

JavaScript 1
alert("Lektion 13");

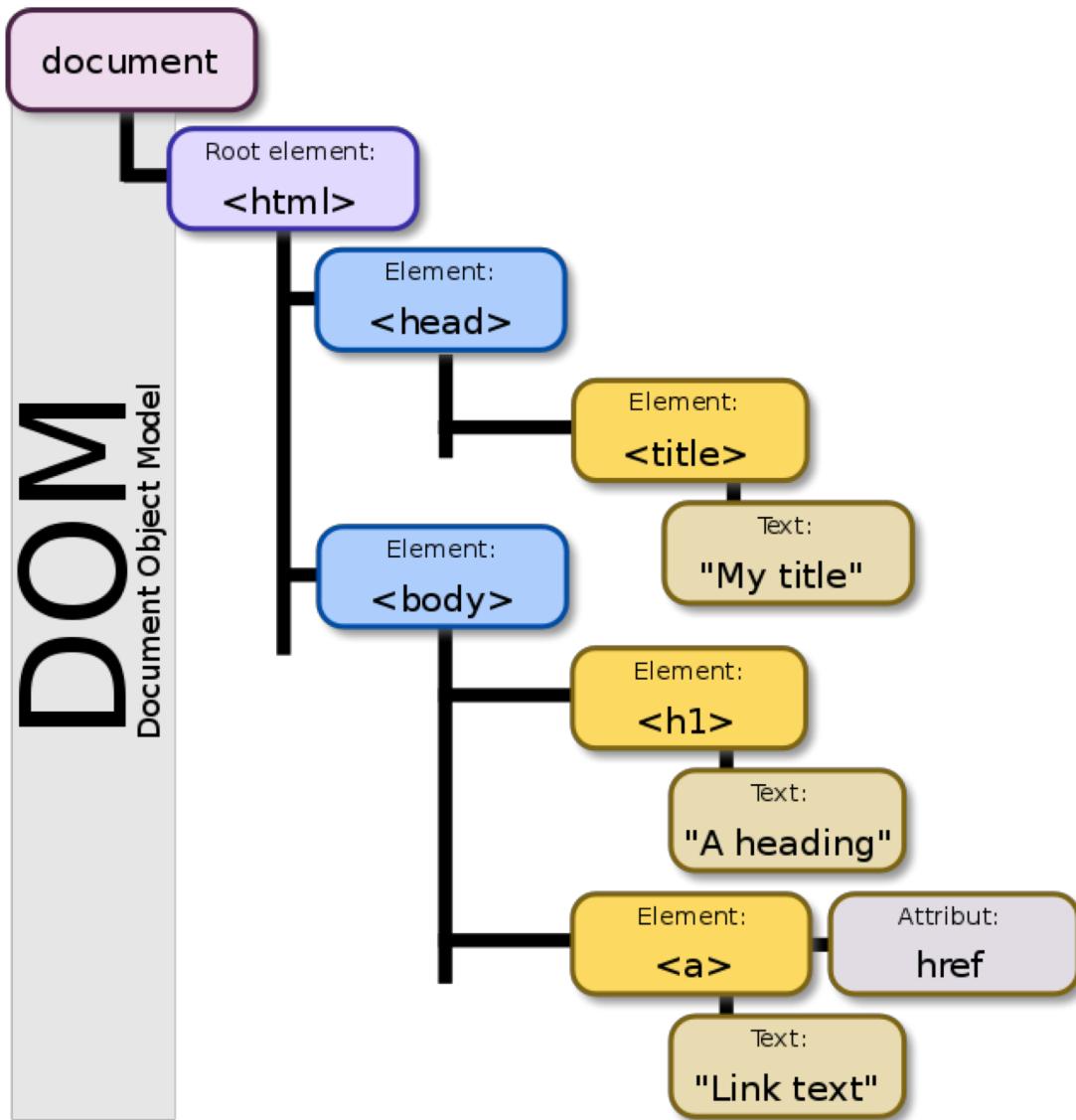
Utbildare: Mahmud Al Hakim

NACKADEMIN

Lektionstillfällets mål

- **Mål med lektionen**
 - **Händelsehantering i JavaScript (Events)**
 - Att förstå events
 - Händelsehantering med HTML-attribut
 - Händelsehantering med DOM
- **Arbetsmetod**
 - Teori och praktik varvas under lektionen

Kort summering av föregående lektion

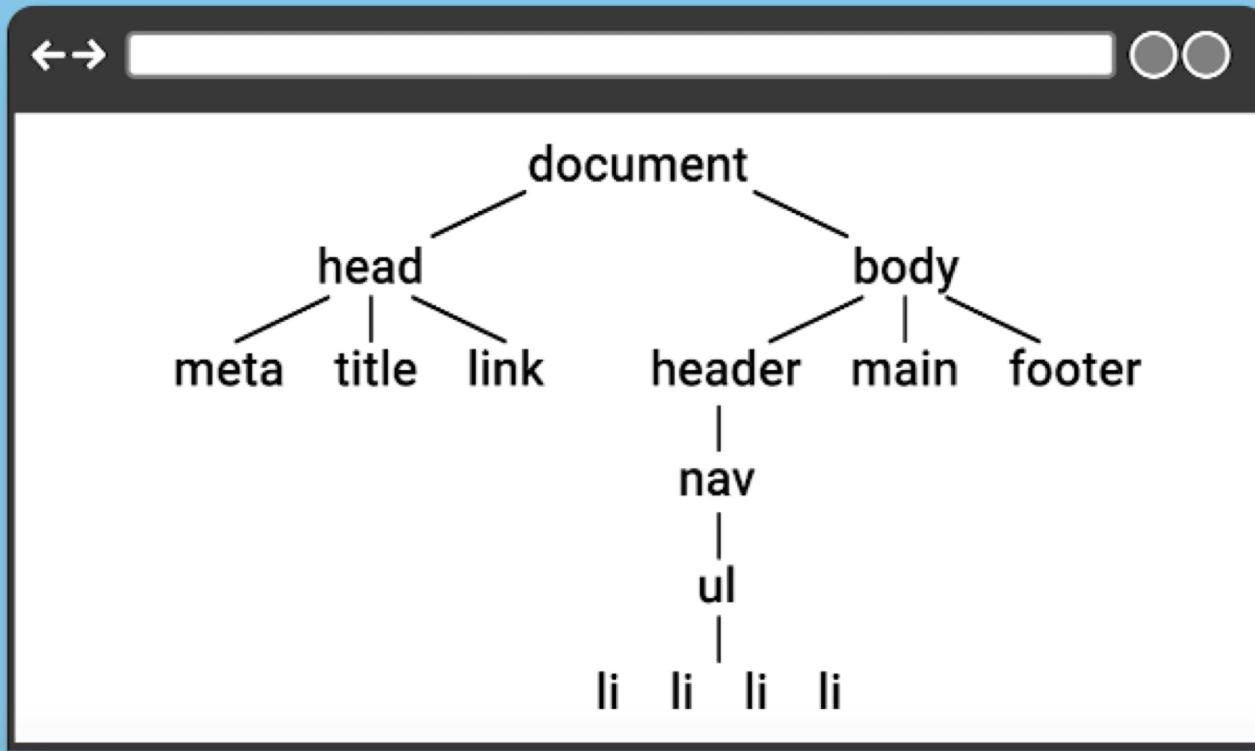


Föregående lektion:

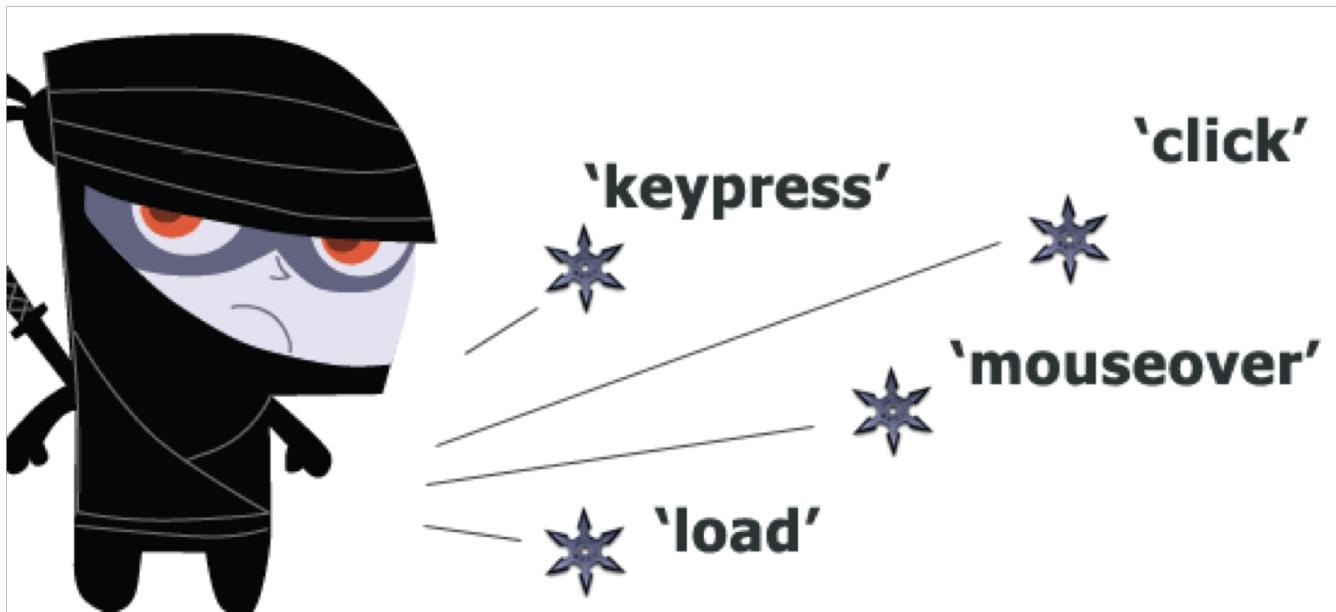
- Vi har jobbat med DOM.

BOM och DOM – Sammanfattning

<https://youtu.be/t3CWlfZUvL8>

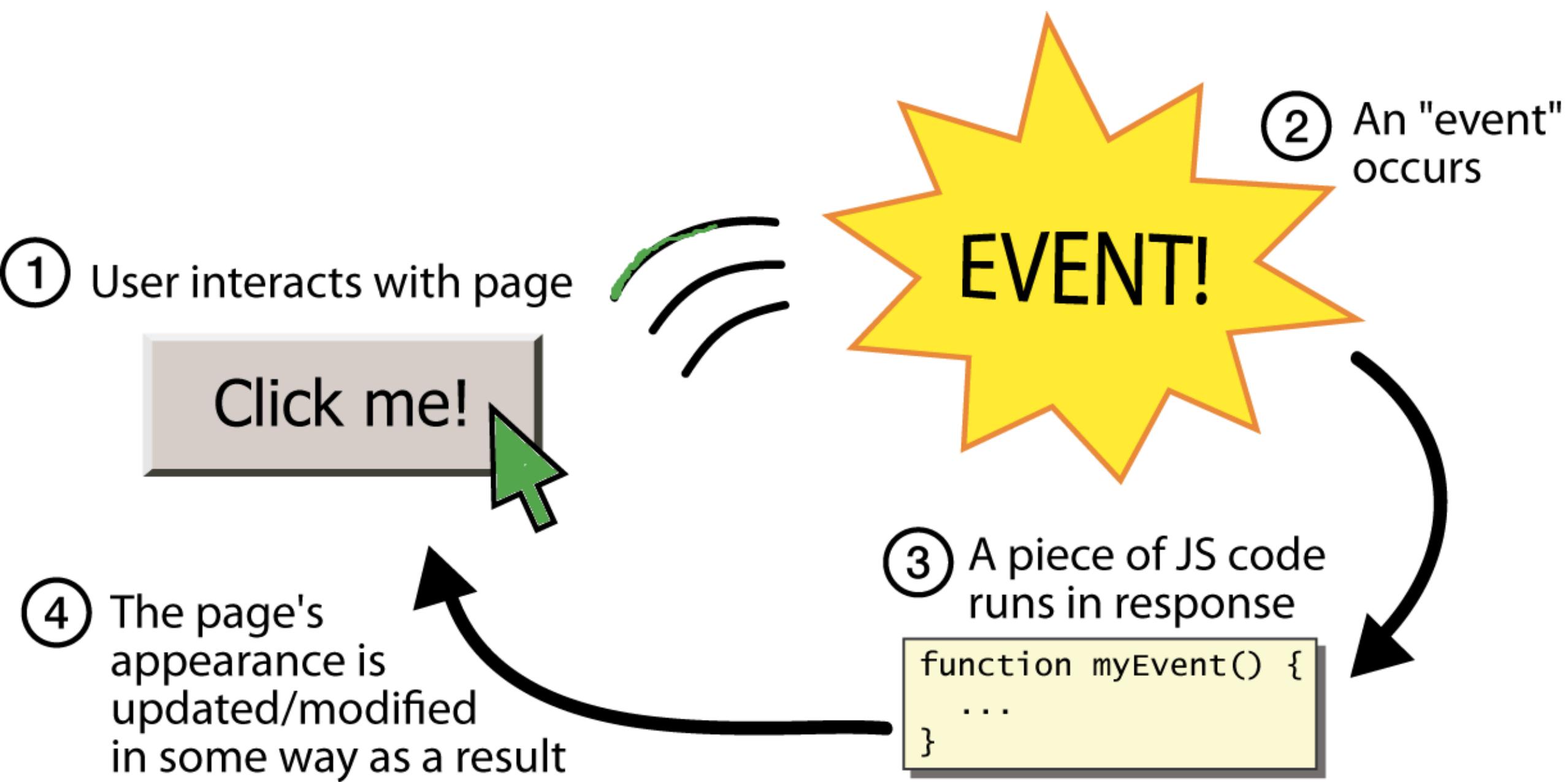


Att förstå events



Bildkälla: <https://multidict.net/clilstore/page.php?id=5599>

- Med JavaScript kan du få webbläsare att reagera på vissa händelser s.k. events.
- Du kan göra dina webbsidor mer levande och interaktiva genom att de förändras på något sätt beroende på vad användaren gör.
- T.ex. När användaren klickar på en knapp eller helt enkelt när sidan laddas.



Tre olika sätt att hantera events i JavaScript

1. HTML Event Attributes

Events hanteras inne i HTML-koden med hjälp av olika attribut som kopplas direkt till HTML-element.

(OBS! Rekommenderas inte. Följer ej SoC-principen, se nästa bild)

https://www.w3schools.com/tags/ref_eventattributes.asp

2. Standard DOM Events

Events hanteras inne i JavaScript-koden med hjälp av olika egenskaper som finns i DOM.

OBS! Vi får bara koppla en enda funktion till varje event.

https://www.w3schools.com/jsref/dom_obj_event.asp

3. DOM EventListener

Events hanteras med hjälp av metoden addEventListener()

Du kan koppla flera olika funktioner till varje event.

Du kan också ta bort events med hjälp av metoden removeEventListener()

https://www.w3schools.com/js/js_htmldom_eventlistener.asp



SoC =
Separation of concerns

- På svenska ungefär...
"att hålla isär saker och ting".
- Rekommendationen när det gäller webbutveckling är att hålla isär JavaScript och HTML.

HTML Event Attributes – Exempel 1



```
<body onload="alert('Welcome');">  
  
</body>
```

HTML Event Attributes – Exempel 2



<body>

```
<button onclick="document.body.bgColor='red'">red</button>
<button onclick="document.body.bgColor='green'">Green</button>
<button onclick="document.body.bgColor='blue'">Blue</button>
```

</body>

HTML Event Attributes – Exempel 3

```
<p id="demo"></p>
```

```
<button onclick="myFunction()">Click me</button>
```

```
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello World";
}
</script>
```



HTML Event Attributes – Exempel 4



```
<p id="demo">
```

Dölj texten genom att klicka på knappen nedan.

```
</p>
```

```
<button
```

```
    onclick="document.getElementById( 'demo' ).style.display='none' ">
```

Dölj

```
</button>
```

HTML Event Attributes – Exempel 5



```
<p onclick="this.style.display='none' ">
```

Dölj texten genom att klicka här.

```
</p>
```

HTML Event Attributes – Exempel 6



```
<p onmousemove="this.style.color='green'">  
    Peka här för att ändra färgen!  
</p>
```

HTML Event Attributes – Exempel 7



```
<button onclick="getTime();">What is the time now?</button>
<h1 id="time"></h1>
<script>
function getTime() {
    document.getElementById("time").innerHTML =
        new Date().toTimeString().substring(0,8);
}
</script>
```

Att arbeta med standard DOM Events

element.onevent = functionName;

The diagram illustrates the components of the event assignment statement. It consists of three main parts: 'element', 'onevent', and 'functionName'. Brackets below each part are labeled: 'ELEMENT' under 'element', 'EVENT' under 'onevent', and 'CODE' under 'functionName'. Below these labels, detailed descriptions are provided for each component.

ELEMENT	EVENT	CODE
DOM element node to target	Event bound to node(s) preceded by word "on"	Name of function to call (with no parentheses following it)

Standard DOM Events – Exempel 1

- Vi startar med HTML-koden nedan

Username:

```
<input type="text" id="username">
```

password:

```
<input type="password" id="password">
```

```
<button>Login</button>
```

```
<p id="feedback"></p>
```

Standard DOM Events – Exempel 1 fort.

```
function checkUsername() {  
    // code to check the length of username  
}  
let elUser = document.getElementById('username');  
elUser.onblur = checkUsername;
```

A reference
to the DOM
element node
is often stored
in a variable.

```
function checkUsername() {  
    // code to check the length of username  
}  
[var el = document.getElementById('username');  
 el.onblur = checkUsername; ——————  
           |—————]
```

The event name is preceded by the word "on."

The code starts
by defining the
named function.

The function
is called by the
event handler on
the last line, but
the parentheses
are omitted.

Standard DOM Events – Exempel 1 fort.

```
function checkUsername() {  
    let elMsg = document.getElementById('feedback');  
    if (elUser.value.length < 5) {  
        elMsg.textContent = "Username must be 5 characters or more!";  
    } else {  
        elMsg.textContent = " ";  
    }  
}
```

Standard DOM Events – Exempel 2

Ange pris exkl. moms:

```
<input type="text" id="price1">
<button id="btnPrice">Beräkna pris inkl. moms</button>
<p id="price2"></p>
```

```
<script>
let price1 = document.getElementById('price1');
let price2 = document.getElementById('price2');
let btnPrice = document.getElementById('btnPrice');
btnPrice.onclick = getPrice;
function getPrice() {
    price2.innerHTML = price1.value * 1.25;
}
</script>
```

Att arbeta med Event Listeners

Adds an event listener to the DOM element node(s)

METHOD

element.addEventListener('event', *functionName* [, *Boolean*]);

ELEMENT

DOM element
node to target

EVENT

Event to bind node(s)
to in quote marks

CODE

Name of function
to call

EVENT FLOW

Indicates something called
capture, and is usually set
to false

Event Listeners – Exempel 1

```
<button id="btnDate">Visa dagens datum</button>
<p id="demo"></p>

<script>
let btn = document.getElementById("btnDate");
btn.addEventListener("click", displayDate);

function displayDate() {
    document.getElementById("demo").innerHTML =
        new Date().toString();
}
</script>
```

Event Listeners – Exempel 2

```
<button id="btnDate">Visa datum och tid</button>
<p id="date"></p> <p id="time"></p>
<script>
let btn = document.getElementById("btnDate");
btn.addEventListener("click", displayDate);
btn.addEventListener("click", displayTime);

function displayDate() {
    document.getElementById("date").innerHTML =
        new Date().toLocaleDateString();
}

function displayTime() {
    document.getElementById("time").innerHTML =
        new Date().toTimeString().substring(0,8);
}
</script>
```

Du får koppla flera funktioner till samma event med event listeners.
Detta är inte tillåtet om du använder standarder DOM Events!

Övning 1

- Nedanstående kod använder HTML-eventet onload för att visa ett meddelande.

```
<body onload="alert( 'Welcome' );">
```

- Skapa ett skript som ersätter detta med DOM event istället.
- Du ska alltså rensa HTML-koden!
- Så här ska koden ser ut

```
<body>  
<script>  
    // Skriv din lösning här  
</script>  
</body>
```

Övning 2

- Nedanstående kod använder HTML-eventet onclick för att ändra bakgrundsfärgen på sidan.

```
<button  
    onclick="document.body.bgColor='red'">  
    Red</button>
```
 - Skapa ett skript som ersätter detta med DOM event istället.
 - OBS! Det finns enbart en enda knapp på sidan.
 - Du ska alltså rensa HTML-koden!
Så här ska koden ser ut
- ```
<button>red</button>
<script>
 // Skriv din lösning här
</script>
</body>
```

# Övning 3

Använd "HTML DOM Event" istället för "HTML Event Attribut" för att få bättre SoC (Separation of concerns) i koden nedan.

```
<p id="demo"></p>

<button onclick="myFunction()">Click me</button>

<script>
function myFunction() {
 document.getElementById("demo").innerHTML = "Hello World";
}
</script>
```

# Övning 4

Använd ”HTML DOM Event” istället för ”HTML Event Attribut” för att få bättre SoC (Separation of concerns) i koden nedan.

```
<p id="demo">
Dölj texten genom att klicka på knappen nedan.
</p>
<button
onclick="document.getElementById('demo').style.display='none'">
Dölj
</button>
```

# Övning 5

Använd ”HTML DOM Event” istället för ”HTML Event Attribut” för att få bättre SoC (Separation of concerns) i koden nedan.

```
<p onclick="this.style.display='none'">
```

Dölj texten genom att klicka på här.

```
</p>
```

# Övning 6

Använd ”HTML DOM Event” istället för ”HTML Event Attribut” för att få bättre SoC (Separation of concerns) i koden nedan.

```
<p onmousemove="this.style.color='green'">
```

Peka här för att ändra färgen!

```
</p>
```

# Övning 7

Använd "HTML DOM Event" istället för "HTML Event Attribut" för att få bättre SoC (Separation of concerns) i koden nedan.

```
<button onclick="getTime();">What is the time now?</button>
<h1 id="time"></h1>
<script>
function getTime() {
 document.getElementById("time").innerHTML =
 new Date().toTimeString().substring(0,8);
}
</script>
```

# Övningar

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_events1](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_events1)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_events2](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_events2)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_events3](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_events3)

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_timing\\_clock](https://www.w3schools.com/js/tryit.asp?filename=tryjs_timing_clock)

[https://www.w3schools.com/howto/tryit.asp?filename=tryhow\\_js\\_typewriter](https://www.w3schools.com/howto/tryit.asp?filename=tryhow_js_typewriter)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_dom\\_html9](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html9)

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_addeventlistener\\_add\\_many2](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addeventlistener_add_many2)

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_addeventlistener\\_parameters](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addeventlistener_parameters)

# Summering av dagens lektion

- Vi har idag jobbat med händelsehantering i JavaScript (Events)
- Vi har lärt oss hur events hanteras med HTML-attribut och DOM.
- Reflektioner kring dagens lektion?
  - Vad tar du med dig från dagens lektion?
  - Finns det något som var extra svårt att förstå?
  - Finns det något som vi behöver repetera?
  - Hur upplevde du dagens arbetsmetoder?

# Framåtblick inför nästa lektion

- Läs: Bok 2, sid. 243-255
- Under nästa lektion kommer vi att ha ett antal övningar som görs i grupp och repetition inför provet.
- Lycka till.



NACKADEMIN