

# UX-design

## Laboration 1

# Laboration 1

- Laborationen finns på PingPong
- Deadline är fredag 1 februari kl 08:59
- Laborationen lämnas in genom att pusha koden till ett git-repo
- Inlämningen ska bestå av tre filer: mds.css, mds.js och demo.html

# Komponentbibliotek

- Laborationen går ut på att skapa ett (vieldigt litet) komponentbibliotek
- Exempel på komponentbibliotek är Bootstrap, Material Components, Ionic, etc.
- Vi ska inte använda någon kod från några andra bibliotek, utan komponenter ska skrivas från grunden
- I “mds.css” ska komponenterna implementeras med BEM
- “mds.js” ska definiera ett objekt med metoder för att initiera komponenterna. I vissa fall behövs ingen JavaScript-kod och i så fall kan metoderna lämnas tomma
- “demo.html” ska vara en sida som visar hur alla komponenterna används

# BEM

- Det har dykt upp en del frågor om BEM så vi går genom det lite grundligare
- BEM är en kodstandard för att skriva komponenter i CSS
- BEM ger oss en standard för hur vi namnger klasser och hur vi skriver CSS selectors
- <http://getbem.com/naming/>
- BEM står för Block-Element-Modifier

# Block

- Ett block motsvarar en återanvändbar komponent
- Klassnamnet får innehålla bokstäver (a-z), siffror och bindestreck (-)
- Alla HTML-element kan vara ett block om de accepterar ett klassnamn

Exempelvis om vi vill skapa en Floating Action Button från Material Design kan vi skapa ett block som heter “fab”

## HTML

```
<button class="fab">...</button>
```

## CSS

```
.fab {  
}
```

# Block

- En CSS selector för ett block får inte vara beroende av andra element/block på sidan
- Inga tag selectors
- Inga ID selectors

# Element

- “Element” är delar av block som inte har någon innebörd var för sig
- Klassnamnet är blocknamnet plus två understreck plus elementnamnet. T ex **“floating-button\_icon”**
- I många fall består en komponent av många olika “element”
- En CSS selector för ett element får inte vara beroende av andra element/block på sidan
- Inga tag selectors
- Inga ID selectors

# Exempel - Element

## HTML

```
<button class="fab">
  <i class="material-icons fab__icon">add</i>
</button>
```

## CSS

```
.fab {
  ...
}

.fab__icon {
  ...
}
```



# Använd inte descendant selectors

- En CSS-regel för ett “element” ska inte definieras som en descendent selector
- Eftersom vi redan kan se vilket block den tillhör genom klassnamnet är en descendant selector överflödig

Följande CSS är inte korrekt användning av BEM

```
.fab .fab__icon {  
  ...  
}
```

# Modifier

- En “modifier” är en flagga för ett element eller ett block som ändrar dess utseende, funktion eller tillstånd
- Klassnamnet för en modifier är blocknamnet eller elementnamnet plus två bindestreck (-) plus namnet på modiern.
- En modifier för ett block blir “**block--modifier**” och en modifier för ett element blir “**block\_elem--modifier**”
- T ex om vi vill göra en Mini FAB kan vi kalla modiern “**mini**” och klassnamnet blir “**fab--mini**”.

# Exempel

<https://codepen.io/argelius/pen/yGWPJE>

Skuggorna är tagna från

<https://gist.github.com/serglo/f9f0be9a66fd6755a0bda85f9c64e85f>

# Text field

- I laborationen ska ni implementera fyra olika komponenter från Material Design med BEM och skriva JavaScript för komponenterna i de fallen där det behövs (funktionalitet som inte går att implementera med CSS)
- En av komponenterna är en “Text field”
- Material Design definierar två varianter: “filled” och “outlined”. Ni behöver bara implementera “filled”
- Ni behöver inte implementera ikoner
- Ni behöver inte heller implementera felmeddelande och helper text

# Text field

<https://material.io/design/components/text-fields.html#filled-text-field>

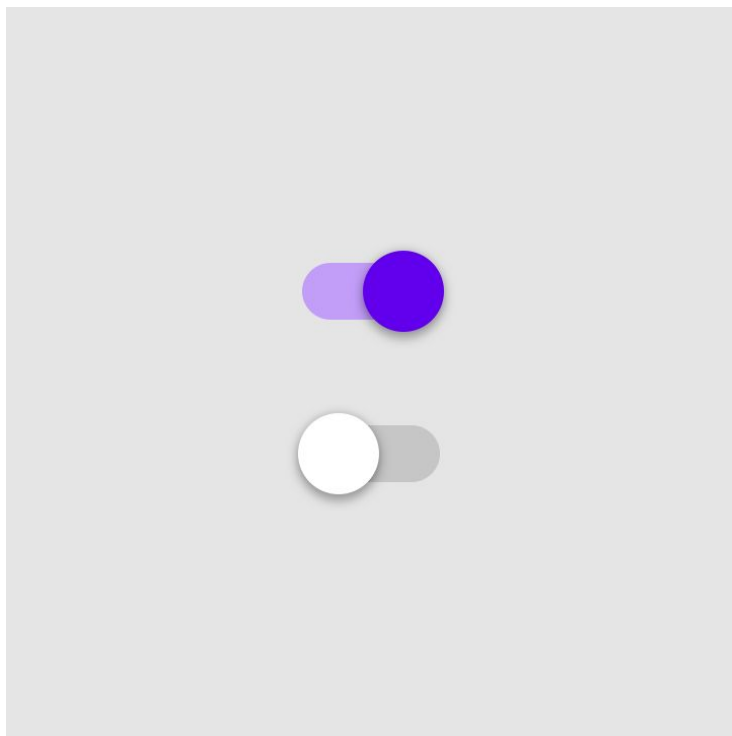
Följande kommer bedömas

- Om komponenten stämmer överens med specifikationen (textstorlek, mått, position)
- Att label-texten uppför sig korrekt
- Att linjen under fältet uppför sig korrekt
- Att animationer används för label och linjen under fältet

Följande kommer ej bedömas

- Färger
- Hur väl animationerna stämmer med specifikationen

# Switch



# Switch

<https://material.io/design/components/selection-controls.html#switches>

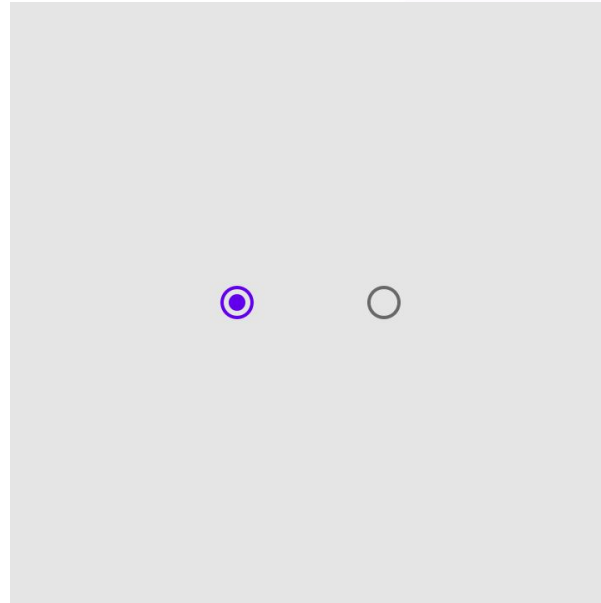
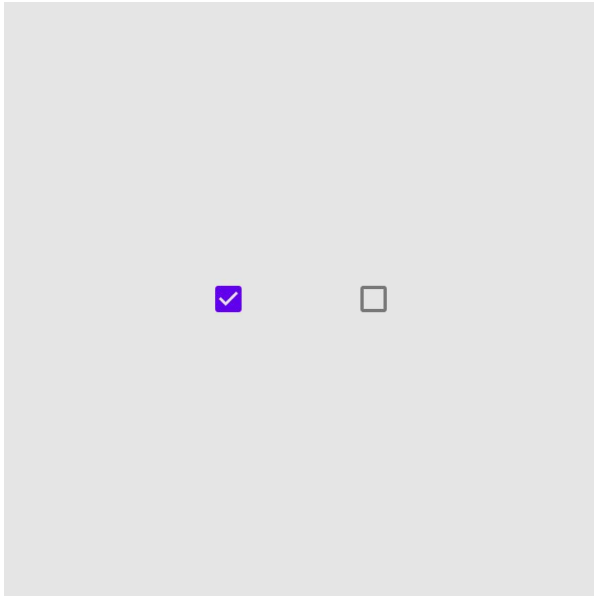
Följande kommer bedömas

- Om komponenten stämmer överens med specifikationen (textstorlek, mått, position)
- Att animationer används när komponenten slås på eller av

Följande kommer ej bedömas

- Färger
- Hur väl animationerna stämmer med specifikationen

# Checkbox och radio button





# Checkbox och radio button

<https://material.io/design/components/selection-controls.html#checkboxes>

<https://material.io/design/components/selection-controls.html#radio-buttons>

Följande kommer bedömas

- Om komponenten stämmer överens med specifikationen (textstorlek, mått, position)

Följande kommer ej bedömas

- Färger

# CSS-bibliotek

- Den största delen av inlämningen är att skapa ett CSS-bibliotek
- Filnamnet ska vara “mds.js”
- Klassnamnen för alla blocken är definierad i PDF:en
- Er uppgift är att skapa en struktur för varje komponent och definiera CSS-regler med BEM
- Ni ska inte använda några externa bibliotek utan “mds.css” ska kunna användas utan att ladda in några andra CSS-filer

# JavaScript-bibliotek

- Ni behöver även skapa ett JavaScript-bibliotek
- Filnamnet ska vara “mds.js”
- Filen ska definiera en global variabel som heter “mds”
- Ni behöver inte exportera objektet på något annat vis
- Ni behöver definiera en metod för varje komponent
- Namnen är definierade i PDF:en
- Ni kan använda jQuery om ni vill

*I den här uppgiften behöver ni antagligen bara skriva JavaScript för “Text field”. De andra komponenterna behöver ingen JavaScript-kod för att fungera.*

*Ni kan ändå definiera tomma funktioner för de övriga komponenterna. Om vi tänker oss att vi vill lägga till funktionalitet i framtiden kan det vara bra att användarna kör funktionen så de slipper ändra sin kod när vi släpper en ny version.*

# demo.html

- Ni behöver även skapa en demo-sida som har filnamnet “demo.html”
- Sidan ska importera “mds.css” och “mds.js” och visa hur alla komponenterna används
- HTML-strukturen kommer vara beroende av hur ni definierat era CSS-klasser

# Exempel

- Här följer ett enkelt exempel på hur “Text field” skulle kunna användas
- Jag har inte definierat några inre element eftersom det beror på hur ni strukturerar er CSS

```
<div class="mds-text-field">  
  ...  
</div>
```

```
<script>  
  mds.textField(document.querySelector('.mds-text-field'));  
</script>
```

# Sammanfattning

- Uppgiften är att skapa ett bibliotek som implementerar en liten del av Material Design-specifikationen
- Komponenterna kommer att bedömas baserat på hur väl de stämmer överens med specifikationen
- Komponenterna behöver inte vara perfekta men om de skiljer väldigt mycket kan ni behöva komplettera
- Användning av färger kommer ej bedömas
- Om ni är osäkra på hur komponenterna ska se ut kan ni jämföra med Material Components for Web

<https://material-components.github.io/material-components-web-catalog/#/component/text-field>