

## Aufgabe 6

### Nr.1:

ZEILE	KOMMENTAR
1	Importiert ein http Modul von Node
3	Ein neuer Namespace wird definiert
4	Auf der Konsole wird Starting Server ausgegeben
5	Variable port des typs number speichert einen string (process.env.PORT), der zur number umgewandelt wird. Dieser string weist den Server darauf hin, auf welchen Listening Port gehört werden soll
6	Wenn die vorher deklarierte Variable port nicht zutrifft, dann soll der folgende port verwendet werden
7	Alternativer Wert für Port wird auf 8100 festgelegt
9	Variable „Server“ vom Typ „http.Server“ wird erstellt und bekommt die Zuweisung http.createServer. Dadurch wird der PC zu einem http-Server.
10	Server wird der addListener request hinzugefügt, der auf die Funktion handleRequest verweist
11	Server wird der addListener listening hinzugefügt, der auf die Funktion handleListen verweist
12	Es wird ein Listener erstellt, der auf den festgelegten Port hört.
14	Die Funktion handleListen wird deklariert. Es werden keine Parameter entgegengenommen, weil (), und keine Werte zurückgegeben, weil :void
15	Auf der Konsole wird Listening ausgegeben
18	Die Funktion handleRequest wird deklariert. Es wird die Anfrage/Anforderung entgegengenommen (_request: http.IncomingMessage) und die die Antwort zurück an den Benutzer (_response: http.ServerResponse)
19	Auf der Konsole wird I hear voices ausgegeben
21	Setzt fest, dass im Header mit bestimmten Attributen geantwortet wird (Styling) (?)

22	Eine Antwort, die dem Browser erlaubt, Code von jedem Ursprung auszuführen
24	Die Antwort der Anfrage wird in die URL geschrieben
26	Die Antwort wird beendet

#### **Nr.2.4:**

Wenn man den Server neu lädt, wird die URL im Parameter `_request` abgespeichert. Danach wird die URL von `_request` ausgelesen und in das HTML geschrieben.

#### **Nr.2.5:**

In Zeile 19 muss der Inhalt von `console.log` in `_request.url` geändert werden, damit der Text im Terminal erscheint.

#### **Nr.2.6:**

Mit Strg+C wird der Server beendet.