

myPT by av Robin Justice Akyüz

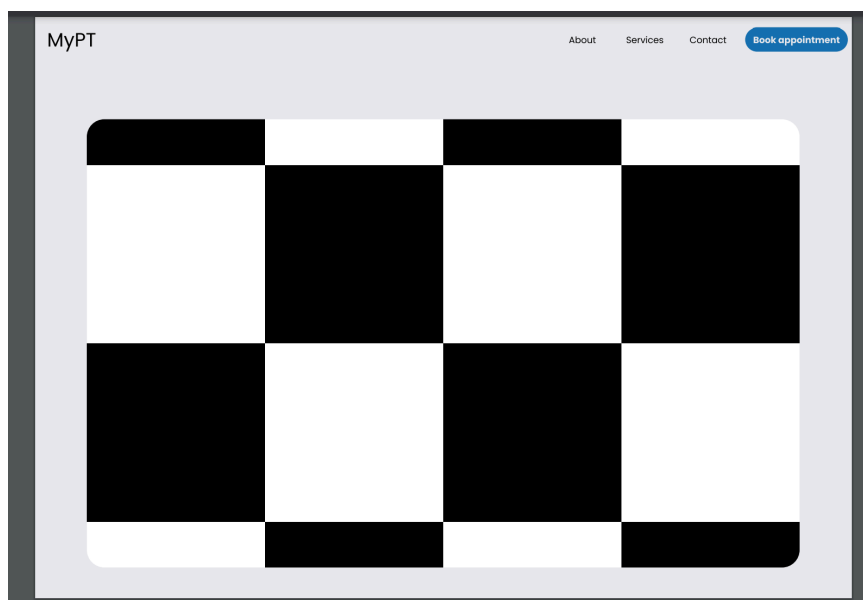
Sammanfattning

Mitt program är en kombination av html och css för att skapa en fin layout till min hemsida för personlig träning. Javascript och jQuery är implementerat vid de interaktiva stegen, där vi klickar på knappar, hoverar över fält osv. Hemsidan ger användaren möjlighet att kontakta mig för att utveckla sig själv och sin hälsa. Det kan gälla maten de äter i form av diet, hur och när de ska träna. Målet är att få en ständig kontakt mellan kund och tränare, där jag ska ge rätt boost för att få kunden att stå på sina egna ben, styrkt med vetenskap.

Framtagande

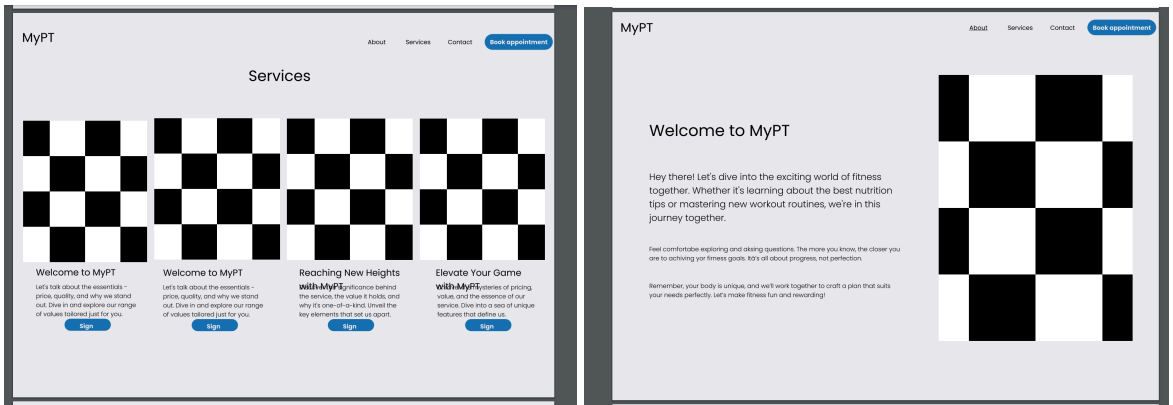
Jag började med att tänka på att göra ett projekt som skulle kunna gynna mig i framtiden, både om jag vill fullfölja projektet i framtiden, eller får generell kunskap om hur man skapar hemsidor.

Det första jag gjorde var att skapa projektet med hjälp av verktyget FIGMA, som jag blivit vid från föregående kurser. Jag tänkte att det var ett enkelt sätt att skissa mina idéer som jag har i huvudet till något konkret. Första idén var att rita på papper, men de kunskaperna har jag tyvärr inte. Här är första "ramen" i figma, där jag har följt ett standard business-template för att skapa layouten.



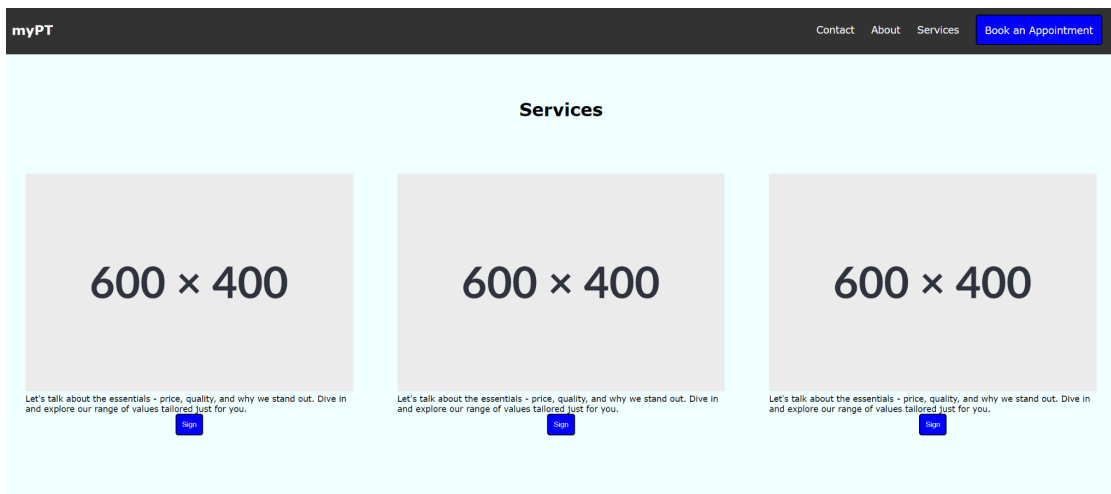
Det ingick att ändra menyerna till det jag önskar, exempelvis att knapparna ska ta en till den relevanta sidan som att boka och ta i kontakt. På "service" delen hade jag endast 2 tjänster, vilka var för att skapa muskler och gå ned i vikt.

Vidare kom jag på att det troligen finns många fler tjänster vi kan lägga till här, så att vi potentiellt får flera kunder. Även i bilden nedan har vi inte bestämt oss för färgerna, utan endast layouten - texterna är också in klistrade som placeholders. Titta bilaga nedanför:

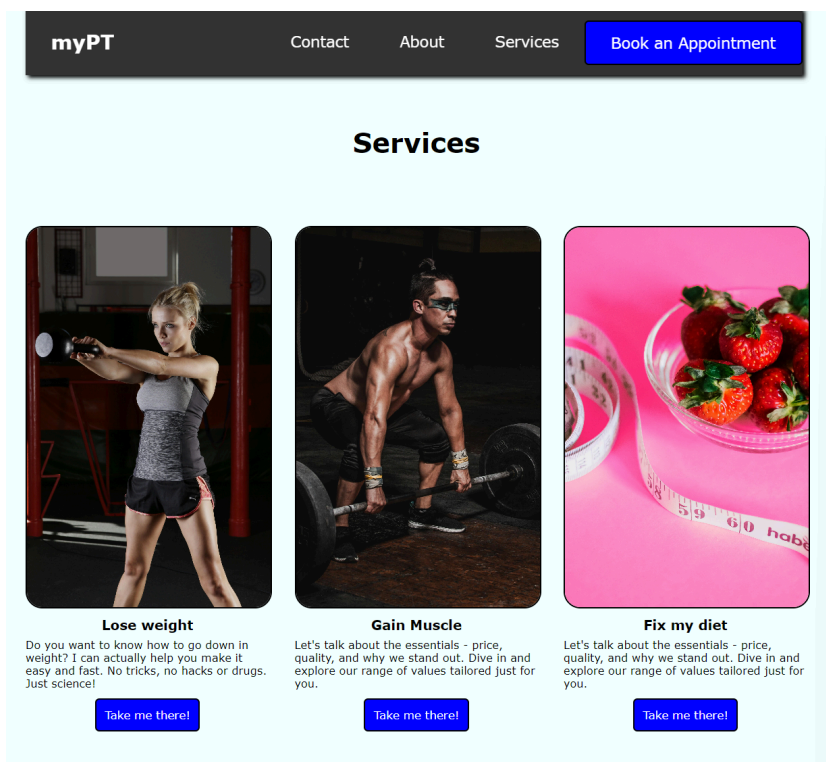


The image shows a wireframe of a contact form titled 'Get in Touch with your trainer'. The form includes fields for 'Name (required)' (split into 'First Name' and 'Last Name'), 'Email (required)', and 'Message (required)'. There is a 'Send' button at the bottom. To the left of the form, there is a short paragraph of text and contact information for 'Justiceakyuz@gmail.com' and 'Kabelverksgatan 32, 125 48 Årsvå, Stockholm'.

Bilden ovan representerar vi kontakt framen som jag kom på sist, vilket nu i efterhand är det viktigaste på hemsidan, här ska kunden kunna skriva ett meddelande till mig om vad jag kan hjälpa dem med. Denna förväntar mig att jag inte ska vara lika snygg när hemsidan är implementerad med programmeringen.



Bilden ovan var den första iterationen av hemsidan vilket tog en otroligt lång tid att fixa. Först var problemet med att få navbaren att ordna så att loggan var till vänster och meny artiklarna var till höger.

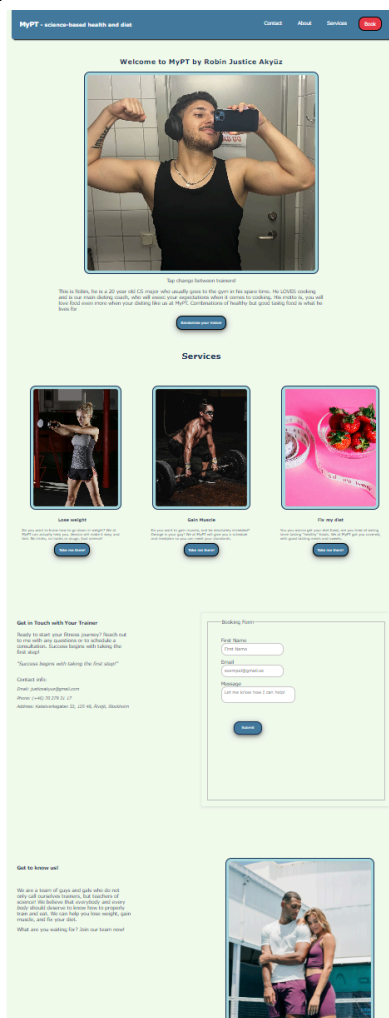


Nu har vi kommit en lång bit med services-delen som visas ovan. Det vi gjort i dessa steg är att fixat css för flexboxen samt bytt ut placeholders till riktiga bilder från hemsidan Pexels. Det största problemet jag stod på här var att bilderna hade olika storlekar. Hur jag löste problemet var att bestämma förälder elementet och sedan använda mig av object-fit.

Form

2 sidor eller mer, beskriv hur koden i det färdiga programmet är uppbyggd och fungerar, ta med exempel på centrala kod-avsnitt samt beskriv hur helheten fungerar med pseudokod. (text och bild).

Min hemsida MyPT, är uppbyggd med standard taggarna i HTML - <head>, <body>, <footer>. Innut i body har jag en <main> tag som innehåller flera avsnitt i sina egna <article> taggar. I varje sådan finns sina egna taggar, som beskriver ett ämne eller funktikon. Hemsidan är indelad i artiklarna under sektionerna: home, about , service och booking. Dessa inledningar finns på navigationsbaren att trycka på. Hemsidan är en och samma långa page, men för att visa min kunskap så delade jag istället upp "contact" till sin egna page istället. Här har jag zoomat ut hemsidan och tagit en skärmdump för att visa den långa hemsidan jämfört med nya sidan som poppar upp av att man trycker på contact.



```
<body>

  <main>

    <article id="navArt">

      <nav class="navbar">

        <div class="logo">MyPT <small>- science-based health and diet </small></div>
        <ul class="navul">

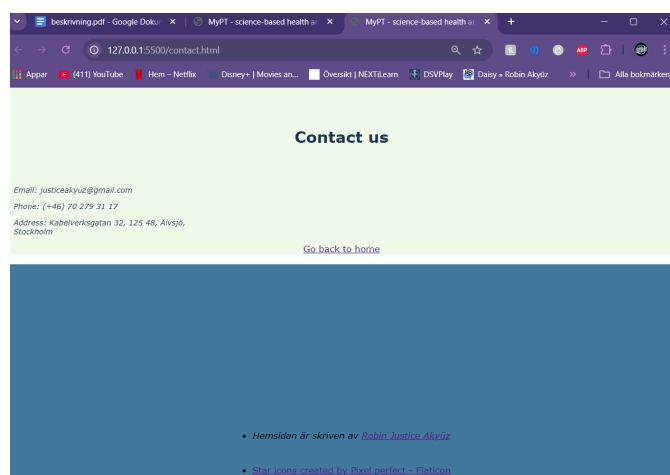
          <li><a href="/contact.html">Contact</a></li>
          <li><a href="#about">About</a></li>
          <li><a href="#services">Services</a></li>
          <li><a href="#book" id="bookBtn">Book</a></li>

        </ul>
      </nav>

    </article>

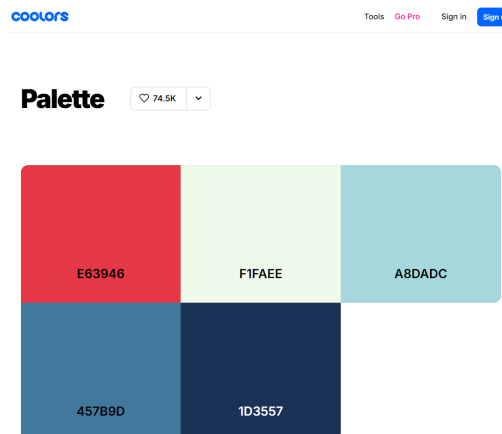
  </main>

</body>
```



Koden visar skillnaderna i navigationen av länkarna -> "/" istället för "#".

Färgerna tog jag från en palette hemsida, colors.co. vilket jag tyckte passade för en träningshemsida, samt färger som jag tycker om!



Det jag spenderat mest tid på är home pagen samt navbaren, där jag har en bild inuti en picture tag på mig. Denna har vidare funktion med JS / jQuery som jag ska prata vidare i nästa kapitel - funktion. Sedan finns det två paragraph texten nedanför som har sina egna id, och meningen är att de ska ändras på med funktioner.

Vidare kan vi diskutera alla sektioner av hemsidan, som finns under artiklarna som jag nämnde innan. "Service"-delen innehåller tre bilder och knappar som ska vara mina produkter eller tjänster jag säljer. De har hover funktion, med css som skuggor, borders etc. Samma gäller knapparna, dock blir de röda när man hovrar över. Det är meningen att i framtiden ska knapparna leda till sina egna pages med mer info etc, men det kände jag inte var relevant att fortsätta med pga av skalan av detta projekt och tiden jag har.

```
.flexContainer {
  display: flex;
  align-items: flex-start;
  justify-content: space-between;
  gap: 30px;
  color: #1D3557;
}

.flexContainer figure {
  display: flex;
  flex-direction: column;
  align-items: center;
  flex: 1;
}

.flexContainer h4 {
  text-align: center;
  font-size: medium;
}

.flexContainer img {
  width: 100%;
  height: 450px;
  object-fit: cover;
  max-width: 350px;
}

.flexContainer figcaption {
  margin-top: 5px;
  margin-bottom: 10px;
  font-size: small;
  text-align: left;
  flex-shrink: 0;
  color: #1D3557;
}

<article>
  <h2>Welcome to MyPT by Robin Justice Akyüz</h2>
</article>

<picture id="pictureHolder">
  
</picture>

<p id="promptText"> Tap change between trainers!</p>

<p id="introText">This is Robin, he is a 20 year old CS major who usually goes to the gym in his spare time. He LOVES cooking and
  <button class="signBtn" id="randomBtn">Randomize your trainer</button>
</p>
</article>

<article id="services">
  <h2>Services</h2>

  <div class = "flexContainer">
    <figure>
      <picture>
        
      </picture>
      <figcaption>
        <h4>Lose weight</h4>
        Do you want to know how to go down in weight? We at MyPT can actually help you. Jessica will make it
        easy and fast. No tricks, no hacks or drugs. Just science!
      </figcaption>
      <button class="signBtn">Take me there!</button>
    </figure>
    <figure>
      <picture>
        
      </picture>
      <figcaption>
        <h4>Gain Muscle</h4>
        Do you want to gain muscle, and be absolutely shredded? George is your guy! We at MyPT will give you a schedule and
      </figcaption>
      <button class="signBtn">Take me there!</button>
    </figure>
  </div>
</article>
```

Koden är inte speciellt avancerad på bilderna ovanför, som beskriver en del av service artikeln. Det jag gjorde var att skapa en container för bilderna gav den flex attributet så att de fick samma avstånd, och sedan kopierade över samma kod från första <figure> taggen och allt innehåll på tre. Sedan är det bara att ändra koden, bilden och funktionen till rätt innehåll - snabb och effektiv programmering. Inspekterar man med hjälp av f12 ser man en vitbubbla som visar (flex), vilket betyder att jag använt mig av den layout metoden. I den första övningen 1&2 använde vi oss av grid, men nu ville jag testa och lära mig flex.

Booking delen är en div med namnet "textContainer" som har en vänster och höger <aside>.* Dessa är också indelade med flexbox. Till vänster har vi asiden som heter "left", och innehåller en paragraph med text, asiden till höger "right", innehåller taggen <form> som bygger upp ett formulär för kunden att kontakta mig på. Koden nedan, visar hur jag har strukturerar formuläret samt delat upp aside till vänster och höger.

```
<article id="book">
  <h2>Booking</h2>
  <div class="textContainer">
    <aside id="left">
      <h3>Get In Touch with Your Trainer</h3>
      <p id="ready1st">Ready to start your fitness journey? Reach out to me with any questions or to schedule a consultation</p>
      <p id="Success">Success begins with taking the first step!</p>
      <label>Contact Info:</label>
      <address id="address">
        <p>Email: justinabryd@gmail.com</p>
        <p>Phone: (+46) 70 279 31 17</p>
        <p>Address: Kabelverksgatan 32, 125 48, Älvsjö, Stockholm</p>
      </address>
    </aside>
    <aside id="right">
      <form>
        <fieldset>
          <legend>Booking Form</legend>
          <label for="firstName">First Name</label>
          <input type="text" id="name" placeholder="First Name" autofocus required pattern="[A-Za-z0-9]{2,}">
          <label for="email">Email</label>
          <input class="nameInput" type="email" id="email" name="email" placeholder="example@gmail.se">
          <label>Message</label>
          <textarea id="message" placeholder="let me know how I can help!" required>
          <button id="submitBtn" type="submit" value="Submit">
        </fieldset>
      </form>
    </aside>
  </div>
</article>
```

*Boken nämnd att div taggen är utdaterad och att man ska använda article, men detta visar jag redan under uppgiften. Jag fick information om hur man gjorde denna del från w3school, som använde sig av div istället.

About sidan är liknande till föregående del, då jag härigenom kopierat och klistrat koden till en vänster och högerled och sedan ändrat genomgående, för att effektivisera processen. Sist är en enkel bild på fem stjärnor med en review-text.

Koden har sett ungefär likadan under processen, då html layouten var gjort först med css och sedan javascriptet. Detta har lett till att det som har blivit ändrat är javascript koden. Det första exemplet var funktionen changePicture() som först var en hel metod inuti jquery click eventet på "#robinImg". Detta gjorde att

funktionen inte kunde användes utanför klickeventet. Som lösning bröt jag ut koden så att klicket istället anropar funktionen `changePicture`. Detta gjorde att jag kunde göra en till funktion som anropar samma funktion, på annat sätt en just att klicka på bilden. Detta blev knappen "randomBtn" som orsakar val av tränare slumpmässigt. Detta finns även kommenterat i smått på rad 85.

Funktion

Låt oss gå igenom JS och jQuery delen av mitt projekt. Som jag nämnde innan är homepagen med bilden på mig och paragrafen det viktigaste i pagen. Bilden går att trycka på, och visar en annan bild istället, samt att undertexten ändras i relevans med bilden. Den är uppbyggt med jQuery med en array som innehåller sökvägen till de tre bilderna. Se nedan:

```
$(document).ready(function(){  
  
    const carouselImg = ["bilder/robin_pt.jpg", "bilder/pexels-jessica.jpg", "bilder/pexels-george.jpg"];  
  
    let counter = 1;  
  
    let robinTxt = "This is Robin, he is a 20 year old CS major who usually goes to the gym in his spare time. He LOVES cooking and is our main c  
    let jessicaTxt = "This is Jessica, she is a 35 year old mom and trainer. She is the head coach of losing weight, and will help you tailor you  
    let georgeTxt = "This is George, he is a 29 year old science based weight lifter and is our 'Master of Weights' and can teach you how to lif
```

När man trycker på bilden anropas funktionen `changePictures()`; igång, det ser vi här:

```
//Vid klick på bildhållaren går den igenom arrayen och ändrar src attributet till bilderna i arrayen.  
$("#robinImg").click(function(){  
  
    changePicture(false);  
  
});
```

Denna funktion tar in en `true` eller `false` som parameter och ett par `if`-satser. I `if`-satserna är "condition":et vilket nummer variabeln `counter` som tidigare är initierad. Funktionen ändrar på `src` från bilden på mig, till den bilden som finns i arrayen på index av `counter`. Här matchas även paragraf texten till, med jquery funktionen `$("#introText").html(variabel)`; där variabel är antingen, `robinTxt`, `JessicaTxt` eller `georgetxt` -igen - beroende på index av arrayen och `counter`.

```

//Vid klick på bildhållaren går den igenom arrayen och ändrar src attributet till bilderna i arrayen.
$("#robinImg").click(function(){

    changePicture(false);

});

//funktionen ändrar bilden, beroende om parametern är true eller false.
function changePicture(randomTrainer){

    if(randomTrainer == true){

        counter = Math.floor(Math.random() * 3);

    }

    $("#robinImg").attr("src", carouselImg[counter]);
    counter++;

    if(counter >= carouselImg.length){
        counter = 0;
    }

    if(counter == 1){
        $("#introText").html(robinTxt);
    }
    else if (counter == 2){
        $("#introText").html(jessicaTxt);
    }else{
        $("#introText").html(georgeTxt);
    }

    $("#robinImg").animate({
        height: '+=30px',
        width: '+=100px'
    },500 );

    $("#robinImg").animate({
        height: '-=30px',
        width: '-=100px'
    },700);

}

```

Den andra funktionen som också anropar changePicture() med hjälp av jQuery:

```

$("#randomBtn").click(function() {

```

```

    changePicture(true);
});

```

Denna metod har med knappen som finns under "#robinIMG", i html koden med namn "#randomBtn". Denna knapp anropar changePicture() med parametern true. Går vi igenom funktionen changePicture() igen ser vi att det finns en if sats som tar in booleanen randomTrainer som parameter. Denna kollar i if satsen:

```

if(randomTrainer == true){

```

```

    counter = Math.floor(Math.random() * 3);

```

```

}

```

Så om randomTrainer är true, tar den en random siffra mellan 0-2 och tilldelar variabeln counter den siffran. Counter får ett nytt värde med if -sats och fortsätter metoden som vanligt.

Skilnaden på funktionerna om man trycker på bilden jämfört med knappen är att ena parametern är false och andra true. Eftersom boolean var false i randomTrainer på första metoden så hoppar den över steget med Math.random();

Jag har lagt två animate event när changePicture() metoden anropas, för att öka css höjden och bredden med x-antal pixlar och y-antal pixlar för höjd. Detta skapar en animation att bilden blir större, men sedan vill jag att den ska gå tillbaka till ursprungsläget, och minskar bilden med samma antal pixlar igen, med liknande funktion. Bilden med funktionen changePicture() visar detta. Sist i dessa animate-funktioner finns det också en siffra -> 500 och 700. Som är en parameter på hur många millisekunder som det tar för animationen att visas.

Jag har två mus-event för submit knappen på mitt formulär för mouseenter och mouseleave, där den använder effekt animationerna transform med inparametrarna 1.2 på och transition med inparametrarna 0.6 sekunder. Ena står för hur större bilden ska bli i skalan, och den andra tiden det ska ta. mouse event funktionen gör samma sak fast inverterat, alltså att den går tillbaka till skalan 1 med samma tid.

Samma knapp, "#submitBtn" har en funktion som har en if-sats där den tittar om inputfälten namn, mejladress och message INTE är tomma med OR operationen. Om if-satsens condition stämmer, alltså om någon av de är tomma, ges en alert. Om if-satsens condition är false, går den vidare till else-satsen och då anropas metoden confirmFunc(). Det är en funktion som skriver ut value:n i confirm fönstret.

```
});  
  
//jag bröt ut funktionen så att click på submitbtn anropar denna function -> mer läsbart + återanvända funktionen i framtiden  
function confirmFunc(){  
    confirm(  
        "Name: " + $("#name").val() + "\n" +  
        "E-mail: " + $("#mejladress").val() + "\n" +  
        "Message: " + $("#message").val()  
    );  
}  
  
//anropar changpicture funktionene på knappen med en randomizer.  
$("#randomBtn").click(function() {  
    changePicture(true);  
});  
  
});
```