

Übung JavaScript switch Expression

Ausgangslage

Bei der Übung ifelse haben Sie einen Code dem Http Code Range zugewiesen. In dieser Übung soll nun der exakte Text zu einem HTTP-Code ausgegeben werden. Die HTTP Codes definieren wir wiederum in einer Liste. Das HTTP-Protokoll definiert die folgenden Informationale und Successful-Codes mit Text: 100 Continue 200 OK 201 Created 202 Accepted 203 Non-Authoritative Information 204 No Content 205 Reset Content 206 Partial Content. Das HTTP-Protokoll definiert die folgenden Redirection-Codes mit Text: 300 Multiple Choices 301 Moved Permanently 302 Moved Temporarily 303 See Other 304 Not Modified 305 Use Proxy. Das HTTP-Protokoll definiert die folgenden Client-Error-Codes mit Text: 400 Bad Request 401 Unauthorized 402 Payment Required 403 Forbidden 404 Not Found 405 Method Not Allowed 406 Not Acceptable 407 Proxy Authentication Required 408 Request Timeout 409 Conflict 410 Gone 411 Length Required 412 Precondition Failed 413 Request Entity Too Large 414 Request-URI Too Long 415 Unsupported Media Type. Das HTTP-Protokoll definiert die folgenden Server-Error-Codes mit Text: 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout 505 HTTP Version Not Supported. Wir empfehlen den Einsatz des Visual Code Editors mit dem Code Runner Plugin und node.js. Als Vorlagen können Sie das folgende Programmgerüst verwenden:

```
<pre>console.log('Start HttpStatusCode ...');
let httpCodes = [100, 200, 300, 404, 500, 600, -1];
for (let i = 0; i < httpCodes.length; i++) {
  // TODO
}
console.log('... HttpStatusCode Done');
```

Vorgehen

Lösen Sie bitte die Aufgabe wie folgt: Erzeugen Sie die Datei HttpStatusCodeText.js z.B. in VisualCode in einem eigenen Verzeichnis z.B. `basic/switch`. Kopieren Sie die obige Vorlage in die Datei hinein. Programmieren Sie die HTTP Codes mit den korrekten Texten anstelle der TODO Sektion in der for-Schleife aus. Speichern Sie die Datei immer wieder und testen Sie das Resultat via Run aus. Testen Sie mit weiteren HTTP Codes und auch invaliden Codes wie z.B. `Gugus`.

Lösung

Mögliche Lösungen finden Sie hier. Javascript ES6 unterstützt mehrere for Schleifenvarianten, die Lösungen zeigen dies auf.

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz. Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsausschluss. Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

simtech-ag.ch/ecmascript-training-kurs-javascript-es6-programmierung-ressourcen-übung-java-switch-expression