

Abonnez-vous à DeepL Pro pour éditer ce document.  
Visitez [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) pour en savoir plus.

Intervenant inconnu 0:00

Hé les gars, c'est moi, vous regardez dev dreamit. Et ceci est le guide complet de CSS Flexbox. Dans cette vidéo, nous allons parcourir l'ensemble du module de mise en page flexible CSS ou CSS Flexbox, d'une manière simple, étape par étape, et bien sûr, nous allons faire tout cela en utilisant des exemples tout au long. Si vous êtes nouveau sur cette chaîne et que vous aimez le contenu, cliquez sur le bouton " J'aime " et abonnez-vous ci-dessous. N'oubliez pas d'activer les notifications en appuyant sur la cloche. Commençons par comprendre ce qu'est CSS Flexbox. CSS Flexbox est un moyen relativement nouveau et amélioré de positionner et d'aligner des éléments dans un conteneur donné. Le but de Flexbox est de simplifier le processus de conception d'une structure de mise en page dynamique ou flexible, d'où le terme de modèle de boîte flexible, sans utiliser des éléments tels que CSS, float ou positionnement. L'objectif de CSS Flexbox est donc de nous faciliter la vie en matière de positionnement et de mise en page. Et comme vous le verrez, c'est un module vraiment, vraiment puissant. Avant l'arrivée de Flexbox, il existait quatre méthodes de base pour disposer les éléments : la mise en page en bloc, qui s'étendait sur toute la largeur de la page, et les éléments flottants. Donc, par exemple, vous pouvez avoir un texte et une image. Et vous pouvez faire flotter le texte vers la droite, afin d'avoir le texte et l'image côte à côte, nous avons également la position pour le positionnement spécifique des éléments. Et enfin, pour revenir en arrière, nous avions le tableau. Donc, littéralement, vous créez un tableau avec lui-même, puis vous le remplissez avec vos éléments. Ok, alors maintenant regardons Flexbox. Ici, sur ce conteneur, nous avons un exemple où nous utilisons Flexbox sur un conteneur avec display flex. Et puis nous écrivons simplement le contenu dans ce conteneur en utilisant l'espace de manière égale. Et comme vous pouvez le voir, avec deux simples lignes, nous avons contrôlé nos éléments en les espaçant de manière égale, grâce à la puissance de CSS Flexbox. Ainsi, chaque fois que vous souhaitez contrôler des éléments disposés en ligne, comme c'est le cas ici, ou même en colonne, et que vous voulez contrôler l'espacement entre ces éléments et la façon dont ils utilisent l'espace lorsque le conteneur grandit ou rétrécit. Ce sont les utilisations pour lesquelles Flexbox a été conçu. Remarquez maintenant que nous avons un conteneur. Et puis nous avons des éléments. css Flexbox a des propriétés de conteneur qui sont appliquées au conteneur réel. Et nous en avons utilisé deux ici pour afficher et justifier le contenu. Ce sont des exemples de propriétés de conteneur. Flexbox possède également des propriétés d'élément qui s'appliquent aux éléments du conteneur. Voyons de quoi il s'agit. Voici donc toutes les propriétés Flexbox. Sur le côté gauche, nous avons les propriétés du conteneur. Et à droite, nous avons les propriétés des éléments. Les propriétés du conteneur sont des éléments tels que l'affichage, la direction de la flexion, l'enveloppement de la flexion, le flux de la flexion, le contenu juste, les éléments de ligne et le contenu aligné. Et pour les propriétés des éléments, nous avons order, flex, grow, flex, shrink, flex basis, flex et align self. Et bien sûr, nous allons passer en revue chacune de ces propriétés en détail. La dernière chose que nous devons comprendre avant de nous plonger dans le code est l'axe principal et l'axe transversal de Flexbox. La mise en page Flexbox est basée sur les directions du flux Flex. Pour comprendre cela, regardons le diagramme suivant. Disons que nous avons notre conteneur Flex, car nous avons défini l'affichage de ce conteneur sur Flex, et automatiquement tous les éléments de ce conteneur deviendront des éléments Flex. Donc, à l'intérieur de ce conteneur, nous avons des éléments que nous avons configurés pour qu'ils soient disposés en ligne. Nous avons donc réglé la direction de flexion sur rouler sur ce conteneur. C'est pourquoi les éléments flexibles sont disposés côte à côte. Donc, parce que la direction de flexion est le roulement, notre axe principal ici sera horizontal en travers comme ceci. Et notre axe transversal sera vertical et ira dans la direction opposée. Si nous tournons notre direction de flexion vers la colonne, et que ces éléments sont empilés les uns sur les autres, alors notre axe principal sera vertical, car c'est la direction dans laquelle nos éléments circulent. Et l'axe transversal sera à l'opposé de cette direction. Dans ce scénario, l'axe transversal sera horizontal. L'axe transversal est donc déterminé en fonction de l'axe principal, quel qu'il soit, et l'axe principal est déterminé par ce que nous avons défini comme direction de flexion. Donc, pour résumer, l'axe principal est la direction principale, les éléments de flexion de la langue sont disposés. Et encore une fois, cela dépend de la propriété de direction de flexion, si elle est définie sur ligne ou colonne, et ensuite l'axe transversal est toujours perpendiculaire à l'axe principal. Maintenant, quel est l'intérêt de connaître les axes principaux et l'axe transversal d'un conteneur flexible ? Eh bien, le fait de les connaître sera utile lorsqu'il s'agira d'utiliser certaines des propriétés du conteneur flexible. En parlant de conteneur flexible, plongeons maintenant dans le code et expliquons chaque propriété du conteneur en détail. Nous ferons ensuite de même pour les propriétés des éléments. Et enfin, nous examinerons également quelques exemples. Et encore une fois, je laisserai des timestamps dans la boîte de description ci-dessous pour chaque section. Ok, alors faisons ça. Ok, donc nous y sommes. Alors dans notre éditeur de texte. Dans le HTML, nous avons ce div avec un ID de container, qui est ce container violet ici. Et à l'intérieur de celui-ci, nous avons quatre divs, chacun avec une classe de item et aussi une classe de 123 et quatre respectivement. Et ce sont ces éléments verts ici. Jetons un coup d'oeil à notre CSS. Bon, il s'agit simplement d'un conteneur de sonde de style. Il a une largeur de 1000 pixels et une hauteur de 400 pixels. Ok, allons-y et transformons ce conteneur en conteneur flexible en définissant l'affichage du conteneur sur flex. Il s'agit maintenant d'un conteneur flexible et tous les éléments du conteneur deviennent automatiquement des éléments flexibles.

Intervenant inconnu 4:57

Maintenant, remarquez que dès que vous définissez le conteneur en flex, nos éléments sont disposés en ligne. Et c'est parce que par défaut, la propriété flex direction d'un conteneur flex, flex direction est définie sur row. Rappelez-vous, flex direction établit l'axe principal de notre conteneur. Donc, par défaut, l'axe principal est horizontal, comme ceci. Nous pouvons changer cela en définissant la propriété flex direction sur column. Ok, donc maintenant les éléments sont empilés les uns sur les autres comme ceci, nous pouvons faire l'inversion de colonne pour inverser les éléments. Donc c'est 4321, au lieu de 1234. Et bien sûr, nous pouvons faire la même chose pour le rouleau. Ok, remettons ça en ligne. Et techniquement, je n'ai pas besoin de déclarer une direction flex pour roll car Remember, c'est la valeur par défaut d'un conteneur flex. Mais pour les besoins de ce tutoriel, nous allons la conserver. D'accord, la prochaine propriété du conteneur flexible que nous allons examiner est l'habillage flexible. Vous pouvez probablement deviner ce qu'elle fait d'après son nom. L'enroulement flexible permet aux éléments de s'enrouler sur une nouvelle ligne. Par défaut, la propriété flex wrap est définie sur no wrap, de sorte que les éléments flex ne s'enroulent pas sur une nouvelle ligne. Donc, si nous ajoutons quelques éléments supplémentaires ici, disons que cela va vers le HTML et ajoutons quelques éléments supplémentaires, nous avons 12. Comme vous pouvez le voir, les éléments continueront à être disposés sur la même ligne. Maintenant, ce n'est pas infini. Vous ne pouvez pas avoir 100 éléments ici et vous attendre à ce qu'ils rentrent tous dans ce conteneur, ils vont déborder à un moment donné. Donc gardez ça en tête. Mais pour les autres éléments, allons encore une fois à notre propriété flex wrap, et voyons ce que nous pouvons faire. Nous pouvons régler cette propriété sur wrap. Et maintenant, comme vous pouvez le voir, les éléments s'enroulent sur une nouvelle ligne, nous pouvons également dire wrap, reverse. Et ce que cela fait, c'est qu'au lieu de rapporter de haut en bas, on éructe de bas en haut, donc on a 123456 et ensuite les éléments s'enroulent sur le haut, comme ça. Donc c'est la propriété de l'itinéraire fixe. Ok, alors demandons à ce qu'on l'enroule et qu'on se débarrasse aussi de ces éléments. Ok. La propriété suivante est le flux de flexion, le flux de flexion n'introduit rien de nouveau, parce que le flux de flexion est simplement une propriété raccourcie pour la direction de flexion et l'enveloppement de flexion. Donc avec le flux de flexion, vous pouvez déclarer ces deux directions de flexion et enveloppes de flexion sur une seule ligne. Notre ligne sera donc la suivante. Et donc c'est exactement la même chose que de dire ceci. Nous pouvons donc, si nous le souhaitons, supprimer ces deux éléments maintenant parce que nous avons déclaré flexflow à la place, mais encore une fois, pour les besoins de ce tutoriel, évidemment l'amour aussi. Maintenant, la prochaine propriété du conteneur Flexbox est le contenu justifié. Et bien sûr, vous devez vous souvenir de celle-ci dans mon exemple précédent. Lorsque vous pensez au contenu justifié, je veux que vous pensiez à votre axe principal. Car la propriété justify content définit la façon dont l'espace le long de l'axe principal est utilisé. Par défaut, il est défini sur flex start, ce qui explique pourquoi les éléments sont regroupés au début ici. Nous pouvons le changer en flex end. Ok, cela les place à la fin. Si vous voulez centrer les éléments, il suffit de dire center. Vous commencez à voir la puissance et la facilité d'utilisation de Flexbox. Encore une fois, si vous trouvez cette vidéo utile, faites-le moi savoir dans la section commentaire ci-dessous. Et bien sûr, n'oubliez pas de liker et de vous abonner.

Intervenant inconnu 7:55

Ok, alors qu'est-ce qu'on peut faire d'autre, on n'a qu'à cliquer sur, disons qu'on ne veut pas que nos articles soient tous regroupés au centre ici.

Conférencier inconnu 8:00

Mais nous voulons utiliser l'espace que nous avons autour de nos articles ici en mettant un peu d'espace entre nos articles. Pour cela, nous disons simplement espace entre et ils peuvent voir que nous avons de l'espace entre nos articles et ils diront que nous ne voulions pas seulement de l'espace entre les articles. Nous voulions aussi de l'espace autour des articles. Eh bien, vous l'avez deviné, on change ça en espace autour. Et maintenant vous pouvez voir que nous avons de l'espace non seulement entre nos éléments, mais aussi autour de nos éléments. Si nous regardons attentivement ici, faisons un zoom, vous pouvez voir que l'espace entre chaque élément est en fait plus grand que l'espace à l'extérieur des éléments. Ok, c'est la même chose ici aussi. Disons que nous voulions que ces éléments soient espacés de manière égale, et bien super change pour espacer de manière égale. Et maintenant vous pouvez voir que l'espacement entre nos éléments est exactement le même que l'espacement autour de nos éléments. Gardez à l'esprit qu'il n'y a pas de rembourrage placé sur ce conteneur. Ok, donc si nous ajoutons un rembourrage de disons 20 pixels. Vous pouvez voir que, même si nous avons réglé le contenu sur un espacement régulier, parce que nous avons maintenant un remplissage sur notre conteneur, vous pouvez voir ici que l'espacement autour de nos éléments semble être plus important que l'espacement entre eux. Et c'est parce qu'il faut garder à l'esprit que nous avons également un padding sur ce conteneur. D'accord, passons simplement à ce rembourrage pour l'instant. Ok, la prochaine propriété du conteneur Flexbox que nous allons examiner est align items, pensez aux line items comme la version du contenu justifié de l'axe transversal. Actuellement, notre axe principal est défini sur l'axe horizontal, car notre direction de flexion est une ligne. Et donc l'axe transversal va être vertical en descendant. Donc aligner les éléments est une question d'espacement de nos éléments le long de cet axe transversal. Actuellement, il est donc en position de début de flexion. Cependant, la valeur par défaut pour les éléments de ligne est en fait stretch. Et la raison pour laquelle ils ne s'étirent pas complètement vers le bas est que nous avons spécifié la hauteur de nos éléments. Ok, hauteur h pixels. Si nous enlevons ceci, vous pouvez voir que les éléments s'étirent vers le bas, enlevez ceci et voyez ce que nous pouvons faire d'autre. Nous pouvons également spécifier la fin de flexion, en allant d'autres personnes à la fin de l'axe transversal, nous pouvons dire centre et maintenant nos éléments l'ont envoyé le long de l'axe transversal. Et pour la dernière valeur d'éléments de ligne ici, disons que nous avons eu un élément avec un texte plus grand de sorte que les éléments ne sont plus alignés en fonction de leur ligne de base. Donc, actuellement, si nous regardons cette liste, en supposant que nous pouvons voir que les éléments sont tous alignés, donc le sous-sol de la 123 et les quatre, allons simplement à HTML. Et il a changé le quatrième élément ici en un h1 à la place. Ok, donc maintenant vous pouvez voir que les éléments un, deux et trois sont alignés sur la ligne de base. Mais le numéro 4 ne l'est pas. Voyons maintenant ce que nous pouvons faire. Avec des articles de ligne. Nous pouvons dire, une ligne de base des articles de ligne, et maintenant chacun des articles est aligné selon leur ligne de base. Ainsi, la valeur de la ligne de base alignera les éléments flexibles sur la ligne de base de leur contenu. Ok, alors débarrassons-nous de ce h1 pour notre normal pour dire ça. Et appuyons sur aligner les éléments pour aller vers le centre. C'est donc la propriété Aligner les éléments. Et cela nous amène à la dernière propriété du conteneur Fitz, qui est align content. L'alignement du contenu n'a aucun effet lorsqu'il n'y a qu'une seule ligne d'éléments, comme c'est le cas ici. Donc, allons-y et ajoutons quelques éléments supplémentaires à notre conteneur. Ok, il y en a maintenant 12. Et vous pouvez voir qu'ils s'étendent sur deux lignes. Voyons maintenant ce que fait la propriété line content. D'accord, puisque nous savons que plus d'éléments sont enveloppés sur une nouvelle ligne, nous devons maintenant réfléchir à la façon dont nous distribuons l'espace non seulement entre ces éléments, mais aussi autour d'eux. Or, ces éléments, et ces éléments, constituent collectivement notre contenu, d'où le terme de contenu allié. Donc disons que nous voulons que notre contenu soit poussé vers le début, nous disons simplement, vous l'avez deviné, comme début et sortons que notre contenu est maintenant au début.

Intervenant inconnu 11:35

Et bien sûr, nous pouvons spécifier la fin de la flexion, ainsi que le centre. l'espace autour de l'espace entre l'espace entre les éléments ont été autour. Et enfin space evenly aussi. Pour qu'il y ait une quantité égale d'espace

Intervenant inconnu 11:54

entre les éléments, mais aussi autour d'eux. Voilà donc toutes les propriétés du conteneur flexible. Nous n'avons même pas encore touché aux éléments et regardez à quel point c'était puissant et facile de disposer notre contenu exactement comme nous le voulions, surtout avec les valeurs de propriétés sémantiques simples, comme centre ou espace autour de l'espace uniformément, vraiment, vraiment puissant. Ok, alors passons à l'examen des propriétés des éléments flex. Et puis nous allons regarder deux exemples en utilisant ce que nous avons appris, nous allons construire un jeu appelé en utilisant Flexbox. Et nous allons créer un menu de navigation avec Flexbox également. Ok, donc nous avons réinitialisé notre code. Et voici à quoi ressemblent notre conteneur et nos éléments, à peu près la même configuration que précédemment, où nous avions un conteneur pour les éléments, et chaque élément a une classe de 123 ou quatre. Ok, la première propriété d'élément Flexbox que nous allons examiner est l'ordre. Maintenant, par défaut, nos éléments sont disposés selon l'ordre de la source. Donc, nous avons 123, et quatre, parce que innerHTML. C'est ce que nous avons spécifié 123 ou quatre, mais la propriété order nous permet de contrôler et de changer l'ordre des éléments. Par défaut, la propriété order est fixée à zéro. Donc, ici, nous sélectionnons chaque élément à 123, et quatre, et nous définissons l'ordre à zéro sur tous ces éléments. Mais si nous allons de l'avant et changeons ceux-ci, changeons le 1212 en 10 325. Et pour le numéro 4, allons-y pour le gros 100. Et sans conteneur particulier, l'ordre est changé, nous avons 132 et quatre. Et la façon dont cela fonctionne est que ce qui a l'ordre le plus élevé est poussé plus loin vers la fin, quand il a le nombre le plus bas est au début. Donc très bien, le Mach numéro 1 a un ordre de 1. Donc il est au début, le second ordre le plus élevé est le numéro 5. Ok, le numéro trois, c'est pourquoi le numéro trois vient ensuite, puis c'est le numéro deux avec un ordre de 10, c'est pourquoi le numéro deux est en troisième position. Et enfin, l'élément numéro quatre a un ordre de 100, c'est pourquoi il est à la fin là. Ils peuvent également spécifier des nombres négatifs. Donc si vous avez donné à ipv4 un ordre de moins 100, il est maintenant inférieur à un, c'est pourquoi pour l'instant parce que le premier élément, il est également important de noter que ces nombres sont relatifs les uns aux autres. Donc, mettons tout ça à 1. Ainsi, si nous fixons 123 et 4 à 10, vous verrez qu'ils restent en position 123 et 4, car les cotes sont relatives les unes aux autres. D'accord, c'est la propriété de l'ordre. Allons-y et remettons-les à zéro. La prochaine propriété que nous allons examiner est la propriété flex grow. Avec cette propriété, nous pouvons indiquer à nos éléments comment ils doivent grandir pour utiliser l'espace restant. Donc regardons notre conteneur. Nous avons des éléments, et puis nous avons cet espace supplémentaire ici. Donc flex grow est tout au sujet de la façon dont notre élément devrait croître pour utiliser cet espace supplémentaire. Donc, par exemple, si la dépense ici était de 400 pixels, comment nos éléments devraient se développer pour utiliser cet espace supplémentaire de 400 pixels. Par défaut, la valeur de flex grow est de zéro, ce qui signifie qu'il n'y a pas de croissance du tout. Donc allons-y et ajoutons flexco zéro à nos articles. Ok, donc le walker flickks, groves zéro. Et comme prévu, nos objets ne grandissent pas du tout pour utiliser cet espace supplémentaire. Allons-y et mettons-les tous à un et vous pouvez déjà voir que quelque chose commence à se produire sur notre conteneur. Maintenant, quand nous disons flex quatre à un, cela signifie essentiellement que chaque élément doit croître de manière égale, de sorte que l'espace supplémentaire est distribué uniformément à chacun d'eux. Et donc si nous regardons, nos éléments ont maintenant grandi de manière égale pour utiliser l'espace restant de sorte qu'ils remplissent maintenant le conteneur. En d'autres termes, ils ont tous grandi à un taux égal de un. Ils ont donc tous la même taille. Si vous dites que l'un de ces deux, deux, et il fait juste le nombre deux ici, alors faisons grandir

Intervenant inconnu 15:11

à Ok, donc maintenant les éléments un à quatre ont un flex, grow un, et l'élément deux a un flex grow de deux. Et comme vous pouvez le voir, l'élément numéro deux est visiblement plus grand que les éléments un, trois et quatre.

Intervenant inconnu 15:21

Cela signifie que l'élément deux occupera deux fois plus d'espace que les éléments un, trois et quatre.

Intervenant inconnu 15:29

C'est très important de le savoir. Mais cela ne veut pas dire que l'article 2 aura une largeur double de celle des articles 1, 3 et 4. Vous pouvez voir que bien que l'article 2 soit plus grand que les articles 1, 3 et 4, il n'a pas une taille double. Et donc ça peut être un peu déroutant, mais il doit se souvenir de l'espace restant que nous avons pour les articles un, trois et quatre, parce qu'everflex grow one doit avoir un espace égal pour chacun. Donc la quantité d'espace qu'ils obtiennent devrait être la même, mais l'élément deux devrait avoir deux fois plus d'espace. Ok, donc c'est la propriété de flex grow. OK, allons-y et remettons-les à zéro, pour que nos éléments ne grandissent pas du tout. Et maintenant, la prochaine propriété que nous allons examiner est flex shrink, et flex shrink, comme vous pouvez probablement le deviner par le nom, indique à nos éléments à quel rythme ils doivent se rétrécir. Donc pour cet exemple, allons-y et ajoutons quelques éléments supplémentaires dans notre conteneur. Ok, donc maintenant nous avons 12 articles en tout. Et comme vous pouvez le voir, nos éléments ont rétréci pour tenir dans ce conteneur. Maintenant, la valeur par défaut de flex shrink est de 1. Alors allons-y et ajoutons-la à nos éléments. Ok, donc maintenant les articles 123, et quatre, etc... reflètent de un, nous ne prendrons pas la peine de m'ajouter aux articles cinq à 12, car par défaut, ils affectent tous un rétrécissement de un de toute façon. D'accord, le rétrécissement flexible avec une valeur de un signifie essentiellement que tous les éléments se rétrécissent à un taux égal, de sorte que s'ils doivent un jour se rétrécir, comme c'est le cas dans cet exemple, ils conservent tous la même taille, comme vous pouvez le voir, les éléments un à 12 ont tous exactement la même taille. C'est donc la vitesse à laquelle l'article se rétracte qui est à l'origine de la rétraction flexible. Donc disons que nous voulons que l'article numéro trois se rétrécisse plus rapidement que le reste, eh bien, changeons juste le flex shrink de l'article numéro trois à deux. Et comme vous pouvez le voir, l'article trois est maintenant plus petit que le reste pour vous montrer bêtement. Ils peuvent aussi dire, "En fait, je ne veux pas, je ne me souviens pas du tout d'un rétrécissement". Eh bien, nous avons simplement dit flex shrink sur lui, nous voulons zéro, ils peuvent le voir, l'un conserve sa largeur, et tous les autres éléments rétrécissent. Ok, donc c'est la propriété flex shrink. Ok, alors passons à flex shrink à zéro sur tous les éléments. Nous ne voulons pas qu'ils rétrécissent du tout. Et allons-y et supprimons nos autres éléments ici. Nous sommes donc de retour au formulaire. Ok, la prochaine propriété d'élément flexible que nous allons apprendre est la base de flexion. Pensez à la base flexible comme la taille de nos éléments. Si nous n'avons pas spécifié d'espaces publics, comme nous ne l'avons pas fait ici, alors la taille des éléments se réduit à la largeur. Et si aucune largeur n'est spécifiée, alors la taille de l'élément est ramenée au contenu de l'élément. Alors donnons un sens à tout cela. Dans notre exemple, nous n'avons pas spécifié d'espaces flexibles, mais nous avons spécifié une largeur. Allons donc dans notre classe d'éléments. Ok, vous pouvez voir que nous avons une largeur de 150 pixels, donc chacun de ces éléments a une largeur de 150 pixels. Mais si nous n'avions pas spécifié de largeur, la largeur des éléments serait en fait la taille du contenu. Donc maintenant, nous allons remettre la largeur. Et spécifions sur tous nos éléments une base flexible. C'est une base flexible. Et prenons 200 pixels. Nos articles utilisent maintenant une base flexible. Ils sont donc tous de 200 pixels, contrairement à la largeur. On peut donc s'en débarrasser. Nous n'en avons plus besoin car nos éléments sont dimensionnés sur cette base flexible que nous avons définie ici. Maintenant, nous devons savoir que les espaces flexibles sont limités par la largeur maximale et la largeur minimale. En d'autres termes, si nous donnons à nos éléments une largeur maximale de, disons, 50 pixels, vous pouvez voir que chaque élément a maintenant une largeur maximale de 50 pixels. Donc, ce qui se passe ici, c'est que, bien que les espaces flexibles soient de 200 pixels avant que les éléments puissent atteindre 200 pixels, ils s'arrêtent à 50 pixels parce que nous avons une largeur maximale de 50 pixels. Donc la base du flex est limitée par la largeur maximale. Elle est également limitée par la largeur minimale. Alors débarrassons-nous de cette largeur minimale. Fixons-la à 500 pixels. Et donc maintenant chaque élément a une largeur de 500 pixels. Veuillez noter que la largeur maximale et la largeur minimale remplacent les espaces flexibles. Quel est l'intérêt des espaces flexibles ? Eh bien, lorsque nous avons un débordement, comme ici, la base flexible vient à la rescousse. Parce que si nous définissons la base de flexion à 500 pixels, et enlevons ceci, et définissons également un rétrécissement de flexion à un. Vous pouvez voir maintenant que nous n'avons pas le même problème. Mais les éléments ne pouvaient pas grandir à 500 pixels parce qu'il n'y avait pas assez d'espace. Ils sont donc rétrécis pour s'adapter. Si le conteneur ou la fenêtre d'affichage s'agrandit, les éléments s'agrandissent également jusqu'à 500 pixels. La base flexible est donc utile car elle ne fait pas déborder nos éléments. D'accord, c'est la propriété flex basis. Ensuite, nous avons la propriété flex. Maintenant, flex est la propriété raccourcie pour flex grow flex shrink espaces publics. Alors allons-y et commentons tous ces éléments ici. Et déplaçons également les espaces flex. Et maintenant nous pouvons déclarer flex grow flex shrink des espaces flex sur une seule ligne en disant flex, l'ordre est grow.

Intervenant inconnu 20:23

Et ensuite, c'est le rétrécissement. Et enfin, c'est comme les espaces. Et par défaut, c'est réglé sur 01, et auto. Donc disons que nous voulons que nos articles grandissent à un taux de 1, nous ne voulons pas qu'ils rétrécissent. Donc, mettez-le à zéro, et avec chaque élément de 200 pixels, ok, obtenez exactement la même chose ici. Mais nous avons tout déclaré sur une seule ligne en utilisant flex. Et cela nous amène à notre dernière propriété d'élément flex, qui est alignée auto alignée auto nous permet de spécifier n'importe quel élément unique et de l'aligner. Donc par exemple, allons-y et ciblons l'élément numéro trois. Et c'est la ligne self two en tant que goflex. Et comme vous pouvez le voir, nous avons ciblé uniquement l'élément trois, et l'élément trois est aligné à la fin. Donc, c'est la ligne de la propriété elle-même. Et avec ça, les gars, nous avons passé en revue toutes les propriétés de conteneur flex et toutes les propriétés d'élément flex. Utilisons maintenant ce que nous avons appris pour construire deux petites choses. Commençons par la carte de jeu. Ok, donc première chose, utilisons Flexbox. pour créer ce petit code de jeu. Dans notre HTML, nous avons ce div avec un ID de conteneur d'image, qui a cette section supérieure ici. Et puis nous avons un div avec une classe de info, qui est cette section de texte inférieure ici. Supprimons maintenant nos styles et reconstruisons le tout. Ok, donc tout d'abord, travaillons sur ce conteneur d'image supérieur. Voici à quoi nous voulons ressembler. Et c'est ce que nous avons. Donc, tout d'abord, nous allons nous concentrer sur les styles non flexibles. Ok, donc nous avons donné un arrière-plan de ce gradient linéaire, une largeur et une hauteur de 100 pixels, une marge de zéro auto-impression au milieu, et un rayon de bordure de huit pixels en haut à gauche. Et à droite. Avec ce premier exemple, je voulais vous montrer comment centrer un élément avec Flexbox. Donc, d'abord, nous allons donner à notre conteneur d'image un affichage de flex. Ok, donc maintenant ce conteneur est un conteneur flex, et cette image devient un élément flex. Et nous voulons déplacer cet élément flex vers le centre. Rappelez-vous, par défaut, la direction du flex est le rouleau. Et donc l'axe principal est en travers comme ceci. Donc si nous voulons déplacer l'élément au centre, en travers lorsque la cible justifie le contenu, et que nous disons centre, alors il est maintenant au milieu. Et nous voulons aussi aligner verticalement, ce qui signifie que nous ciblons l'axe transversal. Et pour cibler cela, nous disons align items, center. Et maintenant notre élément flex est parfaitement centré à l'intérieur de notre conteneur flex. Il n'y a pas de travail sur la section du bas ici. Donc rappelez-vous, c'est ce à quoi nous voulions ressembler. Une fois encore, allons-y et ajoutons nos styles non flexibles. Ok, donc ici nous lui avons donné une largeur de 400 pixels pour correspondre à la largeur de notre conteneur. Nous avons donné une hauteur de 100 pixels à ce gradient noir. Et nous avons également spécifié un rayon de bordure de huit pixels pour le bas à droite et à gauche, le premier de tous les textes noirs sur un fond noir ne fonctionne pas. Donc allons-y et définissons un texte blanc. Et allons-y et donnons à cette boîte de saisie un affichage de flex. Et je vois que chacun de nos éléments est disposé en ligne. Mais nous voulons qu'ils s'empilent les uns sur les autres. Nous allons donc définir la direction de flexion sur colonne, et je vois que les éléments s'empilent les uns sur les autres. Je veux aussi qu'ils espacent les éléments de la meule. Donc je vais dire justifier le contenu, espacer autour. Et rappelez-vous que l'axe principal est maintenant vertical parce que nous avons réglé la direction de flexion sur colonne, c'est pourquoi le contenu justifié espace nos images verticalement. Et enfin, pour aligner le tout horizontalement au centre. Nous ciblons align items, center. Et maintenant vous voyez que nous avons notre carte de jeu en utilisant Flexbox. Ok, alors maintenant regardons la création d'un menu de navigation avec Flexbox. Voici ce que nous voulons créer. On a notre logo sur la gauche. Et à droite, nous avons nos liens, bien espacés. Dans le HTML, nous avons un en-tête. Cet en-tête va être notre conteneur flexible. À l'intérieur, nous avons un div avec l'ID de logo, qui contient l'image et une balise F avec nos liens. Ces deux éléments seront donc les éléments flexibles. Cela peut être le conteneur flexible. Allons-y et jetons un coup d'oeil au CSS. Et voyons à quoi cela ressemble sans le style Flexbox. Ok, donc voilà ce que j'ai. Avant de commencer, vous remarquerez peut-être qu'il y a déjà un peu d'espace sur le côté. Et c'est parce que cet en-tête a un peu de rembourrage appliqué à elle. Si vous l'enlevez, vous verrez que nos éléments sont maintenant alignés sur les côtés. C'est la raison pour laquelle nous avons déjà un peu d'espace ici. Ok, donc la première chose que nous voulons faire est de transformer ce conteneur en un conteneur flexible en réglant l'affichage sur flex. Définissons notre direction de flexion et notre enveloppement de flexion en utilisant la propriété raccourcie de flex flow. D'accord, nous allons donc opter pour le flex flow rolling ou wrap et je veux un peu d'espace entre les éléments retirés, nous voulons ceci. D'accord, nous avons donc l'argent, le côté et encore beaucoup d'espace entre les éléments. Maintenant j'ai de l'espace entre nos articles. Et enfin, on va centrer ces éléments sur l'axe transversal. Nous allons donc dire "éléments de ligne, centrez la latence", centré sur l'axe transversal. Ok, passons maintenant à nos éléments.

Intervenant inconnu 25:19

Et la première chose que nous voulons faire est d'appliquer un contour à chacun d'entre eux afin que nous puissions voir avec quoi nous travaillons. Ok, donc on applique un gris uni d'un pixel sur le logo, et la même chose pour les éléments de navigation. Et je veux que ce logo occupe 40% de notre en-tête. Je vais donc définir la base flexible du logo à 40%. Et pour les éléments de navigation, je veux que ce soit 60 %. Voilà ce que nous avons. Maintenant, cette balise nav avec les liens à l'intérieur, je vais la régler sur un affichage de flex également. Et enfin, vous pouvez voir que nous avons des liens, mais aussi qu'il y a un peu d'espacement entre eux.

Conférencier inconnu 25:50

en utilisant, nous avons cet espace libre supplémentaire ici. Nous voulons que nos articles grandissent pour qu'ils utilisent cet espace.

Intervenant inconnu 25:56

Nous allons donc simplement dire flex grow one, et ils peuvent voir qu'ils grandissent tous pour utiliser cet espace de manière égale. Enfin, allons-y et supprimons nos contours. Et voilà, c'est ce que nous voulions. Et c'est ce que nous avons. Ok les gars, voilà comment utiliser Flexbox pour construire un menu de navigation simple, encore une fois les gars, si vous avez trouvé cette vidéo utile, s'il vous plaît, écrasez le bouton "j'aime" en bas, appuyez sur le bouton "s'abonner" aussi. N'oubliez pas de cliquer sur la cloche de notification pour être informé dès qu'un nouveau contenu est publié. Faites-moi savoir dans la section des commentaires ci-dessous comment vous comptez utiliser Flexbox et aussi ce que vous avez aimé dans cette vidéo et comment elle vous a aidé. Ok, les gars. C'est tout pour cette leçon. Continuez à vous entraîner et je vous verrai au prochain cours.

Transcrit par https://otter.ai