/\*==================== Mostrar Menu ====================\*/

const showMenu = (toggleId, navId) =>{

    const toggle = document.getElementById(toggleId),

    nav = document.getElementById(navId)

    // Valida que as variaveis existem

    if(toggle && nav){

        toggle.addEventListener('click', ()=>{

            nav.classList.toggle('show-menu')

        })

    }

}

showMenu('nav-toggle','nav-menu')

/\*==================== Remover Menu Mobile ====================\*/

const navLink = document.querySelectorAll('.nav\_\_link')

function linkAction(){

    const navMenu = document.getElementById('nav-menu')

    navMenu.classList.remove('show-menu')

}

navLink.forEach(n => n.addEventListener('click', linkAction))

/\*==================== Sessoes de Scroll Link Ativo ====================\*/

const sections = document.querySelectorAll('section[id]')

function scrollActive(){

    const scrollY = window.pageYOffset

    sections.forEach(current =>{

        const sectionHeight = current.offsetHeight

        const sectionTop = current.offsetTop - 50;

        sectionId = current.getAttribute('id')

        if(scrollY > sectionTop && scrollY <= sectionTop + sectionHeight){

            document.querySelector('.nav\_\_menu a[href\*=' + sectionId + ']').classList.add('active-link')

        }else{

            document.querySelector('.nav\_\_menu a[href\*=' + sectionId + ']').classList.remove('active-link')

        }

    })

}

window.addEventListener('scroll', scrollActive)

/\*==================== Mostrar Scroll Top ====================\*/

function scrollTop(){

    const scrollTop = document.getElementById('scroll-top');

    // When the scroll is higher than 560 viewport height, add the show-scroll class to the a tag with the scroll-top class

    if(this.scrollY >= 200) scrollTop.classList.add('show-scroll'); else scrollTop.classList.remove('show-scroll')

}

window.addEventListener('scroll', scrollTop)

/\*==================== Tema escuro ou claro ====================\*/

const themeButton = document.getElementById('theme-button')

const darkTheme = 'dark-theme'

const iconTheme = 'bx-sun'

// Previously selected topic (if user selected)

const selectedTheme = localStorage.getItem('selected-theme')

const selectedIcon = localStorage.getItem('selected-icon')

// We obtain the current theme that the interface has by validating the dark-theme class

const getCurrentTheme = () => document.body.classList.contains(darkTheme) ? 'dark' : 'light'

const getCurrentIcon = () => themeButton.classList.contains(iconTheme) ? 'bx-moon' : 'bx-sun'

// We validate if the user previously chose a topic

if (selectedTheme) {

  // If the validation is fulfilled, we ask what the issue was to know if we activated or deactivated the dark

  document.body.classList[selectedTheme === 'dark' ? 'add' : 'remove'](darkTheme)

  themeButton.classList[selectedIcon === 'bx-moon' ? 'add' : 'remove'](iconTheme)

}

// Activate / deactivate the theme manually with the button

themeButton.addEventListener('click', () => {

    // Add or remove the dark / icon theme

    document.body.classList.toggle(darkTheme)

    themeButton.classList.toggle(iconTheme)

    // We save the theme and the current icon that the user chose

    localStorage.setItem('selected-theme', getCurrentTheme())

    localStorage.setItem('selected-icon', getCurrentIcon())

})

/\*==================== Reduz o tamanho para A4 ====================\*/

function scaleCv(){

    document.body.classList.add('scale-cv')

}

/\*==================== Remover o tamanho quando o curriculo e baixado ====================\*/

function removeScale(){

    document.body.classList.remove('scale-cv')

}

/\*==================== Gerar PDF ====================\*/

// PDF generated area

let areaCV = document.getElementById('area-cv')

let resumeButton = document.getElementById('resume-button')

// Html2pdf options

let opt = {

    margin:       0,

    filename:     'curriculo.pdf',

    image:        { type: 'jpeg', quality: 1.00 },

    html2canvas:  { scale: 4 },

    jsPDF:        { format: 'a4', orientation: 'portrait' }

  }

// Function to call areaCv and Html2Pdf options

function generateResume(){

    html2pdf(areaCV, opt)

}

// Quando o botão é pressionado, irá executar as três funções:

resumeButton.addEventListener('click', () =>{

    // 1. A classe .scale-cv é adicionada ao corpo, que reduz o tamanho dos elementos

    scaleCv()

    // 2. The PDF is generated

    generateResume()

    // 3. The .scale-cv class is removed from the body after 5 seconds to return to normal size.

    setTimeout(removeScale, 2000)

})