

Funktioner

Hoisting, Scopes and Callbacks

Function Declaration

```
function myFunction(){  
    //Do Stuff  
}
```

Namngiven funktion

Function Expression

```
var myFunction = function{  
    alert('Not Anonymous!');  
}
```

Fortfarande
namngiven
funktion

Anonymous function

```
(function(){  
    alert( 'Anonymous!' );  
});
```

Helt anonym

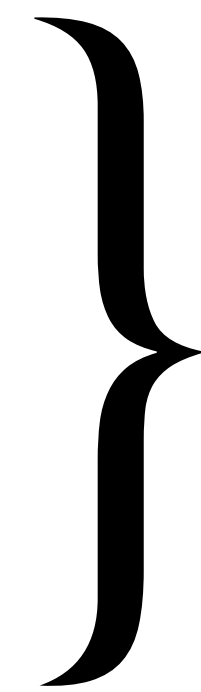
Scope

```
var a = 1;
```

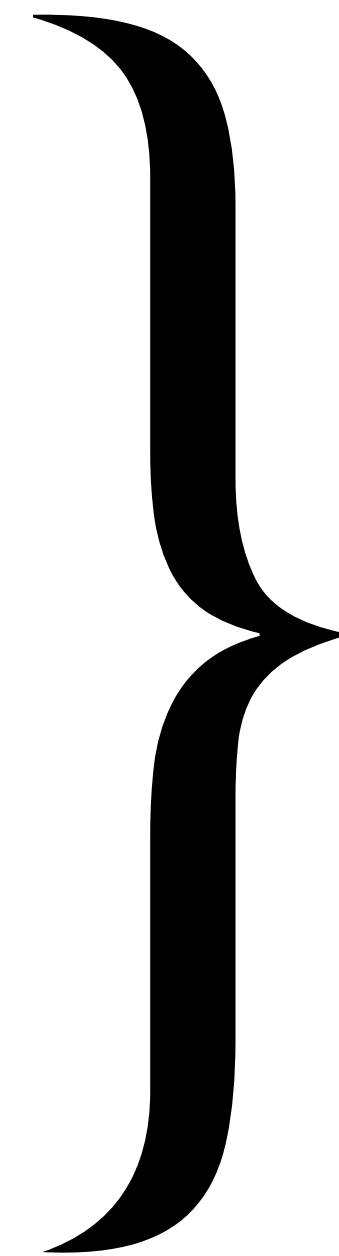
```
function myFunction() {
```

```
    //Do stuff
```

```
}
```



Funktionens
scope



Global
Scope

Scope access

- Varje funktion har ett eget scope
- Funktionen har tillgång till **alla yttre scope**
- **Global** har dock **inte** tillgång till funktionens scope

Scope

```
var name = "Jesper";  
  
function sayMyName(){  
    var name = "Nully";  
  
    console.log(name);  
  
}
```

Funktionens
variabel prioriteras
inuti funktionen!

Funktioner “skjuts upp”

`myFunction();` `<- funktionen kan köras`

`function myFunction(){}
myFunction();`

`myFunction2();` `<- funktionen kan inte köras, inte deklarerad`

`var myFunction2 = function(){}
myFunction2();`

Hoisting

- Function declaration samt variable declaration “**hoistas**”
- De deklarereras **i toppen av scopet**
- “Skjuts upp” i scopet

Hoisting

- Samma gäller för variabler

```
function myFunction(){
```

```
  var a = 1;
```

```
  console.log("Värdet på a: " + a + "Värdet på b: " + b );
```

```
  var b = 2;
```

```
}
```

undefined



Variabel Hoisting

```
function myFunction(){  
    var a = 1;  
  
    var b;    < – deklarerad men inte tilldelad  
  
    console.log("Värdet på a: " + a + "Värdet på b: " + b );  
  
    b = 2;  
  
}
```

Hoisting

```
function myFunction(){  
  var a = 1;  
  
  var b = 2;  
  
  console.log("Värdet på a: " + a + "Värdet på b: " + b );  
}
```

best practice att deklarera i
början av scopet

Funktioner som argument

- Funktioner kan returnera värden
- Funktioner kan användas som parametrar
- Kan skickas runt nästan hur som helst

setInterval

- Inbyggd funktion som gör något i ett visst intervall
- **Tar två parametrar: en funktion samt ett intervall**
- `setInterval(function, 1000);`



Millisekunder

Exempel namngiven funktion

```
var myFunction = function(){  
    console.log( 'Every Second' );  
}
```

Funktion som parameter

```
setInterval(myFunction, 1000);
```

A blue rectangular box containing the text "Funktion som parameter" has a black arrow pointing from its bottom-left corner to the "myFunction" argument in the "setInterval" function call of the code below.

Exempel anonym funktion

- Använd en anonym funktion som argument

```
setInterval(function(){  
    console.log('Every second!')  
}, 1000);
```


Callbacks

- Funktioner kan skickas med som argument
- Funktioner som skickas med som argument kan kallas på när funktionen körts klart
- Såkallade **callbacks**

Callback

```
var myFunction = function(value, callback){  
    //Do something with value  
    callback();  
}
```

Callbacks

```
function myFunction(value, callback){
```

```
    value + 1;
```

```
    callback(value);
```

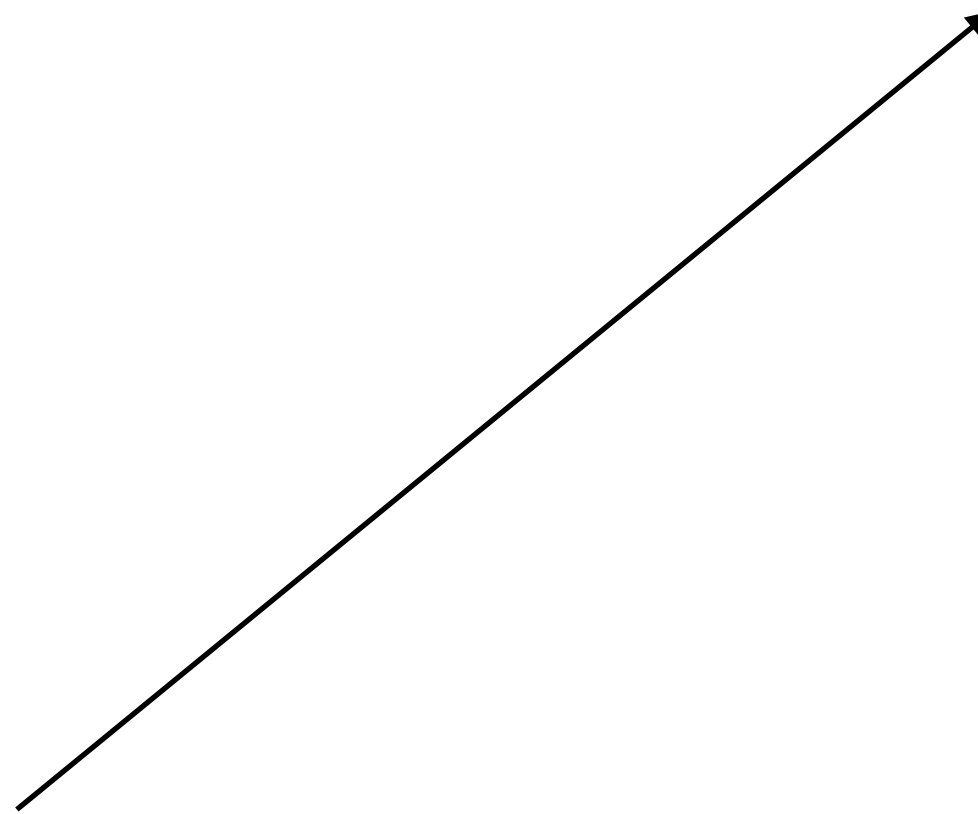
```
}
```

```
function logger(value){
```

```
    console.log(value);
```

```
}
```

```
myFunction(10, logger);
```



Funktioner i funktioner

```
function myOuterFunction(){  
  function myInnerFunction(){  
  }  
  myInnerFunction(); ← Kalla direkt på den inre funktion  
}  
  
myOuterFunction(); ← Båda funktionerna körs
```

Closures

- När vi skapar en funktion inuti en annan funktion skapas automatiskt en **closure**
- Den inre funktionen har tillgång till den yttre funktionens variabler **även fast den yttre funktionen har körts klart**
- Funktionen “kommer ihåg” var den skapades.

Closures

```
function myOuterFunction(){  
  var name = 'Jesper';  
  
  function myInnerFunction(){  
  
  }  
  
  return myInnerFunction();  
  
}
```

```
var inner = myOuterFunction();
```

inner(); <— har tillgång till variabeln name

Övningar :O

- Finns numera på vår favoritplats GitHub!
- <https://github.com/FEND16/javascript1>
- Övningar finns under '**exercises**'
- **Skicka ett slack-meddelande med ditt GitHub-namn så lägger jag till dig i FEND16-organisationen på GitHub**