



Opgaver – Closures og Callbacks

Opgave (Closures - 1)

Givet følgende kode:

```
function buildFunctions(){
    var arr=[];
    for (var i=0; i<3; i++){
        arr.push(function() {console.log(i);})
    }
    return arr;
}

var fs = buildFunctions();
fs[0]();
fs[1]();
fs[2]();
```

- a) Hvad er det forventede output? Giv et bud og test koden i VSCode.
- b) Sandsynligvis blev du lidt overrasket (mange gjorde i hvert fald!) – Redegør for hvorfor der udskrives som der gør (Hint: der er tale om closures fænomenet).
- c) Ændre koden så resultatet bliver som forventet (Hint: benyt let og en lokal variable)
Redegør for hvorfor koden nu virker efter hensigten 😊

Opgave (Closures – 2, function factories)

Givet følgende kode:

```
function makeGreeting(language) {
    return function(firstname, lastname){
        if (language==='en'){console.log('Hello ' + firstname + ' ' +
lastname)};
        if (language==='dk'){console.log('Hej ' + firstname + ' ' +
lastname)};
        if (language==='es'){console.log('Hola ' + firstname + ' ' +
lastname)}
    }
}
```

```

    }
}

var greetEnglish = makeGreeting('en');
var greetDanish = makeGreeting('dk');
var greetSpanish = makeGreeting('es');

greetEnglish('John', 'Doe');
greetDanish('Henrik', 'Høltzer');
greetSpanish('Pablo', 'Fuentes');

```

- Hvad er det forventede output? Giv et bud og test koden i VSCode.
- Redegør med Closures fænomenet in mente hvad der sker i koden.
- Omskriv den anonyme funktion (vores function expressions) til ES6 (Arrow-function).

Opgave (Closures – 3, function factories)

- Lav en ny funktion *makeAdder(x)* der skal være en "function factories", idet den skal returnere et nyt funktions objekt (via et function expressions). Den anonyme funktion der skal returneres skal tage et nyt argument og addere med x.
- Test din kode med:

```

var add5 = makeAdder(5);
var add10 = makeAdder(10);
var addHello = makeAdder("Hello ");

console.log(add5(2)); // 7
console.log(add10(2)); // 12
console.log(addHello("Henrik")); // Hello Henrik

```

Opgave (Callback function)

- Afprøv nedenstående kode I VSCode:

```

function tellMeWhenDone(callback){
    // .... some work
    callback();
}

tellMeWhenDone(function(){console.log("Im' done!");});

```

```
tellMeWhenDone(function(){console.log("All done!");});
```

b) Find selv på et eksempel, hvor du afprøver "templaten" på en callback function 😊