

# Sten-sax-påse

HTML, CSS och Javascript

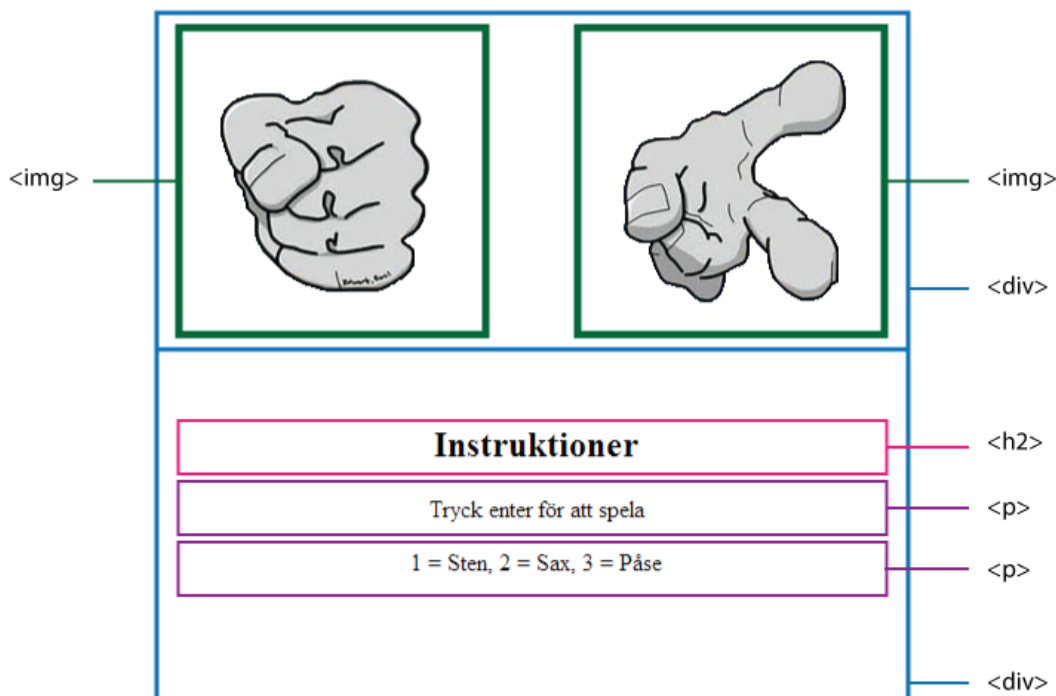
## Krav

Chrome som webbläsare

Om du inte har kodat en hemsida förut så kan det bli mycket på en gång. Men när du är klar kommer du veta hur man skriver kod för att göra en hemsida, lärt dig hur man kan göra en hemsida 100 gånger snyggare med hjälp av CSS och gjort ditt första spel! Hur grymt som helst!

## HTML

Så här vill vi att hemsidan ska se vara uppbyggd. Två bilder bredvid varandra och lite text som förklarar hur man spelar spelet. För att kunna göra det måste vi använda oss av 4 olika "taggar". Vi har bara skrivit ut taggnamnet i ritningen för att vi lättare ska kunna få en överblick.



## Hur använder man taggar?

För att börja en tagg så skriver man bara: `< namnet på taggen >`

Sen när man vill avsluta en tagg skriver man: `</ namnet på taggen >`

Så här används våra taggar:

- `<div>` En div är som en box som ramar in allt som är i den. I en div kan man ha andra taggar. `</div>`
- `<h2>` Skriv texten du vill ska synas mellan taggarna så här! h2 står för header 2. Alltså Rubrik nivå 2. Vill man ha en större rubrik kan man använda h1 och vill man ha en mindre rubrik använder man h3. `</h2>`
- `<p>` Precis som h2-taggen är det bara skriva det man vill ska synas på skärmen här. `</p>`

Sedan kan det finnas special fall som till exempel `<img>`. För att använda `<img>` måste man ge den en bild att skriva ut. Det gör man så här:

```

```

`<img>` är ett undantag för regeln med att alla taggar måste avslutas med `</ namnet på taggen >`.

## Håll koll på din kod!

För att ha nå innehållet i de olika taggarna kan man ge dem namn i form av klasser och id.

- `id="Vad du vill att den ska heta"`.  
Ett id måste vara unikt! Ingen annan får heta samma sak.
- `class="Namnet på klassen den tillhör"`.  
Flera taggar kan ha samma class. Om man ändrar något i klassen kommer det att ändras på alla som är med i samma class.

## CSS

CSS använder man för att få sin sida att se ut precis som man vill. Vi ska bara ändra så att texten är i mitten och att det blir lagom med mellanrum mellan allt.

För att snygga till en tagg som är medlem i en class skriver man:

```
.namnetpåklassen{  
  
}
```

(Notera att det står PUNKTnamnetpåklassen!)

För att snygga till en tagg som har ett id skriver man:

```
#idnamnet{  
  
}
```

Man kan även ändra på alla med samma taggnamn så här:

```
taggnamnet{  
  
}
```

# Javascript

Det är här alla funktioner till spelet skapas! För att skapa en funktion i javascript skriver man:

```
function vadfunktionenheter() {  
    /* Allt din funktion ska göra */  
}
```

Vi ska totalt skapa 5 stycken funktioner.

## change\_image\_player2()

För att ha vår motspelare ska ändra val hela tiden behöver vi funktionen start\_stop\_player2().

Vi börjar med att hitta player2s bild genom att skriva:

```
var player2_image = document.getElementById("player2");
```

För att ändra bilden till stenhanden skriver vi:

```
player2_image.src = "Sten.jpg";
```

Men vi vill att den ska kunna vara sten, sax eller påse. För att göra det måste vi räkna ut en variabel som ändras varje gång funktionen ropas på och som är antingen 1, 2 eller 3.

Det gör vi genom att skriva:

```
id = id % 3 + 1;
```

% är superbra att kunna. Den heter modulus och är en väldigt konstig men smart sak.

Om vi skriver:

```
id = 1%3  
alert(id)
```

Kommer att skriva ut: 1

```
id = 4%3  
alert(id)
```

Kommer också att skriva ut en: 1

```
id = 5%3  
alert(id)
```

Kommer att skriva ut: 2

Det är för att modulus försöker trycka in 3 så många gånger i siffran framför som möjligt och får den inte plats skriver den ut det som är kvar. Konstigt men väldigt nyttigt för oss!

Men hur vet den vad id är varje gång? Jo, för att hålla koll på vad id är måste vi skapa en global variabel. Det gör vi genom att skriva:

```
var id = 0;
```

utanför funktionen. Då kommer alla funktioner kunna se vad den är hela tiden. = 0 skriver vi bara för att vi vill att den ska vara 0 ifrån början.

För att välja en annan bild varje gång använder vi oss av if - else if- else funktioner. Så här:

```
if( id == 1 ){
    player2_image.src = "Sten.jpg";
}
else if(id == 2){
    player2_image.src = "Sax.jpg";
}
else {
    player2_image.src = "Pase.jpg";
}
```

Så där! Vår första funktion är klar. Men för att hela tiden ropa på den måste vi skapa en till funktion. Vi döper den till start\_stop\_player2();

### **start\_stop\_player2()**

Det är väldigt enkelt att ropa på en funktion flera gånger med ett tidsintervall men det är också lite farligt. För om vi skulle ropa på samma funktion flera gånger utan att kunna stänga av den måste man stänga ner vår fina sida för att få den att sluta. Det vill vi inte. Därför måste vi spara ett id när vi startar funktionen. Som tur är är det väldigt enkelt!

Skapa en global variabel högst upp i filen som heter stop\_id. Sen kan vi använda oss av:

```
stop_id = setInterval(change_image_player2, 100);
```

Testa och se vad som händer när du ropar på funktionen från body i html. Skriv så här:

```
<body onload="start_stop_player2();">
```

Skulle vi vilja stänga av så den inte kallar på change\_image\_player2 hela tiden skriver vi bara:

```
clearInterval(stop_id);
```

För att vi lätt ska kunna ropa på samma funktion och starta och stoppa player2 behöver vi använda oss av ett if-else statement och booleans. Booleans kan ha värdet sant eller falskt (true/false) och är väldigt bra att använda om man vill gå in i if varannan gång och else varannan gång. Vi skriver så här:

```
if(player2_stopped == true){
```

```

        stop_id = setInterval(change_image_player2, 100);
        player2_stopped = false;

    } else {
        clearInterval(stop_id);
        player2_stopped = true;
    }
}

```

För att det här ska funka måste vi skapa en ny global variabel som heter `player2_stopped`.

## input()

Den här funktionen gör så att vi kan läsa av vilken tangent som någon trycker på.

För att det ska fungera måste vi först skriva om `<body>` i .html-filen till:

```
<body onKeyPress="input();" >
```

Det kommer göra så att vår funktion kallas på varje gång någon trycker på en tangent!

Nu tillbaka till javascript. För att få tangenten skriver vi:

```
x = event.keyCode;
```

Vi kan lätt testa vilken tangent som tryckts på genom att skriva:

```
alert(x);
```

Hitta vilket nummer tangent 1,2,3 och enter har! Vi behöver dom när vi ska bestämma vilken bild du har valt när du spelar och för att starta/stoppa spelet.

Om vi trycker på tangent 1,2 eller 3 vill vi:

- Stoppa spelare 2
- Välja rätt bild för spelare 1.
- Räkna ut vem som vann!

Vi skriver då till exempel:

```

if(x == 49){
    start_stop_player2();
    player1_image.src = "Sten.jpg";
    the_winner_is();
}

```

När vi trycker på enter vill vi starta/stoppa spelare 2. Hur gör vi det?

## **the\_winner\_is()**

Vann du? För att ta reda på det behöver vi veta vilken bild som spelare 1 och spelare 2 valde. Det kan man ta reda på genom att skriva:

```
var player1 = document.getElementById("player1").src;
var player1s_choice =
player1.substring(player1.lastIndexOf('/')+1);
```

Andra raden gör bara så att vi får ut filnamnet på bilden. Sten.jpg till exempel.

För att lätt säga vem som vann gör vi en variabel winner som vi ger värdet 1 om spelare 1 vinner, 2 om spelare 2 vinner och 0 om ingen vinner.

Sen är det bara att kolla alla fall och sätt winner till rätt värde. Här är ett exempel:

```
if (player1s_choice == "Sten.jpg" && player2s_choice == "Sax.jpg") {
    winner = 1;
}
```

För att enkelt visa vem som vann ändrar vi färgen på ramen på bilden som vann. Så här gör vi det:

```
if (winner == 1) {
    border1.borderColor="#00ff00";
    border2.borderColor="#ffffff";
}
```

Du kommer nu få en grön ram runt spelare 1 och ingen ram runt spelare 2. (Den är egentligen vit men det syns ju inte).

## **onload()**

Den här funktionen är bara till så att vi kan ändra bordercolor. Allt du behöver skriva är:

```
border1 = document.getElementById("player1").style;
border2 = document.getElementById("player2").style;
```