100DaysOfCode – Applied visual design

Day2

1-

CSS de base: amélioration de la compatibilité avec les replis du navigateur

Lorsque vous travaillez avec CSS, vous rencontrerez probablement des problèmes de compatibilité de navigateur à un moment donné. C'est pourquoi il est important de prévoir des solutions de secours pour les navigateurs afin d'éviter tout problème potentiel.

Lorsque votre navigateur analyse le CSS d'une page Web, il ignore les propriétés qu'il ne reconnaît pas ou ne prend pas en charge. Par exemple, si vous utilisez une variable CSS pour attribuer une couleur d'arrière-plan sur un site, Internet Explorer ignorera la couleur d'arrière-plan car il ne prend pas en charge les variables CSS. Dans ce cas, le navigateur utilisera la valeur qu'il a pour cette propriété. S'il ne peut trouver aucune autre valeur définie pour cette propriété, il reviendra à la valeur par défaut, ce qui n'est généralement pas idéal.

Cela signifie que si vous souhaitez fournir une solution de secours du navigateur, il vous suffit de fournir une autre valeur plus largement prise en charge immédiatement avant votre déclaration. De cette façon, un navigateur ancien aura quelque chose sur lequel se replier, alors qu'un navigateur plus récent interprétera simplement la déclaration qui viendra plus tard dans la cascade.

2-

CSS de base: Modifier une variable pour une zone spécifique

Lorsque vous créez vos variables dans **:root**, ils définissent la valeur de cette variable pour toute la page.

Vous pouvez ensuite écraser ces variables en les définissant à nouveau dans un élément spécifique.

3-

CSS de base: utilisez une requête multimédia pour modifier une variable

Les variables CSS peuvent simplifier la manière dont vous utilisez les requêtes multimédia.

Par exemple, lorsque votre écran est plus petit ou plus grand que votre point d'arrêt d'interrogation de média, vous pouvez modifier la valeur d'une variable et appliquer son style à chaque utilisation.

**Introduction aux défis de la conception visuelle appliquée**

La conception visuelle dans le développement Web est un sujet vaste. Il combine la typographie, la théorie des couleurs, les graphiques, l'animation et la mise en page pour vous aider à diffuser le message d'un site. La définition de bon design est un sujet très discuté, avec de nombreux livres sur le sujet.

À la base, la plupart des contenus Web fournissent des informations à l'utilisateur. La conception visuelle de la page peut influencer sa présentation et l'expérience de l'utilisateur. En développement Web, HTML donne une structure et une sémantique au contenu d'une page, tandis que CSS en contrôle la mise en forme et l'apparence.

Cette section présente certains des outils de base utilisés par les développeurs pour créer leurs propres conceptions visuelles.

1. Applied Visual Design: Créer une balance visuelle à l'aide de la propriété text-align

Cette section du programme porte sur la conception visuelle appliquée. Le premier groupe de défis s'appuie sur la présentation de carte donnée pour illustrer un certain nombre de principes fondamentaux.

Le texte représente souvent une grande partie du contenu Web. CSS propose plusieurs options pour l'aligner sur la propriété text-align.

text-align: justifier; Toutes les lignes de texte, à l'exception de la dernière, se rencontrent aux bords gauche et droit de la zone de ligne.

text-align: center; centre le texte

text-align: right; aligne le texte à droite

Et text-align: left; (valeur par défaut) left-aligne le texte.

1. Applied Visual Design: Ajustez la largeur d'un élément à l'aide de la propriété width

Vous pouvez spécifier la largeur d'un élément à l'aide de la propriété width en CSS. Les valeurs peuvent être exprimées en unités de longueur relative (telles que em), en unités de longueur absolues (telles que px) ou en pourcentage de l'élément parent contenant. Voici un exemple qui modifie la largeur d'une image en 220px:

img {  
  width: 220px;  
}

1. Applied Visual Design: Ajuster la hauteur d'un élément à l'aide de la propriété height

Vous pouvez spécifier la hauteur d'un élément à l'aide de la propriété height en CSS, similaire à la propriété width. Voici un exemple qui modifie la hauteur d'une image en 20px:

img {

   height: 20px;

}

1. Applied Visual Design: utilisez la balise forte pour rendre le texte en gras

Pour rendre le texte en gras, vous pouvez utiliser la balise forte. Ceci est souvent utilisé pour attirer l'attention sur un texte et symboliser son importance. Avec la balise strong, le navigateur applique le code CSS de font-weight: bold; à l'élément.

1. Applied Visual Design: utilisez la balise u pour souligner du texte

Pour souligner du texte, vous pouvez utiliser la balise u. Ceci est souvent utilisé pour indiquer qu'une section de texte est importante ou un élément à garder en mémoire. Avec la balise u, le navigateur applique le CSS de text-decoration: underline; à l'élément.

1. Applied Visual Design: utilisez la balise em pour mettre du texte en italique

Pour mettre en valeur le texte, vous pouvez utiliser la balise em. Ceci affiche le texte en italique, car le navigateur applique les CSS du style de font-style: italic; à l'élément.

1. Applied Visual Design: Utilisez la balise s pour barrer le texte

Pour biffer du texte, c'est-à-dire lorsqu'une ligne horizontale traverse les caractères, vous pouvez utiliser la balise s. Cela montre qu'une partie du texte n'est plus valide. Avec la balise s, le navigateur applique le CSS de text-decoration: line-through; à l'élément.

Applied Visual Design: Ajuster la taille d'un en-tête par rapport à une balise de paragraphe

La taille de la police des balises d’en-tête (h1 à h6) doit généralement être supérieure à la taille de la police des balises de paragraphe. Cela permet à l'utilisateur de comprendre visuellement la disposition et le niveau d'importance de tout ce qui se trouve sur la page. Vous utilisez la propriété font-size pour ajuster la taille du texte d'un élément.

Applied Visual Design: Ajouter une ombre à un élément semblable à une carte

La propriété box-shadow applique une ou plusieurs ombres à un élément.

La propriété box-shadow prend des valeurs pour offset-x (la distance entre laquelle pousser l'ombre horizontalement et l'élément), offset-y (la distance entre laquelle pousser l'ombre verticalement et l'élément), le rayon de flou, le rayon d'étalement et la couleur. valeur, dans cet ordre. Les valeurs de flou et de rayon sont optionnelles.

Voici un exemple de CSS permettant de créer plusieurs ombres avec des couleurs floues, essentiellement transparentes :

box-shadow: 0 10px 20px rgba (0,0,0,0,19), 0 6px 6px rgba (0,0,0,0,23);

Applied Visual Design: Diminuer l'opacité d'un élément

La propriété opacity en CSS est utilisée pour ajuster l'opacité, ou inversement, la transparence d'un élément.

Une valeur de 1 est opaque, ce qui n’est pas du tout transparent.

Une valeur de 0,5 correspond à la moitié de la transparence.

Une valeur de 0 est complètement transparente.

La valeur donnée s'appliquera à tout l'élément, qu'il s'agisse d'une image avec une transparence ou des couleurs de premier plan et d'arrière-plan d'un bloc de texte.

Applied Visual Design: Diminuer l'opacité d'un élément

La propriété opacity en CSS est utilisée pour ajuster l'opacité, ou inversement, la transparence d'un élément.

Une valeur de 1 est opaque, ce qui n’est pas du tout transparent.

Une valeur de 0,5 correspond à la moitié de la transparence.

Une valeur de 0 est complètement transparente.

La valeur donnée s'appliquera à tout l'élément, qu'il s'agisse d'une image avec une transparence ou des couleurs de premier plan et d'arrière-plan d'un bloc de texte.

Applied Visual Design: utilisez la propriété text-transform pour mettre le texte en majuscule

La propriété text-transform en CSS permet de modifier l'apparence du texte. C'est un moyen pratique de s'assurer que le texte d'une page Web apparaît de manière cohérente, sans avoir à modifier le contenu textuel des éléments HTML réels.

Le tableau suivant montre comment les différentes valeurs de transformation de texte modifient le texte d'exemple "Transforme-moi".

Valeur Résultat

minuscule "transformez-moi"

MAJUSCULE "TRANSFORM ME"

capitaliser "Transformez-moi"

initial Utiliser la valeur par défaut

inherit Utiliser la valeur de transformation de texte de l'élément parent

none Par défaut: Utiliser le texte d'origine

Applied Visual Design: Définition de la taille de police pour plusieurs éléments de titre

La propriété font-size est utilisée pour spécifier la taille du texte dans un élément donné. Cette règle peut être utilisée pour plusieurs éléments pour créer une cohérence visuelle du texte sur une page. Dans ce défi, vous allez définir les valeurs de toutes les balises h1 à h6 pour équilibrer les tailles de titre.

Applied Visual Design: Définition de l'épaisseur de police pour plusieurs éléments de titre

Vous définissez la taille de la police de chaque balise de titre dans le dernier défi. Ici, vous allez ajuster le poids de la police.

La propriété font-weight définit l'épaisseur ou la minceur des caractères dans une section de texte.

Day 5

 Ajuster l'état de survol d'une balise d'ancrage

Ce défi concernera l'utilisation de pseudo-classes. Une pseudo-classe est un mot clé qui peut être ajouté à des sélecteurs, afin de sélectionner un état spécifique de l'élément.

Par exemple, le style d'une balise d'ancrage peut être modifié pour son état de survol à l'aide du :hoversélecteur de pseudo-classe. Voici le CSS pour changer le colorrouge de la balise d'ancrage en rouge pendant son survol:

2-

Applied Visual Design: Modifier la position relative d'un élément

CSS traite chaque élément HTML comme sa propre boîte, généralement désignée sous le nom de CSS Box Model. Les éléments de niveau bloc démarrent automatiquement sur une nouvelle ligne (pensez aux titres, aux paragraphes et aux divs), tandis que les éléments en ligne se trouvent dans le contenu environnant (comme des images ou des étendues). La disposition par défaut des éléments de cette manière est appelée la normal flowde un document, mais CSS propose la propriété de position pour le remplacer.

Lorsque la position d'un élément est définie sur relative, cela vous permet de spécifier comment CSS doit le déplacer par *rapport*à sa position actuelle dans le flux normal de la page. Il s'associe aux propriétés de décalage CSS de leftou right, et topou bottom. Celles-ci indiquent le nombre de pixels, de pourcentages ou d'EM pour déplacer l'élément *loin* de son emplacement habituel. L'exemple suivant déplace le paragraphe de 10 pixels du bas:

p {   
  position: relative;   
  bottom: 10px;   
}

Changer la position d'un élément en relatif ne le supprime pas du flux normal; les autres éléments autour de lui se comportent toujours comme si cet élément était dans sa position par défaut.

**Remarque Le**  
positionnement vous donne beaucoup de flexibilité et de puissance sur la disposition visuelle d'une page. Il est bon de se rappeler que, quelle que soit la position des éléments, le balisage HTML sous-jacent doit être organisé et pris en charge s'il est lu de haut en bas. C’est ainsi que les utilisateurs malvoyants (qui utilisent des appareils fonctionnels comme les lecteurs d’écran) accèdent à votre contenu.

Conception visuelle appliquée: éléments de poussée à gauche ou à droite avec la propriété float

L'outil de positionnement suivant n'utilise pas réellement position, mais définit la floatpropriété d'un élément. Les éléments flottants sont supprimés du flux normal d'un document et poussés vers l' élément parent leftou rightleur élément parent. Il est couramment utilisé avec la widthpropriété pour spécifier la quantité d'espace horizontal requise par l'élément flottant.

Le balisage donné fonctionnerait bien comme une mise en page à deux colonnes, avec les éléments sectionet asideles uns à côté des autres. Donnez au #leftpoint un floatde leftet l' #rightarticle un floatde right.

## Applied Visual Design: Modifier la position des éléments qui se chevauchent avec la propriété z-index

Lorsque les éléments sont positionnés de manière à se chevaucher, l'élément figurant plus tard dans le balisage HTML apparaîtra, par défaut, au-dessus des autres éléments. Cependant, la propriété z-index peut spécifier l'ordre dans lequel les éléments sont empilés les uns sur les autres. Ce doit être un entier (c'est-à-dire un nombre entier et non une décimale), et des valeurs plus élevées pour la z-indexpropriété d'un élément le déplacent plus haut dans la pile que celles ayant une valeur inférieure.

## Applied Visual Design: Centrer un élément horizontalement à l'aide de la propriété margin

Une autre technique de positionnement consiste à centrer un élément de bloc horizontalement. Une façon de faire est de définir sa marginvaleur sur auto.

Cette méthode fonctionne aussi pour les images. Les images sont des éléments en ligne par défaut, mais peuvent être modifiées pour bloquer des éléments lorsque vous définissez la propriété display à block.

## Applied Visual Design: En savoir plus sur les couleurs complémentaires

La théorie des couleurs et son impact sur le design constituent un sujet de fond et seules les bases sont abordées dans les défis suivants. Sur un site Web, la couleur peut attirer l'attention sur le contenu, susciter des émotions ou créer une harmonie visuelle. L'utilisation de différentes combinaisons de couleurs peut réellement changer l'apparence d'un site Web. De plus, une palette de couleurs adaptée à votre contenu peut être réfléchie.

La roue chromatique est un outil utile pour visualiser les relations entre les couleurs. Il s'agit d'un cercle dans lequel des teintes similaires sont voisines et des teintes différentes plus éloignées. Lorsque deux couleurs sont opposées sur la roue, elles sont appelées couleurs complémentaires. Ils ont pour caractéristique que s’ils sont combinés, ils s’annulent et créent une couleur grise. Cependant, lorsqu'elles sont placées côte à côte, ces couleurs apparaissent plus vibrantes et produisent un fort contraste visuel.

Quelques exemples de couleurs complémentaires avec leurs codes hexadécimaux sont:

red (#FF0000) and cyan (#00FFFF)  
green (#00FF00) and magenta (#FF00FF)  
blue (#0000FF) and yellow (#FFFF00)

Ce modèle est différent du modèle de couleur obsolète RYB que beaucoup d'entre nous ont appris à l'école, qui a des couleurs primaires et complémentaires différentes. La théorie des couleurs moderne utilise le modèle RVB additif (comme sur un écran d'ordinateur) et le modèle CMJ (K) soustractif (comme dans l'impression). Lisez [ici](https://en.wikipedia.org/wiki/Color_model) pour plus d'informations sur ce sujet complexe.

Il existe de nombreux outils de sélection des couleurs disponibles en ligne qui permettent de trouver le complément d’une couleur.

**Remarque**  
Pour tous les problèmes de couleurs: l'utilisation de la couleur peut être un moyen puissant d'ajouter un intérêt visuel à une page. Cependant, la couleur seule ne doit pas être utilisée comme seul moyen de transmettre des informations importantes car les utilisateurs malvoyants peuvent ne pas comprendre ce contenu. Ce problème sera traité plus en détail dans les défis Applied Accessibility.

Applied Visual Design: en savoir plus sur les couleurs tertiaires

Les écrans d’ordinateur et les écrans de périphériques créent différentes couleurs en combinant des quantités de lumière rouge, verte et bleue. C'est ce que l'on appelle le modèle de couleur additif RVB dans la théorie des couleurs moderne. Le rouge (R), le vert (G) et le bleu (B) sont appelés couleurs primaires. Le mélange de deux couleurs primaires crée les couleurs secondaires cyan (G + B), magenta (R + B) et jaune (R + G). Vous avez vu ces couleurs dans le défi Couleurs complémentaires. Ces couleurs secondaires sont le complément de la couleur primaire non utilisée dans leur création et sont opposées à cette couleur primaire sur la roue chromatique. Par exemple, le magenta est composé de rouge et de bleu et complète le vert.

Les couleurs tertiaires sont le résultat de la combinaison d’une couleur primaire avec l’un de ses voisins de couleur secondaire. Par exemple, le rouge (primaire) et le jaune (secondaire) font orange. Cela ajoute six couleurs supplémentaires à une roue chromatique simple pour un total de douze.

Il existe différentes méthodes de sélection de différentes couleurs qui permettent une combinaison harmonieuse du design. Un exemple pouvant utiliser des couleurs tertiaires est appelé jeu de couleurs complémentaire complémentaire. Ce schéma commence par une couleur de base, puis l'associe aux deux couleurs adjacentes à son complément. Les trois couleurs fournissent un contraste visuel fort dans un dessin, mais sont plus subtiles que d'utiliser deux couleurs complémentaires.

Voici trois couleurs créées à l'aide du schéma à complément fractionné:

| **Couleur** | **Code Hex** |
| --- | --- |
| Orange | # FF7D00 |
| cyan | # 00FFFF |
| framboise | # FF007D |

Day 6

## Applied Visual Design: Ajustez la couleur de divers éléments en couleurs complémentaires

Le défi Couleurs complémentaires a montré que les couleurs opposées sur la roue chromatique peuvent se rendre plus vives lorsqu'elles sont placées côte à côte. Cependant, le fort contraste visuel peut être choquant s'il est utilisé de manière excessive sur un site Web et peut parfois rendre le texte plus difficile à lire s'il est placé sur un arrière-plan de couleur complémentaire. En pratique, l’une des couleurs est généralement dominante et le complément sert à attirer l’attention visuelle sur certains contenus de la page.

## Applied Visual Design: Ajuster la teinte d'une couleur

Les couleurs ont plusieurs caractéristiques, notamment la teinte, la saturation et la légèreté. CSS3 a introduit la hsl()propriété comme moyen alternatif de choisir une couleur en indiquant directement ces caractéristiques.

**La teinte** est ce que les gens considèrent généralement comme une "couleur". Si vous représentez un spectre de couleurs commençant du rouge à gauche, passant du vert au milieu au bleu et du bleu à droite, la teinte correspond à l'emplacement d'une couleur le long de cette ligne. Dans hsl()Teinte, utilise un concept de roue de couleur au lieu du spectre, où l'angle de la couleur sur le cercle est indiqué par une valeur comprise entre 0 et 360.

**La saturation** est la quantité de gris dans une couleur. Une couleur entièrement saturée ne contient pas de gris et une couleur faiblement saturée est presque entièrement grise. Ceci est donné en pourcentage avec 100% complètement saturé.

**La légèreté** est la quantité de blanc ou de noir dans une couleur. Un pourcentage allant de 0% (noir) à 100% (blanc) est donné, 50% étant la couleur normale.

Voici quelques exemples d'utilisation hsl()avec des couleurs de luminosité normales complètement saturées:

| **Couleur** | **HSL** |
| --- | --- |
| rouge | hsl (0, 100%, 50%) |
| jaune | hsl (60, 100%, 50%) |
| vert | hsl (120, 100%, 50%) |
| cyan | hsl (180, 100%, 50%) |
| bleu | hsl (240, 100%, 50%) |
| magenta | hsl (300, 100%, 50%) |

## Applied Visual Design : Ajuster le ton d'une couleur

L' hsl()option en CSS facilite également l'ajustement du ton d'une couleur. Mélanger du blanc avec une teinte pure crée une teinte de cette couleur et ajouter du noir fera une nuance. Vous pouvez également créer une nuance en ajoutant du gris ou en teintant et en ombrant. Rappelez-vous que les 's' et 'l' hsl()signifient respectivement saturation et légèreté. Le pourcentage de saturation change la quantité de gris et le pourcentage de luminosité détermine la quantité de blanc ou de noir dans la couleur. Ceci est utile lorsque vous avez une teinte de base que vous aimez, mais que vous en avez besoin de différentes variantes.

## Applied Visual Design: Créer un dégradé linéaire CSS graduel

L'application d'une couleur sur des éléments HTML ne se limite pas à une seule teinte plate. CSS fournit la possibilité d'utiliser des transitions de couleur, également appelées dégradés, sur des éléments. Ceci est accessible via la fonction de la backgroundpropriété linear-gradient(). Voici la syntaxe générale:

background: linear-gradient(gradient\_direction, color 1, color 2, color 3, ...);

Le premier argument spécifie la direction à partir de laquelle la transition de couleur commence - il peut être exprimé en degré, où 90 degrés crée un dégradé vertical et 45 degrés est incliné comme une barre oblique inverse. Les arguments suivants spécifient l'ordre des couleurs utilisées dans le dégradé.

Exemple:

background: linear-gradient(90deg, red, yellow, rgb(204, 204, 255));

## Applied Visual Design: Utiliser un dégradé linéaire CSS pour créer un élément rayé

La repeating-linear-gradient()fonction est très similaire à linear-gradient()la différence majeure de répéter le motif de dégradé spécifié. repeating-linear-gradient()accepte une variété de valeurs, mais pour des raisons de simplicité, vous travaillerez avec une valeur d'angle et des valeurs d'arrêt de couleur dans ce défi.

La valeur de l'angle est la direction du dégradé. Les arrêts de couleur sont comme des valeurs de largeur qui indiquent l'endroit où une transition a lieu et sont donnés avec un pourcentage ou un nombre de pixels.

Dans l'exemple présenté dans l'éditeur de code, le dégradé commence par la couleur yellowde 0 pixel qui se fond dans la deuxième couleur blueà 40 pixels du début. Etant donné que l’arrêt de couleur suivant est également à 40 pixels, le dégradé passe immédiatement à la troisième couleur green, qui se fond dans la quatrième valeur de couleur, redà 80 pixels du début du dégradé.

Pour cet exemple, il est utile de penser que les couleurs s'arrêtent par paires, toutes les deux couleurs se confondant.

0px [yellow -- blend -- blue] 40px [green -- blend -- red] 80px

Si toutes les deux valeurs de couleur d'arrêt ont la même couleur, le mélange n'est pas perceptible, car il se situe entre la même couleur, suivi d'une transition difficile vers la couleur suivante, de sorte que vous obtenez des rayures.

## Applied Visual Design: Créer une texture en ajoutant un motif subtil en tant qu'image d'arrière-plan

Une façon d'ajouter de la texture et de l'intérêt à un fond et de le rendre plus visible est d'ajouter un motif subtil. La clé est l’équilibre, car vous ne voulez pas que l’arrière-plan se détache trop et qu’il s’éloigne du premier plan. La backgroundpropriété prend en charge la url()fonction afin de créer un lien vers une image de la texture ou du motif choisi. L'adresse du lien est entourée de guillemets entre parenthèses.

En utilisant l'URL de https://i.imgur.com/MJAkxbh.png, définissez le backgroundde toute la page avec le bodysélecteur.

## Applied Visual Design: utilisez la propriété scale de transformation CSS pour modifier la taille d'un élément

Pour changer l'échelle d'un élément, CSS a la transformpropriété, ainsi que sa scale()fonction. L'exemple de code suivant double la taille de tous les éléments de paragraphe de la page:

p {   
  transform: scale (2);   
}

## Applied Visual Design: utilisez la propriété scale de transformation CSS pour redimensionner un élément au survol

La propriété css transform dispose d'une variété de fonctions qui vous permettent de redimensionner, déplacer, faire pivoter, incliner, etc., vos éléments. Lorsqu'elle est utilisée avec des pseudo-classes telles que :hover celle spécifiant un certain état d'un élément, la propriété transform peut facilement ajouter de l'interactivité à vos éléments.

Voici un exemple pour redimensionner les éléments de paragraphe à 2,1 fois leur taille d'origine lorsqu'un utilisateur les survole :

p: survol {   
  transformer: échelle (2.1);   
}

## Applied Visual Design: Utilisez la propriété de transformation CSS skewX pour incliner un élément le long de l'axe X

La fonction suivante de la transformpropriété est skewX()qui incline l’élément sélectionné le long de son axe X (horizontal) d’un degré donné.

Le code suivant incline l'élément de paragraphe de -32 degrés le long de l'axe des abscisses.

p {   
  transform: skewX (-32deg);   
}

## Applied Visual Design: utilisez skewY, propriété de transformation CSS, pour incliner un élément le long de l'axe des ordonnées

Étant donné que la skewX() fonction incline l'élément sélectionné le long de l'axe des X d'un degré donné, il n'est pas surprenant que la skewY()propriété biaise un élément le long de l'axe des Y (vertical).

Applied Visual Design: Créer un graphique à l'aide de CSS

En manipulant différents sélecteurs et propriétés, vous pouvez créer des formes intéressantes. L'un des plus faciles à essayer est la forme d'un croissant de lune. Pour ce défi, vous devez travailler avec la box-shadowpropriété qui définit l'ombre d'un élément, ainsi que la border-radiuspropriété qui contrôle l'arrondi des coins de l'élément.

Vous allez créer un objet rond et transparent avec une ombre nette légèrement décalée sur le côté. L'ombre sera en fait la forme de la lune que vous voyez.

Afin de créer un objet rond, la border-radiuspropriété doit avoir une valeur de 50%.

Vous souvenez peut - être un défi plus tôt que la box-shadowpropriété prend des valeurs pour offset-x, offset-y, blur-radius, spread-radiuset une valeur de couleur dans cet ordre. Les valeurs blur-radiuset spread-radiussont facultatives.

Applied Visual Design: créer une forme plus complexe à l'aide de CSS et HTML

L'une des formes les plus populaires au monde est la forme du cœur. Dans ce défi, vous en créerez une en utilisant du CSS pur. Mais d’abord, vous devez comprendre les éléments ::beforeet les ::afterpseudo-éléments. Ces pseudo-éléments sont utilisés pour ajouter quelque chose avant ou après un élément sélectionné. Dans l'exemple suivant, un ::beforepseudo-élément est utilisé pour ajouter un rectangle à un élément avec la classe heart:

.heart :: before {   
  content: "";   
  couleur de fond: jaune;   
  rayon de bordure: 25%;   
  position: absolue;   
  hauteur: 50px;   
  largeur: 70px;   
  en haut: -50px;   
  à gauche: 5px;   
}

Pour que les ::beforeet ::afterpseudo-éléments fonctionnent correctement, ils doivent avoir une contentpropriété définie . Cette propriété est généralement utilisée pour ajouter des éléments tels qu'une photo ou un texte à l'élément sélectionné. Lorsque les ::beforeet ::afterpseudo-éléments sont utilisés pour créer des formes, la contentpropriété est toujours nécessaire, mais il est défini sur une chaîne vide.

Dans l'exemple ci-dessus, l'élément de classe hearta un ::beforepseudo-élément qui produit un rectangle jaune avec heightet widthde 50px et 70px, respectivement. Ce rectangle a des coins arrondis en raison de son rayon de bordure de 25% et est positionné absolument à 5px de leftet à 50px au-dessus topde l'élément.

Applied Visual Design: Apprenez comment fonctionnent les propriétés @ Propriétés de l'animation et des images clés CSS

Pour animer un élément, vous devez connaître les propriétés de l'animation et la @keyframesrègle. Les propriétés de l'animation contrôlent le comportement de l'animation et la @keyframesrègle contrôle le déroulement de cette animation. Il y a huit propriétés d'animation au total. Ce défi restera simple et couvrira les deux plus importants en premier:

animation-namedéfinit le nom de l'animation, qui sera utilisé plus tard @keyframespour indiquer à CSS quelles règles vont avec quelles animations.

animation-durationdéfinit la durée de l'animation.

@keyframesest comment spécifier exactement ce qui se passe dans l’animation sur la durée. Cela se fait en donnant des propriétés CSS pour des "images" spécifiques pendant l'animation, avec des pourcentages allant de 0% à 100%. Si vous comparez cela à un film, les propriétés CSS pour 0% indiquent comment l'élément s'affiche dans la scène d'ouverture. Les propriétés CSS pour 100% indiquent comment l'élément apparaît à la fin, juste avant la liste des crédits. Ensuite, CSS applique la magie pour faire basculer l'élément sur la durée donnée afin de jouer la scène. Voici un exemple pour illustrer l'utilisation des @keyframespropriétés d'animation:

#anim {  
  animation-name: colorful;  
  animation-duration: 3s;  
}  
@keyframes colorful {  
  0% {  
    background-color: blue;  
  }  
  100% {  
    background-color: yellow;  
  }  
}

Pour l’élément avec l’ animid, l’extrait de code ci-dessus définit le paramètre animation-namesur colorfulet le paramètre animation-durationsur 3 secondes. La @keyframesrègle est ensuite liée aux propriétés d'animation portant le nom colorful. Il définit la couleur bleue au début de l'animation (0%) qui passera au jaune à la fin de l'animation (100%). Vous n'êtes pas limité aux seules transitions début-fin, vous pouvez définir les propriétés de l'élément pour tout pourcentage compris entre 0% et 100%.

Créez une animation pour l'élément avec l'identifiant recten définissant les animation-nameoptions arc-en-ciel et animation-duration4 secondes. Ensuite, déclarez une @keyframesrègle et réglez le background-colordébut de l'animation ( 0%) sur bleu, le milieu de l'animation ( 50%) sur vert et la fin de l'animation ( 100%) sur jaune.

Applied Visual Design: utiliser une animation CSS pour modifier l'état de survol d'un bouton

Vous pouvez utiliser CSS @keyframespour changer la couleur d'un bouton dans son état de survol.

Voici un exemple de modification de la largeur d'une image en survol:

<style>  
  img:hover {  
    animation-name: width;  
    animation-duration: 500ms;  
  }  
  
  @keyframes width {  
    100% {  
      width: 40px;  
    }  
  }  
</style>  
  
<img src="https://bit.ly/smallgooglelogo" alt="Google's Logo" />

Notez que msreprésente millisecondes, où 1000ms est égal à 1s.

Utilisez CSS @keyframespour changer le background-colorde l' buttonélément de sorte qu'il devient #4791d0quand un utilisateur survole. La @keyframesrègle ne devrait avoir qu'une entrée pour 100%.

Applied Visual Design: Modifier le mode de remplissage d'une animation

C'est génial, mais ça ne marche pas encore. Remarquez comment l'animation se réinitialise une fois 500mspassé, ce qui ramène le bouton à sa couleur d'origine. Vous voulez que le bouton reste en surbrillance.

Cela peut être fait en définissant la animation-fill-modepropriété sur forwards. Le animation-fill-modespécifie le style appliqué à un élément lorsque l'animation est terminée. Vous pouvez le définir comme suit:

animation-fill-mode: forwards;

Définissez la animation-fill-modepropriété sur button:hoverpour forwardsque le bouton reste en surbrillance lorsqu'un utilisateur le survole.

Applied Visual Design: créer un mouvement à l'aide d'une animation CSS

Lorsque les éléments ont une valeur spécifiée position, par exemple fixedou relative, les propriétés de décalage CSS right, left, topet bottompeut être utilisé dans les règles d'animation pour créer un mouvement.

Comme illustré dans l'exemple ci-dessous, vous pouvez pousser l'élément vers le bas, puis vers le haut en définissant la toppropriété de l' 50%image clé sur 50px, mais en la définissant sur 0px pour la première image clé ( 0%) et la dernière ( 100%).

@keyframes rainbow {  
  0% {  
    background-color: blue;  
    top: 0px;  
  }  
  50% {  
    background-color: green;  
    top: 50px;  
  }  
  100% {  
    background-color: yellow;  
    top: 0px;  
  }  
}

Ajouter un mouvement horizontal à l' divanimation. A l'aide de la leftpropriété offset, ajoutez à la @keyframesrègle l'arc-en-ciel commence à 0 pixel à 0%, passe à 25 pixels à 50%et se termine à -25 pixels à 100%. Ne remplacez pas la toppropriété dans l'éditeur - l'animation doit avoir un mouvement vertical et horizontal.

Applied Visual Design: Créer une direction visuelle en fondant un élément de gauche à droite

Pour ce défi, vous allez changer l' opacityélément d'un élément animé pour qu'il disparaisse progressivement au fur et à mesure qu'il atteint le côté droit de l'écran.

Dans l'animation affichée, l'élément arrondi avec l'arrière-plan du dégradé se déplace vers la droite de la marque 50% de l'animation selon la @keyframesrègle.

Ciblez l'élément avec l'id de ballet ajoutez la opacitypropriété définie sur 0.1 à 50%, l'élément disparaîtra alors qu'il se déplace vers la droite.

Conception visuelle appliquée: animer des éléments de manière continue à l'aide d'un nombre d'animations infini

Les défis précédents concernaient l'utilisation de certaines propriétés d'animation et de la @keyframesrègle. Une autre propriété d'animation est la animation-iteration-count, qui vous permet de contrôler le nombre de fois que vous souhaitez effectuer une boucle dans l'animation. Voici un exemple:

animation-iteration-count: 3;

Dans ce cas, l'animation s'arrêtera après avoir été exécutée 3 fois, mais il est possible de l'exécuter en continu en définissant cette valeur sur infini.

Pour que la balle continue de rebondir à droite sur une boucle continue, définissez la animation-iteration-countpropriété sur infini.

Applied Visual Design: créer une pulsation CSS à l'aide d'un nombre infini d'animations

Voici un autre exemple d'animation continue avec la animation-iteration-countpropriété qui utilise le cœur que vous avez conçu dans un défi précédent.

L'animation de battement de coeur d'une seconde comprend deux pièces animées. Les heartéléments (y compris les éléments :beforeet :after) sont animés pour changer de taille à l'aide de la transformpropriété, et l'arrière div- plan est animé pour changer sa couleur à l'aide de la backgroundpropriété.

Faites en sorte que votre cœur batte en ajoutant la animation-iteration-countpropriété à la fois pour la backclasse et la heartclasse et en définissant la valeur sur infini. Les sélecteurs heart:beforeet heart:aftern'ont pas besoin de propriétés d'animation.

Conception visuelle appliquée: animer des éléments à des taux variables

Il existe différentes façons de modifier les taux d’animation d’éléments animés de la même manière. Jusqu'à présent, ceci a été réalisé en appliquant une animation-iteration-countpropriété et en fixant des @keyframesrègles.

À titre d’illustration, l’animation de droite consiste en deux "étoiles" dont la taille et l’opacité diminuent toutes les 20% de la @keyframesrègle, ce qui crée l’animation scintillante. Vous pouvez modifier la @keyframesrègle pour l’un des éléments afin que les étoiles scintillent à des rythmes différents.

Modifiez le taux d'animation pour l'élément avec le nom de classe de star-1en modifiant sa @keyframesrègle à 50%.

Conception visuelle appliquée: animer plusieurs éléments à des taux variables

Dans le défi précédent, vous avez modifié les taux d'animation de deux éléments animés de la même manière en modifiant leurs @keyframesrègles. Vous pouvez atteindre le même objectif en manipulant animation-durationplusieurs éléments.

Dans l'animation exécutée dans l'éditeur de code, il y a trois "étoiles" dans le ciel qui scintillent au même rythme sur une boucle continue. Pour les faire scintiller à des vitesses différentes, vous pouvez définir la animation-durationpropriété sur différentes valeurs pour chaque élément.

Définissez les animation-durationéléments avec les classes star-1, star-2et star-3sur 1s, 0.9s et 1.1s, respectivement.

Applied Visual Design: Modifier le minutage d'une animation avec des mots-clés

Dans les animations CSS, la animation-timing-functionpropriété contrôle la rapidité avec laquelle un élément animé change au cours de la durée de l'animation. Si l'animation est une voiture se déplaçant d'un point A à un point B dans un temps donné (votre animation-duration), le animation-timing-functionmessage indique comment la voiture accélère et décélère au cours du trajet.

Un certain nombre de mots-clés prédéfinis sont disponibles pour les options courantes. Par exemple, la valeur par défaut est ease, qui commence lentement, accélère au milieu, puis ralentit à la fin. D'autres options incluent ease-out, qui est rapide au début puis ralentit ease-in, qui est lent au début, puis accélère à la fin, ou linearqui applique une vitesse d'animation constante tout au long.

Pour les éléments avec id of ball1et ball2, ajoutez une animation-timing-functionpropriété à chacun et définissez-les #ball1sur linearet #ball2sur ease-out. Remarquez la différence entre la manière dont les éléments bougent pendant l'animation mais se terminent ensemble, puisqu'ils partagent la même chose, à savoir animation-duration2 secondes.

Applied Visual Design: découvrez le fonctionnement des courbes de Bézier

Le dernier défi a introduit la animation-timing-functionpropriété et quelques mots clés qui modifient la vitesse d’une animation au cours de sa durée. CSS offre une option autre que les mots-clés qui offre un contrôle encore plus fin sur la façon dont l’animation se déroule, grâce à l’utilisation de courbes de Bézier.

Dans les animations CSS, les courbes de Bézier sont utilisées avec la cubic-bezierfonction. La forme de la courbe représente le déroulement de l'animation. La courbe vit sur un système de coordonnées 1 par 1. L'axe des abscisses de ce système de coordonnées est la durée de l'animation (considérez-la comme une échelle de temps), et l'axe des ordonnées correspond au changement de l'animation.

La cubic-bezierfonction se compose de quatre points principaux qui sont assis sur cette grille 1 par 1: p0, p1, p2, et p3. p0et p3sont définis pour vous - ce sont les points de début et de fin qui sont toujours situés respectivement à l’origine (0, 0) et (1, 1). Vous définissez les valeurs x et y pour les deux autres points et leur emplacement dans la grille dicte la forme de la courbe que l'animation doit suivre. Cela se fait en CSS en déclarant les valeurs x et y du p1et des p2points d'ancrage « » sous la forme: (x1, y1, x2, y2). En résumé, voici un exemple de courbe de Bézier en code CSS:

animation-timing-function: cubic-bezier(0.25, 0.25, 0.75, 0.75);

Dans l’exemple ci-dessus, les valeurs x et y sont équivalentes pour chaque point (x1 = 0,25 = y1 et x2 = 0,75 = y2). Si vous vous souvenez de la classe geometry, le résultat est une ligne qui va de l’origine au point (1 , 1). Cette animation est un changement linéaire d'un élément pendant la durée d'une animation et est identique à l'utilisation du linearmot clé. En d'autres termes, cela change à une vitesse constante.

Pour l'élément avec l'id de ball1, remplacez la valeur de la animation-timing-functionpropriété par linearsa cubic-beziervaleur de fonction équivalente . Utilisez les valeurs en points données dans l'exemple ci-dessus.

Applied Visual Design: utiliser une courbe de Bézier pour déplacer un graphique

Un défi précédent portait sur le ease-outmot-clé décrivant un changement d'animation qui accélère d'abord puis ralentit à la fin de l'animation. A droite, la différence entre le ease-outmot clé (pour l'élément bleu) et le linearmot clé (pour l'élément rouge) est illustrée. Des progressions d'animation similaires à celles du ease-outmot clé peuvent être obtenues à l'aide d'une fonction de courbe de Bézier cubique personnalisée.

En général, la modification des points d'ancrage p1et de direction p2entraîne la création de différentes courbes de Bézier, qui contrôlent la progression de l'animation dans le temps. Voici un exemple de courbe de Bézier utilisant des valeurs pour imiter le style de soulagement:

animation-timing-function: cubic-bezier(0, 0, 0.58, 1);

Rappelez-vous que toutes les cubic-bezierfonctions commencent par p0at (0, 0) et se terminent par p3at (1, 1). Dans cet exemple, la courbe se déplace plus rapidement sur l’axe Y (commence à 0, passe à la p1valeur y de 0, puis passe à la p2valeur y de 1) qu’elle n’atteint l’axe des X (0 pour commencer, puis 0 pour p1, jusqu'à 0,58 pour p2). En conséquence, le changement dans l'élément animé progresse plus rapidement que le temps de l'animation pour ce segment. Vers la fin de la courbe, la relation entre la modification des valeurs x et y s'inverse - la valeur y passe de 1 à 1 (pas de changement) et les valeurs x passent de 0,58 à 1, ralentissant ainsi la progression de l'animation par rapport à la durée de l'animation.

Pour voir l’effet de cette courbe de Bézier en action, changez l’ animation-timing-functionélément de ider avec id de reden une cubic-bezierfonction avec x1, y1, x2, y2 valeurs réglées respectivement à 0, 0, 0.58, 1. Cela fera progresser les deux éléments à travers le animation similaire.