LEKCE: Display - typ a chování prvků na stránce

Opět tak trochu opakování základů, ale také spousta nových informací.  
  
Nauč se, jak **pomocí vlastnosti displayměnit typy prvků**, což významně ovlivňuje, jak se prvky na stránce chovají a zobrazují.

Výpisky

Display

* Jak se prvky chovají na stránce
* Výchozí nastavení na nějakou hodnotu, všechny prvky jsou nějakého typu
* Blokový, řádkový a blokově-řádkový typ
* Všechny nadpisy, odstavce, seznamy, div, … = blokové prvky
  + Zabírá celou dostupnou šířku, pokud nenastavíme nějakou jinou, atypickou
  + Roztáhne se na celou šířku, kterou má k dispozici od svého rodiče (minimálně v rámci body)
  + Jsou vždycky pod sebou, pokud to nenastavíme jinak
  + skládají se pod sebe, i když by se vešly vedle sebe
* řádkové prvky = prvky, které jsou součástí řádku textu
  + nejběžněji odkaz
  + jsou vždy součástí řádku (jako slova ve větě), skládají se na řádku za sebou, vedle sebe
  + zabírají jen minimální šířku, která je potřeba
  + nebo značka span (obdoba divu – není vidět, nic neznamená, nemá sémantický význam, je to například pokud chci podbarvit nějaký text, tak jej obalím do spanu)
  + nejdou jim nastavit některé vlastnosti box-modelu (padding, margin, výška, šířka, …)
  + respektive tyto vlastnosti je možné nastavit, ale ne všechny se projevují způsobem, jaký bychom chtěli
* řádkově-blokové (hybridní) prvky = slučují vlastnosti řádkových a blokových prvků
  + jako řádkový se chová proto, že je součástí řádku a skládá se vedle sebe na řádek, z blokových si zae bere to, že mu je možné nastavit vlastnosti box-modelu = roztáhne řádek, odsadí řádky tak, aby se tam vešel
* výchozí vlastnost mohu měnit pomocí vlastnosti display – ta má několik desítek hodnot – nás zajímají block, inline a inline-block
* ke změně dochází, pokud potřebujeme sémanticky užít nějaký prvek, typicky odstavec textu, ale potřebuji jich více vyrovnat za sebe – my bychom mohli místo značky pro odstavec (p) užít tu značku span, ale z nějakého sémantického důvodu prostě ten odstavec potřebujeme
* vlastnost display má ještě jednu hodnotu, která slouží ke skrývání prvků

Skrývání prvků

* display: none (ve smyslu zobrazit: nijak) – schová nám celý prvek
* děláme to proto, že máme na stránkách javascript (technicky se zajišťuje js + v rámci css)
* prvek se chová, jako by v prohlížeči neexistoval
* možné použít také v responzivním webdesignu (zobrazení na mobilech atd.)
* to zobrazování lze také řídit vlastností visibility: hidden/visible
  + výchozí je visible
  + když nastavím hodnotu hidden, prvek je skrytý, ale v dokumentu fakticky zabírá místo, jenom není vidět (udělá se tam prostě bílé místo)
* opacity (průhlednosti
  + 0 = je úplně průhledný (není vidět)
  + 1 = neprůhledný = je celý vidět
* Prakticky je to nastavování změny display například pro potřeby menu
* Když děláme to menu, tak Luděk si automaticky vynuluje margin i padding, aby mu to nedělalo problémy
* . menu a:hover (je to odkaz ve stavu když na něj najedu myší) - pseudotřída