



**LAUREA**  
UNIVERSITY OF APPLIED SCIENCES  
*Together we are stronger*

# JavaScript Events

Dynamic Web Apps - 2017

Mika Stenberg / Meija Lohiniva



**JavaScript**

# 1. What will be discussed today

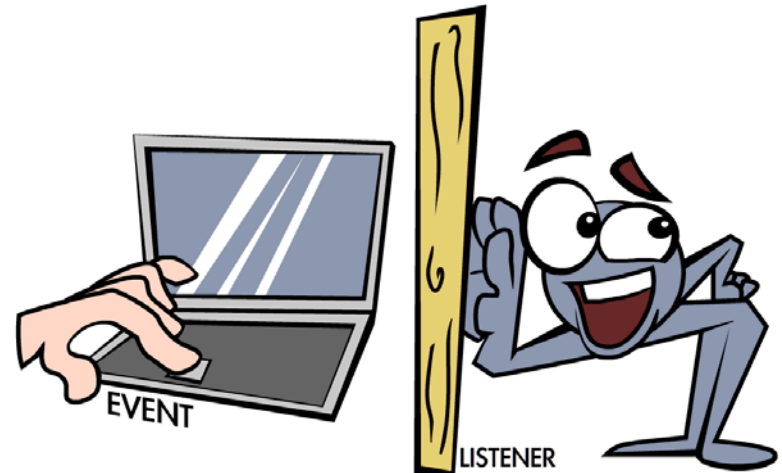
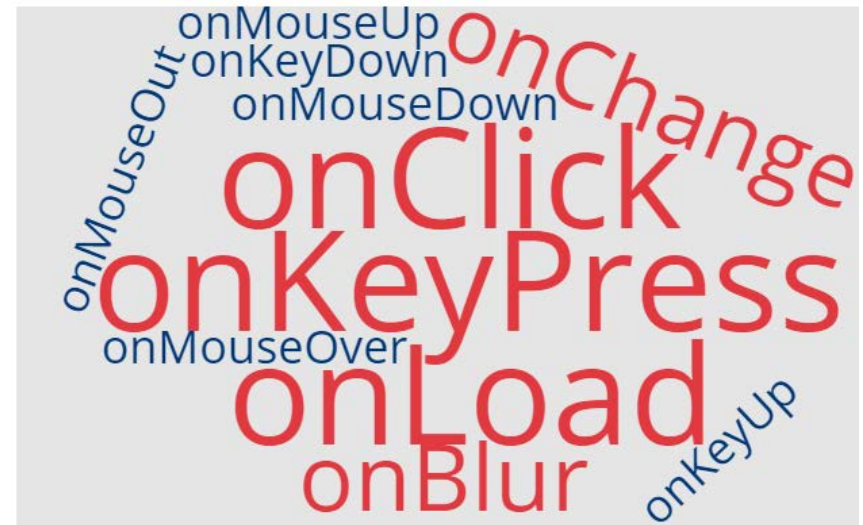
1. Introduction
2. HTML Events
3. Handling Events in JS
4. Example
5. Common Events in JS
6. Adding listeners dynamically

# 1. Introduction

- ▶ So far we've learned that the browser will run any JavaScript code whenever it encounters one while loading the page.
- ▶ With an exception: code within functions will only be run when the function gets called.
- ▶ This Chapter introduces a way to call those functions, other than from a block of code itself: **events**.

## 2. HTML Events

- ▶ HTML events are "**things**" that happen to a page or its elements.
- ▶ These can be something like:
  - ▶ A web page has finished loading.
  - ▶ An input field was changed.
  - ▶ A button was clicked.
  - ▶ A form was submitted.
- ▶ JavaScript can "**react**" (execute code) on these events.
- ▶ Common tasks are checking or validating an input.



(Picture: <https://yfain.github.io/Java4Kids/>)  JavaScript

### 3. Handling Events in JS

- ▶ HTML allows event handler attributes, with JavaScript code, to be added to HTML elements.
- ▶ Some examples could be:

Javascript code

```
<button onclick="alert('Click')">Click me</button>  
<input onfocus="myFunction()"></input>  
<form onsubmit="validateForm()"></form>  
<button onmouseover="alert('On me!')"  
onmouseout="alert('Off me')">Nada</button>
```

An HTML attribute

## 4. Example

- While we can write the code directly into the event, it is usually easier to call for a named function.
- Then we declare the programming code later on as a function:

```
<!DOCTYPE html>
<html>
<body>

<p>What is the name(s) of your browser?</p>

<button onclick="myFunction()">Try it</button>

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

<script>
function myFunction() {
    document.write ( "Name is " + navigator.appName);
    document.write("<br>");
    document.write ("Code name is " + navigator.appCodeName);
}
</script>

</body>
</html>
```

## 5. Common Events in JS

- ▶ Common events are listed below.
- ▶ Full list of events can be found [online](#).

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

## 6. Adding Listeners Dynamically

- ▶ In some cases, one might want to add event listeners dynamically through JavaScript.
- ▶ This can be done using `addEventListener`-method.
- ▶ Removing the listener is done using `removeEventListener` -method.

```
// Get reference to an element
element = document.getElementsByTagName('h1')[0];

// Add
element.addEventListener("click", function(){ alert("Hello
World!"); });

// Remove
element.removeEventListener("mousemove", myFunction);
```



## 6. Adding Listeners Dynamically

- ▶ Use `addEventListener` if you want to attach several listeners.
- ▶ Not supported in IE <9 (you can use `attachEvent` for earlier versions).

```
var element = document.getElementById('a');  
  
element.onclick = doSomething_1;  
//This overwrites previous onclick  
element.onclick = doSomething_2;  
  
//These are both ok  
element.addEventListener('click', doSomething_3);  
element.addEventListener('click', doSomething_4);
```

# Questions or comments?