|  |  |
| --- | --- |
| **Sie wissen wer HTML erfunden hat**  Ende der sechziger Jahre: Arpanet (Advanced Research Projects Agency Network) wurde im Auftrag der US-Luftwaffe von einer kleinen Forschergruppe unter der Leitung des MIT (Massachusetts Institute of Technology) entwickelt, mit dem Ziel ein dezentrales Netzwerk zwischen verschiedenen US-Universitäten zu erschaffen.  **Sie können die wichtigsten und auch die aktuelle HTML Versionen nennen.** 1992: CERN entwickelt erste Version von HTML (Hypertext Markup Language).  2014: HTML 5. Überarbeitung des DOM und Einführung von vielen neuen Tags  2016: HTML 5.1  2017: HTML 5.2 **Sie kennen die Frontend und Backendentwicklung**    **Sie kennen die 3 Kerntechnologien der Webentwicklung**  HTML, CSS und JavaScript.  **Sie kennen das HTML-Element**  <html lang="de"> <html/>  **Sie kennen den Aufbau der HTML-Seite**  HTML einfach und verständlich erklärt  **Sie kennen Text-Elemente von HTML**  <p></p>, <h1></h1>, <h2></h2>…  **Sie kennen Listen-Elemente von HTML**  <ul>  <li>text</li>  <li>text</li>  <li>text</li>  </ul>  **Sie können Bilder einfügen und kennen die Attribute**  <img src=bild.jpg alt=beschreibung>  **Sie können Hyplerlinks einfügen**  <a href="Link">Link-Text</a> **Sie können die HTML-Seite mit semantischen Elementen strukturieren** <header> <nav> </nav> </header> <main> <article> </article> </main> <footer> </footer> **Sie können den DocType erklären**  Die erste Zeile <!doctype html> ist die Dokumenttypangabe, im Beispiel diejenige von HTML5. Die Dokumenttypangabe verrät dem Browser, welche Version von HTML er zu erwarten hat.Sie sehen keine direkte Auswirkung, ob Sie die Dokumenttypangabe schreiben oder nicht. Bei komplexeren Dokumenten mit CSS kann die Darstellung aber ohne Dokumenttypangabe nicht wie gewünscht aussehen. Also: Wenn Sie die Dokumenttypangabe immer am Anfang schreiben, sind Sie auf der sicheren Seite. **Ihnen ist bewusst, dass HTML eine Spezialform von XML ist.**  HTML ist eine Spezialform von XML. HTML ist der wichtigste Baustein der Webentwicklung und wird verwendet, um die Struktur einer Seite zu definieren. XML kann Daten zwischen Servern transportieren und wird oft zusammen mit HTML oder anderen Anwendungen verwendet.  **Sie kennen das HTML-Grundgerüst (html, head, body)**  <!DOCTYPE html> <html lang="de"> <head> <meta charset="utf-8" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <title>Titel</title> </head> <body> </body> </html>  **Sie können CSS Inline, Internal und External gebrauchen.**  **Inline**: <h1 style="color:blue;">Ich bin ein Titel</h1> **Internal**:  <head> <style> h1 { color: blue; } </style>  </head> **External**:  <head> <link href="style.css" rel="stylesheet" />  </head>  **Sie kennen den Element-Selektor** | **Sie kennen den Universal-Selektor**  \*{}  **Sie wissen was unter «CSS Margin Collapse» verstanden wird**  Es wird immer das grössere margin verwendet, funktioniert nur bei Top und Bottom  **Sie können die Abmessungen von Emelenten und Bildern bestimmen**    **Sie können den Seiteninhalt zentrieren**  margin-left: auto; margin-right: auto; ODER Text-align:center;  **Sie kennen Block Elemente**  <address><article><aside><blockquote><canvas><dd><div><dl><dt><fieldset><figcaption><figure><footer><form><h1>-<h6><header><hr><li><main><nav><noscript><ol><p><pre><section><table><tfoot><ul><video> 🡪100% der elternbreite, visuell als blöcke formatiert  **Sie kennen Inline Elemente**  <a><abbr><acronym><b><bdo><big><br><button><cite><code><dfn><em><i><img><input><kbd><label><map><object><output><q><samp><script><select><small><span><strong><sub><sup><textarea><time><tt><var> 🡪 belegt nur den Platz des Inhalts, verursacht keine zeilenumbrüche, höhen und breiten gelten nicht, paddings und margins nur rechts und links  **Sie kennen Inline-Block Elemente**  display: inline-block; Verfügt über die Inline und Block style Optionen.  **Sie kennen Pseudoelemente**  H1::first-letter; h1::first-line; h1::before; h1::after;  **Sie kennen die statische Positionierung**  Hierbei werden alle Elemente wie gewohnt nacheinander angeordnet, so wie sie im HTML-Dokument notiert wurden. Position: static;  **Sie kennen die relative Positionierung**  position: relative; 🡪 Damit platzieren bzw. verschieben Sie ein Element relativ zur aktuellen Position mit den CSS-Eigenschaften top, bottom, left und right und entsprechenden Wertangaben. Die anderen Elemente werden nicht beeinflusst – sprich: Die anderen Elemente bleiben an der Position, als würde sich das verschobene Element noch an derselben Stelle befinden.  **Sie kennen die absolute Positionierung**  position: absolute; 🡪 Hiermit ziehen Sie das Element aus dem Dokumentenfluss heraus. Hierbei können Sie mit den CSS-Eigenschaften top, bottom, left und right das Element absolut im nächstgelegenen Eltern-Element bzw. Webbrowser- Fenster platzieren – unabhängig davon, wo das HTML-Element im HTML-Dokument notiert wurde. Alle anderen Elemente tun jetzt so, als würde das absolut verschobene Element nicht mehr zum Dokumentenfluss gehören, und die eventuell so entstandene Lückqe wird mit dem nachfolgenden Element "aufgefüllt", bzw. es rückt nach.  **Sie kennen die fixe Positionierung**  position: fixed; 🡪 Die fixe Positionierung verhält sich zunächst wie die absolute Positionierung, allerdings mit dem deutlichen Unterschied, dass diese fixe Position absolut zur linken oberen Kante des Webbrowser-Fensters gemessen wird. In der Praxis bedeutet dies im Gegensatz zur absoluten Position, dass sich ein fixiertes Element nicht mehr mitbewegt, wenn das Webbrowser-Fenster gescrollt wird.  **Sie kennen die sticky positionierung**  position: sticky; Diese Funktion ist eine Hybridform aus einer relativen und fixen Positionierung. Das Element verhält sich hierbei zunächst so, wie ich es bei der relativen Positionierung beschrieben habe, bis eine gewisse Grenze wie der obere oder untere Bildschirmrand erreicht wurde, wo das Element dann "kleben bleibt" und sich wie bei der fixen Positionierung verhält.  **Sie kennen die Kombinatoren (Nachfahrenselektor, Kindselektor, Nachbarselektor, Geschwisterselektor)**  Kindselektor: e > f Nachfahrenselektor: e f. Nachbarselektor: e + f. Geschwisterselektor: e ~ f  **Sie kennen das Vokabular für die Flexbox (Flexcontainer, Flexitems, Hauptachse, Querende Achse).**  **Sie kennen die möglichen Ausrichtungen der Flexitems.**  justify-content: flex-start; flex-end center space-between space-around space-evenly  **Sie wissen wie man Zwischenräume definiert.**  Justify-content: space-around; justify-content:space-between justify-content: space-evenly;  **Sie wissen wie man beim Flexcontainer konfiguriert, so dass er nach aussen wie ein Inline-Element verhält.**  Display: inline-flex;  **Sie wissen wie man die Leerraumverteilung auf der querenden Achse im Flexcontainer einstellt.**  align-items: center;"  **Sie wissen wie man die Leerraumverteilung auf der querenden Achse für ein einzelnes Flexitem einstellt.**  align-items: flex-start; /\* Ausrichtung der Flex-Items auf der querenden Achse: am Anfang \*/  align-items: center; /\* Ausrichtung der Flex-Items auf der querenden Achse: zentriert \*/  align-items: flex-end; /\* Ausrichtung der Flex-Items auf der querenden Achse: am Ende \*/  align-items: baseline; /\* Ausrichtung der Flex-Items auf der querenden Achse: auf der Grundlinie \*/  align-items: stretch; /\* Ausrichtung der Flex-Items auf der querenden Achse: gestreckt \*/  **Sie wissen wie man die Leerraumverteilung auf der Hauptachse für ein einzelnes Flexitem einstellt.**  justify-content: space-between; /\* Leerraumverteilung auf der Hauptachse \*/  **Sie können mit den Chrome-Entwicklertools Flexangaben ändern.**  Ja du musst das CSS anpassen in den Entwicklung Tools  **Sie können die Reihenfolge der Flexitems manuel ändern.**  Mit flex order  **Sie können Selektoren kombinieren**    **Sie kennen Id-Selektor und den Klassen-Selektor**  klassenName {} #idName{}  **Sie kennen Attributselektoren**  <input type="text" ...> <input type="submit" ...>  **Sie können Text designen**  Color: red; font-family: Arial; font-size: 20px; font-weight: bold;  **Sie können Selektoren kombinieren**  **Sie können mit Farben arbeiten**  color: farbe;  **Sie kennen Pseudoklassen**  Einer Pseudoklasse wird ein Doppelpunkt vorangestellt. 🡪 a:hover{color: red;}  **Sie kennen Pseudoklassen für Link (LVHA)**    **Sie können den Konflikt mit widersprüchlichen Selektoren lösen (Recht des stärkeren Selektors)**  !important> Id > class > element |