

Extension Function: script

- `java:script(script datei, filter, param)`
- `java:script(script id, inline code, filter, param)`

Einbindung von EcmaScript als Extension.

Einbindung wie [art: script](#) mit folgenden Ausnahmen:

- **es muss "var result" definiert sein da diese den Rückgabewert festlegt**
- Der \$ Scope liegt hier auf dem filter Objekt, da kein Handler zur Verfügung steht.
- Variable root: zeigt auf /* des Ausgabedokuments
- Variable param: übergebener Parameter

Beispiele

```
*****
*** XSL Code
*****
<xsl:stylesheet version="1.0" exclude-result-prefixes="xsl java jart">
  <xsl:output encoding="Windows-1252" method="html"/>
  <xsl:param name="j-j-filter"/>
  <xsl:variable name="script1">
    importPackage(Packages.jart);

    var cob = new CacheObj();
    var result = cob.locked;
  </xsl:variable>
  <xsl:template match="/">
    <html>
      <head>
        <title>Test 4</title>
      </head>
      <body>
        <h1>Test 4</h1>
        <div>
          <xsl:value-of disable-output-escaping="yes" select="java:script("script2.js", $scrip
        </div>
        <div>
          <xsl:value-of disable-output-escaping="yes" select="java:script("test4.js", $j-j-fi
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

*****
*** test4.js:
*****
var result = "test in xsl " + $.params.get("xslFile") + "<br/>";
for(var i = 0; i < parseInt(param); i++){
  result += "[POS:" + i + "]";
}

*****
*** Ergebniss:
*****

Test 4
true
test in xsl test4.xsl
```

[POS:0][POS:1][POS:2][POS:3][POS:4][POS:5][POS:6][POS:7][POS:8][POS:9][POS:10][POS:11][P(