

Bisher haben wir das `document` Objekt zum Selektieren von Elementen im DOM verwendet. Dafür verwendeten wir `.getElement...` und `.querySelector...` Wir nutzen das `document` Objekt auch für Nodes. Es kann aber mehr. Hier eine kleine Auswahl von Eigenschaften und Methoden des `document` Objekts ...

JS

<code>document.URL;</code>	← Gibt die Internetadresse URL des Dokuments zurück.
<code>document.domain;</code>	← Gibt die Domain des Dokuments zurück.
<code>document.characterSet;</code>	← Gibt den Zeichensatz des Dokuments zurück.
<code>document.title;</code>	← Gibt den Titel des Dokuments zurück.



Ein neuer Titel wird mit `document.title = "Neuer Titel";` gesetzt!

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Mein Dokument</title>
  </head>

  <body>
    <div class="inhalt">
      <h1>Dokumenteigenschaften</h1>
      <p id="ausgabe"></p>
    </div>
    <script>
      var ausgabe = "Hier die Eigenschaften: <br>";
      ausgabe = ausgabe + "<br>URL: " + document.URL;
      ausgabe = ausgabe + "<br>Domain: " + document.domain;
      ausgabe = ausgabe + "<br>Zeichensatz: " + document.characterSet;
      ausgabe = ausgabe + "<br>Titel: " + document.title;
      document.getElementById("ausgabe").innerHTML = ausgabe;
      document.title = "Dokument geladen!";
    </script>

  </body>
</html>
```

JS

<code>document.hasFocus();</code>	← Ermittelt ob das Dokument fokussiert ist (true/false).
<code>document.readyState;</code>	← Ermittelt ob ein Dokument vollständig geladen wurde!

```
<script>
  alert(document.readyState);
  function meineFunktion() {alert(document.readyState);}
</script>
<button onClick="meineFunktion();">Fertig</button>
```

JS

`document.designMode = "on";` ← Macht das Dokument editierbar.



Mit `document.designMode = "on";` kann man das gesamte Webdokument im Browser editieren (zum Beispiel einen Text hinzufügen oder löschen)!
Mit `= "off"` wird das Editieren deaktiviert.