

Server.ts Tabelle und Sonstiges

Zeile:	Was die Zeile macht:
1	Es wird ein http Objekt im Code erstellt
2	-
3	Ein neuer Namespace wird deklariert
4	Starting server wird in der Konsole ausgegeben
5	Variable „port“ vom Typ number wird erstellt und wird ein Port zugewiesen
6	Bedingung für „port“ wird deklariert
7	„port“ wird der Wert 8100 zugewiesen
8	-
9	Variable „server“ vom Typ http.server wird erstellt und bekommt die Zuweisung http.createServer(); Dadurch wird mein PC zum Server.
10	Server wird ein „ request“ listener hinzugefügt, der auf die Funktion handleRequest verweist
11	Server wird ein „listening“ listener hinzugefügt, der auf die Funktion handleListen verweist
12	server.listen(port); Erstellt einen listener auf den festgelegten Port (Z.7): 8100
13	-
14	Die Funktion handleListen wird deklariert. Es werden keine Werte übergeben, oder zurückgegeben, da, void.
15	Listening wird in der Konsole ausgegeben
16	-
17	-
18	Die Funktion handleRequest wird deklariert. Parameter _request wird entgegen genommen(Anfrage an den Server: http.IncomingMessage. Parameter _response wird entgegen genommen und die die Antwort zurück an den User. (http.ServerResponses).
19	I hear voices wird in der Konsole ausgegeben
20	-
21	Setzt fest, dass im header mit bestimmten Attributen geantwortet wird. Setzt eine bestimmte header value fest.
22	Erlaubt einer Webapplikation, oder dem Browser, auf einem Server ausgeführt zu werden und dabei auf Ressourcen eines anderen Servers zugreifen zu können.
23	-
24	Die Antwort der Anfrage wird in die Url geschrieben
25	-
26	Die Antwort wird beendet.
27	-
28	-

Server.ts Tabelle und Sonstiges

Aufgabe 6.2.4:

Durch einen Server Reload wird die URL im Parameter `_request` gespeichert. Dann wird das URL Attribut von `_request` ausgelesen und ins HTML geschrieben.

Aufgabe 6.2.5:

In Zeile 19 habe ich den Inhalt von `console.log` angepasst (`_request.url`). Dadurch wird der eingegebene Test im Terminal angezeigt.

Aufgabe 6.2.6:

Mit der Tastenkombination Control - C.