

1. HTML ist keine Programmiersprache, sondern eine „Auszeichnungssprache“.
2. Semantische Elemente geben in HTML den Aufbau an, z.B. <footer>, ...
Das waren vor HTML5 alles div Blöcke mit entsprechenden Ids.
3. CSS ist eine Programmiersprache, die für das Aussehen einer Website zuständig ist.
Die drei Arten für die Einbindung sind:
 - Inline
 - in einem HTML <style> tag
 - Als extra Datei über rel=“stylesheet“
4. ID oder class, Typ
5. Für Laufzeitfunktionen, die nicht statisch sind, Eingabepfung, DOM-Manipulation, ...
6. Hoisting: Variable kann benutzt werden bevor sie deklariert wurde
Blöcke werden mit {} deklariert und alle Dinge in dem Block deklariert, sind außerhalb nicht sichtbar → Hoisting entfällt
7. == transformiert beide Operanden zum gleichen Typ vor Vergleich, === nicht
8. Wenn eine Funktion wie eine Variable behandelt wird, ist sie ein first class citizen
9. Alle Teile des Dokuments werden in einem Baum dargestellt mit Dokument selbst als Wurzel
10. Insg. 12 verschiedene, meist genutzt: Dokumentknoten (Wurzel), Elementknoten (HTML Element), Attributknoten (src, alt,...), Textknoten (Text in HTML Element)
11. element.innerHTML = ...
12. Wartet auf Ereignisse auf der Website und kann über Event-Handler umgesetzt werden
13. NPM = Paketverwalter für js. In der package.json werden alle installierten Pakete angezeigt mit allen Attributen. Dev Abhängigkeiten: npm installiert alle Pakete benötigt für lokale Entwicklung und Tests. Normale Abhängigkeiten: npm installiert alle Pakete, die bei Produktion benötigt sind
14. Eine Website, die nicht immer eine neue Seite laden muss, sondern die immer auf der gleichen Seite läuft.
15. Komponenten sind eine Art Template, das global auf der Website verwendet werden kann.
Komponenten werden immer einzeln geladen.
16. Bundle Size sollte nicht zu groß sein.
17. Programmierparadigma für verteilte Systeme. Get, Post, Put, Delete, Head, Options
18. 1. Ressourcen haben eindeutige Identifikation 2. Standardmethoden gemäß HTTP müssen gegeben sein 3. Statuslose Kommunikation → keine Sitzungen! 4. Unterschiedliche Repräsentation 5. Hypermedia
19. safe → keine Nebeneffekte treten für Client auf, idempotent → mehrfaches Absenden wirkt sich nicht anders aus als einzelner Aufruf
20. 200er → hat funktioniert, 300er → wurde schon gemacht, 400er → hat nicht funktioniert, 500er → Fehler bei Server
21. Asynchrone und event-driven js runtime Umgebung
22. Der Event Loop ist zuständig für die Codeausführung, Sammeln und Verarbeiten von Events, und der Ausführung von Sub-Tasks.
23. Node.js Module sind an sich nur Librarys. Es gibt Built-in, Included und selbst erstellte Module
24. Eine Callback Funktion ist eine Funktion, die eine andere Funktion als Parameter übergeben bekommt.
25. Ein PWA ist eine Webanwendung mit bestimmten Funktionen, ein Mix aus einer strikten Webanwendung und einer nativen App. Die App ist somit von überall erreichbar, muss aber nicht installiert werden. Der Service Worker steht zwischen dem Server und dem Client und ist der zentrale Adapter zwischen den beiden und ist für die Kommunikation verantwortlich.
26. CORS ist eine Lockerung der Sicherheitsregeln von Same Origin Policy (Anfragen in einer Sitzung müssen immer vom gleichen Absender kommen)
27. Arten zur Authentifikation: Basic (Zugangsdaten), Session Based (Jeder Client hat einen Session-ID Cookie mit dem er auf die selbe Sitzung zugreifen kann, wird im Browsercache gespeichert), Token Based (Jeder Client hat einen Token, der auf dem System gespeichert wird)
28. TypeScript ist eine auf ECMAScript-Vorschlägen basierende Skriptsprache. TypeScript wird in JS umgewandelt und läuft überall wo auch JS läuft.

29. Polling (Client fragt Server immer nach neuen Daten), Long-Polling (Server schickt direkt Antwort wenn neue Daten vorliegen), Bidirektional → Socket-Verbindungen (Jeder kann Daten schicken)
30. Netlify

Aufgabe:

1. Element ID geben
2. In JS Code:

```
let Element = document.getElementById('ID');  
Element.style.color = „red“;
```