

4. Übung Webprogrammierung

Aufgaben 1 bis 6 bis 13. April. Lösungen sollen online verfügbar sein und präsentiert werden können.

1. Gruppe: Erweitern Sie Ihren Apachen so, dass auch HTTPS unterstützt wird. Erstellen Sie dazu ein *self-signed certificate*. Notieren Sie die Schritte die Sie durchgeführt haben, bzw. erklären Sie die Schritte die Sie in einem Tutorial gefunden haben. HTTPS mit dem *self-signed certificate* soll auf Port 8443 angeboten werden. Zeigen Sie im Browser, dass Ihr Zertifikat funktioniert.
2. Gruppe: Installieren Sie ein SSL Zertifikation von <https://letsencrypt.org/> für die Verwendung auf Port 443. Was ist der Unterschied zu einem self-signed Zertifikat?
3. Gruppe: Spielen Sie mit <https://www.ssllabs.com/ssltest/>. Welche *Grade* erreicht die e-commerce site Ihres Vertrauens? Welche Sicherheit erreicht Ihre *Meise*? Testen Sie mit dem Skript `testssl.sh` von <http://testssl.sh/> Ihren Server!
4. Jeder: Erstellen Sie eine HTML-Seite *ohne* CSS die in etwa so aussieht wie unten angegeben. Überprüfen Sie das Ergebnis mit dem Validator. Vergleichen Sie das Ergebnis mit zumindest zwei Browsern (Screenshot!)

Mit HTML kann man einen Text auf viele Arten gestalten:

Natürlich kann man einzelne Wörter **fett**, *kursiv* und unterstrichen darstellen. Dies ist sogar mit unterschiedlichen HTML-Tags möglich: **wichtig**, *hervorgehoben*

Aber man kann auch Text ^{hochstellen}, _{tiefstellen}, ~~durchstreichen~~, usw.

Als ProgrammiererInnen haben wir besonderes Interesse, Code darzustellen: `void func(int &x);`

Und wie bringt man Text ohne Umformatieren auf die Seite?

```
<html>
  fix&foxy
</html>
```

Und zwar genau **so**???

Einzelne **Z**eichen oder Wörter können angepasst werden – viel besser geht's es mit CSS, dazu später mehr!

5. Jeder: Erstellen Sie ein Such-Formular, das Anfragen an <http://google.com/search> per GET request richtet (`q=`). Ihr Formular soll die Anzahl der Ergebnisse einschränken können (`num=`, per Dropdown-Auswahl).
Lesen Sie: <http://moz.com/ugc/the-ultimate-guide-to-the-google-search-parameters>
6. Jeder: Was ist der Unterschied zwischen HTTP GET und POST? Was sind die Vor- und Nachteile? Recherchieren Sie! Zeichnen Sie die jeweiligen requests auf und analysieren Sie die Unterschiede!