

PROPRIÉTÉS

- Sur ce site se trouve l'ensemble des propriétés CSS disponibles
- <http://www.css-faciles.com/proprietes-css-liste-alphabetique.php>

A decorative wavy line in light blue and white, running vertically along the left side of the slide.

SÉLECTEURS

SÉLECTIONNER LES ÉLÉMENTS

LES SÉLECTEURS

CSS	HTML
:root	Element racine
#mon_identifiant	<balise id="mon_identifiant"></balise>
.ma-classe	<balise class="ma-classe"></balise>
balise	<balise>
*	Tous les éléments
div, p	<div> ET <p>
div p	<p> DANS <div>
div > p	<p> directement dans <div>
div + p	Le <p> directement après les <div>
div ~ p	Tous les <p> placés juste après un <div>
[attribut]	Tous les attributs attribut
[attribut="valeur"]	Tous les attributs attribut="valeur"

LES SÉLECTEURS

CSS	HTML
:hover	Élément cible en état survol de la souris
:active	Élément cible en état clique de la souris
:focus	Élément cible en état actifs
:checked	Élément cible en état sélectionnés (cases à cocher)
:disabled	Élément cible en état désactivés
:empty	Élément cible vide, sans enfant
::after	Définition d'un pseudo-élément, après le contenu de l'élément cible
::before	Définition d'un pseudo-élément, avant le contenu de l'élément cible
:first-child	Premier élément enfant de l'élément cible
:only-child	Element enfant unique dans l'élément cible
:last-child	Dernier élément enfant de l'élément cible

LES SÉLECTEURS

CSS	HTML
:has(X)	L'élément cible possède un sélecteur
:not(X)	L'élément cible n'est pas un sélecteur
:is(X)	L'élément cible est un sélecteur

http://www.w3schools.com/cssref/css_selectors.asp pour une liste complète

EXERCICE

- Trouver les sélecteurs
 - Les classes « produits » et « clients »
 - Les tableaux dont la classe est « produits »
 - Les paragraphes dans la div (tous niveaux) qui a l'identifiant « accueil »
 - Les paragraphes dans la div (premier niveau) qui a l'identifiant « contact »
 - Les inputs de type « text » dans tous les formulaires
 - Le label qui est juste après un input de type « checkbox », dans tous les formulaires
 - Le label qui est juste après un input de type « checkbox » qui a la classe « special »
 - Les liens dans l'état « survole »
 - Les inputs de type « text » lorsque la zone de saisie est activée (le curseur est dans l'input)
 - Les paragraphes qui ont les classes « rouge » et « arrondi » (combinées)
 - Les paragraphes qui ont la classe « rouge » sans la classe « arrondi » et qui n'ont pas d'enfant

A decorative wavy line in light blue and white, flowing vertically along the left side of the slide.

COULEUR

GESTION DES COULEURS EN CSS3

COULEURS

- Les couleurs (de texte, de bordure, d'arrière-plan) peuvent-être au format
 - Texte anglais
 - red, crimson, black, green, yellow, orange, ...
 - Hexadécimal (Rouge Vert Bleu = Red Green Blue = RGB)
 - RAPPEL : 00 = 0 & FF = 255
 - #FF7514, #004795
 - #FFF → #FFFFFF
 - #751 → #775511

COULEURS

- Les couleurs (de texte, de bordure, d'arrière-plan) peuvent-être au format
 - RGB (Red, Green, Blue)
 - `rgb(255, 117, 20)`
 - `rgb(255 117 20)`
 - RGBA (Red, Green, Blue, alpha pour la transparence)
 - `rgba(255, 117, 20, 0.5)`
 - Comme pour l'hexadécimal, on va de 0 à 255

COULEURS

- Les couleurs (de texte, de bordure, d'arrière-plan) peuvent-être au format
 - HSL (Hue Saturation Lightness), plus récent
 - `hsl(25 100% 54%)`
 - `hsl(25, 100%, 54%)`
 - HSLa (Hue, Saturation, Lightness, alpha pour la transparence)
 - `hsla(25 100% 54% / 50%)`
 - `hsla(25, 100%, 54%, 0.5)`
 - La première info est la teinte, en degré (de 0 à 360)
 - En commençant par le rouge
 - On a ensuite la saturation, 100% est la couleur réelle
 - Plus on descend et plus on arrivera sur une teinte de gris
 - Puis la luminosité, 50% est la couleur réelle, 0% et le noir et 100% est le blanc



COULEURS

- Intérêt du HSL ?
 - En CSS, une fonction « calc » existe et permet de faire des calculs simples
 - Et il est possible d'utiliser cette fonction dans les fonctions RGB et HSL

```
hsl(50, 50%, calc(50% + 20%));
```

- Etant donné qu'en HSL, on spécifie la « couleur », la saturation et la luminosité
 - Il devient plus facile de jouer sur les nuances d'une couleur
 - <https://hslpicker.com/>

COULEURS

- Il est également possible de créer des dégradés
 - Avec les fonctions linear-gradient ou radial-gradient
 - linear
 - direction, liste des couleurs & positions
 - radial
 - forme ou taille, liste des couleurs & positions

```
linear-gradient(90deg, hsl(50, 10%, 50%) 0%, rgb(250, 100, 150) 60%, red 100%);
```

- Générateur & explications : <https://cssgradient.io/>



UNITÉS

LES UNITÉS EN CSS3

UNITÉS

- pixel
 - Unité absolue, en pixel
- pourcentage
 - Unité en pourcentage d'un élément parent
 - Possible que lorsque la taille parent est définie
- em
 - C'est un pourcentage de la taille en pixel appliquée qui, par défaut, est généralement de 16px
- rem
 - Même chose que em, mais relative à la taille appliquée au document (root), pas à l'élément parent
- viewport
 - Unité en pourcentage de la « zone de vue », souvent la fenêtre navigateur
 - vw (view-width)
 - vh (view-height)
 - vmin / vmax (width OU height le plus petit / width OU height le plus grand)

UNITÉS

```
:root {  
  font-size: 40px;  
}  
  
body {  
  font-size: 20px;  
}  
  
.parent {  
  width: 200px;  
  height: 80vh;  
}
```

```
.rem, .em {  
  display: inline-block;  
  width: 50%;  
}  
  
.rem {  
  font-size: 0.5rem;  
}  
  
.em {  
  font-size: 0.5em;  
}
```

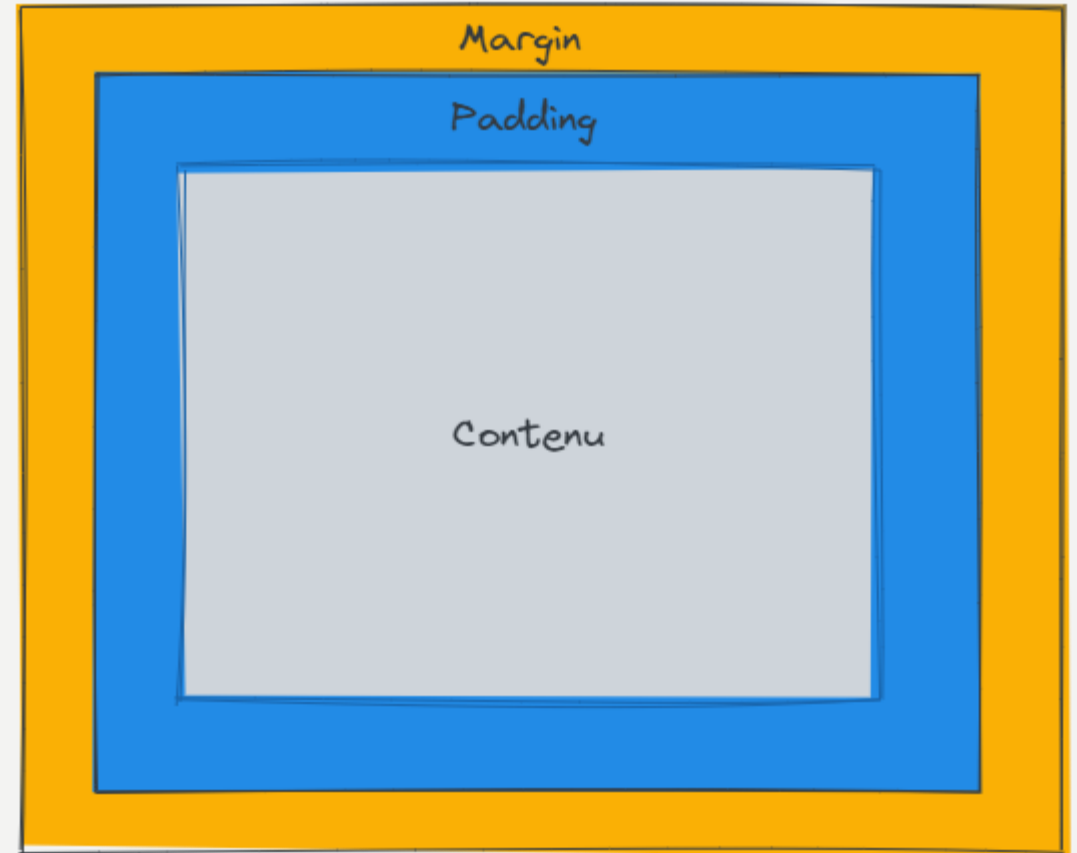
A decorative wavy line in light blue and white, running vertically along the left side of the slide.

MARGES

LES MARGES EN CSS3

MARGES

- Il existe deux types de marges
 - Marges externes (margin)
 - Marges internes (padding)
- Dans le sens des aiguilles d'une montre
 - haut, droit, bas, gauche



A decorative wavy line in light blue and white, running vertically along the left side of the slide.

POSITIONS

LE POSITIONNEMENT EN CSS3

POSITIONNEMENT

- Deux familles d'éléments se distinguent par leur comportement avec les autres
 - Éléments block
 - Éléments inline
- Les éléments « block »
 - Prennent toute la largeur de la page (100%)
 - Se positionnent sous l'élément précédent, qu'il soit block ou inline
 - C'est le cas des p, div, section, img
- Les éléments « inline »
 - Prennent la place nécessaire
 - Se positionnent à côté de l'élément précédent s'il est lui-même inline, sinon en dessous
 - C'est le cas des a, span, b, i, u, input, label, small

POSITIONNEMENT

- On peut modifier le comportement de ces éléments en précisant la propriété
 - « display »
 - block, inline ou inline-block
 - Ou « float »
 - right, left

```
img.je-me-met-a-cote {  
  float: left;  
}
```

```
div.je-me-met-en-ligne {  
  display: inline;  
}
```

```
a.je-suis-block-mais-je-me-met-en-ligne {  
  display: inline-block;  
}
```

```
small.apres-vous {  
  display: block;  
}
```

POSITIONNEMENT

- On peut modifier le comportement en modifiant la propriété « position »
 - static
 - Positionnement par défaut
 - fixed
 - L'élément reste figé sur la page, peu importe sa longueur
 - relative
 - L'élément est positionné relativement par rapport à sa position originale
 - absolute
 - L'élément se positionnera à une position absolue donnée
 - Absolue par rapport à son premier parent positionné (autre que static, valeur par défaut)
 - Si pas de parent positionné, absolue par rapport au document (à la page)
 - sticky
 - L'élément est positionné en fonction du scroll de l'utilisateur
 - C'est un mélange entre fixe et relative : il est relative jusqu'à ce que l'utilisateur scroll dessus

```
div.je-vais-ou-je-veux {  
  position: absolute;  
  top: 20px;  
  left: 50px;  
}
```

POSITIONNEMENT – FLEXBOX

- On peut modifier le comportement en utilisant la nouvelle technique « flexbox »
 - En utilisant la propriété « display »
 - Sur l'élément parent

```
.parent-flexible {  
  display: flex;  
}
```

POSITIONNEMENT – FLEXBOX

- On utilise d'autres propriétés pour adapter le contenu
 - flex-direction Imposer la direction (colonne ou ligne)
 - flex-wrap Permettre de retourner à la ligne si pas assez de place
 - justify-content Justifier le contenu sur l'axe principal (largeur par défaut)
 - align-items Justifier le contenu sur l'axe secondaire (hauteur par défaut)
- Chaque élément enfant d'un *flex* peut décider de ne pas suivre le mouvement
 - align-self Changer l'alignement pour cet élément
 - order Ordonner cet élément
 - flex-grow Dimension d'un élément (1, 2, 3, etc.)

POSITIONNEMENT – FLEXBOX

```
.parent-flexible-enfants-centres {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
.mouton-noir {  
  align-self: flex-end;  
}
```

Pour un génération (et démo) CSS : <https://www.cssportal.com/css-flexbox-generator/>

Pour une documentation complète : https://www.w3schools.com/css/css3_flexbox.asp

POSITIONNEMENT – GRID

- On peut modifier le comportement en utilisant la nouvelle technique « grid »
 - En utilisant la propriété « display »
 - Sur l'élément parent

```
.parent-grille {  
  display: grid;  
}
```

POSITIONNEMENT – GRID

- On utilise d'autres propriétés pour adapter le contenu

- grid-template-columns Liste des colonnes, séparées par un espace
- grid-template-rows Liste des lignes, séparées par un espace

Il est possible d'utiliser la fonction repeat pour répéter plusieurs lignes ou plusieurs colonnes identiques

- grid-template-area Défini des zones utilisables

- Chaque élément enfant d'une *grid* peut décider où se placer

- grid-column-start Colonne de départ
- grid-column-end Colonne de fin
- grid-column Colonne de début / colonne de fin
- grid-area Zone à utiliser

POSITIONNEMENT – GRID

- Pour les templates, une nouvelle unité `fr` (fraction) est disponible
 - Proportion de place disponible

```
grid-template-columns: 1fr 2fr 1fr;
```

```
grid-template-columns: 25% 50% 25%;
```

Pour un génération (et démo) CSS : <https://cssgrid-generator.netlify.app/>

Pour une documentation complète : https://www.w3schools.com/css/css_grid.asp

POSITIONNEMENT – GRID

```
.grid {
  display: grid;
  grid-template-columns: 80px auto;
  grid-template-rows: auto 1fr;
  grid-template-areas:
    "menu header"
    "menu content";
}

nav {
  grid-area: menu;
}

header {
  min-height: 100px;
  grid-area: header;
}

main {
  grid-area: content;
}
```

```
.grid {
  display: grid;
  grid-template-columns: 80px auto;
  grid-template-rows: auto 1fr;
}

.grid .menu {
  grid-column: 1;
  grid-row: span 2;
}

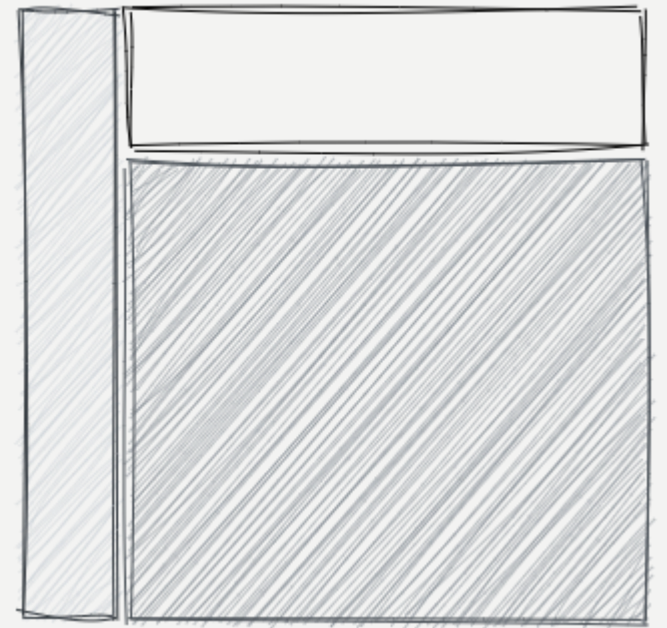
.grid .header {
  grid-column: 2;
  grid-row: 1;
}

.grid .content {
  grid-column: 2;
  grid-row: 2;
}
```

grid-row: 1 / 3;

EXERCICE

- Sur la page d'accueil
 - Mettre la taille de police à 14px sur la racine
 - Background général gris (#F0F0F0)
 - Mettre le titre 1 en taille 1,5rem, couleur hsl 226, 34%, 24%
 - Le header en background blanc, hauteur minimum de 100px
 - Padding 20px sur header et contenu
 - Définir le gras du texte en CSS, et le mettre en rouge (retirer la balise ou)
 - Appliquer une couleur de fond blanc crème (#F9F9F9) à tout le menu
 - Menu 100% de hauteur, 80px de largeur
 - Les liens seront des icônes : remplacer le texte par une lettre (« A », « C »)
 - Faire en sorte que les liens soient les uns sous les autres, centrés
 - Faire en sorte que les liens ne soient plus soulignés ni coloriés par le navigateur
 - Avec une couleur gris clair #a9abac
 - Changer la couleur des liens au survol, les passer en vert hsl 153, 48%, 49%



EXERCICE

- Formulaires
 - Retirer les "< br />" du formulaire, s'il y en a (utiliser autre chose pour le retour à la ligne)
 - Les labels doivent avoir la même largeur (200px)
 - Couleur hsl 232, 14% 68%
 - Taille 0,9rem
 - Police Roboto (Google Font à charger)
 - Mettre en forme le bouton submit
 - Arrondis 4px
 - Couleur de fond vert hsl 153, 48%, 49%
 - Couleur de texte blanc rgb 255, 255, 255
 - Marges internes 8px en haut et en bas, et 22px sur les côtés
 - Taille de police 0.95rem

EXERCICE

- Formulaires (suite)
 - Les inputs, select, textarea
 - Même largeur
 - Arrondis 4px
 - Police Montserrat (Google Font à charger)
 - Couleur de fond blanc
 - Couleur de texte noir (#363636)
 - Bordure 1px solid hsl 0, 0%, 86%
 - Couleur placeholder hsl 0 0% 50%
 - Padding haut bas 0,5em – 1px
 - Padding droite gauche 0,75em – 1px
 - Au survol, changer la couleur de la bordure : #B5B5B5
 - Au focus, ajouter une ombre -1px 3px 10px 0 rgb(0 0 0 / 6%)
 - Au focus visible, ajouter une outline
 - offset 3px
 - width 1px
 - style dashed
 - color hsl 0, 0% 85%

PSEUDO-ÉLÉMENTS

- Un élément qui peut contenir d'autres éléments peut avoir 2 pseudos-éléments
 - Un avant (::before)
 - Un après (::after)
- Chaque pseudo-élément doit avoir un contenu, même vide

```
p::after { content: "" }
```

```
p::after { content: "a" }
```

- Il est possible de le designer en CSS, un pseudo-élément est comme un élément !
 - Sauf qu'il n'interagit pas avec l'utilisateur

Un input ne pouvant pas avoir de contenu ne peut donc pas avoir de pseudo-élément !

EXERCICE – ALLER PLUS LOIN

- Travailler les positions et les pseudo-éléments
- Ajouter une bulle de couleur (verte) sur un lien
 - La bulle doit se présenter devant le texte du lien



TRANSITIONS

LES CHANGEMENTS D'ÉTAT EN CSS3

TRANSITIONS

Changer la couleur d'un texte au survol, en créant un effet de transition

Définir la durée de la transition

Définir les propriétés de la transition

```
p {  
  background-color: blue;  
  transition-property: background-color; /* Transition sur background-color */  
  transition-duration: 1s; /* Durée 1s */  
}  
  
p:hover { background-color: green; }
```

TRANSITIONS

On peut appliquer une accélération sur la transition (fonction de timing)

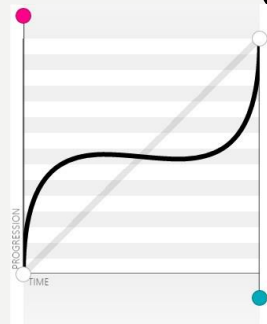
- ease Rapide au début, ralenti à la fin
- linear Vitesse constante
- ease-in Lent à l'approche, accélère à la sortie
- ease-out Rapide à l'approche, ralenti à la sortie
- ease-in-out Approche et sortie lente

```
{ transition-timing-function: ease; }
```

```
{ transition-timing-function: cubic-bezier(0,1.1,1,-0.1); }
```

Aller plus loin avec la fonction cubic-bezier (<https://cubic-bezier.com/>)

- Paramétrez vos accélérations
 - Très rapide à l'approche
 - Légère régression
 - Puis très rapide à la sortie



TRANSITIONS

Une seule propriété qui regroupe toutes les propriétés de transition ?

- La propriété « transition »

```
{ transition: background-color 1s cubic-bezier(0,1.1,1,-0.1); }
```

Plusieurs transitions en même temps ?

```
{ transition: background-color 2s ease, width 1s linear; }
```

```
{ transition: all 2s ease-in; }
```

EXERCICE

- Animer le changement de couleur des liens au survol
- Animer la bulle sur le lien : utiliser transform scale
- Animer la couleur de bordure des inputs au focus
- Animer le outline des inputs au focus-visible

A decorative wavy line in light blue and white, flowing from the top left towards the bottom left of the slide.

ANIMATIONS

LES ANIMATIONS D'ÉTAT EN CSS3

ANIMATIONS

Plus de détails que les transitions, on peut passer par autant d'états qu'on souhaite

L'animation se configure en « keyframes » dans laquelle on définit les étapes d'animation

On applique l'animation grâce à la propriété « animation »

- On peut y ajouter
 - Une accélération, comme avec les transitions
 - Un nombre d'exécutions (n ou infinite)
 - Une direction

ANIMATIONS

```
@keyframes monanimation {  
  0% { transform: translateX(0px); }  
  50% { transform: translateX(150px); }  
  100% { transform: translateX(150px) rotate(30deg); }  
}  
  
p {  
  width: 200px;  
  height: 150px;  
  background-color: red;  
}  
  
p:hover { animation: monanimation 2s; }
```

EXERCICE

- Lorsqu'une zone de saisie est active
 - Créer un effet « secouer » de la gauche vers la droite
 - Utiliser la propriété "transform" et les valeurs "translateX"



PRÉFIXES

LES PRÉFIXES D'ÉTAT EN CSS3

LES PRÉFIXES

Même si on tend vers une harmonisation des standards, certaines propriétés CSS sont expérimentales

Utilisés avec les propriétés CSS qui n'ont pas encore été standardisées

- Border-radius, transition, transform, animation, @keyframes

Mozilla Firefox

- -moz-transition

Google Chrome & Safari

- -webkit-transition

Opera

- -o-transition

LES PRÉFIXES

```
@-webkit-keyframes monanimation {  
  from { -webkit-transform: translateX(0px); }  
  to { -webkit-transform: translateX(150px); }  
}
```

```
@-moz-keyframes monanimation {  
  from { -moz-transform: translateX(0px); }  
  to { -moz-transform: translateX(150px); }  
}
```

```
@-o-keyframes monanimation {  
  from { -o-transform: translateX(0px); }  
  to { -o-transform: translateX(150px); }  
}
```

```
@keyframes monanimation {  
  from { transform: translateX(0px); }  
  to { transform: translateX(150px); }  
}
```

A decorative wavy line in light blue and white, flowing vertically along the left side of the slide.

RESPONSIVE

LE RESPONSIVE EN CSS3

RESPONSIVE

- Adapter le contenu de son site aux formats Smartphones, Tablettes, ou Imprimantes : les @media queries
- Sur-couche aux sélecteurs CSS existants
- Défini le cadre d'application du style CSS

```
@media screen and (max-width: 1280px)
{
  /* Propriétés CSS pour un écran ne dépassant pas 1280px de largeur */
}
```

- Pour forcer les smartphones à adapter leur taille d'écran en px
 - Utiliser de la balise meta « viewport »

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

RESPONSIVE

color	Gestion de la couleur (bits) minimum ou maximum
height	Hauteur de la fenêtre
width	Largeur de la fenêtre
device-height	Hauteur du périphérique
device-width	Largeur du périphérique
orientation	Orientation du périphérique (portrait ou paysage)
media	Type d'écran
screen	Écran « classique »
print	Impression
all	Tous les types d'écran

https://developer.mozilla.org/fr/docs/Web/CSS/Media_Queries/Using_media_queries

RESPONSIVE

only	Uniquement
and	Et
not	N'est pas
,	Ou

```
/* Sur les écrans, fenêtre longue de 1280px maximum */  
@media screen and (max-width: 1280px) { }
```

```
/* Sur tous les écrans quand la largeur de la fenêtre n'est pas comprise entre 1024px et 1280px */  
@media not all and (min-width: 1024px) and (max-width: 1280px) { }
```

```
/* Sur les imprimantes */  
@media print { }
```

```
/* Sur tous les écrans orientés verticalement */  
@media all and (orientation: portrait) { }
```

EXERCICE

- Sur les écrans de largeur inférieure à 600px
 - Réduire la taille du texte général
 - Cacher le menu

RESPONSIVE

- Les @media queries joue sur l'ensemble de l'écran
- Les @container queries permettent de jouer sur un élément cible
 - Attention, ce n'est pas compatible avec tous les navigateurs !

```
.container {  
  container-type: inline-size;  
  container-name: demo;  
}  
  
@container demo (max-width: 500px) {  
  .container label {  
    background: red;  
  }  
}
```

A decorative wavy line in light blue and white, flowing from the top left towards the bottom left of the slide.

VARIABLES

LE VARIABLES EN CSS3

VARIABLES

- CSS3 est statique
 - Utilisation de LESS, SASS ou autre
 - Les fichiers sont « compilés » en CSS
- Mais, il existe depuis peu la fonction CSS « var »
 - Qui permet de charger la valeur d'une variable préalablement définie
 - Permet par exemple de changer rapidement une couleur

VARIABLES

```
:root {  
  --success-h: 153;  
  --success-s: 48%;  
  --success-l: 49%;  
  --success: hsl(var(--success-h), var(--success-s), var(--success-l));  
  --success-light: hsl(var(--success-h), var(--success-s), calc(var(--success-l) + 10%));  
  --success-dark: hsl(var(--success-h), var(--success-s), calc(var(--success-l) - 10%));  
}  
  
input[type="submit"] {  
  background: var(--success-light);  
}
```

EXERCICE

- Mettre les couleurs & la police en variable