

TERNINGSPIL

Vi skal lave en hjemmeside med et terningspil. Hjemmesiden skrives i HTML. Layoutet laves med CSS, og selve spillet programmeres i Javascript. Først designer vi en spilleplade. Den skal bestå af en terning i et midterstillet hvidt felt på en grøn baggrund:



HTML-koden

Det hvide felt kan laves som en midterstillet tabel med 1 celle, hvor baggrundsfarven er hvid. Her er HTML-koden til spillepladen:

```
<html>

<head>
  <title>Terningspil</title>
  <link rel="stylesheet" type="text/css" href="terning.css">
  <script type="text/javascript" src="spil.js">
  </script>
</head>

<body>
  <table width="20%" align="center">
    <tr>
      <td class="terning">
        <br>
        
        <br>
        <b>Klik på terningen</b>
      </td>
    <tr>
    </tr>
  </table>
</body>

</html>
```

Terningens filnavn er *sekser.gif* som skrevet i ``-koden. Hvis man klikker på terningen (p.g.a. hændelsen “onClick”, skal der aktiveres noget Javascript-kode i funktionen *spil*. Javascript-koden befinder sig i en fil med navnet *spil.js* som angivet i koden `<script>` i HTML-dokumentets hoved. Terningen

er anbragt i en tabel, som er centreret, og som har en bredde på 20% af browservinduets bredde.

Stylesheetet

Farverne defineres i et stylesheet med navnet *terning.css* som angivet i koden `<link>`. Her er stylesheetet:

```
body          {color: black;
               background-color: green;
               }

td.terning    {background-color: white;
               text-align: center}
               }
```

Tekstfarven bliver sort, og baggrundsfarven bliver grøn. Vi definerer en klasse, *terning*, for koden `<td>` (tabelcelle), hvor baggrundsfarven bliver hvid, og hvor tekst og billeder bliver centreret i cellen. For at bruge denne klasse, skal vi indsætte `class="terning"` i `<td>`-koden i HTML-dokumentet. Brug af klassen *terning* bevirker, at det ikke er alle tabelceller, der får hvid baggrundsfarve.

Javascript-koden

Vi skal programmere et spil, som slår med en terning. Det klares ved at indsætte et tilfældigt tal blandt tallene 1, 2, 3, 4, 5 og 6. Vi gemmer det tilfældige tal i variabelen `kast`:

```
kast = Math.floor(6*Math.random()) + 1
```

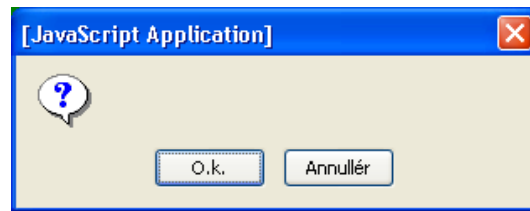
Math.random vælger et tilfældigt tal i intervallet fra 0,000000... til 0,999999... Når man ganger med 6, bliver tallet i intervallet fra 0,000000... til 5,999999... *Math.Floor* fjerner decimalerne, og når man lægger 1 til resultatet, får man et af tallene 1, 2, 3, 4, 5 eller 6.

Spillet skal give 50 kr i gevinst, hvis det slår en sekser, og 10 kr i gevinst, hvis det slår en femmer. Hvis det slår en ener, toer, treer eller firer, skal man tabe 20 kr. Det klares med en *if*-sætning. Her er første udgave af javascript-koden for funktionen *spil*:

```
function spil() {
    kast = Math.floor(6*Math.random()) + 1
    if(kast == 6) {
        gevinst = 50
    }
    else if(kast == 5) {
        gevinst = 10
    }
    else {
        gevinst = -20
    }
    alert('Din gevinst er '+gevinst+' kr')
}
```

Spillet kører kun én gang. Hvis det skal køre flere gange, skal vi bruge en løkke. Der er 2 muligheder: en *for*-løkke og en *while*-løkke. En *for*-løkke kører et bestemt antal gange, mens en *while*-løkke kører,

så længe en betingelse er opfyldt. *While*-løkken er mest smart, da man så kan spille, så længe man ønsker. Hvis vi udskifter *alert*-boksen med en *confirm*-boks, skal vi klikke “O.k.” for at fortsætte eller “Annullér” for at stoppe.



Hvis man klikker på “O.k.”, giver boksen værdien *true*, og hvis man klikker på “Annullér”, giver boksen værdien *false*. Vi gemmer værdien i variabelen *ok*:

```
ok = confirm('Din gevinst er '+gevinst+' kr')
```

While-løkken skal så køre, så længe variabelen *ok* har værdien *true*:

```
function spil() {  
  ok = true  
  while(ok == true) {  
    kast = Math.floor(6*Math.random()) + 1  
    if(kast == 6) {  
      gevinst = 50  
    }  
    else if(kast == 5) {  
      gevinst = 10  
    }  
    else {  
      gevinst = -20  
    }  
    ok = confirm('Din gevinst er '+gevinst+' kr. Fortsæt?')  
  }  
}
```

Sætningen *ok = true* før *while*-løkken er nødvendig, da variabelen skal have en værdi, inden løkken starter.

Hvordan man ændrer billede

Hvis vi vil have javascriptet til at vise, hvad terningen slår, skal vi have 6 forskellige billedfiler, som viser de 6 terninger. Lad os kalde dem *ener.gif*, *toer.gif*, *treer.gif*, *firer.gif*, *femmer.gif* og *sekser.gif*. Hvis javascript-koden skal kunne kommunikere med HTML-dokumentet, skal vi indsætte en *id*-variabel i

-koden. Lad os give *id*-variablen værdien “*terning*”. Det gøres sådan her i HTML-koden:

```

```

Derefter kan man med javascript-variablen *terning.src* ændre navnet på billedfilen. F.eks. kan man udskifte sekseren med en firer ved at skrive følgende i javascript-koden:

```
terning.src = 'firer.gif'
```

Her er hele javascriptkoden for spillet:

```
function spil() {
    ok = true
    while(ok == true) {
        kast = Math.floor(6*Math.random()) + 1
        if(kast == 1) {
            terning.src = 'ener.gif'
            beloeb = -20
        }
        else if(kast == 2) {
            terning.src = 'toer.gif'
            beloeb = -20
        }
        else if(kast == 3) {
            terning.src = 'treer.gif'
            beloeb = -20
        }
        else if(kast == 4) {
            terning.src = 'firer.gif'
            beloeb = -20
        }
        else if(kast == 5) {
            terning.src = 'femmer.gif'
            beloeb = 10
        }
        else {
            terning.src == 'sekser.gif'
            beloeb = 50
        }
        ok = confirm('Din gevinst er '+beloeb+' kr. Fortsæt?')
    }
}
```

Bemærk, at det er *id*-variablen i HTML-koden, som giver javascriptet mulighed for at lave ændringer på hjemmesiden. Her er resultatet at et spil:

