**CSS**

Nu ska vi testa lite CSS. Det används dels för att göra vår HTML snyggare, men också för att göra sidor lättare att förstå och använda.

Det finns tre sorters CSS:

**Inline – För att förändra en HTML-tagg**

Vi ska nu prova att skriva lite CSS för att färga vår text. Det gör vi genom att skriva lite CSS i vår tagg. Kallas för *attribute.* Ibland vill man bara ändra på en särskild tagg.

**<p style=”color:blue”>Vår text</p>**

Lägg märke till att vi sätter CSS-koden som en del av första taggen.

Spara och uppdatera browsern.

Om det fungerade så prova att göra andra stycket rött t.ex. eller annan färg.

**Internal CSS – För att sätta stilen för hela dokumentet**

Istället för att sätta CSS-kod för varje enskild tagg kan man bestämma hur t.ex. alla stycke-taggar ska se ut i hela dokumentet.

För att bestämma hur vår CSS ska styra hela sidan sätter man upp regler inom style-taggar. Oftast sätter man dem i HEAD-delen i vårt HTML-dokument.

<head>

<meta charset=**"utf-8"**>

<title>**Min första sida**</title>

<style type=**"text/css"**>

**p {**

**color:green;**

**}**

</style>

</head>

Här sätter vi att alla stycketaggar ska vara gröna. Eftersom vi har satt annan CSS på våra stycketaggar så kommer inte detta att påverka dom. Ta bort era style-attribut eller från era p-taggar eller skapa nya.

Spara koden och uppdatera browsern.

Nu ska alla taggar som inte har ”inline-CSS” vara gröna.

**Klasser och id**

Ett annat sätt att tilldela flera attribut som färg, storlek m.m. till en eller flera former är klasser och id:n

Skillnaden på klasser och id:n är att klasser kan användas på flera ställen medan id:n är unika och bara kan användas en gång. ID:n kan (i vissa fall) användas flera gånger men bör inte.

**Vi testar med id. Ett id föregås av en hashtag:**

<head>

<style type=**"text/css"**>

**#oneTime {**

**color:red;**

**font-size: 2em;**

**font-style: italic;**

**}**

</style>

</head>

<p id=**"oneTime"**>**Denna text styrs av ett id**</p>

**Vi testar med klass. En klass föregås av en punkt:**

<head>

<style type=**"text/css"**>

**.manyTimes {**

**color:blue;**

**font-size: 3em;**

**font-style: normal;**

**}**

</style>

</head>

<p class=**"manyTimes"**>**Denna text styrs av klass**</p>

**Vi kan också använda flera klasser på en gång:**

Vi skapar en ny klass som gör texten i fetstil:

**.boldText {**

**font-weight: bold;**

**}**

Nu kan vi använda båda klasserna för att få text som är blå, 3em i storlek, normal och fet.

<p class=**"manyTimes boldText"**>**Denna text styrs av klass**</p>

**DIV och External Style Sheet**

Div är en HTML-tag som används för att dela upp en sida i olika delar. Ni kommer att se sedan när ni lär er mer om HTML5 det finns andra taggar som fungerar som en div men har andra namn.

Man kan säga att en div skapar en behållare.

**Övning 1**

Vi ska testa att skapa en sida som är uppdelad i olika delar.

Först ska vi dock prova den tredje varianten av CSS - **External Style Sheet**

Att lägga in CSS i varje dokument är oftast en dålig idé, åtminstone när man kör ”skarpt”.

Har man t.ex. en viss färg eller stil för alla rubriker måste man gå in på varje sida för att ändra.

En bättre idé är att bara ha ett ställe där man ska ändra på. För att lösa det använder man en separat sida där man har sin CSS.

Ta allting från mellan <style>-taggarna och lägg det i en ny fil.

Spara filen som t.ex. style.css. Tillägget css är det viktiga.

För att vi ska tala om att vi vill använda det dokumentet lägger vi in en länk i <HEAD>.

<link rel=**"stylesheet"** type=**"text/css"** href=**"style.css"**>

Det som är viktigt här är det som står vid **href**. Det är namnet på dokumentet vi länkar till. Har du sparat under annat namn så skriv det istället.

Spara och uppdatera browsern.

**Övning 2**

Nu ska vi testa att skapa en sida uppbyggd med <DIV> och CSS. Gör en ny sida med HTML.

Vår HTML-sida ska se ut så här innanför <body>-taggarna:

<div id=**"container"**>

<div id=**"header"**>

Sidhuvud

</div>

<div id=**"content"**>

Innehåll

</div>

<div id=**"sidePanel"**>

Sidpanel

</div>

<div id=**"footer"**>

Sidfot

</div>

</div>

Spara och uppdatera

Vi har i vår HTML delat in vår sida i olika div:s. Utan vår CSS syns dom dock inte.

För att ange var div:arna ska vara, hur de ska se ut, vilken storlek de ska ha m.m. ska vi använda CSS.

**Skapa CSS-fil**

Nu ska vi skapa en CSS-fil för att placera och formatera våra div:ar.

Skapa en fil som heter t.ex. styles2.css. Namnet är inte viktigt men glöm inte tillägget .css.

Glöm inte heller att ha det namnet i länken till filen.

CSS-kod som läggs i nya filen:

**#container**

**{**

**width: 673px;**

**border: solid 1px black;**

**margin: 0px;**

**}**

**#header**

**{**

**color: #FFFFFF;**

**width: 100%;**

**height: 100px;**

**background-color: #778CB3;**

**}**

**#sidePanel**

**{**

**width: 213px;**

**height: 372px;**

**margin-top: 5px;**

**background-color: #778CB3;**

**color: #FFFFFF;**

**padding: 4px;**

**}**

**#content**

**{**

**width: 446px;**

**float: right;**

**height: 380px;**

**margin-top: 5px;**

**background-color: #A0AFCB;**

**}**

**#footer**

**{**

**background-color: #F7CB33;**

**padding: 12px;**

**width: 649px;**

**color: #000000;**

**font-size: 90%;**

**text-align: center;**

**clear: both;**

**border-top: 5px solid #FFFFFF;**

**}**

Spara och uppdatera.

**Float**

För att ni ska se hur float fungerar gör så här:

Markera bort den div som har id=”content”

Markera bort gör man i HTML med <! – och -->

Kallas rem för remark och används för att t.ex. sätta en kommentar som browsern struntar i. Fast som ni ser kan man även tillfälligt plocka bort kod.

I den div som har id=”sidepanel” skriver ni:

**float: right;**

Spara och uppdatera. Nu bör din panel vara till höger. Testa att skriva:

**float: left;**

Ta bort kommatering för att återställa.

**Float i meny**

Float kan också användas i listor för att skapa en horisontell meny.

**Box-modellen**

Boxmodellen används för att sätta ramar runt objekt och text i CSS.

<http://www.w3schools.com/css/css_boxmodel.asp>

Border är en ram som kan vara synlig om vi vill runt t.ex. en text.

Padding är mellanrum mellan texten och ramen som inte syns.

Margin är mellanrum utanför ramen som inte heller syns.

