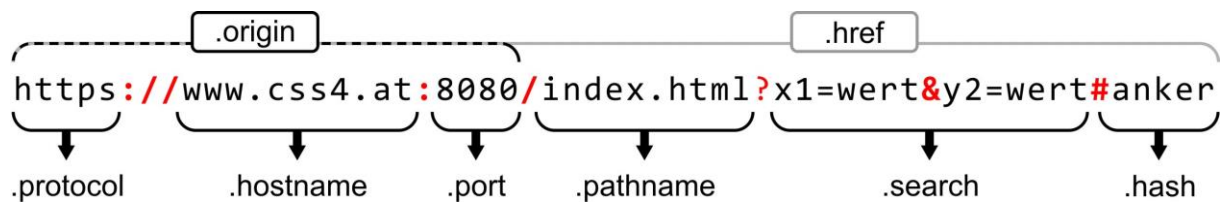


Mit dem `Location` Objekt (Unterojekt von `window`) hat man Zugriff auf die vollständige URL der aktuellen Webseite. Alle Eigenschaften des `location` Objekts können sowohl ausgegeben als auch gesetzt werden! URL steht für eng. Uniform Resource Locator.



JS `location.href` ← Die gesamte URL



Definiert die gesamte URL (vom Protokoll bis zum Anker). Mit `location.origin` bekommt man das Protokoll, den Hostnamen und den Port geliefert.

JS `location.protocol` ← Protokoll bzw. Schema



Liefert bzw. setzt das Protokoll (auch Schema genannt), einer URL. Das kann HTTP, HTTPS, FTP aber auch mailto, file, news usw. sein.

JS `location.hostname` ← Definiert den Hostnamen (z. B. `www.css4.at`)

JS `location.port` ← Port



Die Angabe des Ports erlaubt die Ansteuerung eines TCP-Ports. Wird kein Port angegeben, so wird der Standard-Port des jeweiligen Protokolls verwendet – zum Beispiel bei HTTP 80, bei HTTPS 443 und bei FTP 21. Mit `location.host` wird Hostname und Port definiert.

JS `location.pathname` ← Definiert den Pfad mit Datei (z. B. `/html/index.html`)

JS `location.search` ← Abfragestring z. B. Übergabewerte



Liefert bzw. setzt Übergabewerte in Form eines "Query-String". Mit einem Fragezeichen (?) beginnt der Query-String. Das Fragezeichen wird ebenfalls ausgegeben. Mehrere Übergaben sollten durch ein kaufmännisches Und (&) getrennt werden.

JS `location.hash` ← Definiert den Anker (z. B. `#anker`)

Location Methoden

JS `location.reload()` ← Ladet eine Seite neu

JS `location.replace()` ← Ersetzt eine Webseite durch eine andere.

JS `location.assign()` ← Gleich wie `.replace()` nur ohne 'Zurück-Button'.

History Methoden

JS `history.back()` ← Öffnet die vorherige Seite nochmals (Zurück-Button).

JS `history.forward()` ← Ladet die nächste Seite aus der `history` Liste.

JS `history.go()` ← Springt innerhalb der `history` Liste.
z. B. `history.go(-2)`; springt zwei Seiten zurück.