

Parallel Prozesse/Multithreading

- C++
- Racket
- Java
- Javascript

1 C++

Arbeitsweise:

Das Dokument wird Stück für Stück abgearbeitet.
Es bedarf eines zusätzlichen include: 'omp.h'

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>

int main(void){
    int i;
    omp_set_num_threads(1);
    #pragma omp parallel for
    for(i=0;i<500;++i){
        const int id = omp_get_thread_num();
        printf("Hello World from thread %d\n", id);
        if(id==0){printf("There are %d threads\n", omp_get_num_threads());}
    }
    return EXIT_SUCCESS;
}
```

2 Racket

Arbeitsweise:

Racketscript wird Stück für Stück abgearbeitet.
Im Vergleich mit anderen Sprachen, das kürzeste überhaupt.
Racket ist standardmäßig für Parallelprozesse ausgelegt.

```
#lang racket
(thread
  (lambda()
    (letrec
      [(loop
        (lambda(x)
          (sleep 1)
          (display x)
          (loop(+ x 1))))])
      (loop 0))))
```

3 Java

Arbeitsweise:

Aus der main methode wird die Klasse des Thread aufgerufen und mit start() aktiviert.
Für den Thread ist zwar 'extends Thread' notwendig, doch es muss keine Packet importiert werden.
Standardmäßig wird das Script Stück für Stück abgearbeitet.

```
public class threading{
    public static void main(String[] args)throws Exception{
        Processing process=new Processing();
        process.start();
    }
}
class Processing extends Thread{
    private int counter=0;
    public Processing()throws Exception{}
    public void run(){
        try{
            while(true){
                Thread.sleep(1);
                System.out.println(""+counter);
                counter++;
            }
        }
        catch(InterruptedException e){}
    }
}
```

```
}
```

4 Javascript

Arbeitsweise:

In einem 'worker.js' läuft ein separater Prozess zum Hauptdokument 'main.htm'
Erst wird 'main' vollständig abgearbeitet, dann der oder die Prozesse aufgerufen.
In diesem Beispiel durch 'alert()' verdeutlicht.

```
//document:
main
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Multithreading</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script type="text/javascript" language="JavaScript">
function createThread(){
  var worker = new Worker('worker.js');
  worker.addEventListener('message',function(evt){for(var i=0;i<1000;i++){alert(evt.data);}},false);
  worker.postMessage('Hello World'); // Send data to our worker.
  alert("end of document");
}
</script>
</head>
<body onLoad="createThread()">
</body>
</html>

//document:
worker
// JavaScript Document
self.addEventListener('message',function(evt){self.postMessage(evt.data);},false);
```