JavaScript

Beispiel Code für unterschiedliche fälle

- Order on Resize
- Menu Barrierefrei
- Canvas Draw Lines

Order on Resize

```
const size = "";
const initsize = false;

window.addEventListener("resize", (e) => {

    if(e.currentTarget.innerWidth < 1000 && (size == "notmobile" || initsize == false)) {
        let grids = document.querySelectorAll(".grid-container");

        grids.forEach(element => {
            let images = element.querySelectorAll(".frame-type-image");
            if(images.length > 0){
                let orderfollowup = images.length;
                for(a=0; a<images.length; a++){
                    images[a].parentElement.style.order = a + 1;
                }

                let rest = element.querySelectorAll(".frame:not(.frame-type-image)");
                for(a=0; a<rest.length; a++){
                    orderfollowup = orderfollowup + 1;
                    rest[a].parentElement.style.order = orderfollowup;
                }
            }
        })

        size = "mobile";
        initsize = true;
    }
    if((size == "mobile" || initsize == false) && e.currentTarget.innerWidth >= 1000) {
        let grids = document.querySelectorAll(".grid-container");

        grids.forEach(element => {
            let frames = element.querySelectorAll(".frame");
            for(a=0; a<frames.length; a++){
                frames[a].parentElement.style.order = 0;
            }
        })

        size = "notmobile";
        initsize = true;
    }
});
```

Menu Barrierefrei

```
let subnav = document.querySelectorAll(this.config.elementsNav + ">ul>li");

    subnav.forEach(element => {
        element.addEventListener("mouseenter", (e) => {
            let chnav = element.querySelector("ul");
            if(chnav != null){
                chnav.style.height = chnav.clientHeight + "px";
            }
        })
        element.addEventListener("mouseleave", (e) => {
            let chnav = element.querySelector("ul");
            if(chnav != null){
                chnav.style.height = "";
            }
        })

        element.addEventListener("focusin", (e) => {
            let chnav = element.querySelector("ul");
            if(chnav != null){
                chnav.style.height = chnav.scrollHeight + "px";
            }
        })

        element.addEventListener("focusout", (e) => {
            if(e.relatedTarget != null){
                if(e.relatedTarget.classList != null){
                    if(e.relatedTarget.classList.value == "" || e.relatedTarget.classList.value == "ionav-btn"){
                        let cor = e.relatedTarget.getBoundingClientRect();
                        this.toggleOverlay();
                    }
                }

                if(e.relatedTarget != null){
                    if(e.relatedTarget.parentElement.classList.value != "ionav--children"){
                        if(e.relatedTarget.parentElement.previousSibling != null){
                            if(e.relatedTarget.parentElement.previousSibling.lastChild != null){
                                e.relatedTarget.parentElement.previousSibling.lastChild.style.height = "";
                            }
                        }
                    if(e.relatedTarget.parentElement.nextSibling != null){
                        if(e.relatedTarget.parentElement.nextSibling.lastChild != null){
                            e.relatedTarget.parentElement.nextSibling.lastChild.style.height = "";
                        }
                    }
                    if(e.relatedTarget.previousSibling != null){
                        e.relatedTarget.previousSibling.lastChild.style.height = "";
                    }
                    if(e.relatedTarget.nextSibling != null){
                        if(e.relatedTarget.nextSibling.lastChild != null){
                            e.relatedTarget.nextSibling.lastChild.style.height = "";
                        }
                    }
                }
            }
        }
    })
});

let bb = document.querySelector(".ionav-btn");
bb.tabIndex = 1;
bb.addEventListener("keypress", (e) => {
    if(e.key === "Enter"){
        this.toggleOverlay();
    }
});
```

Canvas Draw Lines

Zeichnet 3 unterschiedliche Linien. Zeichnet bei Veränderung der Bildschirmgröße die Linien neu

```
(function(){

    let joa = document.querySelectorAll('.job-offer-line');
    if(joa != undefined && joa != null){
        joa.forEach((jo) => drawJobOfferLine(jo));
    }

    let ola = document.querySelectorAll('.one-line-line');
    if(ola != undefined && ola != null) {
        ola.forEach((ol) => drawOneLineLine(ol));
    }

    let dia = document.querySelectorAll('.pictureDivider-line');
    if(dia != undefined && dia != null) {
        dia.forEach((di) => drawDividerLineLine(di));
    }

    function drawJobOfferLine(jo) {
        let cvWidth = jo.offsetWidth;
        let fWidth = jo.parentElement.parentElement.offsetWidth;
        let cvHeight = jo.parentElement.offsetHeight;
        jo.width = cvWidth;
        jo.height = cvHeight;
        let cx = jo.getContext('2d');


        cx.lineWidth = 5;
        cx.strokeStyle = '#1B6895';

        cx.beginPath();
        if(fWidth >= 1000){
            cx.moveTo(0, 0);
            cx.lineTo(cvWidth, 0);
            cx.lineTo(cvWidth * 0.75, cvHeight);
            cx.lineTo(0, cvHeight);
        }
        else {
            cx.moveTo(0, 0);
            cx.lineTo(cvWidth * 0.75, 0);
        cx.lineTo(cvWidth * 0.5, cvHeight);
        cx.lineTo(0, cvHeight);
    }
    cx.stroke();
}

function drawOneLineLine(ol){
    let cvWidth = ol.parentElement.offsetWidth;
    let fWidth = ol.parentElement.parentElement.offsetWidth;
    let cvHeight = ol.parentElement.offsetHeight;
    ol.width = fWidth;
    ol.height = cvHeight;
    let cx = ol.getContext('2d');

    cx.clearRect(0, 0, ol.width, ol.height);

    cx.lineWidth = 5;
    cx.strokeStyle = '#1B6895';

    cx.beginPath();
    if(fWidth >= 1000){
        cx.moveTo(fWidth, 0);
        cx.lineTo(cvWidth * 1.85, 0);
        cx.lineTo(cvWidth * 1.6, cvHeight);
        cx.lineTo(0, cvHeight);
    }
    else {
        cx.moveTo(fWidth, 0);
        cx.lineTo(cvWidth * 0.75, 0);
        cx.lineTo(cvWidth * 0.5, cvHeight);
        cx.lineTo(0, cvHeight);
    }
    cx.stroke();
}

function drawDividerLineLine(di){
    let cvWidth = di.parentElement.offsetWidth;
    let cvHeight = di.parentElement.offsetWidth * 0.1;
    di.width = cvWidth;
    di.height = cvHeight;
    let cx = di.getContext('2d');


    cx.lineWidth = 5;
    cx.strokeStyle = '#1B6895';
```

```
        cx.beginPath();
        cx.moveTo(cvWidth, 0);
        cx.lineTo(cvWidth * 0.66 - 10, 0);
        cx.lineTo(cvWidth * 0.33 - 10, cvHeight - 10);
        cx.lineTo(0, cvHeight - 10);
        cx.stroke();

        cx.beginPath();
        cx.moveTo(cvWidth, 10);
        cx.lineTo(cvWidth * 0.66, 10);
        cx.lineTo(cvWidth * 0.33, cvHeight);
        cx.lineTo(0, cvHeight);
        cx.stroke();
    }

    window.addEventListener("resize", (e) => {
        joa = document.querySelectorAll('.job-offer-line');
        if(joa != undefined && joa != null) {
            joa.forEach((jo) => drawJobOfferLine(jo));
        }

        ola = document.querySelectorAll('.one-line-line');
        if(ola != undefined && ola != null) {
            ola.forEach((ol) => drawOneLineLine(ol));
        }

        dia = document.querySelectorAll('.pictureDivider-line');
        if(dia != undefined && dia != null) {
            dia.forEach((di) => drawDividerLineLine(di));
        }
    });
})()
```