

jQuery

# Vad är jQuery?

JavaScript-bibliotek som förenklar samspel  
mellan JavaScript och HTML

## Varför jQuery?

- Smidigt och enkelt
- Kompatibilitet över alla webbläsare
  - Tonvis av plugins
  - Bra dokumenterat

# Hur använder man jQuery?

- Ladda ner från <http://jquery.com/>
  - Inkludera med `<script src="">` taggen i `<head>`
  - Skriv din kod på samma ställe som traditionellt JavaScript, i .js-filer
- jQuery-kod ÄR JavaScript-kod, fast förenklad. Det går bra att blanda jQuery och traditionellt JavaScript!

```
<script src="js/jquery.min.js"></script>
```

# Syntax

- Alla jQuery-statements börjar med **\$**
- Syntaxen för ett statement är (i de flesta fall) uppdelad i två delar:
  - Selector (hitta elementet/elementen)
  - Metod (gör något med dem)

## jQuery-symbol

↓  
`$("#testDiv").addClass("testClass");`

↑  
**Selector**

↑  
**Metod**

# Selectors

```
$("#testDiv").addClass("testClass");
```

- Använder sig av CSS selectors, på samma sätt som `querySelectorAll()`

- jQuery har även egna selectors (t ex `:first`)

- Man kan behandla flera element samtidigt utan att behöva loopa igenom en array som i traditionell JavaScript

- Till skillnad från traditionell JavaScript så avbryts inte scriptet om ett element inte hittas (graceful failure)

# Selectors - exempel

-Hitta med Id

```
$("#testDiv").addClass("testClass");
```

-Hitta med CSS-klass

```
$(".testDiv").addClass("testClass");
```

-Hitta med taggnamn

```
$("div").addClass("testClass");
```

# Selectors - exempel

-Manipulera flera element samtidigt

```
$("#div").addClass("testClass");
```

-Om vi hade gjort samma sak i traditionellt JavaScript

```
var divs = document.getElementsByTagName("div");  
for (var i = 0; i < divs.length; i++) {  
    divs[i].className += "testClass";  
}
```

# Metoder

```
$("#testDiv").addClass("testClass");
```

-Det finns metoder för att hantera

- Attribut
- CSS
- Manipulering (av innehåll)
- Traversering
- Events
- Effekter (imorgon)
- Ajax (nästa kurs)



# Attribut

-Lägg till/ändra attribut

```
$("input[type=checkbox]").attr("checked","true");
```

-Ta bort attribut

```
$("input[type=checkbox]").removeAttr("checked");
```

-Ändra attribut value

```
$("#userName").val("Martin");
```

-Hämta attribut value och lägg i variabel

```
var userName = $("#userName").val();
```

# Attribut

-Ändrat attribut height och width

```
$("#testDiv").height(50);  
$("#testDiv").width(100);
```

-Hämta attribut height och width

```
var divHeight = $("#testDiv").height();  
var divWidth = $("#testDiv").width();
```

# CSS

Lägg till och ta bort CSS-klass

```
$("#testDiv").addClass("testClass");  
$("#testDiv").removeClass("testClass");
```

Ändra CSS-styling

```
$("#testDiv").css("background-color", "yellow");  
$("#testDiv").css("border", "1px solid red");
```

Ändra CSS-styling (med object)

```
$("#testDiv").css({  
    backgroundColor: "yellow",  
    border: "1px solid red"  
});
```

# CSS

Hämta CSS-styling och lägg i variabel

```
var divBg = $("#testDiv").css("background-color");
```

```
var divBg = $("#testDiv").css({backgroundColor});
```

# JavaScript Objects

- En del av JavaScript. Inte specifikt för jQuery!
- Som en array fast med ett namn för varje värde istället för en siffra. Man skulle kunna säga “en lista med variabler”.
- Startar med { och slutar med }

```
var styling = {  
    backgroundColor: "yellow",  
    border: "1px solid red"  
};
```

```
$("#testDiv").css(styling);
```



# Manipulering (av innehåll)

-Ersätt allt HTML-innehåll i elementet

```
$("#testDiv").html("Ny text");
```

-Lägg till innehåll i början av elementet

```
$("#testDiv").prepend("Text i början");
```

-Lägg till innehåll i slutet av elementet

```
$("#testDiv").append("Text i slutet");
```

-Töm elementet

```
$("#testDiv").empty();
```

# Manipulering (av innehåll)

-Andra manipuleringsmetoder värda att nämna

`.wrap();`

`.after();`

`.before();`

`.remove();`



# Traversering

- Att traversera innebär att stega fram och tillbaka i en trädstruktur (t ex DOM)

- I JavaScript kan man stega mellan olika element i DOM

- Välj elementets förälder (steget ovan i DOM)

```
$("#userName").parent().submit();
```

- Hitta ett element i elementet (nedanför i DOM)

```
$("form").find("#userName").val("Samuel");
```

# Events

-Funkar med anonyma och deklarerade funktioner

-Bind eventet **click** till ett element (**deklarerad funktion**)

```
$("input[type=button"]').bind("click", myFunction);
```

-Bind eventet **click** till ett element (**anonym funktion**)

```
$("input[type=button"]').bind("click", function(){  
    //do stuff  
});
```

-Ta bort ett eventet **click** från ett element

```
$("input[type=button"]').unbind("click");
```

# Förkortningar av events

-Istället för `.bind()` kan man använda förkortningar för respektive event

-Musklick

```
$("#input[type=button]").click(function(){  
    //do stuff  
});
```

-Hover

```
$("#div").hover(function(){  
    //do stuff  
});
```

-Submit för formulär

```
$("#form").submit(function(){  
    //do stuff  
});
```

# Förkortningar av events

-Sidan färdigladdad (kan användas med andra element också)

```
$(document).ready(function(){  
    //do stuff  
});
```

-Förkortning av ovan (sidan färdigladdad)

```
$(function(){  
    //do stuff  
});
```

-Samma event i traditionellt JavaScript

```
window.onload = function(){  
    //do stuff  
}
```

# Trigga events

- Om man inte anger en funktion som parameter triggrar man eventet istället för att binda det

- Triggra knappens click-event

```
$("#input[type=button]").click();
```

- Skicka formuläret

```
$("#form").submit();
```

