

KLÍČ K POCHOPENÍ CSS

Porozumění základním principům kaskádových stylů CSS je pro webový design zásadní a přitom může být snadné. Pro úspěšnou implementaci kaskádových stylů je sice postačující zvládnutí uživatelsky přátelsky navržených, snadno zapamatovatelných a jednoduše aplikovatelných definic CSS vlastností, avšak porozumění několika základním principům CSS může vést k naprosté kontrole nad webovými dokumenty jak po stránce strukturální tak po stránce prezentační.

Klíčem k pochopení kaskádových stylů je pochopení koncepce vizuálního formátování, v originálu nazvaná "Visual formatting model". Přestože je tato koncepce vizuálního formátování určena především výrobcům UA programů / prohlížečů (vzhledem k tomu, že prakticky detailně popisuje algoritmy a postupy vedoucí ke správnému vykreslení stylovaných dokumentů), má pro autory dokumentů a běžné uživatele CSS značný význam, neboť na základě její znalosti mohou CSS vlastnosti aplikovat s naprostou přesností a také pochopit chování prohlížečů v různých situacích.

Výchozím předpokladem je, že dokumenty jsou jejich uživateli vnímány smyslově – zrakem, sluchem nebo hmatem, přičemž pro tyto způsoby vnímání jsou UA programy / prohlížeči zpracovány podle použitého média. Na základě tohoto předpokladu vizuální formátovací model rozděluje média do tří hlavních skupin: *vizuální*, *zvuková*, *hmatová*.

Ačkoli se některé CSS vlastnosti u těchto skupin médií liší, koncepce vizuálního formátování považuje vizuální média za hlavní typ média, na který vztahuje všechny specifikace postupů, podle nichž UA programy / prohlížeče zpracovávají strom dokumentu. Je to proto, že vizuální média jsou pro zpracování webových dokumentů médii typickými a přizpůsobení pro zvuková a hmatová média je snadno proveditelné prostřednictvím několika rozšiřujících CSS vlastností (v každé skupině médií existuje sada vlastností vztahující se pouze k danému typu média nebo k určitému typu zařízení) a jednak proto, že dokument musí být teoreticky stejně před prezentací nejdříve vždy vykreslen (a tedy např. i pro ryze zvuková média), byť na virtuální ploše.

Poznámky:

1. je třeba si uvědomit, že koncepce vizuálního formátování nepostihuje absolutně všechny aspekty formátování a UA programům / prohlížečům poskytuje i volný prostor pro vlastní interpretaci pravidel a tedy nelze

očekávat, že různé, plně v souladu se specifikací sestrojené prohlížeče zobrazí tentýž dokument zcela shodně

2. koncepce vizuálního formátování vychází z toho, že UA programy / prohlížeče přiřadí hodnoty všem vlastnostem všech prvků podle pravidel kaskády již během vytváření stromu dokumentu a mají tedy před prezentací dokumentu k dispozici jednak všechny prvky, které se mají na stránce prezentovat, uspořádané ve stromu dokumentu a jednak všechny vlastnosti popisující jejich způsob ztvárnění; posledním krokem je pak jejich prezentace, které UA programy / prohlížeče provedou podle postupů určených vizuálním formátovacím modelem
3. stylové vlastnosti pro zvukovou prezentaci mohou být použita současně s vizuálními vlastnostmi (mixovaná média) nebo jako zvuková alternativa k vizuální prezentaci; hmatová zařízení patří do skupiny vizuálních médií

VIZUÁLNÍ FORMÁTOVACÍ MODEL CSS

Podle koncepce vizuálního formátování dokumentu:

1. se dokument vždy vykresluje na nějaké zobrazovací ploše („plátně“), přičemž:
 - u průběžných médií (typicky okno prohlížeče) je touto plochou *jedna souvislá plocha* neomezené délky a šířky, k níž UA programy / prohlížeče zpravidla poskytují uživateli průzor, kterým lze dokument prohlížet (nejčastěji okno, či část obrazovky); při změně velikosti průzoru mohou UA programy / prohlížeče rozložení dokumentu v průzoru změnit a je-li průzor menší než počáteční rámeček dokumentu, UA programy / prohlížeče by měly uživateli poskytnout prostředky k posunu průzoru nad dokumentem (nejčastěji posuvníky, ale i jiný způsob odpovídající typu zařízení, konvencím grafického uživatelského rozhraní, operačnímu systému či technickým možnostem)
 - u stránkovaných médií (typicky tiskárna, projektor,...) je touto plochou *sada jednotlivých stránek*, mezi něž je obsah dokumentu rozdělen a s každou stránkou se pak pracuje jako se samostatnou plochou
2. každý prvek ve stromu dokumentu vytváří žádný a více rámců

3. vzhled a umístění rámců na zobrazovací ploše jsou závislé na:
 - rozměrech rámců
 - pozičním schematu (normální pozicování, obtékající pozicování, absolutní pozicování)
 - vzájemných vztazích prvků ve stromu dokumentu
 - vnějších okolnostech (velikost průzoru, skutečné rozměry objektů,...)
4. v CSS je každý rámec umístěn v trojrozměrném prostoru, kde třetím rozměrem je osa z probíhající ve směru pohledu pozorovatele, takže rámce se mohou na sebe vrstvit

RÁMCE

CSS popisuje rámce specifikované koncepcí vizuálního formátování, podle které se výsledek ztvárnění každého prvku umísťuje do jednoho nebo několika pravoúhlých oblastí – tzv rámců (boxů).

Základní vlastností pro vizuální formátovací model CSS je vlastnost *'display'* pro vizuální formátování aplikovatelná na všechny prvky. Tato vlastnost popisuje, jak je prvek zobrazen na „plátně“ (obrazovka počítače, tisková stránka, apod.). Implicitní hodnotou vlastnosti *'display'* je hodnota *inline*, což znamená, že prvek vytváří jeden nebo více řádkových rámců, avšak pravidla výchozího CSS stylu UA programu / prohlížeče mohou tuto hodnotu u různých prvků přebít a pak se implicitní hodnotou vlastnosti *'display'* těchto prvků stává hodnota určená výchozím stylem UA programu / prohlížeče.

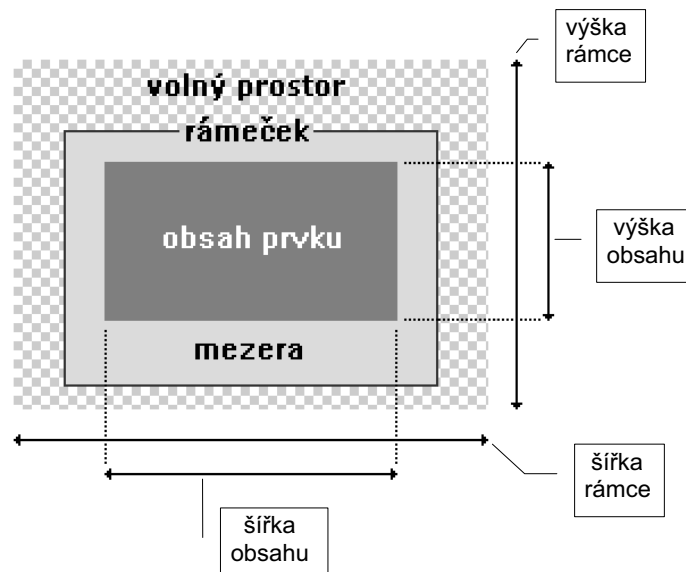
Všechny rámce mají základní jádro vytvořené obsahem prvku s volitelnými okolními oblastmi mezery, rámečku a okraje takto:

1. každý rámec je oblast tvaru pravoúhlého čtyřúhelníku (tzv. box)
2. vnější hranice každého rámce jsou vždy neviditelné a vymezují plochu, kterou rámec obsazuje
3. jádrem každého rámce je oblast obsahu (text, obrázek atd.)
4. kolem obsahu mohou být další dvě oblasti:
 - viditelný rámeček (oblast rámečku - *border*)
 - mezera, tj. odsazení obsahu od rámečku (oblast odsazení - *padding*)
5. mezi rámečkem a vnější hranicí rámce může být volný prostor (oblast okraje – *margin*), který zaručuje, že viditelné oblasti sousedních bloků se nebudou dotýkat; platí přitom pravidlo, že volné prostory (okraje)

sousedních bloků se slučují, přičemž sloučený volný prostor (okraj) odpovídá většímu z obou sousedních volných prostorů (okrajů)

6. rozměry všech oblastí popisují příslušné vlastnosti CSS a také odpovídající typ formátování (blokové nebo řádkové formátování)
7. pro rozměry (výšku a šířku) všech oblastí kolem obsahu (odsazení, rámeček, volný prostor) existují samostatné vlastnosti, takže jejich velikost může být na každé straně jiná

Všechny oblasti rámce vytvářené ve vizuálním formátovacím modelu CSS pro prvek znázorňuje následující obrázek:



Poznámky:

1. rozměry volného prostoru, rámečku a mezery se nastavují prostřednictvím vlastností *'margin'* (volný prostor), *'border'* (rámeček) a *'padding'* (mezera)
2. celkový rozměr rámce (výška a šířka) jsou dány součtem rozměrů velikosti obsahu prvku (např. zformátovaného textu nebo obrázku), mezery, rámečku a volného prostoru

3. neexistuje žádná možnost jak nastavit explicitně celkové rozměry rámece
4. volný prostor je vždy průhledný, takže rodičovský prvek je skrze něj viditelný
5. oblast mezery používá stejné pozadí jako prvek sám (nastavené vlastnosti *'background'*)
6. neexistuje žádná možnost jak samostatně změnit pozadí mezery či volného prostoru
7. každý rámeček je umístěn vzhledem k (tzv.) obsahujícímu rámci, přičemž počáteční rámeček vytváří kořen dokumentu

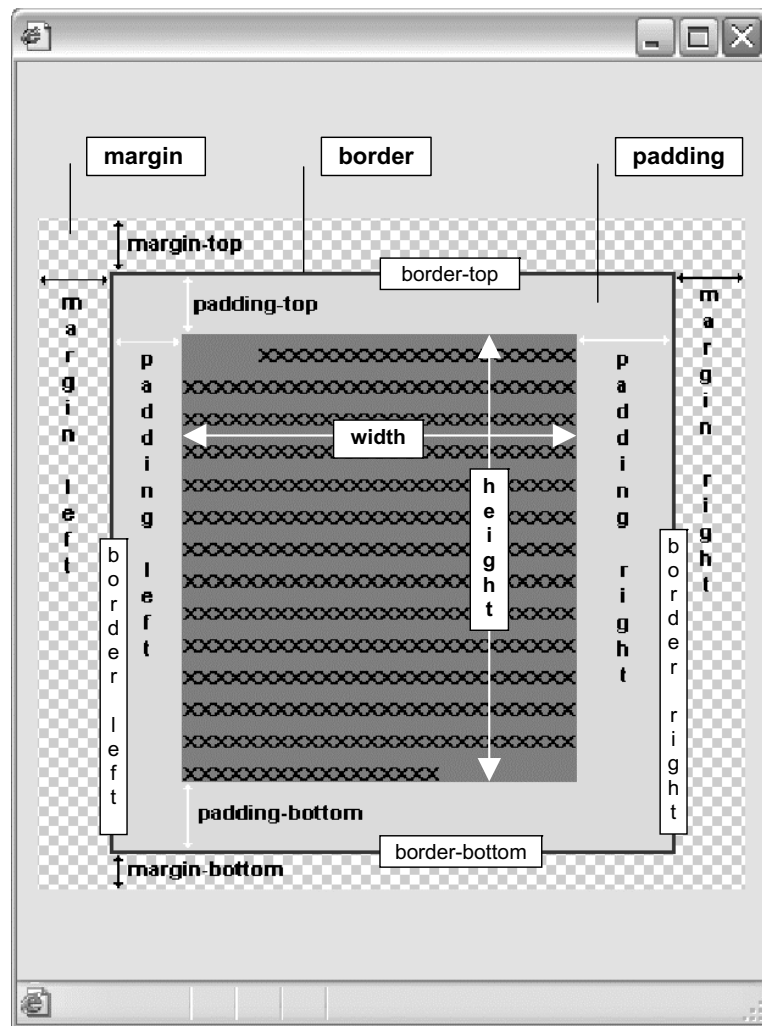
Šířku a výšku oblasti obsahu prvku nastavují vlastnosti **width** a **height**. Pomocí vlastností **min-width**, **max-width** a **min-height**, **max-height** lze zadat doporučenou šířku a výšku obsahu prvku.

Pro nastavení každé z jednotlivých okolních součástí oblasti obsahu prvku (okraj, rámeček, mezera) jsou k dispozici čtyři vlastnosti:

okraj	rámeček	mezera
margin-top	border-top-width	padding-top
margin-right	border-right-width	padding-right
margin-bottom	border-bottom-width	padding-bottom
margin-left	border-left-width	padding-left

které nastavují příslušné velikosti. Integrujícími vlastnostmi **margin** (pro okraj), **border-width** (pro rámeček) a **padding** (pro mezeru) lze zapsat najednou hodnotu všech čtyř vlastností. Hodnotou integrující vlastnosti je vždy jeden až čtyři údaje, udávající příslušnou šířku v pořadí ve směru hodinových ručiček počínaje horním. Je-li čísel méně, chápou se stejně, pouze se rozměry chybějících hodnot doplní vždy podle rozměru hodnoty protilehlé. Je-li uveden jediný rozměr, bude platit pro všechny čtyři strany.

U rámečku lze navíc nastavit vlastnosti **border-color** (barva rámečku) a **border-style** (provedení rámečku). Implicitní hodnotou rámečku je hodnota "none", při níž rámeček není viditelný. Integrujícími vlastnostmi **border-top**, **border-bottom**, **border-left**, **border-right** lze nastavit kompletní provedení příslušné hrany rámečku (tloušťku, barvu a styl, např. **border-left: 3px #00F double**). Největším zjednodušením je vlastnost **border**, kterou lze použít když mají všechny čtyři strany rámečku stejné vlastnosti.



Obrázek A – Prvky formátovacího modelu kaskádových stylů

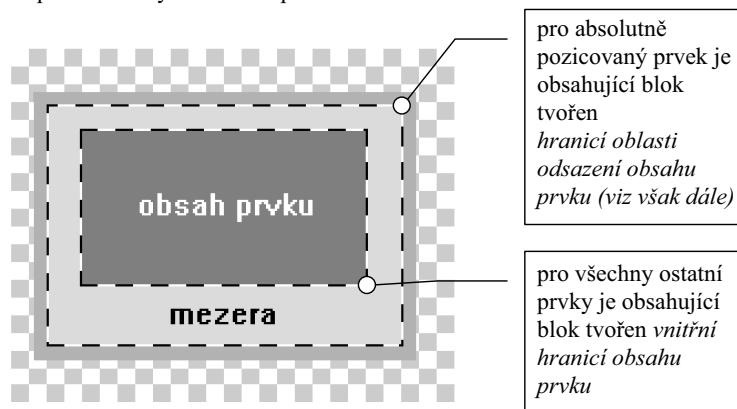
Obrázek vpravo znázorňuje, které rozměry nastavují jednotlivé vlastnosti.

Poznámky:

1. vlastnosti *'width'* a *'height'* definují šířku a výšku **obsahu prvku** (vnitřní rozměry prvku) – rozměry rámce explicitně nelze zadat a jsou vždy rozměry vypočítanými na základě ostatních hodnot oblastí rámce
2. hodnotu vlastností vizuálního formátovacího modelu lze zadat absolutně nebo v procentech vzhledem k aktuální šířce *obsahu prvku* nebo povolit hodnotou *'auto'* a ponechat na prohlížeči, aby si daný rozměr přizpůbil podle potřeby
3. je-li některý z rozměrů některé oblasti nulový, odpovídající hrany sousedních oblastí splynou
4. okraje mohou mít i zápornou hodnotu; mají-li okraje zápornou hodnotu, hrana okraje se posune až dovnitř oblasti rámečku, odsazení či obsahu; ostatní oblasti tak přesahují vnější hranu prvku; umístění prvku se však stále řídí touto vnější hranou - záporné okraje tak způsobí, že obsah prvku může přesahovat svůj obsahující blok

Doporučení:

Alespoň jedna z vlastností určujících šířku/výšku obsahu prvku ve vizuálním formátovacím modelu by měla být ponechána na hodnotě *'auto'* (implicitní hodnota). To umožňuje prohlížeči lépe se vypořádat se zobrazením prvku v různých situacích. Ve většině případů se jako volná ponechává výška obsahu prvku.

**Obrázek B – Ilustrace k obsahujícím blokům****TYPY RÁMCŮ**

Specifikace CSS definuje několik typů rámců, které mohou být dle CSS vytvářeny. Typ rámce je určen vlastností *'display'* (viz str. 84), která ve vizuálním formátovacím modelu CSS ovlivňuje chování rámce. V zásadě se všechny prvky dokumentu dělí do dvou kategorií:

- **blokové prvky** jsou ty prvky zdrojového dokumentu, které jsou vizuálně formátovány jako víceřádkové bloky (např. odstavce)

v CSS se prvek stává blokovým, má-li jeho vlastnost *'display'* jednu z hodnot: *block*, *list-item*, *compact* a *run-in* (částečně), *table*
blokové prvky **vytvářejí blokové rámce** (boxy)

- **řádkové prvky** jsou ty prvky zdrojového dokumentu, které nevytvářejí bloky obsahu (jejich obsah se nachází v řádku, např. obrázek `` nebo zdůrazněná část textu `` v odstavci)

v CSS se prvek stává řádkovým, má-li jeho vlastnost *'display'* jednu z hodnot: *inline*, *inline-table*, *compact* a *run-in*
řádkové prvky **vytvářejí řádkové rámce** (boxy); obsahem řádkového prvku nesmí být nikdy blokový prvek (např. uvnitř prvku `span` se nesmí nacházet prvek `div`)

V CSS prvky jejichž obsah se nachází ve značce typu kontejner vytváří tzv. **neanonymní rámce**. V (X)HTML mají všechny prvky, které nejsou samostatnými značkami (jako BR) pro vlastnost *'display'* (určující typ rámce) svoji přirozenou hodnotu, kterou není třeba implicitně určovat (např. značky P, DIV, FORM, H1, UL, OL,... jsou blokové; blokovým prvkem je i prvek LI jehož hodnota vlastnosti *'display'* je *list-item* a TABLE jehož hodnota vlastnosti *'display'* je *table*).

V CSS úseky obsahu, které se nenachází v žádném kontejneru vytváří tzv. **anonymní rámce**. Takové úseky obsahů typicky vznikají kdy nějaký prvek obsahuje v sobě jiný prvek a v tom případě úseky obsahu před a za ním vytvářejí úseky neležící v kontejneru.

Neanonymní a anonymní rámce mohou vznikat jak uvnitř blokových prvků tak uvnitř řádkových prvků.

Typy a charakteristiky CSS rámců uvádí souhrnně následující tabulka.

Typy a charakteristiky rámců (boxů) CSS

rámce	blokové	neanonymní	neanonymní blokové rámce vytvářejí blokové prvky jejichž obsah se nachází v kontejneru prvku a v jejich obsahu se nenachází jiný prvek (obsah není smíšený)
		anonymní	anonymní blokové rámce vznikají je-li obsah blokového prvku smíšený, tj. jsou v něm jak blokové prvky, tak řádkové úseky obsahu, a ty se sjednotí tak, že se řádkové úseky „zabalí“ do virtuálního anonymního kontejneru, např. <code><div>nějaký text<p>text odstavce</p>další text</div></code> se sjednotí do virtuální konstrukce <code><div></code> <code><anonymní kontejner>nějaký text</anonymní kontejner></code> <code><p>text odstavce</p></code> <code><anonymní kontejner>další text</anonymní kontejner></code> <code></div></code> vlastnosti anonymního kontejneru se dědí od jeho rodičovského prvku, nedědičné vlastnosti mají své výchozí hodnoty
	řádkové	neanonymní	neanonymní řádkové rámce vytvářejí řádkové prvky jejichž obsah se nachází v kontejneru prvku a v jejich obsahu se nenachází jiný prvek (obsah není smíšený)
		anonymní	anonymní řádkové rámce vznikají je-li obsah řádkového prvku smíšený, tj. je v něm několik řádkových úseků tvořených řádkovými prvky a úseky jimž neodpovídá žádný řádkový prvek a ty automaticky vytvářejí anonymní řádkové rámce, např. <code><p>nějaký texttext odstavcedalší text</p></code> se sjednotí do virtuální konstrukce <code><p></code> <code><anonymní řádkový úsek>nějaký text</anonymní řádkový úsek></code> <code>text odstavce</code> <code><anonymní řádkový úsek>další text</anonymní řádkový úsek></code> <code></p></code> vlastnosti anonymní řádkového úseku se dědí od jeho rodičovského prvku, nedědičné vlastnosti mají své výchozí hodnoty
	zanořené		prvky s vlastností 'display: run-in' vytvoří specifické rámce, které prvek „zanoří“ na začátek následujícího rámce, protože <i>run-in</i> rámce (box) se chová takto: <ul style="list-style-type: none"> • následuje-li za run-in rámcem blokový rámec, který není plovoucí a není absolutně pozicován, stane se run-in rámec prvním řádkovým rámcem blokového rámce • v ostatních případech se z run-in rámce stane blokový rámec run-in prvky dědí vlastnosti po svých rodičích ve stromu dokumentu, nikoli po prvku, jehož se stanou součástí typickým použitím run-in prvků je vytváření „zatahovaných“ nadpisů
	kompaktní		prvky s vlastností 'display: compact' vytvoří specifické rámce, které prvek umístí na levém nebo pravém okraji vně prvního řádkového rámce následujícího bloku, protože <i>compact</i> rámec (box) se chová takto: <ul style="list-style-type: none"> • následuje-li za compact rámcem blokový rámec, který není plovoucí a není absolutně pozicován, je kompaktní rámec formátován jako řádkový rámec na jedné řádce jehož výsledná šířka se porovnává s jedním z postranních okrajů blokového rámce; o tom, zda s pravým, či levým, rozhoduje vlastnost 'direction' prvku tvořícího obsahující blok a je-li řádkový rámec stejně široký, nebo užší než okraj, je umístěn na tomto okraji • v ostatních případech se z compact rámce stane blokový rámec prvky, které nelze formátovat do jednoho řádku, nemohou být umístěny na okraj následujícího bloku; typickým použitím compact prvků je vytváření „jednořádkových“ marginálií

OBSAHUJÍCÍ BLOKY

Umístění a velikosti rámců jsou v CSS odvozeny od hran pravoúhlého bloku nazývaného **obsahující blok**. Obecně:

- obsahující blok v němž spočívá kořenový prvek (tj. celý dokument), vytváří UA program / prohlížeč a nazývá se *počáteční obsahující blok*
- pro všechny ostatní prvky s výjimkou absolutně pozicovaných prvků je obsahující blok tvořen *vnitřní hranicí obsahu prvku* nejbližšího předka
- pro absolutně pozicovaný prvek s vlastností 'position: fixed' je obsahující blok tvořen *průzorem*
- pro absolutně pozicovaný prvek s vlastností 'position: absolute' je obsahující blok tvořen nejbližším předkem s 'position' jinou než 'static' (tj. s 'position: relative|absolute|fixed') následovně:
 - a. v případě, že je nejbližší předek blokový, je obsahující blok tvořen *hranicí oblasti odsazení obsahu prvku* (oblast padding)
 - b. v případě, že je nejbližší předek řádkový, je obsahující blok závislý na vlastnosti 'direction' nejbližšího předka:
 - i. je-li 'direction: ltr' pak levým horním rohem obsahujícího bloku je levý horní roh prvního řádkového rámce vytvořeného nejbližším předkem; pravým dolním rohem obsahujícího bloku je pravý dolní roh posledního řádkového rámce vytvořeného nejbližším předkem
 - ii. je-li 'direction: rtl' pak pravým horním rohem obsahujícího bloku je pravý horní roh prvního řádkového rámce vytvořeného nejbližším předkem; levým dolním rohem obsahujícího bloku je levý dolní roh posledního řádkového rámce vytvořeného nejbližším předkem
 - c. nemá-li prvek předka, je obsahující blok tvořen *hranicí počátečního obsahujícího bloku*
- průběžně vytvářené rámce slouží jako obsahující bloky pro všechny své podřízené rámce (rámec tedy "zakládá" obsahující blok pro své následovníky)
- každý rámec je umístěn vzhledem k obsahujícímu bloku, ale obsahujícím blokem není omezován; to znamená, že ho může přesáhnout

Poznámky:

1. pojmy **rámec** a **obsahující blok** nejsou totéž, přestože rámce zakládají obsahující blok pro své potomky
2. pojem **obsahující blok rámce** znamená ten obsahující blok, ve kterém rámec existuje, nikoli ten, který vytváří
3. šířka počátečního obsahujícího bloku může být určena vlastností 'width' kořenového prvku dokumentu - jestliže má tato vlastnost hodnotu *auto*, stanoví počáteční šířku UA program / prohlížeč, např. shodnou s aktuální šířkou průzoru; výška počátečního obsahujícího bloku může být určena vlastností 'height' kořenového prvku dokumentu - jestliže má tato vlastnost hodnotu *auto*, roztáhne se výška obsahujícího bloku na výšku obsahu celého dokumentu
4. počáteční obsahující blok nemůže být pozicován a nemůže být obtékán; to znamená, že UA program / prohlížeč musí u kořenového prvku ignorovat vlastnosti *position* a *float*

POZICOVÁNÍ

Při vykreslování stromu dokumentu se ztvárněné prvky dokumentu vykreslují v přirozeném sledu tak jak jdou za sebou, to znamená, že řádkové rámce se kladou vedle sebe zleva doprava a dolů a sloupcové pod sebe a obojí pak postupně od začátku ke konci dokumentu. Tento způsob vykreslování prvků dokumentu, tzv. normální řazení, je stejný bez ohledu na to zda je dokument vykreslován s použitím nebo bez použití kaskádovitých stylů. Kaskádovité styly CSS však mají tu specifickou vlastnost, že obsahují mechanismus umožňující aplikovat v normálním řazení prvků dokumentu výjimky. Tento mechanismus výjimečného řazení prvků dokumentů v CSS se nazývá **pozicování**.

Nutno zdůraznit, že:

1. velkou většinu uspořádání obsahu dokumentů plně řeší normální řazení dokumentových prvků a výjimky z tohoto řazení řeší vesměs pouze specifické situace, které by šlo v normálním řazení simulovat stěží nebo s velkou pracností a povětšinou ještě pouze za cenu opuštění čistě

strukturovaného zdrojového tvaru dokumentu (typicky vícesloupcová sazba, marginálie, vsuvky, ...); v CSS pozicování není nic jiného a není třeba v něm cokoliv jiného či dalšího hledat

2. pozicování (X)HTML prvků je možné pouze prostřednictvím kaskádových stylů (bez použití kaskádových stylů výjimky z normálního řazení prvků možné nejsou)
3. při dynamickém přístupu k prvkům dokumentu (tj. ve spolupráci skriptů) umožňují výjimky (a pouze výjimky) v normálním řazení aplikovat různé efekty jako je pohyb, odhalování, skrývání a animování obsahu prvků
4. jestliže kaskádové styly zavádějí do (X)HTML dokumentů DTP, tedy typografická hlediska, s nimiž prakticky končí jednoduchost jazyka HTML (vzhledem k tomu, že přes svoji eleganci kaskádovité styly budou vždy vyžadovat značnou dávku profesionální zdatnosti), pak pozicování navíc vyžaduje absolutní představu o tom, čeho autor chce dosáhnout – nepromyšlené použití pozicování nemá sebemenší smysl a pravděpodobně bude mít i pochybný efekt

Specifickou charakteristikou CSS pozicování je, že umožňuje umístování objektů dokumentu nejenom na ploše v jedné rovině (2D pozicování), ale také umístování objektů dokumentu v různých z rovinách (3D pozicování). Oba způsoby pozicování mají přesně vymezené možnosti a přesná pravidla pro aplikaci.

2D pozicování CSS umožňuje:

1. **posunout** prvek ze své pozice dané normálním řazením
2. **obtékat** prvek obsahem okolních objektů v normálním řazení
3. **zanořit** prvek v normálním řazení na začátek obsahu následujícího bloku
4. **předsadit** obsah aktuálního prvku v normálním řazení do oblasti levého nebo pravého postranního okraje (volný prostor) následujícího prvku, (tj. vně jeho prvního řádkového rámce do prostoru, do něhož jinak nelze žádný obsah umístit)

3D pozicování CSS umožňuje:

1. **zakládat vrstvy** v dokumentu (tj. specifikovat další roviny pro zobrazení prvků dokumentu), přičemž základní vrstvu tvoří počáteční obsahující blok dokumentu
2. **ukotvit objekty dokumentu na přesnou pozici** (vzhledem k obsahujícímu bloku)
3. **ukotvit objekty dokumentu v průzoru**

Základní vlastností pro pozicování v CSS je vlastnost '*position*' (ze skupiny vlastností pro vizuální formátování a pozicování) aplikovatelná na všechny prvky. Tato vlastnost popisuje, jak je prvek umístěn na „plátně“ (obrazovka počítače, tisková stránka, apod.). Implicitní hodnotou vlastnosti '*position*' je hodnota *static*, což znamená, že implicitně jsou prvky v dokumentu pozicovány v přirozeném sledu, tj. v normálním řazení (pro standardní HTML provedení bez možnosti pozicování se používá pojmenování "statické pozicování", tj. '*position: static*' je identické s klasickým HTML formátováním). Další z možných hodnot vlastnosti '*position*' jsou hodnoty *relative* a *absolute/fixed* a společně určují hlavní poziční schema CSS takto:

'position: '	poziční schema CSS
relative	relativní pozicování umožňuje umístění posunem vzhledem k přirozené pozici v dokumentovém sledu
absolute / fixed	absolutní pozicování umožňuje umístění na konkrétní přesně stanovenou pozici
static	statické pozicování řazení v normálním sledu (implicitní pro všechny prvky)

Jednotlivé formy pozicování se realizují volbou hlavního pozičního schématu kombinací s dalšími vlastnostmi takto:

forma pozicování	určující CSS vlastnosti	upřesňující CSS vlastnosti
bez pozicování (normální řazení)	position: static	-
posun	position: relative	top, right, bottom, left
obtékání	position: [static relative] float:[left right]	clear
zanoření	position: static display:run-in	-
předsazení	position: static display:compact	direction: [ltr rtl]
ukotvení na přesnou pozici	position: absolute	top, right, bottom, left
ukotvení v průzoru	position: fixed	top, right, bottom, left
určení vrstvy	z-index: [±]n	-

Posun

position: relative; [top: ; right: ; bottom: ; left: ;]

V normálním řazení posune rámeček prvku ze své momentální vlastní polohy v dokumentu o určenou hodnotu.

Rámeček posouváního prvku je nejprve zformátován v normálním řazení jako by pozicován nebyl a poté je z této své aktuální pozice posunut relativně o zadanou hodnotu. Relativní hodnoty posunu určují stylové vlastnosti 'top' (shora-nadol), 'right' (zprava-doleva), 'bottom' (zdola-nahoru) a 'left' (zleva-doprava) platné pro posouvání prvek. Platí přitom, že left=-right a top=-bottom .

Rámeček relativně pozicováního prvku není z normálního řazení vyňat. Relativní pozicování nemá žádný vliv na vzhled a rozměry prvku - relativně pozicování prvek si udržuje rozměry dané jeho normálním řazením, včetně případných ukončení řádků a prostoru původně pro ukončení řádků rezervováního. Posun rámeček relativně pozicováního prvku nemá žádný vliv na umístění rámeček, který za ním následuje - následující rámeček je umístěn stejně, jako kdyby relativně pozicování prvek vůbec nebyl posunut. Z toho vyplývá, že posun rámeček:

- může způsobit překryv rámeček v dokumentu
- vytvoří novou vrstvu v níž je rámeček posouván a která je novým obsahujícím blokem pro potomky v normálním řazení tohoto prvku a pozicování následníky
- ponechá v normálním sledu prvků dokumentu původní prostor daný jeho rozměry beze změny (tzn. při úplném odsunu relativně pozicováního rámeček po něm zůstane volný prostor odpovídající jeho rozměrům)

1. relativně pozicování rámeček vznikne nastavením vlastnosti 'position' daného prvku na hodnotu *relative*
2. v kombinaci s dynamickým skriptováním lze s relativním pozicováním dosáhnout animačních efektů
3. typickým použitím relativního pozicování je využití k formátování nadsazeného a podsazeného textu a exponenty a indexy; v tom případě je však třeba vzít v úvahu, že relativní pozicování neovlivňuje výšku původního řádku
4. relativním pozicováním lze nasimulovat víceloupcovou sazbu, je to však pouze optická náhražka (stejně jako použití tabulky pro tento účel), protože obsahy sloupců nemohou mezi sebou automaticky přetékat (základní podmínka víceloupcové sazby) a navíc vyžaduje velmi přesný pracný propočít relativního umístění a přesunutí jednotlivých rámeček tvořících sloupce víceloupcové sazby na správné cílové místo; výhodou je, že při tomto způsobu simulace víceloupcové sazby zůstane zachována čistota struktury dokumentu

Ilustrace: Posun prvku

```
<P style="position:relative; left:25px"> ... </P>  
<P>Odstavec následující za posunutým odstavcem</P>
```