Font

Ales dans google fonte. Rechercher Lato et choisir

Light 300. Et l’importez dans css.

<style>

@import url('https://fonts.googleapis.com/css2?family=Inconsolata:wght@300&family=Lato:wght@300&display=swap');

</style>

On va effectuer le html el section par section.

<div class="navbar">

      <div class="container">

        <h1 class="logo">Loruki.</h1>

         <nav>

            <ul>

                <li> <a href="index.html">Home</a></li>

                <li> <a href="features.html"> Features</a></li>

                <li> <a href="docs.html"> Docks</a></li>

            </ul>

         </nav>

      </div>

    </div>

\*{

    box-sizing: border-box;

    padding: 0;

    margin: 0;

}

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

body{

    font-family: 'Lato' ,sans-serif;

    color: #333;

    line-height: 1.6;

}

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

ul{

    list-style: none;

}

a{

    text-decoration: none;

    color: #333;

}

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement

h1,h2{

   font-weight: 300;

   line-height: 1.2;

   margin:10px 0;

}

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement

p{

    margin: 10px 0 ;

}

img{

    width: 100%;

}

.navbar{

    background-color:#047aed ;

    color: #fff;

}



Ici la couleur des link ne change pas car on a mis en haut la couleur des <a> a en noir

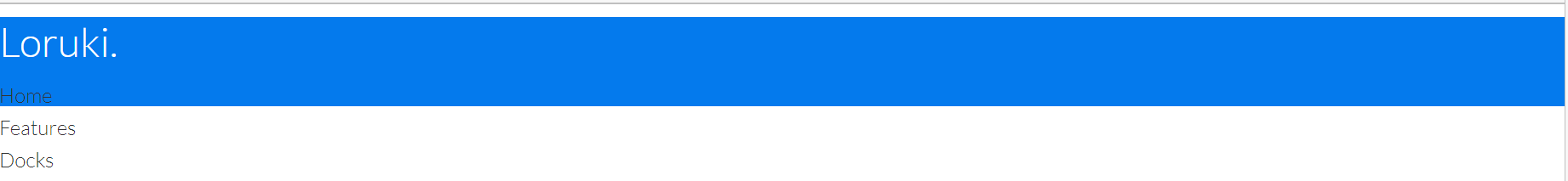
.navbar{

    background-color:#047aed ;

    color: #fff;

    height: 70px;

}



.container{

    max-width: 1100px;

}



.container{

    max-width: 1100px;

    margin:  0 auto;

}

On a créé un contenir et lui donné une mage auto à droite etgayuche.

Une image contenant texte, capture d’écran, Bleu électrique, Police

Description générée automatiquement

Le header avait un padding de 10 px. On va le centrer

.container{

    max-width: 1100px;

    margin:  0 auto;

    overflow: auto;

}

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

.container{

    max-width: 1100px;

    margin:  0 auto;

    overflow: auto;

    padding: 0 48px;

}

Une image contenant texte, capture d’écran, Bleu électrique, Police

Description générée automatiquement

Remarque avec padding : espace dans le contenu.

Padding : valeu1 px valeur2 px = hauts bats et droite et gauche

Padding : valeu1 px valeur2 px = valeur1 pour hauts bats et valeur2 droite et gauche

Padding : valeu1 px valeur2 px valaleur3 px = valeur1 pour hauts , valeur2 droite et gauche, valeur3 bas.

Flexbox :

Maintenant on va considérer le container comme un flexbox. Et ses deux éléments <h1> logo et <nav> sont ces deux flex-items. Dans le div qui conteient le container on va ajouter une class=flex.

 <div class="container flex" >

.flex{

    display: flex;

}

Avec ca tous les élement fils sont aligné sur l’axe paricipal.

Une image contenant texte, Police, Bleu électrique, bleu vert

Description générée automatiquement

On veut placer le logo à gauche et les menus séparément à droite. On uilise la valeur justify content :

Qui détermine comment les items sont répartis entre eux sur l’axe principal qui est ici l’horizontale.

Exple :

.flex{

    display: flex;

    justify-content: center;

}

Une image contenant texte, capture d’écran, Bleu électrique, Police

Description générée automatiquement

On voit que les items sont centré sur l’axe horizontal.

.flex{

    display: flex;

    align-items: center;

}

Align-items aligne chaque item au sein de son conteneur sur l’axe secondaire.

Donc logo et le ul sont acentré dans leur conteneur sur l’axe secondaire.



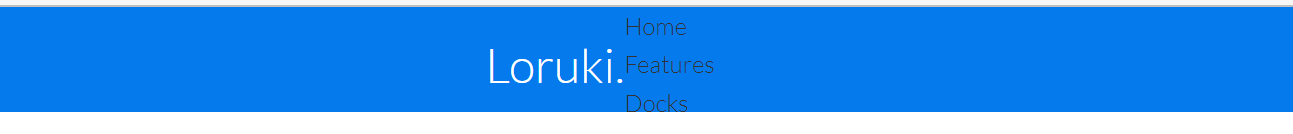
.flex{

    display: flex;

    justify-content: center;

    align-items: center;

}



.flex{

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 100%;

}



C’est la même chose de mettre justify-content dans le .navbar .flex du css

.navbar .flex{

    justify-content: space-between;

}

Maintenant on veut séparer les liens Home, Features et Docs et les disposer horizontalement.

Pour cela on va les appliquer à des propriété de fkexbox.

.navbar ul{

    display: flex;

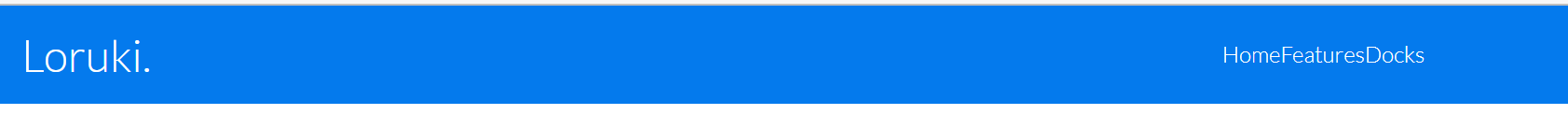
}



.navbar a{

    color: #fff;

}

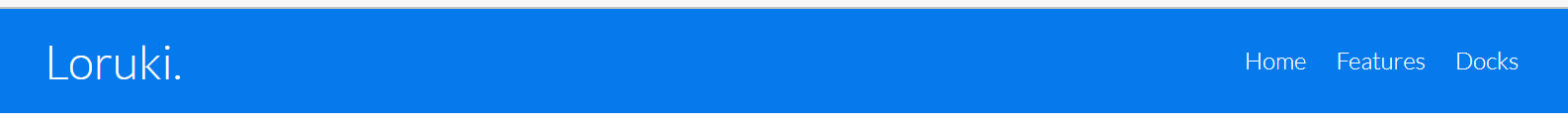


.navbar a{

    color: #fff;

    padding: 10px;

}



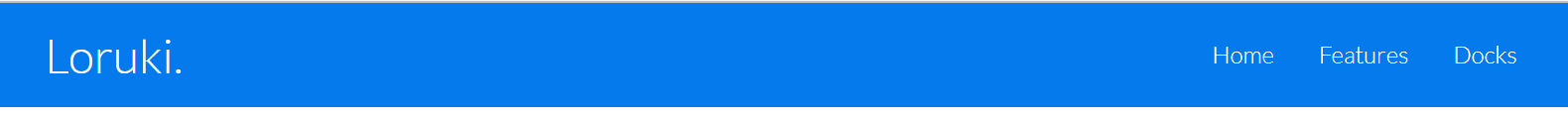
.navbar a{

    color: #fff;

    padding: 10px;

    margin: 10px 5px;

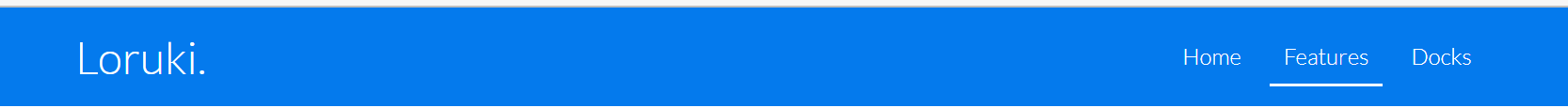
}



.navbar a:hover{

    border-bottom: 2px #fff solid;

}



On pointer la souris sur Features.

Section showcase

   <!--  Show case-->

    <section class="showcase">

      <div class="container grid">

        <div class="shocase-tex">

          <h1> Basier Deployment</h1>

          <p> Deploy web apps of all kinds,

            from large scale enterprise

             APIs to static websites for

            individuals. Fill out the

            form to try a demo of our

             platform</p>

             <a href="features.html" class="btn btn-outline">Read More</a>

        </div>

        <div class="showcase-form card">

          <h2> request a Demo</h2>

          <form>

            <div class="form-control">

            <input type="text" name="name"

             placeholder="Name"

             required>

            </div>

            <div class="form-control">

              <input type="text" name="company"

               placeholder="Company Name"

               required>

              </div>

              <div class="form-control">

                <input type="email" name="email"

                 placeholder="Email"

                 required>

                </div>

     <input type="submit" class="btn btn-primary">

          </form>

        </div>

      </div>

    </section>

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Chaque champ du formulaire est un div de classe form-control et un input de type text sauf pour bouton.

Remarque :

Chaque partie a un div père de classe le nom de la partie.

Exple : Pour la partie navigation le div du navbar contient le container ,le logo et le ul des liens.

Pour la section showcase. On a une section père de classe showcase qui contient un container , un div de text , formulaire,

.showcase{

    height: 400px;

    background-color:#047aed;

}

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

La couleur bacckround-color du showcase est le même que le navbar

21min

.showcase{

    height: 400px;

    /\* background-color:#047aed; \*/

    background-color: var(--primary-color);

    color: #fff;

    position:relative ;

}

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

.showcase h1{

    font-size: 40px;

}

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Flex-grid

Dans le navbar on avait une classe flex et container. Ici aussi dans section on a dans le showcase on une classe container et grid. Grid contient comme fils ‘div.showcase-text ‘et ‘ div.showcase-form card’

<div class="showcase-form card">

 <div class="showcase-text">

.showcase p{

    margin: 20px 0;

}

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

<https://la-cascade.io/articles/css-grid-difference-entre-grilles-explicite-et-implicite>

Dans le navbar on avait une classe flex et container. Ici aussi dans section on a dans le showcase on une classe container et grid. Grid contient comme fils div.showcase-text et div.showcase-form.card.

.grid{

    display: grid;

    grid-template-columns: repeat(2,1fr);

}

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

On voit que on les mis ses fils dans deux colonness.

.card{

    background-color: #fff;

    border-radius: 10px;}

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

.showcase .grid{

    grid-template-columns: 50% 45% ;

}

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

.card{

    background-color: #fff;

    border-radius: 10px;

    box-shadow: 0 3px 10px rgba(0,0,0,0.2);

    padding: 20px;

    margin: 10px

}

Une image contenant texte, capture d’écran, Rectangle, conception

Description générée automatiquement

29 min

On veut utiliser top proriety pour cela on utilise position :relative

.showcase-form{

    position: relative;

    top:60px;

    height:350px ;

}

Une image contenant texte, logiciel, capture d’écran, Page web

Description générée automatiquement

Ici on voit que showcase-form fait sortir du du conatiner. Pour remédier à ça on fait dans .showcase.grid overflow=visible

On a enlever ca :/\* utilities \*/

/\* .container{

    max-width: 1100px;

    margin: 0 auto;

    overflow: auto;

    padding: 0 48px;

} \*/

On obtient :

Une image contenant texte, capture d’écran, logiciel, conception

Description générée automatiquement

.showcase-form{

    position: relative;

    top:60px;

    height:350px ;

    width: 400px;

    padding: 40px;}

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Showcase-form.form-control

.showcase-form .form-control{

   margin:30px 0;

}

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

On voit que form-control s’occupe l’espace entre les chanps input.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ensuite on s’occupe des inputs on a un input type text et un de type email et un type submit.

Une image contenant texte, capture d’écran, logiciel, conception

Description générée automatiquement

.showcase-form input[type="text"],

.showcase-form input[type="email"]{

border:0;border-bottom: 1px solid #b4becb;}

.showcase-form input[type="text"],

.showcase-form input[type="email"]

{

border:0;

border-bottom: 1px solid #b4becb;

width:100%;

}

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

{

border:0;

border-bottom: 1px solid #b4becb;

width:100%;

padding: 3px;

font-size: 16px;

}

Une image contenant texte, capture d’écran, Rectangle, conception

Description générée automatiquement

32min58

On voit si on clique sur le champs on voit le rectangle noir. Pour l’enlever on dire input :focus =none

Une image contenant texte, Police, Rectangle, ligne

Description générée automatiquement

 .showcase-form input:focus{

    outline: none;

}

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

On voit que la couleur du champ est parti quand on y focus.

.btn{

display: inline-block;

padding:10px 30px ;

cursor: pointer;

background-color: var(--primary-color);

color: #fff;

border-radius: 5px;

}

Une image contenant texte, capture d’écran, Rectangle, conception

Description générée automatiquement

Pour la classe btn-outline de Readmore :

.btn-outline{

    background-color: transparent;

    border: 1px #fff solid;

}

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

30 min

.showcase::after{

    content:"sssss";

    /\* Le chowcase avait la position relative \*/

    position:absolute;

    height: 100px;

    bottom: -70px;

    color:white;

    left: 0;

    right: 0;

    background-color: red;

}

Une image contenant texte, capture d’écran, logiciel, ordinateur

Description générée automatiquement

Remarque :

Bottom :-70 px fait descendre de 70px du haut vers le bas. Tout en gardent son height. Donc on aura 30px qui empietine la section showcase.

.showcase::before,

.showcase::after{

    content:"sssss";

    /\* Le chowcase avait la position relative \*/

    position:absolute;

    height: 100px;

    bottom: -70px;

    color:white;

    left: 0;

    right: 0;

    background-color: red;

    transform: skewY(-3deg);

}

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Stats-section**

    <!-- Stats-->

    <section>

      <div class="container">

        <h3 class="stats-heading text-center my-1">

          Welcome to the plattform for building application type with modern architecture and scaling.

        </h3>

        <div class="grid grid-3 text-center my-4">

          <div>

          <i class="fas fa-server fa-3x"></i>

          <h3>10.349.405</h3>

          <p class="text-secondary">Deployment</p>

          </div>

          <div>

            <i class="fas fa-upload fa-3x"></i>

            <h3>987 TB</h3>

            <p class="text-secondary">Publish</p>

          </div>

          <div>

            <i class="fas fa-project-diagram fa-3x"></i>

            <h3>2,343,265</h3>

            <p class="text-secondary">Project</p>

          </div>

        </div>

      </div>

    </section>

Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

On voit aussi que le container ou le section pietine un peu le showcase. On doit le descendre par padding top.

On voit que le h3 est caché :

Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

On a donné a une class grid au contenu des de 3 div. C’est pour cela on a 3 petits rectangle bleux.

.stat{

    padding-top: 100px;

}

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Il faut le mettre à 3 collones.

.grid-3{

    grid-template-columns: repeat(3,1fr);}

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

On a séparé les classes utilities das un fichier css.

Ici on voit la classe .my-1 margin :1rem 0 ;

Une image contenant texte, capture d’écran, Police, Page web

Description générée automatiquement

On va augmenter la taille des chiffres a 35px

.stats .grid h3{

    font-size: 35px;

}

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

.stats .grid p{

    font-size: 20px;

    font-weight: bold;

}

On a augmenté la taille des paragraphe et leur font.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

.cli .grid{

    grid-template-columns: repeat(3,1fr);

}

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

.cli .grid{

    grid-template-columns: repeat(3,1fr);

    grid-template-rows:repeat(2,1fr) ;

}

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Fait apparaitre un scroll bar et plus de place en bas de page.

On a modifié le .grid

.grid{

    display: grid;

    grid-template-columns: repeat(2,1fr);

    gap: 20px;

    justify-content: center;

    align-items: center;

    height: 100%;

}

.cli .grid > \*:first-child{

    grid-column: 1 / span 2;

    grid-row: 1 / span 2;

}

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

53 min.

**Section image cloude**

    <section class="cloud bg-primary my-1 py-2" >

      <div class="conatiner grid">

        <diV class="text-center">

          <h2 class="lg">Extreme Cloud Hosting</h2>

          <p class="load my-1">Cloud Hosting like youv've ,ever seen.Fast, efficient and scalable</p>

          <a href="features.html" class="btn btn-dark">Read More</a>

        </diV>

        <img src="images/cloud.png" alt="">

      </div>

    </section>

  .bg-primary, .btn-primary{

   background-color: var(--primary-color);

   color:#fff;

  }

**Si on change pas le background on ne verra pas l’image du cloud. Car l’image est blanche.**

**Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement**

:root{

    --primary-color:#047aed ;

    --secondary-color:#1c3fa8;

    --dark-color:#002240;

    --light-color:#f4f4f4;

}

On ajoute à root ces couleurs.

ET on ajoute aussi  dans utilities.css:

  .bg-primary, .btn-primary{

   background-color: var(--primary-color);

   color:#fff;

  }

  .bg-secondary, .btn-secondary{

   background-color: var(--secondary-color);

   color:#fff;

  }

  .bg-dark, .btn-dark{

   background-color: var(--dark-color);

   color:#fff;

  }

  .bg-light, .btn-light{

   background-color: var(--light-color);

   color:#fff;

  }

Une image contenant texte, capture d’écran, Police, Bleu électrique

Description générée automatiquement

  .bg-dark, .btn-dark{

   background-color: var(--dark-color);

   color:#fff;

  }

On voit la couleur .btn-dark sur le bouton.

1h04

  .lead{

   font-size: 20px;

  }

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

  .lead{

   font-size: 20px;

  }

  .sm{

   font-size: 1rem;

  }

  .md{

   font-size: 2rem;

  }

  .lg{

   font-size: 3rem;

  }

  .xl{

   font-size: 4rem;;

  }

**Section languages**

 <section class="languages">

      <h2 class="md text-center my-2">

        Supported Languages

        <dv class="containetr flex">

          <div class="card">

            <h4>Node.js</h4>

            <img src="images/logos/node.png">

          </div>

        </dv>

      </h2>

    </section>

**Une image contenant texte, capture d’écran, logo, Police

Description générée automatiquement**

**Ici on va utiliser container et flex car o veut mettre tous les langages sur une même ligne et chacun est card qui conyeinet une image et un h2**

.languages .card{

text-align: center;

margin: 18px 10px 40px;

}

.languages .flex{

    flex-wrap: wrap;

}

.languages .card h4{

  font-size: 20px;

  margin-bottom: 10px;

}

.languages .card:hover{

    transform: translateY(-15px);

}

**Translate : translateY (negative-valeur) fait monter la carte.**

   <!-- Footer -->

    <footer>

      <div class="container grid-3">

        <h1> Loruki</h1>

        <p> Copyright &copy; 2020</p>

        </div>

    </footer>

**1h10**

**Une image contenant texte, capture d’écran, Police, logo

Description générée automatiquement**

 <div class="social">

          <a href="#"> <i class="fab fa-github fa-2x"></i></a>

          <a href="#"> <i class="fab fa-twitter fa-2x"></i></a>

          <a href="#"> <i class="fab fa-facebook fa-2x"></i></a>

          <a href="#"> <i class="fab fa-instagram fa-2x"></i></a>

          <!-- <a href="#"> <i class="fa-twitter fa-2x"></i></a> -->

        </div>

    </footer>

**Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement**

 <footer class="footer bg-dark py-5">

**On a jouter les class footer bg-dark py-5**

**Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement**

Ensuite on change les icones en blacns.

Remarque pour les icones : on

 <i class="fab fa-github fa-2x">

Et

 <i class="fas fa-server fa-3x"></i>

  .bg-primary a,

  .bg-secondary a,

  .bg-dark a,

  .bg-light a{

color: #fff;

  }

**Ensuite on ajoute les icones : les classes grid et grid-3 dans**

 <footer class="footer bg-dark py-5">

      <div class="container grid grid-3">

**Une image contenant capture d’écran, texte, Police

Description générée automatiquement**

**Media queries**

**Si on diminue au bout la largeur de la page on obtient.**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

/\* Media query \*/

/\*  Mobile\*/

@media(max-width: 500px){

.navbar{

    height: 110px;

}

.navbar .flex{

    flex-direction: column;

}

.navbar ul{

    padding: 10px;

background-color: rgb(0,0,0,0.1);

}}

**Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

On voit que navbar est est inf a 500px voir images ci-dessous. Donc on applique flex-drection en column . On a deux élement de flex div du logo et le nav. Qui sont en column.

**Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement**

**Ici navbar est a 509px de height donc on change pas de navbar.**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

Le rgba(0 ,0,0,0.1) : donne une couleur transparent bleue.

En média query Mobile : on augmente la hauteur du Navbar.

On veut pour les tablette que tout soit en colonne. Donc on ajoute toute les classe qui ont la propriété grid.

On ajoute dans le media query que les propriétés css qui on existé avant .

Pour le showcas-form on utilise justify-self a center. Et margin auto. Ce qui centralise tout les éléments.

.showcase-form{

    justify-self: center;

    margin: auto;

}

Pour le .cli

.cli .grid > \*:first-child{

    grid-column: 1 ;

    grid-row: 1;

}

On enleve les span et on fait grid-columet grid-row a 1.