@import url("https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;1,200;1,300;1,400;1,500;1,600;1,700&family=PT+Sans:ital,wght@0,400;0,700;1,400;1,700&display=swap");

:root {

--black: hsla(0, 0%, 0%, 1);

--oxford-blue: hsla(221, 51%, 16%, 1);

--orange-web: hsla(37, 98%, 53%, 1);

--orange-web-transparent: hsla(37, 98%, 53%, 0.95);

--platinum: hsla(0, 0%, 90%, 1);

--white: hsla(0, 0%, 100%, 1);

--modal-duration: 1s;

}

\* {

box-sizing: border-box;

}

body {

display: flex;

margin: 0;

font-family: "PT Sans", "Gill Sans", sans-serif;

}

h1,

h2,

h3,

h4,

h5,

h6 {

font-family: "Kanit", "Gill Sans", sans-serif;

font-weight: 400;

line-height: 1;

}

input {

padding: 0.5rem;

margin: 0.2rem;

border: 0;

}

input:hover {

cursor: pointer;

filter: brightness(130%);

}

.page-content {

width: 100%;

}

.top-nav {

position: relative;

display: flex;

flex-direction: column;

align-items: center;

min-width: 17rem;

max-width: 30rem;

border-right: 1px solid var(--platinum);

background: var(--oxford-blue);

z-index: 100;

}

.logo {

margin: 1rem;

}

.logo img {

border-radius: 50%;

}

.nav-links {

width: 100%;

padding: 0;

list-style: none;

}

.nav-links li {

display: block;

width: 100%;

padding: 0;

margin: 0;

padding: 0.8rem;

border-bottom: 1px solid var(--orange-web);

background: var(--oxford-blue);

color: var(--white);

transition: background 0.2s ease-in-out;

}

.nav-links li:first-of-type {

border-top: 1px solid var(--orange-web);

}

.nav-links li:hover {

cursor: pointer;

background: var(--orange-web);

text-decoration: underline;

}

header {

position: relative;

padding: 1rem;

background: var(--oxford-blue);

text-align: center;

font-size: 130%;

color: white;

text-align: center;

}

header button {

width: 50%;

}

header h1 {

padding: 0 1rem;

max-width: 80%;

margin: 0 auto;

}

header p {

margin: 1rem 0;

}

/\* Main container style \*/

.container {

display: flex;

flex-direction: column;

padding: 1rem;

max-width: 100%;

}

/\* Button styles \*/

button,

input[type="submit"] {

padding: 1rem;

border: none;

border-radius: 0.7rem;

background: var(--platinum);

color: var(--black);

transition: filter 0.2s ease-in-out;

}

button:hover,

button:focus {

cursor: pointer;

filter: brightness(175%);

outline: 0;

}

.toggle {

display: none;

position: fixed;

left: 1rem;

top: 1rem;

width: min-content;

text-align: left;

background: hsla(0, 0%, 0%, 0.5);

border-radius: 20%;

transition: transform 0.4s ease-in-out;

color: var(--white);

}

.toggle:hover {

cursor: pointer;

transform: scale(1.05);

}

.cta-button {

font-size: 1.2rem;

}

/\* Modal styles \*/

.modal-container {

display: none;

justify-content: center;

align-items: center;

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

height: 100%;

width: 100%;

background: hsla(0, 0%, 0%, 0.8);

animation: fade 0.5s forwards;

}

.modal {

display: flex;

flex-direction: column;

min-width: 20rem;

max-width: 25rem;

background: var(--orange-web-transparent);

border-radius: 1rem;

box-shadow: 0 0 10px hsla(0, 0%, 0%, 0.5);

color: var(--black);

overflow: hidden;

}

.show-modal {

display: flex;

animation: fadeIn 0.5s forwards;

}

.hide-modal {

animation: fadeOut 0.5s forwards;

}

.modal-header {

display: flex;

align-items: center;

width: 100%;

padding: 1rem;

background: var(--oxford-blue);

color: var(--white);

}

.modal-header h3 {

text-align: center;

font-size: 1.8rem;

width: 100%;

margin: 0;

}

.close-button {

position: absolute;

flex: 1 0 2rem;

max-width: 3rem;

padding: 0.5rem 0.8rem;

font-size: 1.5rem;

}

.close-button > \* {

pointer-events: none;

}

.modal-content {

padding: 1rem;

text-align: center;

}

.form-inputs {

display: flex;

flex-direction: column;

justify-content: space-between;

}

.form-inputs input {

flex: 1 1 auto;

width: 100%;

background: var(--platinum);

}

.radio-inputs {

display: flex;

flex-direction: column;

margin: auto;

padding: 0.5rem;

font-size: 1.1rem;

}

.radio-inputs label {

text-align: right;

}

.radio-inputs input {

width: auto;

}

.form-inputs input[type="submit"] {

text-align: center;

margin: auto;

}

/\* Animations \*/

@keyframes fadeIn {

0% {

opacity: 0;

}

100% {

opacity: 1;

}

}

@keyframes fadeOut {

0% {

opacity: 1;

}

100% {

opacity: 0;

}

}

/\* Set nav to disappear at small screen size \*/

@media (max-width: 700px) {

.top-nav {

position: fixed;

top: 0;

left: 0;

height: 100vh;

transform: translateX(-100%);

transition: transform 0.3s ease-in-out;

}

.toggle {

display: block;

z-index: 100;

}

.slide-nav {

transform: translateX(0%);

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link rel="stylesheet" href="style.css" />

<script src="https://kit.fontawesome.com/9966c2ab14.js" crossorigin="anonymous"></script>

<title>A Very Professional Landing Page</title>

<link rel="stylesheet" href="model.responsive.css">

</head>

<body>

<nav class="top-nav">

<div class="logo">

<img src="https://randomuser.me/api/portraits/men/51.jpg" alt="user portrait" />

</div>

<ul class="nav-links">

<li><a href="#"></a>Home</li>

<li><a href="#"></a>Portfolio</li>

<li><a href="#"></a>Blog</li>

<li><a href="#"></a>Contact</li>

</ul>

</nav>

<div class="page-content">

<header>

<button id="toggle" class="toggle">

<i class="fa fa-bars fa-2x"> </i>

</button>

<h1>Wow, such professional. Very business.</h1>

<p>

Leverage agile frameworks to provide a robust synopsis for high level

overviews.

</p>

<button class="cta-button" id="open-button">Sign Up!</button>

</header>

<main class="container">

<h2>On the Internet nobody knows you're a dog.</h2>

<p>

Leverage agile frameworks to provide a robust synopsis for high level

overviews. Iterative approaches to corporate strategy foster

collaborative thinking to further the overall value proposition.

Organically grow the holistic world view of disruptive innovation via

workplace diversity and empowerment. Bring to the table win-win

survival strategies to ensure proactive domination. At the end of the

day, going forward, a new normal that has evolved from generation X is

on the runway heading towards a streamlined cloud solution. User

generated content in real-time will have multiple touchpoints for

offshoring.

</p>

<p>

Capitalize on low hanging fruit to identify a ballpark value added

activity to beta test. Override the digital divide with additional

clickthroughs from DevOps. Nanotechnology immersion along the

information highway will close the loop on focusing solely on the

bottom line.

</p>

<h2>OMG MOAR BUSINESS!</h2>

<p>

Podcasting operational change management inside of workflows to

establish a framework. Taking seamless key performance indicators

offline to maximise the long tail. Keeping your eye on the ball while

performing a deep dive on the start-up mentality to derive convergence

on cross-platform integration.

</p>

<p>

Collaboratively administrate empowered markets via plug-and-play

networks. Dynamically procrastinate B2C users after installed base

benefits. Dramatically visualize customer directed convergence without

revolutionary ROI.

</p>

<h2>So, should I business?</h2>

<ul>

<li>Do you have a good company idea?</li>

<li>No? Okay, here's one for free.</li>

<li>App that tracks all cats.</li>

<li>Also pawprint unlock for phone.</li>

<li>Much welcome. Very luck.</li>

</ul>

</main>

</div>

<!-- Modal HTML -->

<div class="modal-container" id="modal-container">

<div class="modal">

<div class="modal-header">

<button class="close-button" id="close-button">

<i class="fa fa-times"> </i>

</button>

<h3>Sign Up!</h3>

</div>

<div class="modal-content">

<p>

Plz sign up heer if you R hooman or dog. If cat, plz email with

lokayshun.

</p>

<form class="modal-form">

<div class="form-inputs">

<label for="name">

<input type="text" id="name" placeholder="Name" />

</label>

<label for="email">

<input type="email" id="email" placeholder="Email" />

</label>

<label for="name">

<input type="password" id="password" placeholder="Password" />

</label>

<label for="name">

<input type="password" id="password-confirm" placeholder="Confirm Password" />

</label>

<div class="radio-inputs">

<label for="cat">I am a cat (=^ェ^=)

<input type="radio" id="cat" value="Cat" name="is-cat" />

</label>

<label for="not-cat">Not a cat (=①ω①=)

<input type="radio" id="not-cat" value="Not a cat" name="is-cat" />

</label>

</div>

<input type="submit" value="Submit" class="submit" />

</form>

</div>

</div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

.first-block {

border-style: solid;

width: 40%;

}

.second-block {

border-style: solid;

height: 2em;

}

.third-block {

float: right;

width: 50%;

}

/\* indice Le positionnement en bas de la page peut se faire par

l'utilisation de position: absolute; et bottom: 0;.

\*/

@media screen and (max-width: 760px){

.first-block { /\*L'élément HTML avec la classe first-block se positionne en bas de la page si la largeur de l'écran est inférieure à 760 px.\*/

border-style: solid;

position:absolute;

bottom: 0;

width: 40%;

}

}

@media print {

.second-block { /\*L'élément HTML avec la classe second-block ne doit faire que 75 % de la taille de son parent lors d'une impression.\*/

border-style: solid;

height: 2em;

width: 755;

}

}

@media screen and (min-width: 1024px), (max-heigth:800px) {

.third-block { /\*L'élément HTML avec la classe third-block devient rouge lorsque la largeur de l'écranfait plus de 1024 px ou lorsque sa hauteur est inférieure à 800 px.\*/

float: right;

width: 50%;

background: red;

}

}

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=, initial-scale=1.0">

<title>Document</title>

<link href="notion media queries.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" media="print" href="print.css" type="text/css" />

<link rel="stylesheet" media="screen" href="screen.css" type="text/css" />

</head>

<body>

<!--commentaire notion media queries

Appliquez la notion

Question

Pour mettre en application ce que nous venons de voir, vous allez créez une page HTML sur votre environnement de travail VSCode et écrire la feuille CSS associée.

Pour cela, vous réutiliserez le code HTML ainsi que la feuille CSS proposés ci-dessous, et devrez écrire les media queries répondant aux question suivantes :

Il est recommandé d'agir sur les différents paramètres d'affichage de la fenêtre du navigateur afin de voir les modifications dynamiques du style grâce aux outils de développement des navigateurs.

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail VSCode.

-->

<body>

<p class="first-block">Je passe en bas de la page sur les petits écrans</p>

<div>

<p class="second-block">Je fais les trois quarts de la taille de mon parent lors d'une impression </p>

</div>

<p class="third-block">Je deviens rouge sur les grands écrans ou lorsque la hauteur de l'écran est inférieure à 800px</p>

</body>

</html>

/\*Le positionnement à droite peut se faire par

l'utilisation de float: right;.\*/

.change-width{/\*change-width : elle devra afficher la bordure de l’élément

et s'assurer que sa largeur soit toujours de 40 % de la largeur du parent.\*/

border-style: solid ;

width: 40%;

}

div {

font-size: 16px;

/\*Cette valeur peut être changée pour modifier

la hauteur des éléments ayant la classe change-height\*/

}

.change-height{/\*change-height : elle devra afficher la bordure de l'élément

et s'assurer que sa hauteur soit toujours le double de la police de l’élément parent (div).\*/

border-style: solid ;

height: 2em;

}

.bottom-right{/\*bottom-right : elle devra placer l'élément à droite de la page,

s'assurer que sa largeur soit toujours de la moitié de la largeur de la page, et le colorer en rouge.\*/

float: right;

background: red;

width: 50%;

}

.change-viewport{/\*change-viewport : elle devra définir une taille de police

correspondant à 7 % de la hauteur du viewport.\*/

height: 7vh;

}

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link href="responsive notion.css" rel="stylesheet" type="text/css" />

</head>

<body>

<!--commentaire:

Appliquez la notion

Question

Pour mettre en application ce que nous venons de voir, vous allez créer une page HTML sur votre environnement de travail VSCode et écrire la feuille CSS associée.

Pour cela, vous réutiliserez le code HTML proposé ci-dessous et devrez écrire les classes CSS suivantes :

change-width : elle devra afficher la bordure de l’élément et s'assurer que sa largeur soit toujours de 40 % de la largeur du parent.

change-height : elle devra afficher la bordure de l'élément et s'assurer que sa hauteur soit toujours le double de la police de l’élément parent (div).

bottom-right : elle devra placer l'élément à droite de la page, s'assurer que sa largeur soit toujours de la moitié de la largeur de la page, et le colorer en rouge.

change-viewport : elle devra définir une taille de police correspondant à 7 % de la hauteur du viewport.

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail VSCode.-->

<p class="change-width">Je change de taille si la taille de l'écran change</p>

<div>

<p class="change-height">Ma hauteur est égale a deux fois celle de la police de mon parent</p>

</div>

<p class="bottom-right">Je suis en rouge et je reste toujours collé à droite</p>

<p class="change-viewport">La taille de ma police change avec le viewport</p>

</body>

</html>

/ Smaller menu when on small screen

// All padding and margin are in em, so they will scale as well

@media (min-width : 900px) {

.menu {

font-size: 1.2em;

}

}

.menu {

padding: 0.5em;

background: #eee;

min-height: 2em;

line-height: 1em;

> ul {

transition: max-height .25s linear;

}

ul {

margin: 0;

padding: 0;

text-align: center;

}

li {

// visibility transition is on li because multiple transition selectors is not well supported

transition: visibility .25s linear;

display: inline-block;

border: 1px solid;

padding: .45em 1.1em;

margin: 0 .3em;

position: relative;

}

}

@media (min-width : 651px) {

.menu {

li { // nested menu

ul {

display: none;

position: absolute;

top: 100%;

margin-top: 1px;

left: -1px;

right: -1px;

}

&:hover ul {

display: block;

}

li {

margin: -1px 0 0 0;

box-sizing: border-box;

width: 100%;

}

}

}

}

@media (max-width : 650px) {

.menu {

> ul {

max-height: 0;

overflow: hidden;

margin: 0 3.5em 0 1em;

}

li {

visibility: hidden;

display: block;

padding: 0.5em 0.6em;

border: none;

}

li { // nested menu

ul {

margin-top: 0.5em;

border-left: 1px solid #000;

}

}

.navbar-handle {

display: block;

}

}

#navbar-checkbox:checked + .menu {

ul {

max-height: 300px;

}

li {

visibility: visible;

}

.navbar-handle {

&, &:after, &:before {

border-color: #aaa;

}

}

}

}

.navbar-checkbox {

display: none;

}

.navbar-handle {

@height: 45px; // just a reference to compute em values on the fly

display: none;

cursor: pointer;

position: relative;

font-size: @height;

padding: .5em 0;

height: 0;

width: 1em \* 75px / @height;

border-top: (1em \* 6px / @height) solid;

&:before, &:after {

position: absolute;

left: 0;

right: 0;

content: ' ';

border-top: (1em \* 6px / @height) solid;

}

&:before {

top: 1em \* 17px / @height;

}

&:after {

top: 1em \* 40px / @height;

}

}

///////////

.menu {

position: absolute;

top: 0;

left: 0;

right: 0;

.navbar-handle {

position: absolute;

font-size: 1.2em;

top: 0.7em;

right: 12px;

z-index: 10;

}

}

body{

margin:0;

font-family:'Times New Roman', Times, serif;

}

.back-menu{

width: 100%;

height:120px;

background-color: rgba(8, 8, 8, 0.836);

margin:0;

position:fixed;

-webkit-box-shadow: 0px 10px 13px -6px rgba(92,92,92,0.66);

-moz-box-shadow: 0px 10px 13px -6px rgba(92,92,92,0.66);

box-shadow: 0px 10px 13px -6px rgba(92,92,92,0.66);

}

.back-menu-center{

width: 75%;

height:100%;

margin: 0 auto;

/\* background-color: chartreuse; \*/

display: flex;

flex-wrap: nowrap;

}

.logo{

width:22%;

height:100%;

/\* background-color: chocolate; \*/

left:0;

overflow: hidden;

display: flex;

}

.logo-link{

color: #fff;

font-size: 17px;

font-weight: 800px;

text-decoration: none;

}

.logo-link img{

width:100px;

}

.logo-link:nth-child(2){

position: relative;

top: 32%;

left: 5%;

}

.menu-items{

/\* background-color: red; \*/

height:70px;

width: 100%;

align-items: center;

display: flex;

top: 50px;

position: relative;

}

.menu-items ul{

list-style: none;

display: flex;

}

li[class^="menu-item-"] a{

margin:12px 22px;

color: #fff;

text-decoration: none;

font-weight: 600;

-webkit-transition: all 0.9s ease-in-out;

-moz-transition: all 0.9s ease-in-out;

-ms-transition: all 0.9s ease-in-out;

-o-transition: all 0.9s ease-in-out;

transition: all 0.9s ease-in-out;

}

li[class^="menu-item-"] a:hover{

color: #dc18ff!important;

}

.menu-item-1 a{

color:rgb(97 159 255)!important;

}

.menu-register{

position: absolute;

top:0;

right:0;

/\* background-color: cornflowerblue; \*/

margin:7px;

}

.register{

color: #fff;

text-decoration: none;

margin:8px;

}

.register:hover{

color: #dc18ff!important;

}

.toggle-menu{

transform: translateX(10%) translateY(100%);

display: none;

}

.toggle-icon{

font-size:30px;

background-color: #fff;

padding:10px;

border-radius: 3px;

cursor: pointer;

}

.toggle-items{

list-style: none;

background-color:#351630;

width:110px;

line-height:47px;

position: absolute;

display: none;

border-radius: 5px;

-webkit-animation-name: zoomIn;

animation-name: zoomIn;

-webkit-animation-duration: 1s;

animation-duration: 1s;

-webkit-animation-fill-mode: both;

animation-fill-mode: both;

}

@-webkit-keyframes zoomIn {

0% {

opacity: 0;

-webkit-transform: scale3d(.3, .3, .3);

transform: scale3d(.3, .3, .3);

}

50% {

opacity: 1;

}

}

@keyframes zoomIn {

0% {

opacity: 0;

-webkit-transform: scale3d(.3, .3, .3);

transform: scale3d(.3, .3, .3);

}

50% {

opacity: 1;

}

}

li[class^="toggle-item-"] a{

color: #fff;

text-decoration: none;

font-weight: lighter;

}

li[class^="toggle-item-"] a:hover{

color: #dc18ff!important;

}

.toggle-item-1 a{

color:rgb(97 159 255)!important;

}

#toggle-btn{

display: none;

}

#toggle-btn:checked +ul{

display: block;

}

@media screen and (max-width:953px) {

.toggle-menu{

display: block;

}

.back-menu-center{

display: none;

}

}

@media screen and (min-width:953px) {

.toggle-menu{

display: none;

}

}

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>responsive</h1>

<input type="checkbox" id="navbar-checkbox" class="navbar-checkbox">

<nav class="menu">

<ul>

<li>Home</li>

<li>About us</li>

<li>

Our company

<ul>

<li>History</li>

<li>Very long and tedious history</li>

</ul>

</li>

<li>Our team</li>

<li>Contact us</li>

</ul>

<label for="navbar-checkbox" class="navbar-handle"></label>

</nav>

<body>

<div class="back-menu">

<div class="back-menu-center">

<div class="logo">

<a class="logo-link" href="#"><img src="https://cdn.logo.com/hotlink-ok/logo-social-sq.png" alt="YourLogo"></a>

<a class="logo-link" href="#">Brand</a>

</div>

<div style="width:78%;position: relative;">

<div class="menu-items">

<ul>

<li class="menu-item-1"><a href="#">Home</a></li>

<li class="menu-item-2"><a href="#">Blog</a></li>

<li class="menu-item-3"><a href="#">About us</a></li>

<li class="menu-item-4"><a href="#">Contact us</a></li>

</ul>

</div>

<div class="menu-register">

<a class="register" href="#">Register</a>

<span style="color: #fff;font-weight: 900;">|</span>

<a class="register" href="#">login</a>

</div>

</div>

</div>

<div class="toggle-menu">

<label for="toggle-btn">

<i class="fas fa-bars toggle-icon"></i>

</label>

<input id="toggle-btn" type="checkbox">

<ul class="toggle-items">

<li class="toggle-item-1"><a href="#">Home</a></li>

<li class="toggle-item-2"><a href="#">Blog</a></li>

<li class="toggle-item-3"><a href="#">About us</a></li>

<li class="toggle-item-4"><a href="#">Contact us</a></li>

<li class="toggle-item-5"><a href="#">Register | login</a></li>

</ul>

</div>

</body>

</html>

</body>

</html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="text responsive.css">

</head>

<body>

<div class="smartphone-menu-trigger" onclick="myFunction(event);"></div>

<nav class="menu" id="navbar" tabindex="0">

<header class="logo">Jump To Content</header>

<ul>

<li tabindex="0"><a class="nav-link" href="#introduction">Introduction</a></li>

<li tabindex="0"><a class="nav-link" href="#comments">Comments</a></li>

<li tabindex="0"><a class="nav-link" href="#colours">Colours</a></li>

<li tabindex="0"><a class="nav-link" href="#backgrounds">Backgrounds</a></li>

<li tabindex="0"><a class="nav-link" href="#borders">Borders</a></li>

<li tabindex="0"><a class="nav-link" href="#margins">Margins</a></li>

<li tabindex="0"><a href="#padding">Padding</a></li>

<li tabindex="0"><a href="#height-width">Height/Width</a></li>

<li tabindex="0"><a href="#box-model">Box Model</a></li>

<li tabindex="0"><a href="#outline">Outlines</a></li>

<li tabindex="0"><a href="#text">Text</a></li>

<li tabindex="0"><a href="#links">Links</a></li>

<li tabindex="0"><a href="#lists">Lists</a></li>

</ul>

</nav>

<main id="main-doc">

<h1 id="main-logo">documentify CSS</h1>

<hr>

<section class="main-section" id="introduction">

<header class="h2">Introduction

</header>

<article>

<h3>What is CSS?</h3>

<ul>

<li>CSS stands for Cascading Style Sheets.</li>

<li>CSS describes how HTML elements are to be displayed on screen, paper, or in other media.</li>

<li>CSS saves a lot of work. It can control the layout of multiple web pages all at once</li>

<li>External stylesheets are stored in CSS files</li>

</ul>

<h3>Why Use CSS?</h3>

<p>CSS is used to define styles for your web pages, including the design, layout and variations in display for different

devices and screen sizes.</p>

</article>

</section>

<hr>

<section class="main-section" id="comments">

<header class="h2">Comments

</header>

<article>

<p>Comments are used to explain the code, and may help when you edit the source code at a later date.Comments are ignored

by browsers. A CSS comment starts with /\* and ends with \*/. You can add comments wherever you want in the code. A

comment can span multiple lines.</p>

<code class="keyboard-example">/\* This is a CSS comment \*/</code>

</article>

</section>

<hr>

<section id="colours" class="main-section">

<header class="h2">Colours

</header>

<article>

<p>The color CSS data type represents a color. A color may also include an alpha-channel transparency value, indicating

how the color should composite with its background. A color can be defined in any of the following ways:</p>

<ul>

<li>Using a keyword (such as blue or transparent).</li>

<li>Using the RGB cubic-coordinate system (via the #-hexadecimal or the rgb() and rgba() functional notations).</li>

<li>Using the HSL cylindrical-coordinate system (via the hsl() and hsla() functional notations).</li>

<li>Using the LCH cylindrical coordinate system, via the lch() functional notation.</li>

<li>Using the Lab coordinate system, via the lab() functional notation.</li>

<li>Using the color() functional notation, to specify a color in a variety of predifined or custom color spaces.</li>

</ul>

</article>

</section>

<hr>

<section class="main-section" id="backgrounds">

<header class="h2">Backgrounds</header>

<article>

<p>The background property in CSS allows you to control the background of any element (what paints underneath the content

in that element). It is a shorthand property, which means that it allows you to write what would be multiple CSS

properties in one. Background is made up of eight other properties:</p>

<ul>

<li>backround-image</li>

<li>background-color</li>

<li>background-position</li>

<li>background-size</li>

<li>background-clip</li>

<li>background-origin</li>

<li>background-blend-mode</li>

<li>background-attachment</li>

<li>background-repeat</li>

</ul>

<p>You can use any combination of these properties that you like, in almost any order (although the order recommended in

the spec is above). There is a gotcha though: anything you don’t specify in the background property is automatically set

to its default</p>

<code class="keyboard-example">body{ background-color: coral;}</code><br>

<code class="keyboard-example">body{ background-size: cover;}</code>

</article>

</section>

<hr>

<section class="main-section" id="borders">

<header class="h2">

Borders

</header>

<article>

<p>The border shorthand CSS property sets an element's border. It sets the values of border-width, border-style, and

border-color. The border property may be specified using one, two, or three of the values listed below. The order of

the values does not matter. The border will be invisible if its style is not defined. This is because the style

defaults to none. As with all shorthand properties, any omitted sub-values will be set to their initial value.

Importantly, border cannot be used to specify a custom value for border-image, but instead sets it to its initial

value, i.e., none. The border shorthand is especially useful when you want all four borders to be the same. To make

them different from each other, however, you can use the longhand border-width, border-style, and border-color

properties, which accept different values for each side. Alternatively, you can target one border at a time with the

physical (e.g., border-top ) and logical (e.g., border-block-start) border properties.Borders and outlines are very

similar. However, outlines differ from borders as outlines never take up space, as they are drawn outside of an

element's content. According to the spec, outlines don't have to be rectangular, although they usually are.</p>

</article>

</section>

<hr>

<section class="main-section" id="margins">

<header class="h2">

Margins

</header>

<article>

<p>Margins are used to create space around elements, outside of any defined borders. The CSS margin properties are used to

create space around elements, outside of any defined borders. With CSS, you have full control over the margins. There

are properties for setting the margin for each side of an element (top, right, bottom, and left). CSS has properties for

specifying the margin for each side of an element:</p>

<ul>

<li>margin-top</li>

<li>margin-bottom</li>

<li>margin-left</li>

<li>margin-right</li>

</ul>

<p>All the margin properties can have the following values:</p>

<ul>

<li>auto - the browser calculates the margin.</li>

<li>length - specifies a margin in px, pt, cm, etc.</li>

<li>% - specifies a margin in % of the width of the containing element.</li>

<li>inherit - specifies that the margin should be inherited from the parent element.</li>

</ul>

<h3>Margin-shorthand Property</h3>

<p>To shorten the code, it is possible to specify all the margin properties in one property.</p>

<h3>Margin Collapse</h3>

<p>Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the

two margins.This does not happen on left and right margins! Only top and bottom margins!</p>

<code class="keyboard-example">h2{ margin-left: 30px;}</code><br>

<code class="keyboard-example">#main{margin: 30px 20px;}</code>

</article>

</section>

<hr>

<section id="padding">

<h2>PADDING</h2>

<article>

<p>Padding is used to create space around an element's content, inside of any defined borders. With CSS, you have full control over the padding.

There are properties for setting the padding for each side of an element (top, right, bottom, and left).CSS has properties for specifying the

padding for each side of an element:</p>

<ul>

<li>padding-top</li>

<li>padding-bottom</li>

<li>padding-left</li>

<li>padding-right</li>

</ul>

<p>All the padding properties can have the following values:</p>

<ul>

<li>length - specifies a padding in px, pt, cm, etc.</li>

<li>% - specifies a padding in % of the width of the containing element.</li>

<li>inherit - specifies that the padding should be inherited from the parent element.</li>

</ul>

<h3>Padding-shorthand Property</h3>

<p>To shorten the code, it is possible to specify all the padding properties in one property.</p>

<h3>Padding and element width</h3>

<p>The CSS width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and

margin of an element (the box model). So, if an element has a specified width, the padding added to that element will be added to the total

width of the element. This is often an undesirable result.</p>

</article>

</section>

<hr>

<section id="height-width">

<h2>HEIGHT-WIDTH</h2>

<p>The height and width properties are used to set the height and width of an element. The height and width properties do not include padding, borders,

or margins. It sets the height/width of the area inside the padding, border, and margin of the element. The height and width properties may have the

following values:</p>

<ul>

<li>length - specifies height and width in px, pt, cm, etc.</li>

<li>% - specifies a height and width in % of the containing element.</li>

<li>inherit - specifies that the height and width should be inherited from the parent element.</li>

<li>auto - This is default. The browser calculates the height and width.</li>

<li>initial- This sets height and width to default value.</li>

</ul>

<div class="keyboard-example">

div {<br>

height: 100px;<br>

width: 500px;<br>

background-color: powderblue;<br>

}<br>

</div>

<p>The problem occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

The max-width property is used to set the maximum width of an element. The max-width can be specified in length values, like px, cm, etc., or in

percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width). Using max-width instead, in this

situation, will improve the browser's handling of small windows. If you for some reason use both the width property and the max-width property on

the same element, and the value of the width property is larger than the max-width property; the max-width property will be used (and the width

property will be ignored).</p>

<table>

<caption>Various height and width properties</caption>

<tr>

<th>PROPERTY</th>

<th>DESCRIPTION</th>

</tr>

<tr>

<td>height</td>

<td>Sets height of a element</td>

</tr>

<tr>

<td>max-height</td>

<td>Sets maximum height of a element</td>

</tr>

<tr>

<td>min-height</td>

<td>Sets minimum height of a element</td>

</tr>

<tr>

<td>width</td>

<td>Sets width of a element</td>

</tr>

<tr>

<td>max-width</td>

<td>Sets maximum width of a element</td>

</tr>

<tr>

<td>min-width</td>

<td>Sets minimum width of a element</td>

</tr>

</table>

</section>

<hr>

<section id="box-model">

<h2>BOX-MODEL</h2>

<article>

<p>All HTML elements can be considered as boxes. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins,

borders, padding, and the actual content. The image below illustrates the box model: </p>

<img src="box-model.PNG" alt="CSS BOX MODEL">

<p>Explaination od different parts:</p>

<ol>

<li>CONTENT - The content of the box, where text and images appear</li>

<li>PADDING - Clears an area around the content. The padding is transparent.</li>

<li>BORDER - A border that goes around the padding and content.</li>

<li>MARGIN - Clears an area outside the border. The margin is transparent.</li>

</ol>

<p>The box model allows us to add a border around elements, and to define space between elements. Consider the following div:</p>

<div class="keyboard-example">

div {<br>

width: 300px;<br>

border: 15px solid green;<br>

padding: 50px;<br>

margin: 20px;<br>

}<br>

</div>

<p>In order to set the width and height of an element correctly in all browsers, you need to know how the box model works. When you set the width and

height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must

also add padding, borders and margins. For example full width and heights of the div above are 470px both.</p>

</article>

</section>

<hr>

<section id="outline">

<h2>OUTLINES</h2>

<article>

<p>An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out". CSS has the following outline

properties:</p>

<ul>

<li>outline-style</li>

<li>outline-color</li>

<li>outline-width</li>

<li>outline-offset</li>

<li>outline</li>

</ul>

<p>Outline differs from borders! Unlike border, the outline is drawn outside the element's border, and may overlap other content. Also, the outline

is NOT a part of the element's dimensions, the element's total width and height is not affected by the width of the outline.</p>

<h3>CSS outline-style PROPERTY</h3>

<p>The outline-style property specifies the style of the outline, and can have one of the following values:</p>

<ol>

<li>dotted - Defines a dotted outline</li>

<li>dashed - Defines a dashed outline</li>

<li>solid - Defines a solid outline</li>

<li>double - Defines a double outline</li>

<li>groove - Defines a 3D grooved outline</li>

<li>ridge - Defines a 3D ridged outline</li>

<li>inset - Defines a 3D inset outline</li>

<li>outset - Defines a 3D outset outline</li>

<li>none - Defines no outline</li>

<li>hidden - Defines a hidden outline</li>

</ol>

<em>None of the other outline properties will have ANY effect unless the outline-style property is set!</em>

</article>

</section>

<hr>

<section id="text">

<h2>TEXT</h2>

<article>

<p>This text is styled with text-formatting properties. Some of which are:</p>

<ul>

<li>text-alignment</li>

<li>text-color OR color</li>

<li>text-decoration</li>

<li>text-transformation</li>

<li>text-shadow</li>

<li>text-spacing</li>

</ul>

</article>

</section>

<hr>

<section id="links">

<h2>LINKS</h2>

<article>

<p>With CSS, links can be styled in many different ways. Links can be styled with any CSS property (e.g. color, font-family, background, etc.). In

addition, links can be styled differently depending on what state they are in. The four links states are:</p>

<ul>

<li>a:unvisited - A normal unvisited link</li>

<li>a:visited - A link user has visited</li>

<li>a:hover - A link when user mouses over it</li>

<li>a:active - A link when it is clicked</li>

</ul>

<em>When setting the style for several link states, there are some order rules:</em>

<ol>

<li>a:hover <strong>MUST</strong> come after a:link and a:visited</li>

<li>a:active <strong>MUST</strong> come after a:hover</li>

</ol>

<h3>Text Decoration</h3>

<p>The text-decoration property is mostly used to remove underlines from links.</p>

<h3>Background Color</h3>

<p>The background-color property can be used to specify a background color for links.</p>

<h3>Link Buttons</h3>

<p>We can combine several CSS properties to display links as boxes/buttons.</p>

<h3>Cursor</h3>

<p>You can make cursor look like different pointers when hovering over a link like cross-hair, resize-pointer, loading-pointer etc.</p>

</article>

</section>

<hr>

<section id="lists">

<h2>LISTS</h2>

<article>

<p>The CSS list properties allow you to:</p>

<ul>

<li>Set different list item markers for ordered lists</li>

<li>Set different list item markers for unordered lists</li>

<li>Set an image as the list item marker</li>

<li>Add background colors to lists and list items</li>

</ul>

<h3>Different List Item Markers</h3>

<p>The list-style-type property specifies the type of list item marker.</p>

<h3>Image as The List Item Marker</h3>

<p>The list-style-image property specifies an image as the list item marker:</p>

<div class="keyboard-example">list-style-image: url('sqpurple.gif');</div>

<h3>Position The List Item Markers</h3>

<p>The list-style-position property specifies the position of the list-item markers (bullet points).<kbd>list-style-position: outside;</kbd> means

that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically. This is default.

<kbd>list-style-position: inside;</kbd> means that the bullet points will be inside the list item. As it is part of the list item, it will be

part of the text and push the text at the start.

</p>

<h3>Remove Default Settings</h3>

<p>The <kbd>list-style-type:none</kbd> property can also be used to remove the markers/bullets. The list also has default margin and padding. To

remove this, add margin:0 and padding:0 to ul or ol elements.</p>

<h3>List - Shorthand property</h3>

<p>The list-style property is a shorthand property. It is used to set all the list properties in one declaration. When using the shorthand

property, the order of the property values are:</p>

<ol>

<li>list-style-type (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot

be displayed)</li>

<li>list-style-position (specifies whether the list-item markers should appear inside or outside the content flow)</li>

<li>list-style-image (specifies an image as the list item marker)</li>

<li>If one of the property values above are missing, the default value for the missing property will be inserted.</li>

</ol>

<h3>Styling List With Colors</h3>

<p>We can also style lists with colors, to make them look a little more interesting. Anything added to the ol or ul tag, affects the entire list,

while properties added to the li tag will affect the individual list items.</p>

<table>

<caption>All CSS List Properties</caption>

<tr>

<th>PROPERTY</th>

<th>DESCRIPTION</th>

</tr>

<tr>

<td>list-style</td>

<td>Sets all the properties for a list in one declaration</td>

</tr>

<tr>

<td>list-style-image</td>

<td>Specifies an image as the list-item marker</td>

</tr>

<tr>

<td>list-style-position</td>

<td>Specifies the position of the list-item markers (bullet points)</td>

</tr>

<tr>

<td>list-style-type</td>

<td>Specifies the type of list-item marker</td>

</tr>

</table>

</article>

</section>

</main>

<script src="script.js></script>

</body>

</html>

body {

background: #fff;

margin: 0;

font-family: "Open Sans", Helvetica Neue, Helvetica, Arial, sans-serif;

color: #000;

padding-left: 240px;

overflow-x: hidden;

}

.menu,

#navbar {

height: 100vh;

width: 240px;

position: fixed;

top: 0;

left: 0;

z-index: 5;

outline: none;

border-right: solid 1px rgba(0, 0, 0, 0.3);

overflow: scroll;

background-color: #fff;

}

.logo {

font-size: 20px;

color: orangered;

font-family: Georgia, "Times New Roman", Times, serif;

font-weight: light;

margin: 30px;

}

.menu ul {

list-style: none;

padding: 0.5em;

margin: 1em;

line-height: 3;

}

.menu ul a {

text-decoration: none;

color: black;

font-family: Georgia, "Times New Roman", Times, serif;

}

.menu ul li {

padding: 0.15em;

background-position: left 15px center;

background-size: cover;

transition: all 0.15s linear;

cursor: pointer;

}

.menu ul li:hover {

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

border-left: solid 5px orangered;

}

main,

section {

margin: 20px;

}

#main-logo {

color: orangered;

text-align: end;

font-size: 8vh;

font-weight: lighter;

font-family: Georgia, "Times New Roman", Times, serif;

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

border-left: solid 5px orangered;

}

#box-model article img {

max-width: 100%;

}

.h2 {

font-size: 24px;

text-transform: uppercase;

font-weight: bold;

}

section h2,

.h2 {

color: orangered;

text-align: center;

padding: 0;

margin: 0;

font-family: Georgia, "Times New Roman", Times, serif;

}

article > h3 {

color: orangered;

}

.keyboard-example {

margin-left: 20vw;

font-family: monospace;

}

table {

border: solid black 1px;

width: 100%;

border-collapse: collapse;

caption-side: bottom;

}

caption {

color: orangered;

margin-top: 7px;

}

tr:nth-child(even) {

background-color: orangered;

color: #fff;

}

th {

color: orangered;

height: 50px;

}

td {

height: 20px;

}

td,

th {

text-align: left;

border: solid 1px rgba(0, 0, 0, 0.3);

}

@media screen and (max-width: 580px) {

body {

padding-left: 0;

}

#main-logo {

font-size: 30px;

}

.menu {

padding-top: 1em;

width: 100%;

box-shadow: 0 0 0 100em #00000000;

display: none;

transition: all 0.1s ease-in-out;

}

.smartphone-menu-trigger {

width: 40px;

height: 40px;

padding: 0;

margin: 0;

position: fixed;

background: orangered;

transition: all 0.3s ease-in-out;

top: 0;

z-index: 6;

}

.active {

display: block;

}

.smartphone-menu-trigger:before,

.smartphone-menu-trigger:after {

content: "";

width: 50%;

height: 2px;

background: #fff;

border-radius: 10px;

position: absolute;

top: 45%;

left: 50%;

transform: translate3d(-50%, -50%, 0);

}

.smartphone-menu-trigger:after {

top: 55%;

transform: translate3d(-50%, -50%, 0);

}

}