

ClientServer: GET und Ajax

1. Vorbereitung

1) Assoziative Arrays

Der Unterschied liegt darin, dass assoziative Arrays Wörter statt Zahlen als Index benutzen.

→ Index wird hier als „Schlüssel“ oder key bezeichnet

Vorteil: Code wird besser lesbar und weniger fehleranfällig

Nachteil: umständlich

2) Zeile 27

- Variable query mit Typ AssocStringString

- es wird gefragt ob die Antwort eine url ist, wenn ja (true) wird die url in ein JavaScript Objekt umgewandelt (query)

3) For in Schleife

- mit ihr kann über alle Eigenschaften eines Objektes iteriert werden

- in Zeile 30 werden die Eigenschaften des Objektes query ausgegeben

- mit jedem Schleifendurchgang wird die Konsole um eine Objekteigenschaft erweitert

- Schleife läuft so lange, wie es verfügbare Objekteigenschaften gibt

4) Access-Control-Allow-Origin

Bedeutet, dass die Datei auf eine Quelle eines anderen Bereichs zugreift und sie auf diese Ursprungsquelle auch zugreifen darf.

4. Ajax

2)

Init: - nimmt Event entgegen und ruft dann setupColorDivs() auf

setupColorDivs: - string Array mit Farben („rot“, „grün“, „blau“)
- divs Variable wird erstellt und greift auf die NodeList zu in der alle div-Elemente stehen
- for-Schleife: solange wie die Länge aller divs,
→ übergibt Hintergrundfarbe und hängt einen EventListener an das Div → ruft bei Click die Funktion handleClickOnDiv auf

handleClickOnDiv: - Event Click wird entgegengenommen
- style holt sich die Information zu dem Aussehen (Farbe) der Divs
→ geht mit sendRequest Funktion weiter

sendRequest: - xhr Daten der URL werden erhalten
- xhr open → URL wird in dem Server localhost 8100 geöffnet

- Info der Farbe wird hinzugefügt
- true: Programm läuft asynchron ab
- an xhr wird ein EventListener angehängt
- bei readystatechange (bei Veränderung) wird die Funktion handleStateChange aufgerufen
- xhr.send() = URL wird abgeschickt

handleStateChange: - if: wenn Status gleich ist wie bei abgeschlossene Anfrage, dann soll in der Konsole ready, type, status, response und text der URL ausgegeben werden

- alert mit der response poppt auf

3)

XMLHttpRequest ist ein JavaScript Objekt, das von Microsoft entwickelt und von Mozilla, Apple, und Google übernommen wurde. Es bietet einen einfachen Weg, Daten von einem URL zu erhalten. Trotz seines Namens kann man mit XMLHttpRequest jede Art von Daten laden, nicht nur XML, und es unterstützt auch andere Protokolle als HTTP.

4)

Kunstwort für Asynchronous JavaScript und XML. Auf JavaScript basierende Technik, die die asynchrone Übertragung von Daten zwischen Client und Server betrifft.