

Programmierung - Lektion 3

...

PC 230908

Reagera på musen - från lektion 2

I simple.js ingår funktioner för att “lyssna” efter hur musen rör sig och om någon knapp är nedtryckt. Här är ett simpelt ritprogram. Vi behöver endast en update funktion.

```
function update(){  
  circle(mouse.x, mouse.y, 10, "red");  
}
```

Ett något (men bara något) bättre ritprogram

Vi vill att vi ska rita om vi håller in vänster musknapp och sudda om man håller in höger musknapp. För att lyssna efter om vänster knapp är nedtryckt kan vi använda `mouse.left`. Vi ritar med röd färg och suddar genom att måla över med vit färg.

```
function update(){  
  if (mouse.left) {circle(mouse.x, mouse.y, 10, "red"); }  
  if (mouse.right) {circle(mouse.x, mouse.y, 10, "white"); }  
}
```

Nytt idag



Loopar

```
circle(100, 200, 40, "green");  
circle(200, 200, 40, "green");  
circle(300, 200, 40, "green");  
circle(400, 200, 40, "green");  
circle(500, 200, 40, "green");  
circle(600, 200, 40, "green");  
circle(700, 200, 40, "green");
```

Om jag vill repetera ett stycke kod flera gånger så kan man skapa en loop. Det finns flera loopar.

I exemplet vill jag rita 7 cirklar och väljer att göra anropa circle 7 gånger. Det är inte en smidig lösning, den skulle inte fungera bra om jag vill rita t ex 1 000 cirklar.

for-loopen

```
for (var x = 100; x<=700; x += 100) {  
    circle(x, 200, 40, "green");  
}
```

När jag vet hur många gånger jag ska upprepa är det oftast en for-loop jag använder.

Koden i förra exemplet kan skrivas som i exemplet till vänster

for-loopen

```
for (initalisering; test; storlek på steg) {  
    // Kod som körs i loopen  
}
```

I JavaScript är for-loopen ganska löst definierad men om vi använder den på det här viset kommer det fungera.

Initialisering kan vi tänka på som startvärdet, vi deklarerar vanligtvis en variabel här.

Test kollas inför varje loop och måste vara ett villkor.

Slutligen kan vi ange hur mycket vi vill att variabeln vi deklarerade i initialisering ska öka.

for-loop Exempel 2

```
for (i = 1; i<=5; i++){  
    text(10, i*25, 20, "Talet är : " + i , "orange");  
}
```

Vad är startvärdet för koden ovanför? Hur många rader text kommer skrivas ut?

Experimentera med koden och se till att du förstår vad som händer med din utskrift innan du kör programmet.

while loopar

```
let x = 50;
while (x <= 700){
  circle(x, 100, 20, "green");
  x += 70;
}
```

Om jag inte vet hur många gånger
gånger jag behöver upprepa en loop så
finns while loopen.

Det här exemplet fungerar egentligen
lika bra med en for-loop.

while loopen

```
while (condition){  
    //kod som körs om condition är  
    sant  
}
```

Om jag inte vet hur många gånger
gånger jag behöver upprepa en loop så
finns while loopen.

while loopen

```
let x = 100;  
let radie = 20  
while (radie <= 500){  
    circle(x, 100, radie, "orange");  
    x += 100;  
    radie=random(600);  
}
```

Här är ett exempel där vi ritar slumpartade cirklar tills radien blir större än 500.

Skillnaden mellan for och while?

- I JavaScript är skillnaden inte så stor, en loop kan antingen göras med hjälp av for eller while.
 - For ska användas när antalet upprepningar är känt
 - While när vi inte på förväg vet hur många upprepningar det tar tills ett villkor är uppfyllt.
- simple.js gör att vi inte behöver använda loopar i samma utsträckning, funktionen update() är i princip en loop.
- Det finns andra loop konstruktioner, JavaScript har även en do-while loop.
- Se upp för oändliga loopar - Spara din kod innan du kör det!

Felaktig kod kan bli dyrt.