Font

Ales dans google fonte. Rechercher Lato et choisire

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

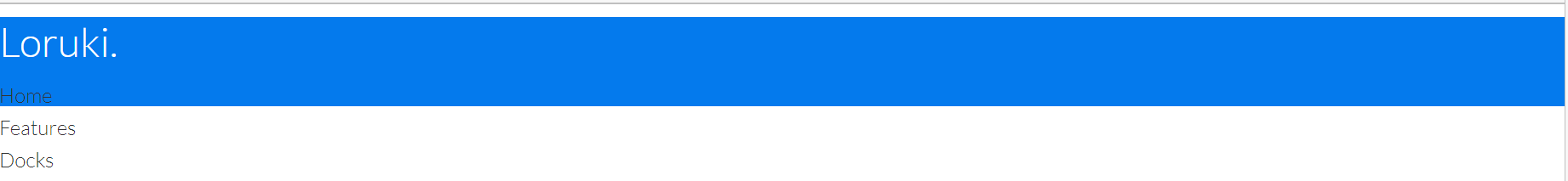
.navbar{

    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.



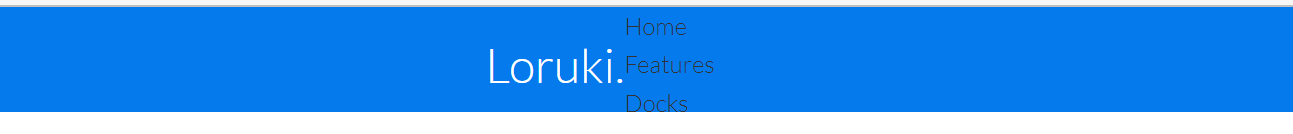
.flex{

    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;

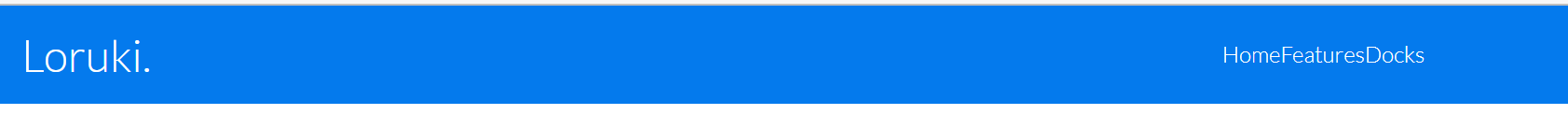
}



.navbar a{

    color: #fff;

}

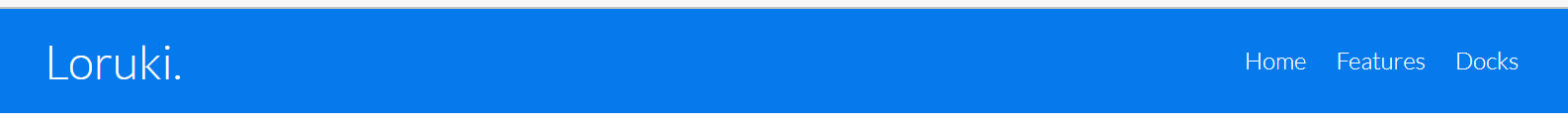


.navbar a{

    color: #fff;

    padding: 10px;

}



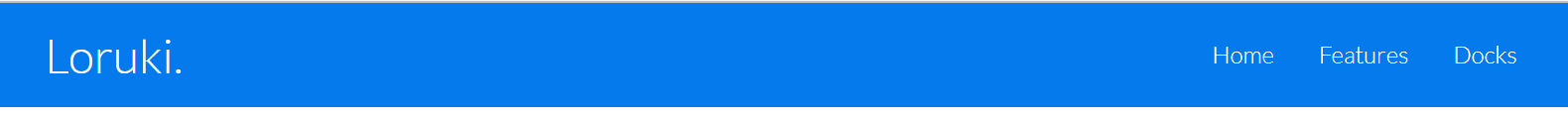
.navbar a{

    color: #fff;

    padding: 10px;

    margin: 10px 5px;

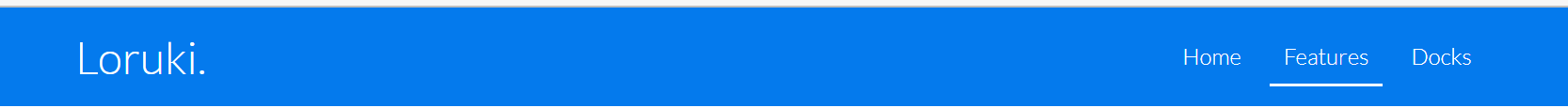
}



.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.