**Commandes en CSS**

Bonnes pratiques CSS

|  |  |
| --- | --- |
| ***Pour les sélecteurs :***  Pas d'Id !!!  Pas de balise a.button  .button a  Eviter les selecteur de descendance .article .title  préférer ">" plus prévisible **.article > .title**  Pas de **+**, **~**, **:nth-child( )**, ...  Sélecteur idéal : une seule classe ! **.maClasseSpecifiqueCSS**  en nommant le visuel, le rendu ou la fonction visuelle  .primary-button (bouton principal)  .highlight-box (élément de mise en avant, de promotion) | ***Pour relier au HTML :***  classes spécifique pour MeP nommée selon le principe :  BEM = Block, Element and Modifier  .block--modifier  .block\_\_element--modifier  *Exemple:*  <ul class="menu">  <li class="menu\_\_tab">  <li class="menu\_\_tab">  <li class="menu\_\_tab menu\_\_tab--actif">  </ul> |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbole | Signification | Code | Utilisation | Exemple |
| box-sizing | il est conseillé de placer cette propriété en premier lieu dans tous les projets CSS  lorsqu'on indique à un bloc une width ou height => fera exactement cette valeur et pas cette valeur + ou - margin, padding, ... | \* {  box-sizing: inherit;  }  body {  box-sizing: border-box;  } |  | Rq: inherited permet d'éviter de casser la cascade (si redéfinition de border-box dans un élément, ses enfants hériterons de la redéfinition !) |
|  |  |  |  |  |
| Sélecteurs |  |  |  |  |
| selecteur {  } | Changement des propriétés d’un élément de la page HTML | selecteur {  prop1: valeurProp1;  prop2: valeurProp2;  } |  |  |
| selecteurA, selecteurB {  } | Appliquer les mêmes modifications des propriétés à plusieurs éléments en même temps. | selecteurA, selecteurB {  prop1: valeurProp1;  prop2: valeurProp2;  } |  |  |
| \* | Sélecteur universel  Regroupe tous les éléments de la page HTML associée | \* {  prop1: valeurProp1;  prop2: valeurProp2;  } | pour modifie de la même manière absolument tous les éléments de la page HTML |  |
| /\* \*/ | Commentaire | /\* Texte du commentaire \*/ |  |  |
| . | Classe  pour modifier de la même façon tous les éléments appartenant à cette classe | #nomclasse {  prop1: valeurProp1;  …  } |  |  |
| # | ID  pour modifier un élément particulier de manière spécifique | .nomid {  prop1: valeurProp1;  …  } |  |  |
| Sélecteurs composés |  |  |  |  |
| selecteurA selecteurB {  } | Sélectionne uniquement les balises de type B incluses dans les balises de type A (enfants directs et indirects) | selecteurA selecteurB {  …  } |  | div p {  color: blue;  } |
| selecteurA + selecteurB {  } | Sélectionne uniquement la première balise de type B qui suit chaque balise de type A (balises sœurs) | selecteurA + selecteurB {  …  } |  |  |
| selecteurA.valeurClasse {  } | Sélectionne les selecteurs de type A possédant la valeur de la classe indiquée | selecteurA.valeurClasse {  } |  |  |
| selecteurA[attributT] {  } | Sélectionne toutes les balises de type A possèdent un attribut de type T | selecteurA[attributT] {  …  } |  |  |
| selecteurA[attributT="valeurV"] {  } | Sélectionne toutes les balises de type A présentant une valeur V à leur attribut T | selecteurA[attributT="valeurV"] {  …  } |  |  |
| selecteurA[attributT\*="valeurV"] {  } | Sélectionne toutes les balises de type A contenant V dans leur attribut T | selecteurA[attributT\*="valeurV"] {  …  } |  | a[title\*="ici"] {  …  } |
|  |  |  |  |  |
| Pseudo-formats |  |  |  |  |
| :hover {  } | Modifier la MeF d'un élément au survol de la souris | selecteur:hover {  prop1: valeurProp1;  …  } |  |  |
| :active {  } | Modifier la MeF (d'un lien) au moment où on clique dessus |  |  |  |
| :focus {  } | Modifier la MeF d'un élément lorsqu'il est sélectionné |  | Le plus souvent appliqué aux liens (lien encadré après lui avoir cliqué dessus)  et pour les formulaires (zone active) |  |
| :visited {  } | Modifier la MeF d'un lien qui a déjà été visité |  |  |  |
| :required {  } | Modifier la MeF d'un élément de formulaire dont la réponse est obligatoire |  |  |  |
| :invalid {  } | Modifier la MeF d'un élément de formulaire dont la réponse est incorrecte. |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | Rq: la spécificité du ciblage des éléments => quelle valeur de propriété sera prioritaire  (ensuite, à priorité égale, c'est la valeur définie en dernier qui sera appliquée) | # (id)  . : [ ] (classe, pseudo-classes, atributs)  p :: (élément et pseudo-élément)  \* (tous les éléments) | => 1000 pts  => 1 pt  => 0,1 pt  => 0 pt |  |
|  |  |  |  |  |
| Propriétés générales |  |  |  |  |
| opacity: | Transparence | opacity: nb: | la valeur doit être comprise entre  0 (totalement transparent)  et 1 (totalement opaque) |  |
| color: rgba( );  background-color: rgba( ); | Couleur transparente | color: rgba(nbR, nbV, nbB, nbA); | la couleur définie par le code RVB aura une transparence de valeur nbA (comprise entre 0 et 1) |  |
| transform: | Permet d'appliquer certains effets comme :  inclinaison | rotate(θdeg); | voir *CSS-avance-Commandes* |  |
| page-break-inside: | gère les sauts de pages et retour à la ligne au sein de l'élément  saut autorisé  saut interdit avant la fin de l'élément | auto; (défaut)  avoid; |  |  |
|  |  |  |  |  |
| Dimensions des Blocs |  |  |  |  |
| width: | Largeur du bloc | width: nbpx;  width: nb%;  width: nbvw; | % => 100% = largeur totale de la page  vw = Viewport Width : portion de fenêtre affichée à l'écran (100vw = la largeur de la fenêtre) |  |
| height: | Hauteur du bloc | height: nbpx;  height: nb%;  height: nbvh; | vh = ViewPort Height |  |
| min-width: | Largeur minimale du bloc | min-width: nbpx; |  |  |
| … |  |  |  |  |
| max-heigth: | Hauteur maximale du bloc |  |  |  |
|  |  |  |  |  |
| Arrière plan |  |  |  |  |
| background-color: | Change la couleur de fond | background-color: red;  background-color: #FE0000;  background-color: rgb(255, 0, 0); | peut s’appliquer à un élément (p, div, …)  peut s’appliquer à toute la page (body) |  |
| background-image: url(" "); | Image de fond  une seule image  plusieurs images | background-image: url("adresseImage");  background-image: url("adresseImage"), url("adresseImage"); | dans le cas de plusieurs images, la première est placée au-dessus de la deuxième, … |  |
| background-attachment: | Fixer l’arrière-plan  fixé  défilant avec le texte (défaut) | background-attachment: fixed;  background-attachment: scroll; |  |  |
| background-repeat: | Répétition du fond  non répété  répété sur la 1re ligne  répété sur la 1re colonne  répété en mosaïque (défaut) | background-repeat: no-repeat;  background-repeat: repeat-x;  background-repeat: repeat-y;  background-repeat: repeat; |  |  |
| background-position: | Position du fond  en haut/bas  gauche/droite  centré | top; / bottom;  left; / right;  center; | il est possible de combiner ces mots pour définir une position plus précise  ex. : top right; |  |
| background: | Combine les 4 dernières valeurs en une seule ligne de code | background: url("adresseImage") fixed no-repeat top right; | les valeurs des propriétés sont séparés par des espaces  leur ordre et leur nombre n’a pas d’importance |  |
|  |  |  |  |  |
| Bordures |  |  |  |  |
| border: | Créer une bordure encadrant l’élément  9 types de bordures | border: taillepx couleur type;  none; (défaut)  dotted; groove;  dashed; ridge;  solid; inset;  double; outset; |  |  |
| border-top:  border-right:  border-bottom:  border-left: | Modifier une seule bordure |  |  |  |
| border-radius: | Arrondir les coins des bordures  Créer des bordures éliptiques | border-radius: taillepx;  border-radius: pourcentage%  border-radius: tailleApx / tailleBpx; |  |  |
| border-radius: | Pour arrondir chaque coin de manière différente | border-radius: TLpx TRpx BRpx BLpx |  |  |
|  |  |  |  |  |
| Ombres |  |  |  |  |
| box-shadow: | Ajouter une ombre à l’élément | box-shadow: décalageHorizontalpx décalageVerticalpx adoucissementDégradépx couleur; | Si on veut placer l'ombre à l'intérieur de l'élément, on rajoutera la 5e valeur  inset;  juste après la couleur. |  |
| text-shadow: | Ajouter une ombre au texte contenu dans l’élément | Idem |  |  |
|  |  |  |  |  |
| Texte |  |  |  |  |
| font-size: | Taille du texte | font-size: Xpx;  font-size: Xem; |  |  |
| font-family: | Police | font-family: policeA, policeB, policeDeSecour; |  |  |
| font-style: | Texte en italique  oblique  normal | font-style: italic;  font-style: oblique;  font-style: normal; | pour annuler l’effet italique de <em> |  |
| font-weight: | Texte en gras  normal | font-weight: bold;  font-weight: normal; | pour annuler l’effet italique de <strong> |  |
| text-decoration: | Texte souligné  barré  avec ligne au-dessus  clignotant  normal | text-decoration: underline;  text-decoration: line-through;  text-decoration: overline;  text-decoration: blink;  text-decoration: none; |  |  |
| text-align: | Alignement du texte :  à gauche  centré  à droite  justifié | text-align: left;  text-align: center;  text-align: right;  text-align: justify; |  |  |
| color: | Donner une couleur au texte | color: red;  color: #FE0000;  color: rgb(255, 0, 0); |  |  |
| line-height: | Interligne | line-height: 30px;  line-height: normal; |  |  |
| @font-face { | Définir une nouvelle police accompagnant la page | @font-face {  font-family: 'nomNewPolice';  src: url('adresseNewPolice.eot');  src: url('adresseNewPolice.eot?#iefix') format('embedded-opentype'),  url('adresseNewPolice.woff') format('woff'),  url('adresseNewPolice.ttf') format('truetype'),  url('adresseNewPolice.svg#airstreamregular') format('svg');  font-weight: normal;  font-style: normal;  } | les fichiers de la nouvelle police sous diférents format doivent accompagner la page  les nouvelles polices peuvent être téléchargées sur www.fontsquirel.com |  |
|  |  |  |  |  |
| Listes |  |  |  |  |
| list-style-type: | Contrôle les puces des listes  pas de puce  disques  cercle  … | none;  disc; (défaut)  circle;  … |  |  |
| list-style-image: | Replacer les puces des listes par une image | list-style-image: url("adresseImage"); |  |  |
| page-break-inside: | les éléments de la liste ne seront pas séparés, pas de retour possible dans une autre colonne au milieu de la liste | page-break-inside: avoid; |  |  |
| backgroung-image: | autre méthode pour remplacer les puce par des images permettant un contrôle plus poussé de l'image (taille, position, transparence, …) | maListe {  list-style-type: none;  }  mesElements {  background-image: url('adresse de l'image');  background-reapeat: no-repeat;  padding-left: 30px;  } |  |  |
| Tableaux |  |  |  |  |
| td {  border: | Donner une bordure aux cellules du tableau |  |  |  |
| table {  border-collapse: | Fusionner les bordures des cellules  bordures séparées  bordures fusionnées | separate; (défaut)  collapse; |  |  |
| table-layout: | contrôle la largeur des colonnes  taille adaptée à chaque colonne  toutes les colonnes de la même taille | auto; (défaut)  fixed; |  |  |
| table {  border-spacing: | espace entre les cellules |  |  | Remarque : ne peut pas se faire avec margin |
| table {  caption-side: | Position du titre du tableau  au-dessus du tableau  en-dessous du tableau | top; (défaut)  bottom; |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Mise en page |  |  |  |  |
|  |  |  |  |  |
| Contenu dépassant du bloc |  |  |  |  |
| overflow: | Contrôle le contenu dépassant les dimenssions du bloc | overflow: visible; (défaut)  overflow: hidden;  overflow: scroll;  overflow: auto; (conseillé) |  |  |
|  |  |  |  |  |
| Marges |  |  |  |  |
| margin: | Taille des marges extérieures | margin: nbpx;  margin: Tpx Rpx Bpx Lpx;  margin: TBpx RLpx; |  |  |
| margin-top:  margin-right:  margin-bottom:  margin-left: | Modifier la taille d'une seule marge exterieure |  |  |  |
| margin: auto; | Centre le bloc dans la page | width: nbpx/%;  margin: auto; | cette commande ne fonctionne que si on a donné une largeur au bloc avant ! |  |
| padding: | Taille des marges intérieure | padding: nbpx;  padding: Tpx Rpx Bpx Lpx;  padding: TBpx RLpx; |  |  |
| padding-top:  padding-right:  … | Modifier la taille d'une seule marge interieure |  |  |  |
|  |  |  |  |  |
| Positionnement |  |  |  |  |
| float: | Elément flottant  retire l'élément du flux (vertical ou horizontal) et le place à la position indiquée :  à gauche  à droite | float: left;  float: right; | les autres objets vont entourer l’élément flottant et non plus simplement s’aligner avec lui (comme dans Word : « carré » et non plus « aligné sur le texte »). |  |
| clear: | Annule l’effet flottant  il place l'élément à la suite de tout élément flottant afin de ne pas se retrouvé accolé à lui, cad reprend le flux normal en dessous de l'élément flottant  nettoie le flux à gauche  à droite  des 2 cotés | clear: left;  clear: right;  clear: both; | **container**.clearfix::after {  content: '';  display: block;  clear: left;  }  *appliquée au conteneur qui contient un élément flottant => ajuste la taille du conteneur afin que l'élément flottant n'en sorte pas.* | technique du clearfix : ajout un dernier élément (chaine vide : "") à notre conteneur parent qui va devoir passer après les éléments flottant (clear: ) et donc permettre d'agrandir le conteneur pour s'assurer qu'il contient bien son/ses éléments flottants, cad que ceux-ci ne dépassent pas du conteneur. |
| display: | Transforme le type d'un élément  bloc en inline  inline en bloc  blocs alignés  masquer l'élément  rend les blocs flexibles | inline;  block;  inline-block;  none  flex; (=> voir prop. MeP plus bas) | propriété à utiliser à bon escient uniquement quand c'est nécéssaire. Le plus gros de la MeP doit être fait avec Flex (=> voir prop. MeP plus bas) |  |
| display: inline-block;  vertical-align: | Contrôle l'alignement des inline-block  base élément avec base parent  distance de la ligne de base  aligné en haut  centré verticalement  aligné en bas | baseline; (défaut)  nb px/%;  top;  middle;  bottom; |  |  |
| vertical-align: | contrôle l'alignement vertical des éléments |  |  |  |
| position: | Impose la position d'un élément  position absolu dans la page  figé à l'écran  autour de sa position par défaut  affiché à l'écran en permanance | absolute;  fixed;  relative;  sticky; => voir ci-contre | sticky = position "normale" dans le flux tant qu'il est visible, puis devient **fixed** lorsque qu'il disparait pour rester afficher à l'écran en permanence |  |
| position: ;  top:  right:  bottom:  left: | Distance par rapport à la référence | nb px/%; | **absolute** / **fixed** : ref = bords de page ou bords du bloc parent si lui aussi a une position définie (abs/fix/rel).  **relative** : ref = position élément par défaut |  |
| position: ;  z-index: | Ordonne la pile des éléments | z-index: nb; | plus la valeur est élevée, plus l'élément sera haut dans la pile |  |
| FlexBox |  | **Toutes les propriétés de cette section s'appliquent au conteneur et non pas aux éléments à organiser eux-mêmes (sauf** align-self**,** order **et** flex**)** |  |  |
| display: flex; | Aligne les blocs d'un conteneur et les rend flexibles ! | conteneur {  display: flex;  } | cette propriété doit être appliquée au conteneur pour que les propriétés suivantes de MeP puissent être utilisées dans le même conteneur |  |
| flex-direction: | Contrôle la direction d'alignement des blocs  en ligne  en ligne en ordre inversé  en colonne  en colonne en ordre inversé | row; (défaut)  row-reverse;  column;  column-reverse; |  |  |
| flex-wrap: | Retour à la ligne des blocs s'ils n'ont pas assez de place  pas de retour à la ligne  retour à la ligne  retour à la ligne en sens inv. | nowrap; (défaut)  wrap;  wrap-reverse; |  |  |
| align-content: | Controler la répartition des lignes dans le cas où les éléments occupent plusieurs lignes  étiré sur tout l'espace  au début, sans interlignes  centré, sans interl.  à la fin, sans intrel.  avec interlignes  idem + espace début et fin | stretch; (défaut)  flex-start;  center;  flex-end;  space-between;  space-around; |  |  |
| justify-content: | Réparti les éléments selon l'axe principal  alignés au début  centrés  alignés à la fin  espacés régulièrement  idem + espace au début et fin | flex-start; (défaut)  center;  flex-end;  space-between;  space-around; |  |  |
| align-items: | Aligne les éléments selon l'axe secondaire  étirés sur tout l'axe  au début  centré  à la fin  sur la ligne de base | stretch; (défaut)  flex-start;  center;  flex-end;  baseline; |  |  |
| display: flex;  margin: auto; | Centrer les éléments dans le conteneur selon les deux axes | element {  display: flex;  margin: auto;  } | code plus rapide que d'utiliser les deux propriétés précédentes en centré (mais résultat identique) |  |
| align-self: | Positionner un des éléments différemment des autres selon l'axe secondaire  mêmes valeurs que  align-items: | element {  align-self: valeur;  } | contrairement aux propriétés précédentes, celle-ci doit s'appliquer à l'élément concerné (et non pas au conteneur) |  |
| order: | Modifier l'ordre d'affichage des éléments (sans toucher au code HTML) | element {  order: nb;  } | les éléments seront rangés dans un ordre croissant.  s'applique aux éléments (et non pas au conteneur) |  |
| flex: | Contrôle la taille relative des éléments alignés  les elts se réduisent pour être contenu dans leur container (même si on leur a défini une taille spécifique) mais ne s'agrandissent pas  les elts ont une taille fixée  les elts s'agrandissent et rétrécissent en fonction de la taille du container | element {  flex: nb;  }  initial; (défaut)  none;  auto; | si un seul élément à une valeur, il s'étire pour combler tout l'espace vide de l'alignement  si plusieurs ont une valeur, ceux-ci s'étirent de manière relative par rapport à leur valeur  s'applique aux éléments (et non pas au conteneur) |  |
|  |  |  |  |  |
| Media Queries |  |  |  |  |
| @media {  } | Modifie la MeF des éléments compris entre les {} si la/les condition(s) est/sont vérifiée(s) | @media all and (max-width: 700px) {  @media all and (max-width: 700px), (orientation: landscape) { | *= pour tous les écrans, si la largeur de la zone d'affichage est inférieure à 700px*  *= … OU que l'orientation est en paysage* |  |
|  | les valeurs pouvant être attribuées à la media sont :  écran classique  mobile  télévision  projecteur  impression  tous les écrans | screen  handheld  tv  projection  print  all |  |  |
| autres règles | les autres règles peuvent être : | width: max-width: min-width:  height: max-height: min-height:  device-width: max-device-width: …  device-height: … …  color: orientation:  (: portrait/landscape) | plusieurs règles peuvent être combinées avec :  only and not , (= OU)  contrairement à media, ces autres règles et leur valeur doivent être entourés par des ( )  ATTENTION : **device-width** et **device-height** doivent être privilégiées pour discriminer les périph. portables |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbole | Signification | Code | Utilisation | Exemple |
| Sélecteurs spécifiques | sélectionne les éléments : |  |  |  |
|  |  |  |  |  |
| Sélecteurs d'attributs |  |  |  |  |
| [attributT] | possédant un attribut T | [attributT] |  |  |
| [T="V"] | possédant un attribut T qui contient totalement V | [attributT="valeur"] |  |  |
| [T\*="V"] | possédant un attribut T qui contient totalement ou partiellement V | [attributT\*="valeur"] |  |  |
| [T~="V"] | possédant un attribut T qui contient V délimitée par des espaces | [attributT~="valeur"] |  |  |
| [T^="V"] | possédant un attribut T qui commence par V | [attributT^="valeur"] |  |  |
| [T$="V"] | possédant un attribut T qui se termine par V | [attributT$="valeur"] |  |  |
| [T!="V"] | qui ne possède pas l'attribut T ou qui possède un attribut T différent de V | [attributT!="valeur"] |  |  |
| Sélecteurs hiérarchiques |  |  |  |  |
| SelecteurA SelecteurB | de type B inclus (directement et indirectement) dans A  = tous les descendants de A de type B | selecteurA selecteurB |  |  |
| A>B | de type B descendants direct de A | selecteurA > selecteurB |  |  |
| A+B | de type B suivant directement A (frères) | selecteurA + selecteurB |  | Rq : avec ou sans espaces, c'est pareil ! |
| A~B | de type B suivant directement ou indirectement A (frères) | selecteurA ~ selecteurB |  |  |
| A:empty | de type A n'ayant pas d'enfant | selecteurA:empty |  |  |
| A:first-child | de type A étant un premier enfant | selecteurA:first-child |  |  |
| A:nth-child( ) | de type A étant un ne enfant | selecteurA:nth-child(nb)  selecteurA:nth-child(2n+1) |  |  |
| A:last-child | de type A étant un dernier enfant | selecteurA:last-child |  |  |
| A:only-child | de type A étant enfant unique | selecteurA:only-child |  |  |
| Pseudo-sélecteurs d'éléments sélectionnés | (utile not. avec tableaux) |  |  |  |
| :first | premier élément de la sélection | sélection:first |  |  |
| :last | dernier élément de la sélection | sélection:last |  |  |
| :even | éléments dont l'indice est pairs | sélection:even |  |  |
| :odd | éléments dont l'indice impairs | sélection:odd |  |  |
| :eq( ) | élément dont l'indice vaut n | sélection:eq(nb) |  |  |
| :gt( ) | éléments dont l'indice est supérieur à | sélection:gt(nb) |  |  |
| :lt( ) | éléments dont l'indice est inférieur à | sélection:lt(nb) |  |  |
| Pseudo-sélecteurs spécifiques aux formulaires |  |  |  |  |
| :input | de type <input> |  |  |  |
| :password |  |  |  |  |
| :button | de type bouton |  |  |  |
| :checkbox |  |  |  |  |
| :checked | ayant été cochés |  |  |  |
| :radio |  |  |  |  |
| :selected | ayant été sélectionnés |  |  |  |
| :reset |  |  |  |  |
| :image | boutons de type image |  |  |  |
| :submit |  |  |  |  |
| :text |  |  |  |  |
| :focus | possédant le focus (un seul élément) |  |  |  |
| :enabled | ayant été validés |  |  |  |
| Pseudo-sélecteurs particuliers |  |  |  |  |
| :header | de type <h > (h1 à h6) | :header |  |  |
| :hidden | cachés | :hiden |  |  |
| :visible | visibles | :visible |  |  |
| :not( ) | qui ne correspondent pas au sélecteur donné en paramètre | :not(selecteur) |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |