

# Introduktion till JS

Det här kan ni nog redan...

# Dagens föreläsning

- Bakgrund
- Våra verktyg
- Grunderna i Javascript
- Lite mer avancerad Javascript

# Bakgrund

Vad är Javascript och varför ska vi bry oss?

# Språket Javascript

- Ursprungligen avsett för att göra webbsidor roligare
  - Bra stöd för att manipulera DOM
  - Bra stöd i de flesta webbläsare
- Körs numera lite över allt
- Har numera ett bra stöd av tredjepartsbibliotek

# Språket Javascript

- Syntaxen liknar C, Java, etc
- Löst typat – inga explicita variabeltyper
- Multiparadigm:
  - Händelsedrivet
  - Funktionellt
  - Prototypbaserat (ungefär som objektorienterat)
  - Imperativt

# Exempel på användning

- Dynamiska webbsidor
- Serverapplikationer med Node.JS
- Mobilapplikationer med Cordova
- Spelutveckling med Unity
- Robotstyrning med Nodebots

# Lite historik



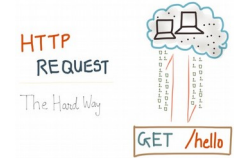
**1995:** Netscape



**1996:** JScript



**1997:** ECMAScript



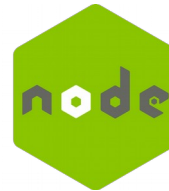
**2005:** AJAX



**2006:** JQuery



**2008:** V8-motorn



**2010:** Node.JS



**2015:** ES 6

# Våra verktyg



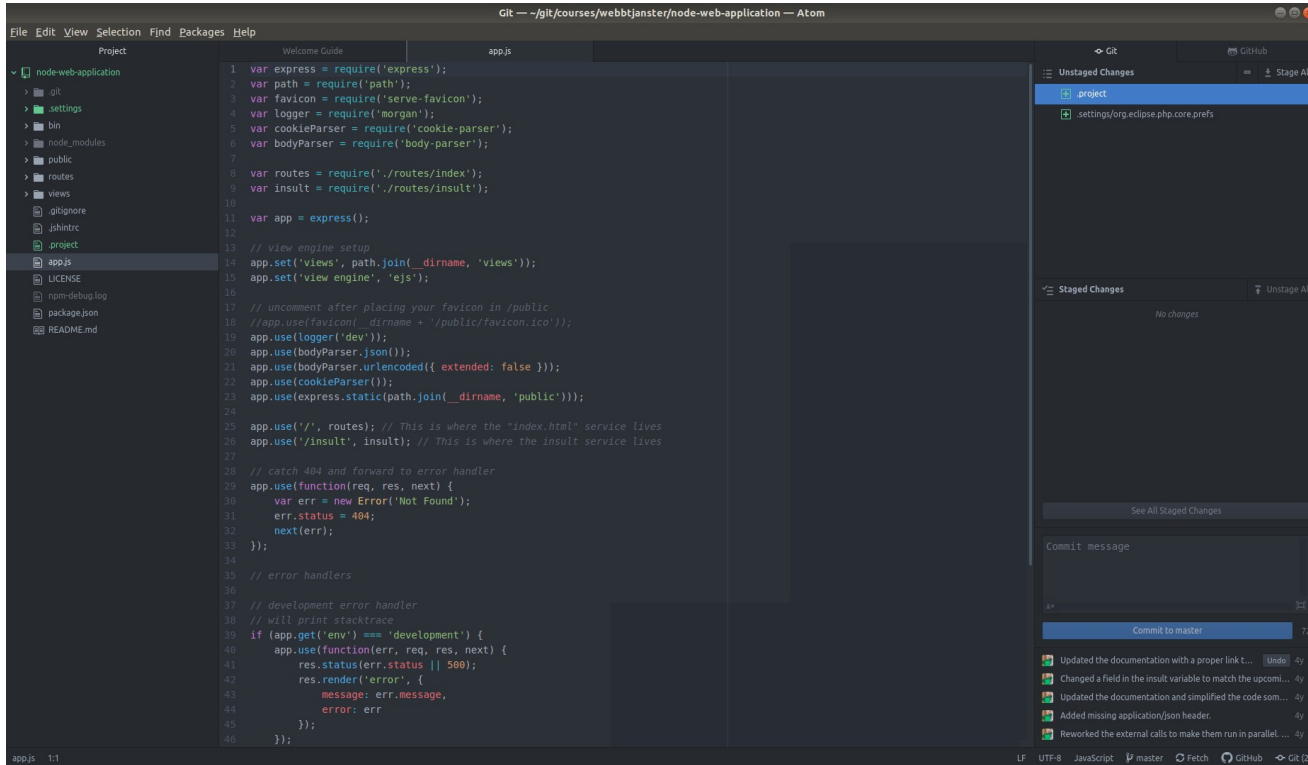
# Våra verktyg

**Textredigerare:** Atom

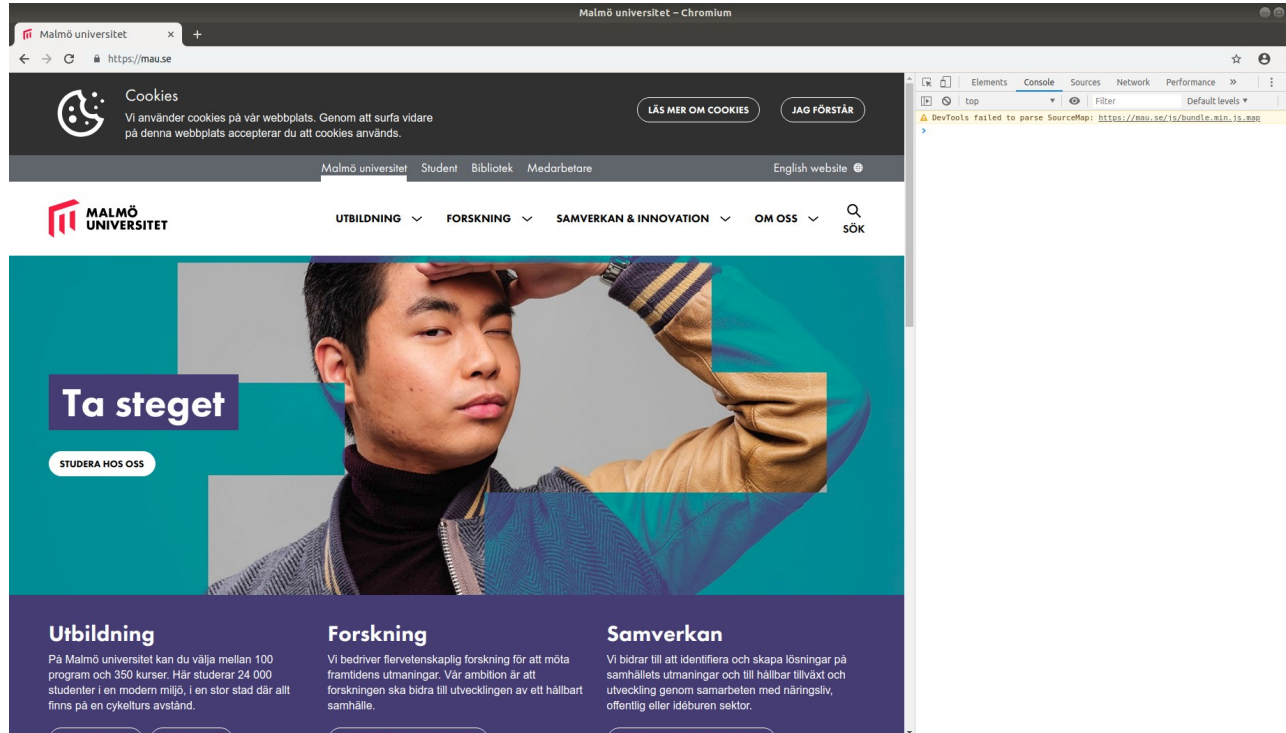
**Webbläsare:** Chromium med Chrome Developer Tools

**Onlineverktyg:** JSFiddle

# Textredigeraren Atom



# Webbläsaren Chromium



# Onlineverktyget JSFiddle

Simple function example - JSFiddle - Chromium

Simple function example - x +


← → ↻ https://jsfiddle.net/maclo/bvwvd0ao/

🔍 ⭐ ⚙

🔗 Run 📁 Save 🛠 Tidy 🗨 Collaborate <> Embed Update New and updated JS/CSS linters

⚙ Settings Sign in

Author



Mindy McAdams  
Gainesville, FL

Fiddle meta

Resources [URL](#) [cdns](#)

Async requests

Other (links, license)

HTML

```
<p>Type a number: <input type="text" id="f" size="4"></p>
<p>Type a different number: <input type="text" id="s" size="4"></p>
<button id="b" class="myButton">Get the Larger</button>
<p>The larger of the two numbers is:
<span id="write"></span></p>
```

JavaScript + No-Library (pure JS)

```
// get the button and make it respond to a click
var theButton = document.getElementById("b");
theButton.onclick = feedTheButton;

// simple function compares two numbers, returns the larger one
function greatestOfTwo( first, second ) {
  if ( first > second ) {
    return first;
  } else {
    return second;
  }
}

// this function runs each time the button is clicked
// the simple function is called within this one
function feedTheButton() {
  // get the two numbers from the text input fields
  // parseText changes string to number
  var box1 = parseInt( document.getElementById("f").value );
  var box2 = parseInt( document.getElementById("s").value );
  // run the function above with one and store what is returned in result
  var result = greatestOfTwo( box1, box2 );
  // write the result into the HTML
  var place = document.getElementById("write");
  place.innerHTML = result;
```

CSS

```
input[type=text] {
  font-family: Consolas, monospace;
  font-size: 1.1em;
  padding: 0.5em;
  text-align: right;
}

/* Inverse button CSS from http://www.bestcssbuttongenerator.com/ */
.myButton {
  -ms-box-shadow: 0px 1px 0px #0f7ffa;
  -webkit-box-shadow: 0px 1px 0px #0f7ffa;
  box-shadow: 0px 1px 0px #0f7ffa;
  background: -webkit-linear-gradient( linear, left top, left bottom, color-stop(0.05, #33bdef), color-stop(1, #019ad2) );
  background: -ms-linear-gradient( top, #33bdef 5%, #019ad2 100% );
  background: -webkit-linear-gradient( top, #33bdef 5%, #019ad2 100% );
  background: -ms-linear-gradient( top, #33bdef 5%, #019ad2 100% );
  background: -webkit-linear-gradient( top, #33bdef 5%, #019ad2 100% );
  background: linear-gradient( to bottom, #33bdef 5%, #019ad2 100% );
  filter: progId:DXImageTransform.Microsoft.gradient( startColorstr="#33bdef", endColorstr="#019ad2", GradientType=0 );
  background-color: #33bdef;
  -ms-border-radius: 5px;
  -webkit-border-radius: 5px;
  border-radius: 5px;
```

Type a number:

Type a different number:

Get the Larger

The larger of the two numbers is:

# Grunderna i Javascript

# Satser

Instruktioner i Javascript kallas *statements* eller *satser*.

Varje sats *kan* avslutas med ett semikolon (eller radbrytning)

```
console.log("hello, world!");
```

Flera satser kan placeras i ett block. Dessa avgränsas med *måsvingar* eller *kryllparenteser*

```
{  
  console.log("hello");  
  console.log("world");  
}
```

# Tilldelning

Variabler får sina värden med hjälp av ett = -tecken

```
a = 1; // Tilldelning  
b = true; // Tilldelning
```

# If- och switch-satser

If-satser används för att hantera villkor:

```
if (a == 2) {  
    // Do something  
} else if (a == 4) {  
    // Do something  
} else {  
    // Do something else  
}
```

Switch-satser fungerar på ungefär samma sätt:

```
switch (a) {  
    case 2:  
        // Do something  
        break;  
    case 4:  
        // Do something  
        break;  
    default:  
        // Do something else  
}
```



# Loopar

Två typer av loopar: while och for

```
while (true) {  
    // Do something  
}
```

```
for (var i = 0; i < 10; i++) {  
    // Do something  
}
```

# Variabler

- Löst typade (sen nästa slide)
- Deklareras på tre sätt:
  - var – äldre sätt, synligt för “alla”
  - let – nyare sätt, synligt endast i aktuellt block
  - const – nyare sätt, konstant och synligt i aktuellt block

# Operationer

## // Aritmetik

```
a = 1 + 2; // Addition
a = 2 - 1; // Subtraktion
a = 2 * 2; // Multiplikation
a = 2 / 2; // Division
a++; // a = a + 1
a--; // a = a - 1
b = !b; // B är nu falskt
```

## // Jämförelser

```
a == 2; // Är a lika med två?
a != 2; // Är a skilt från två?
a < 2; // Är a mindre än två?
a > 2; // Är a större än två?
a && b; // Är a och b båda sanna?
a && !b; // Är a sant och b falskt?
a || b; // Är a eller b sanna?
```

# Variabler

```
var a; // Initialiserad, inget värde
var b = 1; // Initialiserad, har ett värde
let c = true; // Initialiserad, har ett värde
const d = "hello"; // Initialiserad, har ett värde
console.log(a); // Skriver ut "undefined"
console.log(b); // Skriver ut "1"
console.log(c); // Skriver ut "true"
console.log(d); // Skriver ut "hello"

b = 5; // Nytt värde!
console.log(b); // Skriver ut "5"

d = "goodbye"; // Här går det fel!
```

# Datatyper

Javascript har fem datatyper (ish...)

- String
- Number
- Boolean
- Object
- Array

# Strängar - String

En lista av tecken:

```
var a = "hello, world!";
```

Längden av en sträng:

```
console.log(a.length); // Skriver ut "13"
```

Hämta ut ett tecken ur en sträng:

```
console.log(a[2]); // Skriver ut "l"
```

# Tal - Number

Vanliga hel- och flyttal:

```
var a = 1; // Heltal
```

```
var b = 1.5; // Flyttal
```

Och en specialare: NaN!

```
var c = NaN; // Inte ett nummer, men ändå ett nummer.
```

# Sant/falskt - Boolean

Representerar värdena sant och falskt:

```
var iAmWeazel = true;
```

```
var iRBaboon = false;
```



# Objekt - Object

Används för att hålla koll på många värden. Kan innehålla *vad som helst!*

```
var teacher = {  
  name: "Johan",  
  salary: 10000,  
  colleague: {  
    name: "Anton",  
    younger: true  
  }  
};
```

# Vektorer/listor - Array

En typ av objekt som håller koll på värden i en specifik ordning:

```
var a = new Array();  
a[0] = 1;  
a[1] = "two";  
a.push(3);  
console.log(a); // Skriver ut "[1, "two", 3]"  
  
var b = [1, "two", 3]; // Likadan som a!  
console.log(b); // Skriver ut "[1, "two", 3]"
```

# Datatyper

Några bonustyper

- null
- undefined
- function

# Lite mer avancerad Javascript

# Funktioner

Funktioner är återanvändbara samlingar av satser, dvs en eller flera rader av kod. De kan ta in *argument* och ge ett *returvärde*.

```
function a(a, b) {  
  return a + b;  
}
```

# Funktioner som variabler

Funktioner kan även vara variabelvärden:

```
var anka = function (a, b) {  
  return a + b;  
}  
console.log(anka(1, 2)); // Skriver ut "3"
```

Det gör att de kan anropas lite hur som helst:

```
var b = {  
  name: "addition"  
};  
b["add"] = anka;  
console.log(b.add(1, 2)); // Skriver ut "3"
```

# Synlighet

Se tavlan!

# Closures

```
function makeFunc() {  
  var name = 'Mozilla';  
  function displayName() {  
    alert(name);  
  }  
  return displayName;  
}  
var myFunc = makeFunc();  
myFunc();
```



# Batman!



<https://www.destroyallsoftware.com/talks/wat>