

JavaScript-funktioner til beregning af Fibonacci-tal

Vi har tidligere set på beregning af Fibonacci-tal i Delphi. Nu skal vi se, hvordan man laver tilsvarende algoritmer i Javascript.

1 Beregning med rekursiv funktion

```
<HTML>
<HEAD>
<TITLE>Recursive computation (with a function that calls itself) of the nth
number of the Fibonacci sequence </TITLE>
<BODY BGCOLOR="26B81B" TEXT="000000" LINK="2411EE" VLINK="F30C14"
ALINK="DE05FA">

<CENTER>
<H1>Recursive computation (with a function that calls itself) of the nth
number of the Fibonacci sequence</H1>
</CENTER>
</HEAD>

<script language="JavaScript">
<!-- Hide code from old browsers
function Fib(n){
if(n==0)
{return 0}
else if (n==1)
{return 1}
else {
return Fib(n-1) + Fib (n-2)}}
function compute(obj)
{obj.result.value="The " +obj.input.value + "th Fibonacci number is " +
Fib(eval(obj.input.value))}

// End code hiding -->
</script>

<form name="evalForm">
Enter an integer: <input type="text" name="input" size="20"><br>
Result: <input type="text" name="result" size="40"><br>
<input type="button" value="Click For Answer" onClick="compute(this.form)">
</form>

</BODY>
</HTML>
```

Undersøg, om denne Fibonacci-algoritme er rigtig, og se hvilke værdier der bliver beregnet. Sammenlign med Delphi-noten om rekursive og iterative funktioner. Forsøg at tage tid på nogle af de (tidsmæssigt) langvarige beregninger.

Forklar, hvilke objekter og metoder der indgår i funktionen, gerne i et skema, så du får det systematiseret. Sammenlign med de tilsvarende objekter og metoder i Delphi, fx editboks, label, name, caption mv., og de funktioner der bruges i forbindelse med editboksens indhold (edit.text).

2 Beregning med iterativ funktion

```
<HTML>
<HEAD>
<TITLE>
Recursive computation (with a function that calls itself) of the nth number
of the Fibonacci sequence
</TITLE>
<BODY BGCOLOR="26B81B" TEXT="000000" LINK="2411EE" VLINK="F30C14"
ALINK="DE05FA">

<CENTER>
<H1>Iterative computation (with a loop) of the nth number in the Fibonacci
sequence</H1>
</CENTER>
</HEAD>

<script language="JavaScript">
<!-- Hide code from old browsers
function Fib(n){
var i
var earlier=0
var last=1
var result
if(n==0)
{return 0}
else if (n==1)
{return 1}
else {for (i=1; i<=n; i++)
{result=last
last=last + earlier
earlier=result}}
return result}

function compute(obj)
{obj.result.value="The " +obj.input.value + "th Fibonacci number is " +
Fib(eval(obj.input.value))}

// End code hiding -->
</script>

<body>
<form name="evalForm">
Enter an integer: <input type="text" name="input" size="20"><br>
Result: <input type="text" name="result" size="40"><br>
<input type="button" value="Click For Answer" onClick="compute(this.form)">
</form>
<br>

</body>
</html>
```

Undersøg, om denne Fibonacci-algoritme er rigtig, og se hvilke værdier der bliver beregnet. Sammenlign med Delphi-noten om rekursive og iterative funktioner.

Forsøg at tage tid på nogle af de (tidsmæssigt) langvarige beregninger, og sammenlign med den rekursive beregningsform i afsnit 1.

Forklar også her, hvilke objekter og metoder der anvendes. Sammenlign igen med Delphi.

3 JavaScript-opgaver

- 1 Lav en JavaScript-funktion, der kan beregne $n!$
- 2 Lav en JavaScript-funktion med 2 input-bokse, en output-boks og en knap, og prøv om du kan lave regnestykker, fx således at man - efter at have udfyldt de to input-bokse - kan trykke på knappen og få beregnet resultatet i output-boksen.
- 3 Udvid opgave 2, så du kan udføre alle regningsarter. Så har du en lille JavaScript-lommeregner!
- 4 Kan man tilføje en tidsmåling, så man kan få at vide, hvor lang beregningstiden har været i de to fibonacci-eksempler?