

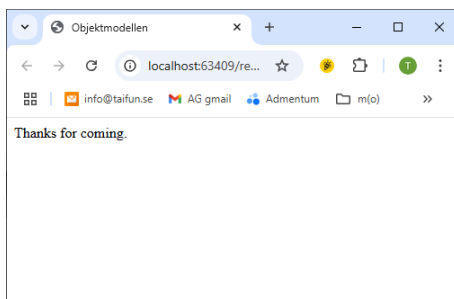
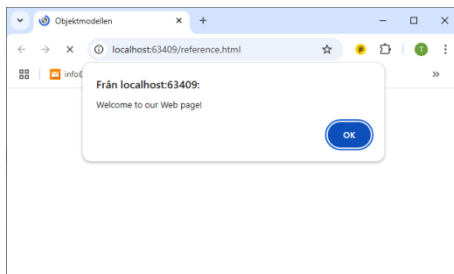
7.1 Objektmodellen

```
reference.html
1 <!-- Reference.html -->
2 <head>
3   <title>Objektmodellen</title>
4   <script>
5     function start()
6     {
7       alert(pText.innerText) // Objektet pTexts attribut
8                             // innerText skrivs ut
9       pText.innerText = "Thanks for coming." // innerText får
10                                           // som variabel ett nytt värde
11     }
12   </script>      <!-- Objektet pText har dynamiskt innehåll -->
13 </head>
14 <body onload = "start()">
15   <!-- HTML-elementet p är ett objekt med referensen id vars -->
16   <!-- värde är pText. Objektet har ett inbyggt attribut som -->
17   <!-- heter innerText som sätts till "Welcome to our ..." -->
18   <p id = "pText">Welcome to our Web page!</p>
19 </body>
```

För att förstå detta avsnitt bättre rekommenderas att läsa om objekt i JavaScript i avsnitt 6.1 (sid 119).

JavaScripts objektmodell

I detta avsnitt behandlar vi den s.k. *objektmodellen* i JavaScript, vilket innebär att JavaScripts objektbaserade karaktär utvidgas till HTML. Det betyder bl.a. att HTML-element behandlas som objekt. Det gör man genom att tilldela elementen attributet **id** och ge det ett värde som sedan kan användas som referens (namn) till objektet. Dessutom har JavaScript definierat attribut (egenskaper) och metoder till objektet som kan användas för att manipulera objektet.



HTML-element som objekt

I scriptet **Reference** (ovan) på rad **17** har t.ex. HTML-elementet **p** definierats som objekt genom attributet **id** vars värde **pText** kan anses som en referens till objektet. Funktionen **start()** refererar till detta objekt två gånger, en gång på rad **7** och en annan gång på rad **9**. Dessutom har JavaScript definierat attributet (egenskapen) **innerText** till objektet **p**. Scriptet **Reference** använder detta attribut på rad **7** för att hämta (läsa) den aktuella texten som är definierat i HTML-elementet **p** som innehåll (rad **17**) och skriva ut den i **alert**-boxen. Sedan används attributet **innerText** för att ändra (skriva) texten på rad **9**. Att denna ändring är möjlig beror på att **innerText** är som attribut i ett objekt en *variabel* vars värde är dynamisk och kan ändras. **innerText** pekar på texten i innehållet av HTML-elementet **p** som i sin tur har referensen (namnet) **pText**. Dvs man skriver:

```
pText.innerText
```

för att komma åt texten i innehållet av HTML-elementet **p**, både för att hämta den och för att ändra den. Detta hade utan JavaScriptets objektmodell inte varit möjligt.