**1. Селектори class и id**

Со селекторите class и id можат да се дефинираат разни стилови независно од XHTML елементите

Да претпоставиме дека ни се потребни три видови на параграфи:

- лево подредени, - десно подредени и - центрирани

нив можеме да ги реализираме на следниов начин

<style type="text/css">

p.levo { text-align : left }

p.desno { text-align : right }

p.centar { text-align : centar }

</style>

Слично со тагот id

<style type="text/css">

#levo { text-align : left }

#desno { text-align : right }

#centar { text-align : centar }

</style>

НАПОМЕНА : иста id вредност може да се појави само еднаш по документ

Можно е имплементирање на повеќе од една класа по елемент

Нека стилот биде вака дефиниран

<style type="text/css">

p.levo { text-align : left }

p.desno { text-align : right }

p.centar { text-align : center }

p.crveno { color : red }

</style>

а вториот параграф

<p class="centar crveno"> Овој параграф ќе биде центриран и црвен. </p>

Ако името на елементот не се специфицира при дефинирање на класата .desno { text-align : right } тогаш таа класа ќе важи за сите елементи

Пример <h1 class = "centar crveno">

**2 Селектор по атрибут**

CSS 2 воведе систем за пристапување до специфични имиња на атрибути и нивни вредности. Засега поддршката не е сигурна во сите прегледувачи. (Не работи кај Internet Explorer 6)

Има четири варијанти на селектори со атрибут:

- **Најширокиот атрибут селектор пристапува до елементи со соодветен атрибут без разлика на неговатa вредност. Синтаксата е:**

**element[attribute]**

Пример: img[title] {border: 3px red;}

Специфицира сите слики во документот кои имаат и атрибут наслов да добијат црвена граница.

**Точна вредност на атрибутот, селектира елементи на база на атрибут со точно дефинирана вредност на атрибутот.**

**element[attribute="exactvalue"]**

Пример: img[title="first grade"] {border: 3px red;}

Само сликите кои во атрибутот title имаат вредност first grade се избрани да имаат црвена граница. Вредноста мора да е идентична како во CSS дефиницијата.

**Парцијална вредност на атрибут**

За атрибути кои прифаќаат вредности со празни места, овој селектор по атрибут овозможува да се селектира само според еден од овие вредности разделени со празни места. (наместо точна идентичност на целата вредност на атрибутот). Тилдата (~) во селекторот го разликува овој селектор од претходниот со егзактната вредност.

**element[attribute~="value"]**

Пример: img[title**~**="grade"] {border: 3px red;}

Овој селектор наоѓа слики кои го содржат зборот grade во листата на вредностите во атрибутот title. Сликите со атрибути title="first grade" или title="second grade" ќе се изберат со овој пример селектор.

**Вредности на атрибут разделени со цртичка (Hyphen)**

Овој селектор е наменет за вредности разделени со цртичка. Селекторот ја бара вредноста која е специфицирана или вредност која почнува така па следи цртичка и нешто друго. Овој тип на селектор е дефиниран со вертикална цртичка (|)..

**element[attribute|="value"]**

Пример: \*[HReflang|="es"] {color: red;}

Овој селектор ги бара сите елементи кај кои атрибутот HReflang е es или започнува со es-. Ќе бидат селектирани елементите кај кои јазикот на линкот кон кој водат се: es, es-ar, или es-es (односно, сите варијации на шпанскиот јазик). (на пример се користи за да се стават знамиња до линковите кон сајтови презентирани во други јазици).

**3 Линкови и привидни класи (pseudo class)**

Каскадните стилови овозможуваат на линковите да им се доделат разни ефекти. Без каскадните стилови ефектите би можеле да се реализираат само со Jава скрипти.

Линковите имаат 4 можни состојби

link – линк кој сеуште не е посетен

visited – линк кој бил посетен

hover – линк на кој во актуелниот момент покажува глушецот

active – селектиран линк

за модификација на приказот на овие состојби се користат привидните класи. Овие ,,фантомски’’ класи не се појавуваат никаде во кодот, туку се базирани на состојбата на овие елементи во самиот документ.

синтаксата за примена на привидните класи е следнава

selector:pseudo-class {property: value}

Различни бои на посетените и непосетените линкови можат да им се доделат со следнава примена на псевдо класите

a:link { color: blue; }

a:visited { color: red; }

a:hover { color: orange; font-style: italic; }

**Правилниот редослед на CSS таговите е :link, :visited, :hover, :active.**

**3.1 Псевдоелементи**

**:first-line**

Аплицира правило за стил на првиот ред од специфицираниот таг. Својствата на :first-line псевдоелементот се ограничени на color, font, background, word-spacing, letter-spacing, text-decoration, vertical-align, text-TRansform, line-height и text-shadow.

**:first-letter**

Додава стил на првата буква на елементот. Може да се менуваат својства за font, color, background, margin, padding, border, text-decoration, vertical-align, text-transform, line-height и float.

**:before** и **:afte**CSS 2 воведе псевдоелементи кои вметнуваат содржина пред и/или после некој елемент и дефинираат стил за таа содржина.

**3.1.1 Универзален селектор**

**\*** {color: blue; } - се постигнува сите елементи да имаат сина боја

**4 Модел на кутија**

Трите најважни концепти на каскадните стилови се пливањето ( floating ), позиционирањето и моделот на кутија.

Овие три концепти го контролираат начинот на кој елементите се аранжираат и прикажуваат

**Padding – от** (поставата), рамката и маргините обично се преддефинирани на 0 но некои читачи за некои елементи имаат други вредности за padding – от и маргиниите.

За избегнување на вакви проблеми некои вредности можат да бидат поставени на 0 со примена на универзалниот селектор \*

\* {

margin: 0;

padding: 0; }

**5 Inline Boxes**

Исто така постои ,,кутија’’ и во редовите со текст. Секој симбол и елемент во линија од текст исто така генерира замислена кутија.

**Content area**

За текст, висината на содржината се одредува според големината на фонтот. За сликите се одредува според висината и ширината на сликата во пиксели.

**Inline box**

Висината на inline box се пресметува според сумата на висината на содржината на елементот плус leading над и

под содржината. Leading е разликата меѓу големината на фонтот и вредностите за висина на линија. Може да е негативна, што значи дека линиите се преклопуваат.За слики, висината на inline box е висината на сликата во пиксели, плус висината на евентуални граници и маргини на img тагот.

**Line box**

Line box се црта околу врвот на највисоката inline box и дното на најниската inline box. Тоа е тоталната висина на линијата со целата нејзина содржина.

**6 Нормален Тек**

Во нормалниот тек кај CSS layout моделот, текстуалните елементи се поставуваат од горе надолу, од лево на десно (кај јазиците кај кои што се чита така).

**7 Својства на граница**

Со својствата на границата се специфицираат стилот и бојата на границите на елементите

ЗАБЕЛЕШКА: Својството "border-color" не делува ако е употребено само, потребна е и употреба на особеноста "border-style"

**8 Својства на маргини**

Со ова својство се дефинира просторот околу HTML елементите

**9 Својство padding**

Со својството padding се дефинира просторот помеѓу границата и содржината на XHTML елемент

**10 Својство dimension**

Ова својство овозможува контрола врз висината и ширината на XHTML елемент и контрола врз растојанието помеѓу редовите

**11 Својства classification**

Ова својства специфицира како и каде ќе биде прикажан некој елемент

- Сите float елементи добиваат својство на блокови (еквивалентно е на display: block).

**11.1 Негативни маргини и преклопување**

Единствена шанса два пливачки објекти да преминат преку границите на блокот во кој се сместени е со доделување негативни маргини.

img { float: left; margin: -10px; }

**11.2 Clearing**

Се користи кога сакаме на некоја од страните на пливачки елемент да нема други објекти, да биде чисто.

Се применува само на блок елементи.

**12 Својство positioning**

Ова својство овозможува специфицирање на позицијата и обликот на XHTML елемент, да смести еден елемент зад друг и да специфицира што ќе се случи кога елементот е преголем за да се смести во доделеното подрачје.