

HTML

12.02.2014

Gwaith: Guten Abend. *noch reinhuscht*

MorgaineSionnach: Hallo

LouisDeLioncourt: huhu

Kaname: Hallo

Freya: kann aber wieder nur mitlesen ^^*

LouisDeLioncourt: Schön, dass ihr zu so später Stunde noch da seid ^^

LouisDeLioncourt: In der letzten Stunde haben wir damit angefangen uns über Layouts ganz ohne Tabellen Gedanken zu machen. Diese sind hauptsächlich aus <div>s aufgebaut und können per css beliebig verformt und gestaltet werden, was wir so in der Form bei Tabellen nicht haben

LouisDeLioncourt: Diese Layouts bezeichnet man auch als "box-models", da sie eben aus einzelnen Boxen zusammengestellt sind

LouisDeLioncourt: <http://dolmorgul.de/HTML-Kurs/index2.html> und so sieht das ganze aus

Freya: *sich vornimmt mal prtokolle zu wälzen*

LouisDeLioncourt: Wir haben oben den head Bereich, eine Navigation an der Seite und den Bereich mit dem tatsächlichen Content / Inhalt in der Mitte. Am Ende kommt noch der footer.

LouisDeLioncourt: Um alles herum ist noch eine div gelegt, die sich wrapper nennt und alles zusammenhält.

LouisDeLioncourt: Wie letztes mal bereits erwähnt könnt ihr die Namen der divs selber frei wählen.

LouisDeLioncourt: Haben das grobe Prinzip alle soweit verstanden, die die letzten Male auch da waren?

Gwaith: *nickt*

Kaname: *nick*

MorgaineSionnach: *das noch nacharbeiten muss, aber es nicht wirklich unlogisch findet*

LouisDeLioncourt: Stellt es euch einfach so vor.. als hättet ihr auf eurem Schreibtisch die einzelnen Bereiche vor euch liegen. diese stellen die <div> dar und enthalten jeweils Inhalte. nun müsst ihr diese nur noch an die richtige position bringen und das macht ihr mit css

LouisDeLioncourt: das bedeutet im umkehrschluss: auch ganz ohne css ist der inhalt eurer seite noch komplett lesbar. dadurch könnt ihr umgestaltungen zb auch ganz leicht vornehmen

LouisDeLioncourt: Ich möchte heute auf ein paar css Feinheiten eingehen, die wir letztes mal nur angerissen haben: padding, margin, float und clear

Freya: *ich les auch diesmal ^^ nach winks bin im Bettchen*

LouisDeLioncourt: padding ist der Innenabstand. damit könnt ihr angeben, wie groß der abstand von einem Element zu seinem Inhalt sein soll

LouisDeLioncourt: nacht

HTML

12.02.2014

LouisDeLioncourt: hier ist der code für euch zum mitmachen <http://pastebin.com/pYSU611U>

LouisDeLioncourt: padding könnt ihr für jede seite einzeln angeben. **padding-left: 10px;** würde bedeuten, dass vom linken rand nach innen 10 pixel abstand gehalten werden soll.

LouisDeLioncourt: das selbe kann man für padding-top padding-right und padding-bottom machen.

LouisDeLioncourt: probiert das zum beispiel einmal an der div id="navi" aus

LouisDeLioncourt: *in die runde schau*

Gwîndôr: <<< Schwarzlicht ^-^ (Freya - verlässt Chat)

MorgaineSionnach: *nickt Klappt*

Kaname: *gespannt zuhör*

LouisDeLioncourt: sehr gut ^^

Gwaith: *Hier auch.*

LouisDeLioncourt: anstatt jetzt alle 4 seiten einzeln anzugeben, könnt ihr das ganze aber auch in einem schreiben

Fehler: User '_padding' ist nicht in diesem Raum!

LouisDeLioncourt: **padding: 30px;** würde auf allen seiten 30px abstand einstellen

LouisDeLioncourt: aber auch unterschiedliche werte sind möglich ^^ **padding: 10px 5px 5px 20px;** würde oben 10, rechts 5, unten 5 und links 20 px Abstand generieren. Die Zahlen gebt ihr hier im Uhrzeigersinn an.

LouisDeLioncourt: auch mal ausprobieren bitte ^^

LouisDeLioncourt: **Margin** ist das genaue Gegenteil. Margin ist der **Außenabstand** eines Elements.

MorgaineSionnach: Was passiert, wenn ich zwei Einstellungen habe, die sich widersprechen? Beispielsweise eine vordefinierte Größe der div und den Innenabstand, der jedoch wesentlich kleiner/größer ist?

LouisDeLioncourt: Alles andere funktioniert hier aber haargenau wie beim padding: margin: 20px; macht an allen Seiten 20 px Abstand, margin-left:20px; nur auf der linken seite

LouisDeLioncourt: die abstände addieren sich dann hinzu und übergehen den wert der vordefinierten gröÙe

MorgaineSionnach: *nickt*

LouisDeLioncourt: margin und padding könnt ihr aber auch im negativen Bereich verwenden. margin-left: -200px; macht beispielsweise was anderes als margin-left: 200px;

LouisDeLioncourt: soweit das prinzip von margin und padding verstanden? Am besten probiert ihr euch da ein bisschen aus. Mir hilft der inspector hier auch immer sehr ^^

MorgaineSionnach: Wie soll ein negativer Abstand aussehen? *soweit alles verstanden hat, sich das aber nicht vorstellen kann*

HTML

12.02.2014

LouisDeLioncourt: er geht in die andere richtung ^^ margin-left: -200px; würde das element nach links, statt nach rechts schieben

LouisDeLioncourt: ist nicht ganz korrekt, aber manchmal hilft es doch

LouisDeLioncourt: Wenn ihr es selbst in eurem Projekt ausprobiert werdet ihr das sicher schnell verinnerlichen

LouisDeLioncourt: Kommen wir zum vorletzten Befehl: **float**

LouisDeLioncourt: Für gewöhnlich ist der Fluss aller Elemente so vorgesehen, dass sie untereinander erscheinen. Das könnt ihr z.B. bei den Bildern und dem Text sehen

LouisDeLioncourt: mit float können wir die elemente um andere elemente fließen lassen und so zb. text ein Bild von rechts umfließen lassen, so wie man es aus zeitungem

kennt <http://www.elmastudio.de/wp-content/uploads/2010/12/css-floats-01.jpg>

LouisDeLioncourt: In unserem aktuellen Layout findet sich das float bei der Navi-Box. dort steht **float: left;** was soviel heißt wie.. alle nachfolgenden Elemente umfließen das Element von rechts (achtung umgekehrt denken ^^)

Fehler: User '_float' ist nicht in diesem Raum!

LouisDeLioncourt: **float: right;** würde bedeuten, alle nachfolgenden Elemente umfließen es von links

LouisDeLioncourt: das bezieht sich nur auf alle Elemente, die nach diesem kommen, nicht auf die davor.

LouisDeLioncourt: versucht s mal bei den bildern z.B. ^^ ``

LouisDeLioncourt: Das war eigentlich schon alles, was man zu float sagen kann ^^ ich hoffe es wird beim ausprobieren klarer..

LouisDeLioncourt: Was nun noch fehlt ist das **clear**. clear gibt es in 3 verschiedenen formen: clear:left; clear:right; und clear:both;

LouisDeLioncourt: clear ist dafür da *trommelwirbel* um float aufzuheben ^^

LouisDeLioncourt: wie bereits erwähnt wirkt sich float ja auf alle nachwirkenden Elemente aus. Im falle unseres Layouts würde es z.B. auch den footer mit fließen lassen. clear:both; in der footer <div> verhindert das aber und löst den Fluss hier auf.

LouisDeLioncourt: *schaut in hoffentlich nicht ganz so verwirrte Gesichter ö.ö*

Kaname: *Schläfe reib*

Gwaith: *nickt - das klingt durchaus logisch für sie*

MorgaineSionnach: *Ah!!! hat es jetzt hinbekommen, dass die Bilder von Louis und Mirai nicht mehr komisch aneinanderkleben*

LouisDeLioncourt: okay, das gehen wir nächste woche nochmal durch ^^

HTML

12.02.2014

MorgaineSionnach: *findet das wunderbar logisch*

LouisDeLioncourt: versucht bitte bis zum nächsten mal ein <div> layout zusammenzuschustern. Dabei dürft ihr euch gern an meiner vorlage orientieren nur etwas anders sollte es schon aussehen