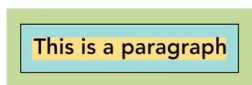


1b Box model

- o nakreslite krabicový BOX model CSS elementov
- o popíšte jednotlivé parametre BOX modelu,
- o vysvetlite rozdiel medzi padding a margin,
- o prevedte výpočet celkovej šírky a výšky elementu,
- o objasnite pojem "margin collapse",
- o pojednajte o 2 spôsoboch riešenia problému "margin collapse"



- width & height
- padding
- border
- margin
- background

Ak nastavím divu šírku 300 px a výšku 100 px v skutočnosti je to viac, lebo sa k nim pripočíta padding = vnútorný okraj, border = orámovanie a margin = vonkajší okraj od strán

Ak ich nenastavíme prehliadač doplní pre margin a padding svoje hodnoty a border sa nezobrazí.

Príklad: odstavec šírka 300px margin 10 px, padding 5 px, border 2 px , dokopy šírka bude 334 px

Každá strana sa dá nastaviť samostatne pomocou:

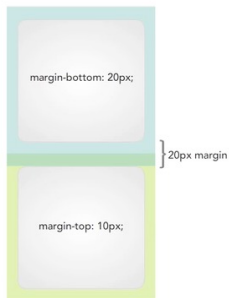
margin-top, margin-right, margin-bottom, margin-left a rovnako aj padding a border.

!!!POZOR!!! Pozadie (farba alebo obrázok) sa "rozťahuje" iba POD elementom, paddingom a borderom, NIE POD MARGINOM.

Collapsing margins - zlievanie vonkajších okrajov objektov ktoré sú nad sebou v zvislom smere.

Riešenie:

- Jednému z elementov dáme požadovaný margin
- Pri vnorených elementoch dáme nenulový padding
- Pri elementoch nad sebou nastavíme border pre jeden z nich – ten medzi nimi



2a Pozicionovanie

- pojednajte o pozicionovaní elementov na stránke,
- objasnite vlastnosť Position a vymenujte jej možnosti,
- vysvetlite relatívne pozicionovanie a jeho vplyv na ostatné elementy,
- vysvetlite absolútne pozicionovanie a jeho vplyv na ostatné elementy,
- vysvetlite fixné pozicionovanie a jeho vplyv na ostatné elementy,
- objasnite dedičnosť vlastnosti position a jej vplyv na rodiča a deti

Normálny tok stránky

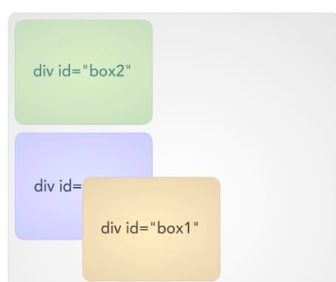
Je taký, ako je stránka písaná v kóde = zobrazuje sa v tom poradí zhora zľava doprava dole.

Dá sa narušiť s css vlastnosťami ako je float a position, z-index.

Position- slúži nastavenie umiestnenie prvku a môže byť:



```
#box1 {  
  position:relative;  
  width: 100px;  
  left: 120px;  
}
```



```
#box1 {  
  position:absolute;  
  width: 100px;  
  left: 50px;  
  top: 200px;  
}
```

1)

static-východiskové-normálny tok dokumentu, nemožno nastaviť súradnice ;

2) relative-prvok je v normálnom toku, ale je posunutý o zadané súradnice: **top, left**. Ostatné prvky sa zobrazia tak, ako by posunutý nebol; Pôvodné miesto prvku ostáva prázdne.

3) absolute-prvok sa zobrazí v polohe podľa súradníc vzhľadom na rohy nadriadeného elementu/tagu, ten je body alebo najbližší ktorý má nastavenú : position- absolute/relative. Ostatné elementy ho nevidia, a zaujmú jeho pôvodné miesto. Posúva sa slovami **top, left, bottom, right**.

4)fixed-zostáva na mieste ktoré má určené voči oknu prehliadača (neskrolluje sa, zostáva na tom istom mieste). Posúva sa slovami top, left, bottom, right.

!3 a 4 vytrhne prvok z normálneho toku, ostatné ho nevidia!

Hodnoty posunu možno zadať ako:

- o vzdialenosť - v bežných jednotkách
- o v percentách - z rozmerov obmedzujúceho bloku

3bFloat parameter

- pojednajte o normálnom toku stránky
- vysvetlite vplyv parametra Float na normálny tok stránky,
- ukážte na vlastnom príklade vplyv Float na ostatné elementy,
- objasnite vplyv Float na nadriadený element,
- popíšte obnovenie normálneho toku stránky,
- pojednajte o iných vplyvoch na normálny tok stránky

FLOAT - určuje obtekanie objektu a môže mať tieto hodnoty:

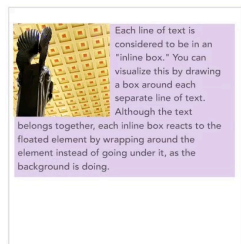
1. none-prvok nie je obtekaný
2. right/left-doprava/doľava

Ak dám viac elementom za sebou, skladajú sa vedľa seba, a keď už v riadku nie je miesto, skladajú sa do ďalšieho ale TAK, ABY boli čo najvyššie!!! POZOR NA TO.

Všetkým elementom s float nastav aspoň width!

!float ruší normálny tok stránky, vytrhne element z toku a stáva sa že nadriadený element ani elementy za ním ho nevidia!

Na obnovenie normálneho toku stránky sa vyrába trieda (napr.: obnov, cistic,...nie anglické názvy) .obnov{clear:both;} a triedu použijeme na mieste kde chceme obnoviť tok stránky (prázdny znak, zlom riadka,...)



img {float:left;}



```
#box1, #box2, #box3 {
float:right;
}

br.clear {
clear:both;
}
```

Add a
 element after the last floated element (any element will do, spans and br elements are often used due to their neutral effect on layout)

Strategies for containing floats

1. Float the container
> may cause layout issues
2. Clear contained element
> may require non-semantic markup

4b Programovanie animácií

- popíšte využitie programovacieho jazyka v animácii,
- opíšte syntax jazyka Action Script,
- na príklade vysvetlite udalosti, akcie a ciele,
- objasnite pojem premenná a predvedte jej definovanie,
- pojednajte o cykloch a ich využití v animáciách,
- vysvetlite vetvenie programu na príklade v Action Scripte,
- predvedte príkazy ovládania časovej osi.

Využitie je na vkladanie objektov, Ovládanie ich správania, spustenie opakujúcich sa udalostí, vytvorenia hier pre webové stránky, mobilné telefóny a tablety. Dosť podobné s javascriptom odpovede.

Syntax js

- príkazy malými písmenami
- metódy z viac anj slov majú veľké písmená na začiatku OKREM prvého
- názvy premenných sami ale podobne datumNarodenia
- !CASE sensitive
- na konci bodkočiarka alebo {}
- na medzery je necitlivý

```

/*premenne*/
var i:Number;
var colorT:ColorTransform = new ColorTransform();
var jumping:Boolean = false;
var meno:String = "Jano";

/*priradenie nacuvaca objektom na kliknutie = udalost */
random_mc.addEventListener(MouseEvent.CLICK, zmenFarbu);
boarder_mc.addEventListener(MouseEvent.CLICK, skoc);

/*funkcie co sa stanu po kliknuti = akcie*/
function zmenFarbu(event:MouseEvent):void
{
    /*cyklus = opakovanie 6x*/
    for(i = 0; i<6; i+1) {
        colorT.blueOffset = Math.round(Math.random() * 510) - 255;
        colorT.redOffset = Math.round(Math.random() * 510) - 255;
        colorT.greenOffset = Math.round(Math.random() * 510) - 255;

        boarder_mc.transform.colorTransform = colorT;

        colorT.blueOffset = Math.round(Math.random() * 510) - 255;
        colorT.redOffset = Math.round(Math.random() * 510) - 255;
        colorT.greenOffset = Math.round(Math.random() * 510) - 255;

        mountain_mc.transform.colorTransform = colorT;
    }
}

function skoc(event:MouseEvent):void
{
    /*vetvenie programu cize podmienka if*/
    if(x<3)
    {
        event.target.y += 10;
        jumping = false;
        event.target.rotation += 45;
    }
    else
    {
        event.target.y -= 10;
        jumping = true;
        event.target.rotation -= 45;
    }
}

```

aby typ movieclip sa správal ako tlačítko
boarder_mc.buttonMode = true;
random_mc.buttonMode = true;
Ovládanie časovej osi je pomocou gotoAndPlay(číslo frejmu) a gotoAndStop.

6a Tagy v HTML

- pojednajte o HTML jazyku,
- rozdeľte HTML tagy podľa miesta v stránke a opakovania v kóde,
- pojednajte o tagoch používaných v hlavičke stránky,
- vymenujte blokové tagy
- predved'te textové tagy,
- rozdeľte a predved'te tagy typu zoznam,

- popíšte tagy súvisiace s tabuľkami,
- pojednajte o tagoch súvisiacich s odkazmi,
- predvedzte tagy typu obrázok,
- pojednajte o tagoch používaných vo formulároch.

Tagy su značky alebo teda slovná zásoba html.

- párove <p>.....</p>
- nepárové <img.....>

Základná štruktúra stránky:

```
<!DOCTYPE html>
<html>
<head>
<title> .....</title>
<meta charset="UTF-8">
..
</head>
<body>
..
</body>
</html>
```

V hlave sa nachádza:

- titul stránky = <title> </title>
- štýly CSS
- Javascripty
- meta tagy

Kódovanie v EU je < meta charset="UTF-8">

HTML Atribúty

pridajú dodatočné vlastnosti tagu, napríklad:

```
<html lang="sk" class="no-js">
</html>
```

Komentáre <!-- toto je komentar -->

Textové tagy

su tie ktoré zobrazia text

- Odstavec <p>1.odstavec </p>
- Nadpisy <h1>nadpis </h1> (h1-h6)
- Adresa


```
<address>
SOSE TT <br>
Sibirska 1
</address>
```
- Ostatné:
 - Prázdny znak
 - Horný index <p>R²</p>
 - Dolný index <p>R₂</p>
 - Predformatovaný text (zobrazí ho tak ako ho napíšete)


```
<pre>
SOSE      Trnava
      Sibirska
</pre>
```
 - Span (slúži na oddelenie časti vety ktorá je dôležitá)


```
<p>toto je <span>dolezity</span></p>
```
 - Citácia v riadku (na stránke sa zobrazuje v úvodzovkách)


```
<q cite:"http://sme.sk/jak.html">
      Toto je citacia
</q>
```
 - Blocquote - to iste ako citácia, ale platí pre odstavec.

Vkladanie obrázku

 (tag je nepárový a riadkový) a má atribúty: pre čítačky nevidomých sa zadáva

alt="text nad obrazkom"

Rozmery, pričom ak zadáte len jeden, druhý sa prepočíta automaticky width="120px" height="400px"

- obrázok sa správa ako inline prvok = viac vedľa seba ak sa zmestia
- dôležitý obrázok obkolesíme tagom <figure></figure>

popis obrázka figure je <figcaption>Popis</figcaption>

FORMULÁRE hľadaj dole v samostatnej otázke

7b E-knihy

- pojednajte o e-knihách a ich použití,
- rozdeľte formáty e-knih a priradte každému miesto použitia, výhody a nevýhody,
- popíšte spôsob tvorby e-knihy pomocou kancelárskych balíkov,
- pojednajte o profesionálnych aplikáciách na tvorbu e-knih,
- pojednajte o konvertovaní e-knih do iných formátov,
- popíšte nástroje synchronizácie e-knih pre prenosné zariadenia a ich operačné systémy,
- popíšte aplikácie pre tvorbu e-knih v iOS alebo Android.

Tu treba hovoriť to čo ste sa učili na GDM. Chceme zvýrazniť: hlavným formát e-knih je .epub. Ten sa zobrazí na každom zariadení inak, a to podľa veľkosti nastaveného fontu. Text sa dá zväčšovať a zalamovanie riadkov sa automaticky prispôbi. Tým sa líši od ostatných formátov.

Pomocou kancelárskeho balíka vytvoríme PDF súbor tak, že ho dáme uložiť ako PDF. Niektoré operačné systémy ponúkajú možnosť čokoľvek uložiť do PDF tak že objekt dáme vytlačiť a miesto tlačiarne vyberiem PDF.

Profesionálna aplikácia treba podpísať adobe InDesign.

Na konvertovanie sa používa bezplatná aplikácia Calibri. Otvoríme v nej súbor - aj office - vyberieme si výsledný formát- Najčastejšie epub, nastavíme obrázok titulky, autora, vlastníka práv a ostatné informácie. Dáme konvertovať.

9a Pokročilé funkcie animácií

- pojednajte o symboloch v animáciách a ich využití,
- opíšte druhy symbolov a ich nastavenie,
- popíšte pojem maska a ukážte na vlastnom príklade,
- vysvetlite úpravu animácie podľa vlastných požiadaviek (pohyb po krivke, transformácia pomocou bodov),
- popíšte scénu a organizáciu animácie,
- popíšte možnosti vkladania zvuku do animácie.

Symbol môže byť akýkoľvek objekt, ktorý vytvoríme a cez pravé tlačítka z neho urobíme symbol. Všetky symboly sa vkladajú do knižnice, majú svoj názov, a Dajú sa kedykoľvek umiestniť do plochy. Sú 3 typy:

- movieclip = má vlastnú časovú os, má najpokročilejšie nastavenia
- button = tlačítka, má 4 stavy a v každom môže inak vyzeráť. 4.stav je len plocha, kde bude reagovať za kliknutie
- graphics = grafický objekt, minimum použitia

Maska je to isté čo photoshop- kľúčová dierka cez ktorú pozeráme na danú vrstvu.

Posledné verzie flashu majú pokročilý editor pohybu, cez ktorý sa dá nastaviť akákoľvek zmena, transformácia objektu, pohyb, spôsob príchodu a odchodu objektu, zrýchlenie, spomalenie, rotácia.

Každá Animácia po otvorení má scene1, steny sa dajú pridávať cez okno scén, v každej môžeme mať samostatnú animáciu alebo jej časť a potom ich spájať presúvaním poradia jednotlivých scén.

Zvuk do animácie musíme najprv importovať, cez menu file import alebo kontrol R. Zvuk sa umiestniť do knižnice bolo snímku keď ho chceme pridať pridáme kľúčový snímok a v ňom v paneli vlastnosti vyberieme zvuk z ponuky.

9b Javascript

- pojednajte o Javascripte, jeho využití vo vývoji stránok a v súčasnosti,
- predvedte deklaráciu rôznych typov premenných,
- predvedte základné aritmetické a logické výrazy,
- zdefinujte vlastnú funkciu obsahujúcu cyklus,
- vymenujte základné objekty v JavaScripte,
- pojednajte o udalostiach jazyka JavaScript,
- vysvetlite možnosti vloženia JavaScriptu do www stránky,
- vysvetlite a na príklade ukážte základné hlášky,
- vytvorte ukážkový príklad HTML stránky s použitím JavaScriptu,

- definujte pojem ASCII tabuľka.

JavaScript + CSS + moderný prehliadač = DHTML (dynamická HTML)

Tak vzniká dynamická stránka schopná reagovať na akcie užívateľa bez odosielania údajov na server. Reakciami sa mení štandardný statický nemenný obsah na dynamický. Typickým príkladom môže byť zmena výzoru stránky, rolovacie menu alebo aj výpočet veku návštevníka stránky a podobne. Interaktivita ("dynamickosť") stránky môže byť zabezpečená dvoma spôsobmi:

- použitím skriptovacieho jazyka bežiaceho v prehliadači návštevníka na zmenu výzoru stránky po akciách spôsobených najmä myšou, klávesnicou alebo v určených časových intervaloch = JS
- použitím skriptovacieho jazyka bežiaceho na serveri. V tomto prípade je časté čítanie údajov z databáz, ukladanie užívateľom zadaných dát do databáz, vyhľadávanie údajov a podobne = PHP

JS je skriptovací jazyk, ktorý je súčasťou HTML a je vykonaný na strane klienta. Pracovný priestor je LEN na strane prehliadača. S JS sa dá teda meniť len obsah stránky a prehliadač. Vie pracovať s objektami html aj css vlastnosťami.

Nemôže pracovať s databázou, prístupovať k Hardwaru. Umiestnenie JS:

- do hlavičky stránky samotný skript = ak chceme zabezpečiť, aby návštevník stránky nemal možnosť nič na stránke urobiť, kým sa nenahrá script medzi `<script>.....</script>` = NA PREMENNÉ A ŠTARTOVNÉ FUNKCIE
- do hlavičky odkazom na externý súbor s príponou .js
 - `<script type="text/javascript" src="knihovna.js"></script>` = NA VLASTNÉ FUNKCIE
- do tela stránky medzi `<script>.....</script>`, zvyčajne pred jej koniec pred `</body>`
- inline zápis ako atribút tagu, `Urad Slana`

Spustenie js

- pri udalosti = klik, prechod myšou nad
- po určitom čase alebo aj opakovane
- po načítaní stránky

```
alert("Hello world"); ✓
Alert("Hello world"); ✗
```

Syntax js

- príkazy malými písmenami
- metódy z viac anj slov majú veľké písmená na začiatku OKREM prvého getElementById
- názvy premenných sami ale podobne datumNarodenia
- !CASE sensitive
- na konci bodkočiarka alebo {}
- na medzery je necitlivý

```
alert("Hello world");
alert ( "Hello world" ) ;
alert
( "Hello world"
);
alert ("Hello world");
```

Premenné

Slúžia na uloženie rôznych hodnôt podľa vašich potrieb. JavaScript nevyžaduje ich zdefinovanie dopredu, ale je lepšie, ak ich zdefinujeme na začiatku. Spôsoby: !!!! CASE SENSITIVE!!! Nesmie sa začínať číslom a BEZ diakritiky, medzier. Povoleny znak_tento.

textova premenna

```
var newText=document.getElementById("myInput").value;
```

číslo

```
var mesiac=4;
```

premenna typu pole

```
var pole=["pizza", "chocolate", "crisps"];
```

hláška

```
alert("Page loaded!");
```

vetvenie programu pomocou if = rozhodovanie s podmienkou

```
if (document.getElementById("weather").value=="rain") {
  alert("prsi");
}
else {
  alert("neprsi!");
}}
```

cyklus for

```

for (var i=0; i<pole.length; i++) {
    arrayString=arrayString+pole[i]+" ";
}

```

Ak chceme nejaký výpočet robiť viac krát, oplatí sa zdefinovať vlastnú funkciu.

```

function nazovFunkce() {
    // príkazy funkce
}

```

Udalosti

Sú akcie človeka, ktorý si prehlíada stránku a pracuje s klávesnicou a myšou. Môže to byť:

- Stlačenie konkrétnej klávesy
- Stlačenie myši
- Prejazd myšou nad elementom atd

Napr. pridanie reakcie na kliknutie(=udalost) sa vykona funkcia (akcia)

```

document.getElementById("myButton").onclick=function() {...}

```

Operácie s premennými

- | | |
|--|---------------------------|
| • matematické operácie: +, -, *, /, % (zvyšok po celočíselnom delení) | score += 10; |
| • ++ (zvýšenie hodnoty o 1), -- (zníženie hodnoty o 1) | += -= *= /=
a = a + 1; |
| • operácie s textovými tetazcami: + (spojene textov) | a += 1; |
| • relácie: == (je rovno), != (nerovno), < (menší), <= (menší rovno), >, >= (väčší rovno) | a++; |
| • logické operácie: && (and - a zároveň), (or - alebo) | ++a; |

Knižnice js: dnes sa používajú knižnice vstavaných funkcií ako je jQuery. Uľahčujú prácu a stačí ich pripojiť ako odkaz do hlavičky. Sú v nich hotové rolovací menu apod.

10a HTML5

- pojednajte o nových možnostiach, ktoré priniesol HTML5,
- predvedte deklaráciu dokumentu typu HTML5,
- vymenujte a popíšte HTML5 elementy uľahčujúce orientáciu v stránke,
- predvedte CSS vlastnosť pre elementy na orientáciu v stránke,
- objasnite elementy figcaption a figure,
- pojednajte o elementoch section a article, ako aj o ich vhodnej kombinácii a použití,
- vysvetlite zmenu pre tučné písmo a kurzívu v HTML5,
- pojednajte o nových atribútoch a ich hodnotách pre element input,
- popíšte element INPUT typu dátum, číslo, rozsah, farba a ich vzhľad v prehliadačoch.

<!DOCTYPE html>

HTML5 priniesol aj nové tagy na zjednodušenie písania stránok, lepšiu orientáciu v stránke a zrýchlenie vkladania multimédií.

1. Tagy na orientáciu v stránke = miesto divov:

- header
- footer
- nav = navigácia
- aside = bočný panel
- main = hlavný obsah stránky
- section = sekcia a môže obsahovať viac článkov
- article = článok
- <code></code> je sa používa na zobrazenie kódu programu alebo niečoho. Pozor, znaky ako < musíme zadať pomocou jeho ekvivalentu, ktorý nájdeme na nete.
- figure = okolo dôležitého obrázka
- figcaption = popis obrázka

!Všetkým okrem obrázkových aby ich staršie prehliadače správne zobrazili, nastavme v CSS *display:block*;

2. Multimédiá = konečne môžeme vkladať naše súbory

a) Video

```

<video width="300" height="300" controls>
    <source src="video.webm" type="video/webm">

```

```
<source src="video.mp4" type="video/mp4">
<p>Video si<a href="video.webm">stiahnite</a>.</p>
</video>
```

b) Audio

```
<audio src="audio.ogg" controls></audio>
```

3. Formulár - pridáva typy pre input, type = ... v časti Formuláre

11b Odkazy a viditeľnosť objektov

- pojednajte o pojme odkaz,
- vymenujte základné typy odkazov,
- objasnite pojem absolútny odkaz a ukážte jeho realizáciu,
- objasnite pojem dokument - relatívny odkaz a ukážte jeho realizáciu,
- objasnite pojem stránko - relatívny odkaz a ukážte jeho realizáciu,
- predvedte odkaz typu email na príklade ho zrealizujte aj s nápovedným textom,
- vysvetlite 4 stavy odkazu a navrhnete ich formátovanie,
- opíšte možnosti nastavenia viditeľnosti objektov,
- popíšte a predvedte nastavenia poradia objektov.

Odkazy:

tá časť stránky na ktorú sa da kliknut a presmeruje nás na inú stránku. Textový odkaz na FB je
Facebook

Atribúty odkazu:

```
<a href="cesta_odkazu.html" target="_blank">odkaz</a>
_parent _self _top
```

blank otvorí odkaz v novom tabe. Odkaz môže byť aj obrázok, čokoľvek.

Druhy odkazov

- Absolútne – celá cesta k odkazu na webe napr. <http://sme.sk>
- Relatívne – odkazy na moje súbory, ktoré sú v blízkych priečiňkoch napr. video.html, zmluva.pdf

Zadávanie cesty k súborom v iných priečiňkoch

Súbory vnútri ďalšieho priečinka s názvom obrázky adresujeme ako [obrazky/pozadie.jpg](#).

Cesta do nadriadeného riečinka sa robí s dvoma bodkami za sebou [../../docs/fero.jpg](#)

Správanie odkazov

Odkazy môžeme nastaviť tak že keď sme na nad nimi myšou zmenia svoj stav. Robí sa to pseudotriedami, pričom je dôležité ich poriadie. Musia byť aj viaceré naraz ALE v poradí:

link visited hover active

a:link je základný stav odkazu, a:visited je stav už navštíveného odkazu, a:hover je stále odkazu v okamžiku keď je nad ním myš. Príklad: html časť hore pre menu pod sebou. CSS:

```
a {
    text-decoration:none;
    display: block;
}
a:link, a:visited {
    color: #fff;
    background-color: #0000dd;
}
a:hover, a:focus {focus je pre posun s TABulátorom
    color: #0000dd;
    background-color: #fff;
}
.active {
    color: #fff;
    background-color: #00dddd;
}
```

12a Formulár v HTML

- pojednajte o formulári a jeho prvkoch z hľadiska jeho úlohy a dizajnu,
- vymenujte a popíšte jednotlivé prvky, ktoré obsahuje formulár,
- vysvetlite úlohu tagu fieldset a legend a navrhnete jeho nastavenie pomocou CSS,

- vymenujte a opíšte elementy formulára typu input a typu textarea,
- pojednajte o hodnotách atribútu TYPE, ktoré elementu INPUT priniesol HTML5,
- pojednajte o podpore typov INPUT HTML5 v súčasných prehliadačoch,
- objasnite ako zabezpečiť grafickú interaktivitu prvkov formulára
- vysvetlite 2 metódy komunikácie formulára s databázou, ako aj pojem akcia.

FORMULÁRE

Slúžia na zadávanie údajov a informácií a ich odoslanie za účelom spracovania, prihlásenia atd. Dáva sa do blokového tagu `<form>...</form>`

Ak je vo formulári viac oblastí (osobné údaje, záujmy, povolanie ...), rozdelíme formulár na oblasti = `<fieldset>.....</fieldset>`. Legend je jej popis.

Základom formulára sú aktívne prvky = tag `<input>`. Je nepárový!!!! a inline element!!!

Správny zápis (odporúčaný):

```
<label for="meno">Meno:
  <input type="typ" id="meno" name="meno" placeholder="sem meno" />
</label>
```

Typy inputu:

- | | |
|------------------------|--|
| 1. Text | 10. Button |
| 2. Password | 11. Odosielacie submit |
| 3. Email | 12. Reset |
| 4. Time | 13. Zafajkávacie checkbox – aj viac naraz |
| 5. Date | 14. Výberové radio – len jedno |
| 6. Number (min aj max) | 15. Atribút placeholder je šedý text v políčku ako nápoveda pre toho čo vyplňa |
| 7. Search | |
| 8. Image | |
| 9. Url | |

Textarea je viacriadkový text a je to párový blokový tag.

```
<textarea rows="4" cols="50">dlhy text .....</textarea>
```

Štýly formulárov

Nastavujeme hlavne šírku, farby písma a pozadia pre label, legend, a ich rozostupy pomocou margin. V html kóde treba spárovať popisok label pomocou `for="meno"` s `id="meno"` v input elemente.

Je výhodné používať triedy na nastavenie rovnakej šírky input.

Zaškrtávacie alebo výberové políčka zostanú vedľa seba.

Na zvýraznenie toho poľa, ktoré práve vypisujeme, použijeme pseudotriedu focus

```
input.text:focus, textarea:focus {...};
```

Na rozostupy NEPOUŽÍVAŤ `
` ale MARGINY!!!!

input atribút for sa viaže na ID v inpute, alebo sa dá použiť stratégia:

```
<label>
  <input type.. > patriaci k text labels inputu
</label>
```

a daj mu v css

```
label {cursor:pointer;}
```

Rozdiel je v tom, že teraz funguje aj kliknutie na text v labeli ako na input, hneď sa zaškrtnie a podobne. Pridané atribúty:

- checked znamená že bude vybraný hneď
- pre select je to selected

odosiela sa value hodnota po kliknutí na button. Toto sa dá použiť na input aj check, select, radio.

```
<input type="file" .../>
```

v css mu daj

```
input[file]:focus {outline:none;}
```

Pre označení niečoho napr. v formulári dáme

```
::selection (bgclor, color, text-shadow:none;}
```

2 metódy posielania údajov sú:

- GET - zapísané vybrané údaje sú pridané do adresy v prehliadačido 3000 znakov, takže ich hocikto môže vidieť.
- POST - táto metóda je bezpečnejšia, pretože žiadne údaje nie sú viditeľné, neobmedzený počet údajov, nedá sa bookmarknúť.

```
<form action="/action_page.php" method="get">
```

14b Tabuľky v HTML

- pojednajte o tabuľke ako prvku webstránky a jej úlohe v histórii vývinu stránok,
- vymenujte a popíšte html tagy súvisiace s tabuľkou,
- vysvetlite úlohu tagu tbody, thead, tfoot, caption,
- predvedte spájanie susedných buniek tabuľky v oboch smeroch,
- pojednajte o vlastnostiach tabuľky nastavovaných pri jej vytvorení,
- pojednajte optimálnom postupe nastavenia tabuľky pomocou tried,
- popíšte možnosti CSS3 pri úprave vzhľadu tabuľky.

Tabuľky slúžia na zobrazenie porovnaní výrobkov, tabuliek.

V minulosti slúžili ako neviditeľná KOSTRA stránky, ktorá držala texty obrázky odkazy pokope. Bolo to neprehľadné v kóde a ťažko sa robili. Dnes sa to NEROBÍ!!!!!!

Tabuľka začína a končí tagom <table>...</table>

Ide sa po riadkoch a každý riadok je <tr>.....</tr>

Každá bunka je <td>.....</td>

Spájanie buniek

Spájanie stĺpcov

```
<td colspan="4">smer vedľa seba</td>
```

Spájanie riadkov

```
<td rowspan="4">smer pod seba</td>
```

Pri colspan aj rowspan musíme zmazať bunky ktoré sme už použili.

Spájané môžu byť aj riadky a stĺpce naraz, vtedy sa za seba dá aj rowspan aj colspan. Nezáleží na poradí.

Dôležité bunky tabuľky = hlava tabuľky

Miesto <td> sa použije <th>Obsah</th>

Slovo Obsah sa zobrazí v strede a hrubým písmom.

Postup pri tvorbe: V html!

Dajte tabuľke atribúty id(meno) alebo triedu, summary (čoho sa týka) a vynulujeme už v html border, cellpadding, cellspacing.

```
<table border="0" cellpadding="0" cellspacing="0" id="kamery"
summary="Porovnanie kamier">. Pri dôležitých tabuľkách dajme aj popis tabuľky = býva nad ňou. !!!ZA
<table>!!!
```

```
<caption>Camera Specs</caption>
```

Tabuľka je jediné miesto, kde sa dá nastaviť zvislé zarovnanie obsahu bunky s
vertical-align: top, bottom, middle

V časti CSS:

1. Pre tabuľku nastavíme šírku, zarovnanie, okraj pozadie, fonty atd.

```
table #kamery {
width: 500px;
border: 1px solid #333;
margin: 0 auto;
border-collapse: collapse;
background: #574F3E;
}
```

2. Nastavíme bunky hlavičky th

3. Nastavíme popis caption

4. Nastavíme vlastnosti bunky td

```
table#kamery td {
padding: 1.5em 1em;
border: 1px solid #333;...
}
```

5. Na oddelenie riadkov tr striedame farby ich pozadia pomocou pseudotried :nth-child(odd) /: nth-child(even) alebo nth-child(2n+1) ...

```
table#kamery tr:nth-child(odd) {
background: #8C7E62;
}
```

6. Na orientáciu v tabuľke dame zmenu farby riadka, nad ktorým je kurzor pseudotriedou :hover

```
table#kamery tr:hover {
background: #333;
}
```

Pietro Alessandro 25/11/15 21:42

Comment [1]: nepárne riadky budú mať túto farbu

16a Základy CSS

- objasnite účel a pojem kaskádových štýlov CSS,
- objasnite správnu syntax CSS pravidiel,
- uveďte 3 možnosti umiestnenia CSS pravidiel a objasnite ich poradie dôležitosti,
- vysvetlite rozdiel medzi ID a class v CSS,
- predvedte komplexné selektory a viacnásobné CSS pravidlá,
- objasnite špecifickosť CSS pravidiel a jej vplyv na elementy, poradi ich aplikovania
- vysvetlite pojem dedičstvo – inheritance,
- popíšte a predvedte ako sa robí presun štýlov embedded do externého a naopak,
- popíšte o postupe pri plánovaní stránky a jej tvorbe z hľadiska koncepcie kedysi a dnes a porovnaj ich,

SYNTAX CSS PRAVIDLA

body je selektor = označiť, čoho sa pravidlo týka

```
body {
background-color: white;
color: black;
padding: 10px;
}
```

2 - {
3 - vlastnosť (tu background-color, color a padding)
4 - : dvojbodka musí byť
5 - hodnota vlastnosti (tu biela, čierna a 10px) a za každou musí byť
BODKOČIARKA;
6 - }
skrátенý zápis
p {color:red; font: Arial;}
Môže sa nachádzať na 3 miestach:

2. Priamo v tagu pomocou slova style
<h1 style="color:red;">Nadpis bude cerveny</h1>
 - o volá sa INLINE štýl, vyhýbať sa!!!
3. V hlavičke stránky medzi slovami
<style type="text/css">
h1 {color:red;}
</style>
 - volá sa EMBEDDED, zriedka!!!!
4. V externom súbore s príponou css. Je to textový súbor, kde sú zapísané rovnaké pravidlá, ako v hlave (bod 2). V hlave stránky je odkaz na tento súbor zobrazený ako
<link rel="stylesheet" type="text/css" href="mojstyl.css" />
a v samotnom súbore sú len CSS pravidlá.

Selektory

- Jednoduché: TAGY:
- 1) h1 {color:red;}, p, div, img, ul...
 - Triedy: meno triedy, trieda bude modrá
 1. .trieda {color:blue;}
 - !Najprv treba triedu vyrobiť, potom použiť! v html takto:
<h1 class="trieda">Modrý nadpis</h1>
 - !Dá sa použiť viac krát na stránke pre viac elementov!
 - !Jeden element môže mať viac tried napr class="trieda vpravo tlacitko"
 - ID štýly – pre elementy s ID, VYSKYTUJE SA LEN PRE 1!!!ELEMENT na stránke
#obsah {color:blue}
 - Pseudotriedy - rôzne. Začínajú dvojbodkou. Pri odkazoch sú štyri:
 2. a: link
 3. a: visited
 4. a: hover
 5. a: active
- Ďalšie pseudotriedy sú:
- o input:focus - na miesto formulára, do ktorého práve píšeme
 - o ::before h1 - je miesto PRED elementom h1; obsah sa pridáva s content: "bla"
 - o ::after h1 - je miesto ZA elementom h1

Viacnásobné selektory

- * {margin:0; padding:0;}
 - VŠETKY ELEMENTY= !!! prvé pravidlo CSS na stránke!
 - vymaže okraje, treba potom nastaviť!

- `h1,h2,h3,li {font-family:Arial;}`
- Vnútročné selektory
 1. `div p {color:red;}` = odstavec v dive
 2. `#obsah #clanok p {width:300px;height:200px;}` odstavec v elemente s ID clanok a ten je v elemente s ID obsah
 3. `aside h1.velky {font-weight:bold;}` nadpis 1 v elemente aside s triedou velky
 4. `div > p` priamy potomok divu = všetky odstavce v dive nevnorené do iného tagu.
 5. `div > h2 + p` najbližší súrodeneц h2 v dive = jason sibling
 6. `p:first-line`, `p:first-letter` je od CSS1, v CSS3 sa píše `p::first-line`, `p::first-letter`
 7. `li:first-of-type` a ...
 8. `li:last-of-type` a ...
 9. `p:only-of-type` ...

Síla selektorov

Ak sa ten istý element vyskytne viac krát v CSS, platí pravidlo o SILE pravidla:

Selector	Specificity		
Inline style	1, 0, 0, 0	<code>#sidebar p</code>	0, 1, 0, 1
ID	0, 1, 0, 0	<code>blockquote p</code>	0, 0, 0, 2
Class, pseudo-class, attribute	0, 0, 1, 0	<code>#sidebar blockquote p</code>	0, 1, 0, 2
Type (tag), pseudo-element	0, 0, 0, 1		

Dedičnosť Inheritance

Vlastnosti elementov stránky sa dajú rozdeliť do dvoch skupín, podľa toho, či ich zdedia, aj elementy v jeho vnútri

17a CSS3

- predvedte transformácie objektov pomocou CSS3 (skosenie, otočenie, posun a zmenu veľkosti),
- popíšte CSS3 animácie a prechody medzi stavmi vlastností elementu,
- pojednajte o použití CSS3 vlastností v starších typoch prehliadačov a predvedte ich realizáciu,
- pojednajte o CSS3 selektoroch vyberanie elementov podľa obsahu,
- pojednajte o CSS3 selektoroch riadiacich sastrovom dokumentu,
- popíšte pseudotriedy pre prácu s formulármi,
- predvedte výber elementov nasledujúcich za iným elementom.

Použití sa dajú 2D a 3D transformácie.

2D sú

Zväčšenie

```
.scale {
  transform: scale(1.5);
}
```

Skosenie

```
.skew {
  transform: skew(-15deg);
}
```

Posun

```
.translate {
  transform: translate(0, 50%);
}
```

Otočenie

```
.rotate {
  transform: rotate(-15deg);
}
```

Kombinácie:

```
.skoszvacis {
  transform: scale(1.5) skew(-15deg);
}
.scale-2 {
  transform: scale(1.5); transform-origin: top left;
}
```

3D sú hlavne rotácia = okolo osi XYZ
čiže rotateX(15deg)

Animácie = zmeny stavu = transitions

Umožňujú jednoduché animácie Bez javascriptu. Treba zadať:

1. Vlastnosť, ktorá sa mení = animuje
transition-property: width, border, left;
2. Ako dlho trvá zmena v sekundách, prípadne oneskorenie transition-delay
transition-duration: 0.8s;
3. Spôsob ako sa to deje = rovnomerne, zrýchlene atd.
transition-timing-function: linear, ease-in, ease-out, ease-in-out,
step-start, step-end,

Pozn. transform-origin: bod, okolo ktorého sa zmena deje.

Môžeme použiť aj keyframes a určiť kedy sa aká zmena udeje:

```
@keyframes nase_animace {  
  0%{vlastnost: hodnota;}  
  20%{vlastnost: hodnota;}  
  40%{vlastnost: hodnota;} 60%{vlastnost: hodnota;}  
  80%{vlastnost: hodnota;}  
  100%{vlastnost: hodnota;}  
}
```

18b CSS3

- pojednajte o nových možnostiach, ktoré priniesol CSS3,
- predvedte zaoblenie rohov elementu,
- predvedte tieň elementu a tieň textu,
- popíšte zadávanie farieb pomocou rgb a rgba,
- predvedte farebný prechod pomocou CSS,
- objasnite použitie viacerých obrázkov pozadia,
- vysvetlite rôzne spôsoby rozťahnutia obrázku pozadia,
- predvedte viacstĺpcový text s nastaviteľnými medzerami medzi stĺpcami,
- vysvetlite použitie vlastných fontov do stránky.

CSS3 pridalo oproti predošlej verzii niekoľko možností. Staršie prehliadače to nemusia všetko podporovať, a tak takmer každú vlastnosť by bolo treba napísať viac x a to najprv s predponami:

-webkit pre safari chrome a edge, moz- pre mozilla, -o- pre operu a až potom všeobecný zápis pre moderné prehliadače text-shadow: 1px

Zaoblenie rohov elementu border radius

```
border-radius: 10px; border-radius: 15% 15% 0 0;
```

TIP: zo štvorca urobíme kruh s border-radius:50%;

Tiene box-shadow

```
box-shadow: 5px 5px 10px rgba(0, 0, 0, .5);
```

kde hodnoty znamenajú: posun doprava, dole, rozmazanie a farba tieňa. Ak pred farbou dáme slovo INSET znamená vnútorný tieň. Môžeme ich vrstviť na seba, prvý je ten najviac hore.

Tiene textu text-shadow

```
.tien{  
  text-shadow: 1px 2px 5px #444444;  
}
```

rovnako ako u box-shadow

Prechod farieb

```
background: linear-gradient(45deg, lightgreen, darkgreen);
```

45 deg je uhol, nepovinné. ellipse alebo circle je tvar, nepovinné. Dá sa určiť stred prechodu

```
.box-3 {  
  background: radial-gradient(at top left, lightgreen, darkgreen);  
}
```

Viacnásobné obrázky pozadia

Viac farieb aj viac obrázkov pozadia. Vyhráva ten nižšie napísaný.

Rozťahovanie pozadia

`background-size: (cover/contain) _vertikalny_rozmer_ _horizontalny_rozmer_;`
contain bude vtedy, ak chceme obrázok pozadia vidieť vždy celý.

Zmena správania pozadia

`box-sizing: content-box | border-box | padding-box`

Zmení odkiaľ a kam sa umiestni pozadie.

Priehľadnosť

opacity: 0 až 1 !!! pozor, platí pre celý element nielen pre jeho pozadie, ale aj text.

Použitie vlastných fontov @font-face

```
@font-face {
  font-family: _nazov_rodiny_;
  src: url(_cesta_k_suboru_s_pismom_) format(_format_suboru_);
}
h1 {
  font-family: _nazov_rodiny_;
}
```

Viac stĺpcový text

```
column-width: _sirka_stlpca_;
column-count: _pocet_stlpcov_;
column-gap: _sirka_odsadenia_mezi_;
column-rule: _vlastnosti_ciary_mezi_;
```

Stačí zadať počet.

RGBa farba

```
color: rgba(255, 0, 0, 0.5);
```

Text Overflow

je označenie textu, ktorý sa nezmestil do svojich rozmerov.

```
text-overflow: ellipsis;
overflow: hidden;
white-space: nowrap;
```

This is some long text that ...

19b Pozadia v CSS

- objasnite parameter background v CSS,
- pojednajte o farbách pozadia a kombinácií farieb a obrázkov na pozadí,
- vymenujte rozdiely obrázka pozadia a bežného obrázka na stránke,
- popíšte stav, kedy použijeme pozadie aj pre html, aj pre body,
- vysvetlite na obrázku princíp umiestňovania obrázkov pozadia pomocou percent,
- objasnite veľkosť a umiestnenie obrázku pozadia pomocou CSS3,
- na príklade ukážte zneviditeľnenie textu rovnakého v pozadí pre prehliadač,
- objasnite či a ako sa dajú dať 2 rôzne obrázky do pozadia,
- pojednajte o použití obrázka tvoreného viacerými obrázkami nad alebo vedľa seba ako pozadia a poukážte na výhody tohto riešenia.

Background

Dá sa nastaviť pre každý element, prípadne pre celú – body, môže byť:

1. Farba background-color
2. Obrázok (alebo aj viac) background-image url(adresa/cesta)
 - a. je silnejší než farba
 - b. aby sa obrázok neopakoval background-repeat: (no-repeat)
 - c. ak chceme aby obrázok prikryl celý objekt (stránku) **background-size: cover** alebo 100% (pomenej), ostatné možnosti sú rozmery napr. 300px 100px (šírka výška), alebo contain.
 - d. aby sa obrázok neskrolloval background-attachment: fixed
 - e. pozícia pozadia: background-position:
 - top left center right bottom (zľava doprava aj zhora nadol; prednastavená je center center)
 - 10px 2em = prvé číslo je zľava, druhé zhora

- ak sú napr.obrázky 2 a meníme ich position, bude background-position: 20px 30px, 40px 20px; prvé dve pre prvý obrázok, a potom pre druhý, podobne repeat
 - **v % pričom 0% 0% umiestni obrázok doľava hore a 100% 100% doprava dole**
3. Obrázkom pozadia môžeme nahradiť odrážku v nečíslovanom zozname, no-repeat
 4. Ak je viac obrázkov pozadia, prvý uvedený je vždy navrchu.
 5. Obrázkom pozadia môžeme nahradiť napríklad nadpis, ale v html nechajme aj jeho text napr. slovo kontakt `<h2>Kontakt</h2>` ALE aby sa tento text nebil s pozadím, posunieme ho doľava s `h2 {text-indent:-1000em;}` Tým zabezpečíme, že text bude na stránke viditeľný pre čítačky, a bude sa aj tlačiť. (obrázky pozadia sa netlačia).
 6. Farba alebo obrázok pozadia sa používa aj ako dynamický prvok = ak sme niekde s myšou nad niečím na zvýraznenie kde sme, alebo v akom stave je odkaz. Vždy sú na to pseudotriedy hover: alebo focus:
 - a. na odkaze `a:hover` {zameníme farby}
 - b. na zvýraznenie riadku tabuľky kde sme s kurzorom `tr:hover` {zameníme farby}
 - c. vo formulári na zvýraznenie poľa, ktoré vyplňame `input:focus` {background....}
 7. často použijeme pod odkaz obrázok, a aby sa ich nestahovalo veľa, spoja sa 3 obrázky do jedného, a ten sa pod odkazom posúva podľa stavu odkazu pomocou kľúčových slov


```
a:link, a:visited {background-position: left;}
a:hover {background-position: center;}
a:active {background-position: right;}
```



21a Vlastnosti textu

- pojednajte o rodinách fontov z hľadiska ich použitia na stránke,
 - objasnite dôvod používania viacerých fontov v stránke,
 - vymenujte, popíšte a porovnajte základné rodiny web fontov
 - vysvetlite optimálne riešenie použitia fontov vo web stránke, miesto a spôsob, kde sa udávajú,
 - vysvetlite použitie vlastných fontov do stránky,
 - vymenujte jednotky pre tlačené stránky a popíšte miesto ich použitia,
 - vymenujte jednotky pre zobrazené stránky a popíšte miesto ich použitia,
 - porovnajte jednotku em a ex,
 - pojednajte o parametri line-height, jeho nastavení a jeho jednotkách
1. Pre `<body>` nastavíme rodinu fontov- viac, ak by zariadenie nemalo nainštalované prvé fonty, posledný je vždy- serif alebo san-serif `{font family: "Verdana", "Arial", san-serif;}`
 2. Pre body nastav veľkosť fontu v percentách 100%
 3. Každému textovému tagu nastavíme font-size v „em“ (1em je 16px pre PC, každé zariadenie má svoju veľkosť em)
 4. Pre odstavce a nadpisy nastavíme výšku riadka „line-height“ v percentách alebo násobok napr 1.5
 v percentách alebo násobok sa počíta z veľkosti fontu! Line-height určuje vzdialenosť 2 riadkov od seba a zároveň vycentruje 1-riadkový text na výšku. **Pre bežný odstavec sa nastavuje výška riadka 1.4 pre obrazovky a 1.2 pre tlač.**
 5. Centrovanie textu :
 - Horizontálne=text-align : center, inherit, justify, left, right
 - a. Vertikálne
 - i. jednoriadkový text je vycentrovaný pomocou line-height väčší ako 1
 - ii. viac riadkový text sa centruje 2 cestami: (stačí vedieť jednu)
 1. text dáme do divu a ten do ďalšieho. Tomu vrchnému dáme vlastnosť `display:table` a tomu vnútri dáme `display:table-cell` a textu nastavíme `vertical-align:middle`
 2. Nadriadenému dáme `position:relative;` tomu vnútri nastavíme `position:absolute; top:50%; left:50%;` a textu nastavíme `#vert p {transform: translate(-50%, -50%);}`

Ostatné vlastnosti textu

1. Farba písma
 - a. color: darkblue alebo
 - b. sa použije namiešanie farieb color: #AA00CC alebo

c. color: rgba(255(R),10(G),20(B),0.1(priehľadnosť))

2. Font font-style: italic, normal, oblique
- font-weight: 100–900, bold je 700, normal je 400
 - font-variant: normal, small caps,
 - text-transform: capitalize/lowercase/uppercase,
 - text-decoration: line-through, none, overline, underline
 - word-spacing: určuje vzdialenosť medzi slovami
 - letter-spacing: určuje vzdialenosť medzi písmenami
 - text-indent: určuje odsadenie textu prvého riadka

Väčšina vlastností textu je dedičná.

22a Zoznamy

- pojednajte o zoznamoch v html, o základných typoch, ich formátovaní,
- objasnite možnosti použitia nečíslovaného zoznamu ako horizontálneho alebo vertikálneho menu,
- pojednajte o príčinách tvorby menu ako nečíslovaných zoznamov,
- opíšte postup, ktorý premení nečíslovaný zoznam na menu,
- vysvetlite použitie (pseudo)sudo class selektorov pre tvorbu odkazov,
- navrhните vlastné vertikálne menu a načrtnite postup jeho krokov.

ol, ul, li vysvetliť, pri ol je aj type="a" alebo i, A, l a aj start="3" pre začatie číslovania od 3/c.

Tvorba vodorovného alebo zvislého menu

Časť HTML

1. Napíšeme texty odkazov
Hory
Lesy
2. Spravíme z nich linky

```
<a href="hory.html">Hory</a>
<a href="lesy.html">Lesy</a>
```
3. Z každého odkazu spravíme položku nečíslovaného zoznamu

```
<ul>
  <li><a href="hory.html">Hory</a></li>
  <li><a href="lesy.html">Lesy</a></li>
</ul>
```

Časť CSS

- Odstránime podčiarknutie odkazov pomocou

```
ul li a {text-decoration: none;}
```
- Odstránime z nečíslovaného zoznamu odrážky pomocou

```
ul {list-style-type: none;}
```
- Nastavíme farbu písma a farbu alebo obrázok pozadia pre odkaz

```
ul li a {
  color: ...;
  background-color: ....;
}
```
- Nastavíme odkazu vlastnosť ktorá zabezpečí že bude zaberáť celú šírku akú má mať nielen šírku textu

```
ul li a {display: block;}
```
- Pri vodorovnom menu dáme odkazy vedľa seba pomocou – sú 2 techniky, pomocou
 - ```
ul li {display: inline;}
```

 alebo s float (presnejšie)
  - ```
ul li {float: left;}
```

 alebo right

Výhodou použitia zoznamu na tvorbu menu je:

1. veľmi ľahko pridáme alebo odstránime novú položku
2. v kóde toto menu vždy rýchlo nájdeme.

Na označenie aktívneho odkazu stránky na ktorej sa práve nachádzame, použijeme triedu .aktivna {...} a v nej nastavíme kurzor aby sa nemenil pomocou cursor:default;

25b Media Queries

- pojednajte o rôznych typoch médií na zobrazenie stránok,
- predvedte zadefinovanie Media queries v CSS pre všetky typy zobrazenia,
- predvedte deklaráciu jednotlivých typov,

- vymenujte a objasnite zásady, ktoré sa dodržiavajú pri tvorbe pravidiel pre typ print,
- vymenujte a objasnite zásady, ktoré sa dodržiavajú pri tvorbe pravidiel pre typ mobil,
- vymenujte a objasnite zásady, ktoré sa dodržiavajú pri tvorbe pravidiel pre typ tablet,
- na vlastnom príklade nastavte fonty, odkazy alebo nepotrebné časti stránky v CSS.

MediaQueries patria do CSS3!!! Vytvárajú iné správanie stránky pri zobrazení na mobile, tablete aj pri tlači.

Rozhoduje slovo `media="screen/projection/print"`, potom rozmery displeja resp. prehliadača a jeho orientácia.

Pre tlač

1. nezobrazujeme pozadia – farby ani obrázky
2. veľkosti písma sú v pt (tlačené body)
3. farby písma sú čierne
4. nezobrazujeme hlavičky, bočné panely, vodorovné ani zvislé menu, päťu, nepotrebné obrázky
5. obrázky zmenšíme na veľkosť v mm
6. ak chceme sami zalomiť stránku na nejakom mieste (nadpise), dáme mu triedu
`.zlom {page-break-before: always;}`

```
<link href="styles/tozai.css" rel="stylesheet"
type="text/css" media="screen">
<link href="styles/print.css" rel="stylesheet"
type="text/css" media="print">
```

Pre tablety a mobily

7. môžeme vytvoriť samostatné css súbory a naimportovať ich pomocou
`@import url(phone.css) only screen and (min-width: 361px);`
8. alebo umiestniť do hlavného css
`@media only screen and (max-width: 1360px)and (min-width:600px) {
 header h1 {
 font-size: 36px;
 }
}`
9. použiť môžeme rozmery displeja, maximum aj minimum, aj ich kombinácie pomocou slov `and` a `or`, prípadne orientáciu na výšku – šírku. Podpora v prehliadačoch je na obrázku.
10. Spájanie viac podmienok je pomocou slov `and` alebo `or`
`@media ((max-width: 1360px) and (min-width:600px)) {
 ...
}`
 - Orientáciu displeja zadávame pomocou
`orientation: portrait alebo landscape`

23 24 Audio video otázky!

23aSpracovanie zvuku

- definujte pojem zvuk a vysvetlite jeho základné parametre,
- definujte pojmy vzorkovacia frekvencia a bitová hĺbka prechodu,
- vysvetlite pojem kodek, uveďte ich príklady, použitie, výhody, nevýhody,
- pojednajte o kodekoch a formátoch zvuku z hľadiska veľkosti výsledného súboru a zariadenia, pre ktoré je určený, 5
- vymenujte základné programy na prácu so zvukom a opíšte ich prostredie,
- objasnite pojmy audiostopa, mute, solo, fade in, fade out, strih zvuku,
- pojednajte o audio efektoch,
- popíšte možnosti exportu audio programov.

Film, videoklip

Je sled viacerých obrázkov prehrávaných v čase doprevádzaný zvukom. Parametre:

- Audio
- Video
 - a. Rozlíšenie – v px
 - b. Pomer strán – 4:3, 16:9, ostatné
 - c. Frame rate – počet snímkov za sekundu. EU 25fps, USA 29,98 až 30, kino 24 a ich násobky
 - d. Riadkovanie
 - i. Prekladané / interlaced = vzniklo kvôli televízii a faktu, že obraz je kreslený po riadkoch a spočíva v tom, že obraz je tvorený striedaním nepárnych a párnych riadkov, takže "blikne" 2x častejšie ako neprekladaný. Možnosti
 1. Upper field first (najprv nepárne riadky)
 2. Lower field first (najprv párne)

- ii. Neprekladané = progressive scan = v plnom rozlíšení, použite vždy tento!
- e. Pixel aspect ratio = pomer strán pixelu. Býva:
 - a. 1,0 (square)
 - b. 1,0928 (PAL DV) = televízia 4:3
 - c. 1,4568 (PAL DV Widescreen) = televízia 16:9
- f. Kodek
- g. Formát
- h. Kontajner

KODEK = počítačový program alebo hardverové zariadenie, ktoré **kóduje** a **dekóduje** video do/z určitého formátu, za účelom zmenšenia veľkosti súboru.

!!! POZOR! Často sa zmieňa pojem formát a kodek.

Formát je štandard (definícia) napr. MPEG-4 Part2 a kodek je jeho jeden typ algoritmu napr. DivX.

Druhy kodekov

1. Bezstratové - Huffy, FFV1, LCL

2. Stratové :

- a) Divx, Xvid
- b) FFmpeg
- c) WMV=windows media video
- d) x264

Audio kodeky: FLAC, Apple Lossless, Windows Media Audio 9 series, AAC(AAC), LAME(MP3)

Formáty:

video: MPEG-1,2,4, H264 (MPEG-4 AVC), Theora, WMV

audio: AAC, Apple Lossless, AC-3, FLAC, MP3, WAV, WMA

!!!KONTAJNER!

Video býva distribuované spolu s audiom v kontajneri. Najznámejší kontajner je Audio Video Interleave AVI.

Obsahuje video vo formáte MPEG-4 a audio vo formáte MP3. Niektoré kontajnery majú možnosť viac audiostóp aj titulkov!

=napr. Matroska. Sú:

AVI, Matroska (MKV), Quicktime, MP4, WMV

Zvuk a jeho vlastnosti

ZVUK = vlnenie hmotného prostredia.

Vlastnosti zvuku:

- b) Frekvencia od 20Hz do 20 000Hz
- c) Intenzita = sila zvuku v dB = udáva, koľkokrát je zvuk silnejší, ako najmenší počuteľný zvuk Z_0 .
 - i. $A_z = 20 \log (Z/Z_0)$ v dB
- d) Pravidelnosť - ak má v sebe opakujúce sa frekvencie a ich násobky je harmonický, inak dysharmonický.
- e) Počet kanálov
 - a. 1 = mono
 - b. 2 = stereo
 - c. priestorový zvuk 5+1, 7+1. 1 je basový reproduktor, najlepšie umiestnený na zemi v rohu. 5+1 sú satelitné reproduktory v rohoch miestnosti a vpredu v strede, 7+1 sú ešte 2 na stranách uší. Tieto systémy sa označujú Dorby Digital.

Digitalizácia zvuku = ako sa zvuk dostane do PC:

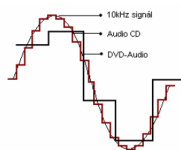
Vzorkovanie = zvuk sa naseká v čase, vyrobia sa vzorky, ktoré majú nejakú hodnotu. Musí sa urobiť s frekvenciou

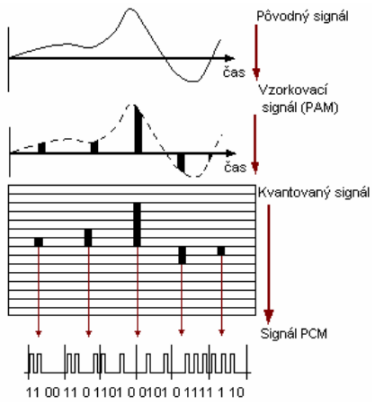
MINIMÁLNE 2x takou, aká je maximálna = $2 \times 20\,000\text{Hz} = 40\,000$ x za sekundu.

vzorkovacia frekvencia CD 44100Hz, DVD 48000Hz alebo 96000Hz

Kódovanie = každá vzorka sa premení do dvojkového kódu na 16 bitov (a viac).

Digitálny zvuk je prehrávaný prepínaním v čase = časový multiplex, napr. Stereo chvíľku ide ľavý kanál, chvíľu pravý, ale je to tak rýchlo, že počujeme naraz 2 kanály.





Kvalita digitálneho záznamu	vzorkovacia frekvencia	rozišenie, kanály
Telefónna kvalita	8000Hz	8 bit - mono
Rozhlasová a CD kvalita	44 100Hz	16 bit - stereo
DVD Audio kvalita	192 000Hz	24 bit - 5.1 surround sound

Programy na prácu s videom

Prehrávače – WMP, KM Player, VLC, QuickTime

Kodeky

Konvertory = Handbrake (do MP4), ProCoder, Video Converter, A-tube catcher ... menia video z 1 formátu do 2.

Editory = slúžia na strih, efektovanie, titulkovanie, zmeny stôp, úpravy zvuku, postprodukcii. Sú:

Amatérske – Movie Maker, VirtualDub, Quicktime

Poloprofesionálne = Sony Vegas, Adobe Premiere, Adobe AfterEffects, Lightworks, FinalCUT

Profesionálne

Programy na prácu s audiom

Prehrávače iTunes, WMP

Kodeky

Konvertory = iTunes, WMP

Editory = slúžia na strih, efektovanie

Free – Audacity,

Platené – Adobe Audition, FLStudio, Traktor

Poloha kamery určuje aj pocity divákov: poloha NAD = pocit nadradenosti, POD = pocit podriadenosti.

Pravidlo tretín: poloha snímaného objektu na priesečníkoch tretinovej mriežky

Pravidlo uhlopriečok: objekty s hranami – hrany by mali byť rovnobežné s uhlopriečkami

3D modelovanie

3D modelovanie je proces vytvárania matematického trojdimenzionálneho objektu pomocou špecializovaného softvéru

Použitie 3D modelovania a vizualizácie

- o 3D vizualizácie interiérov
- o exteriérové 3D vizualizácie
- o produktová 3D vizualizácia
- o 3D vizualizácie krajiny
- o 3D animácie

Základné pojmy

Osi X, Y a Z

Osi sú použité na znázornenie konkrétnych smerov v trojrozmernom priestore. Univerzálnym symbolom použitým pre súradnicové osi sú tri navzájom kolmé úsečky, pretínajúce sa v počiatku súradnicového systému (bod $[0,0,0]$) a označené písmenami X, Y a Z.

- X predstavuje pohyb vo vodorovnom smere, teda doprava a doľava.
- Y je pohyb smerom k pozorovateľovi a preč od neho.
- Z reprezentuje zvislý smer, teda pohyb smerom hore a dole.

Modely

Pri geometrickom modelovaní z pohľadu geometrických prvkov, z ktorých sa 3D model vytvára hovoríme o modeloch:

- analytická reprezentácia (body, krivky, plochy a matematické vzťahy medzi nimi)
- drôtový 3D model
- plošný 3D model (polygonálny model – základ je polygón, uvažuje sa plocha)
- objemový 3D model (konštruktívna geometria – pomocou 2D a 3D primitív, uvažuje sa celý objem)
- voxelový alebo dekompozičný model (rozloženie na elementárne kocky, uvažuje sa celý objem)

Typy modelov

Analytická reprezentácia

alebo **hraničný spline model** sa skladá s bodov, kriviek a spline plôch. Takéto modely nesú úplnú informáciu pre popis objektu.

Takto reprezentované objekty bývajú veľmi reálne a pamäťovo nenáročné.

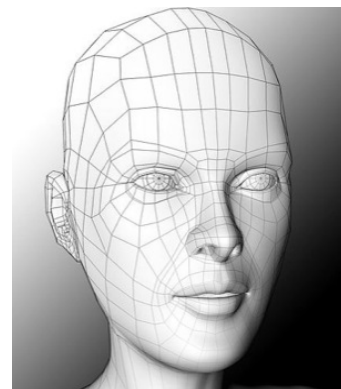
Polygonálny model

využíva ako základ **vertex**, čo je bod v trojrozmernom priestore.

Spojením dvoch vertexov dostávame **úsečku**, pridaním ďalšieho bodu získame **trojuholník**, čo je najzákladnejší polygón.

V zásade sa využívajú hlavne trojhorné polygóny, takže najčastejším útvarom polygonálneho modelu je trojuholník alebo štvoruholník.

Polygonálny model je najpoužívanejší.



Drôtový model

Priestorový drôtový model nazývaný aj Wireframe model je tvorený bodmi, ktoré sú spojené krivkami. Ide o opis bodov a kriviek spájajúce tieto dva body.

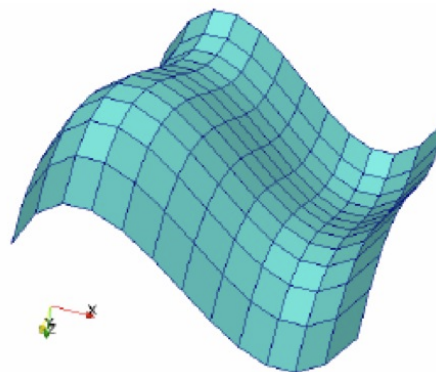
Nevýhoda: chýbajú údaje o stenách medzi krivkami a o priestore (objemu) ohraničeného stenami.



Plošný model

Modeluje telesá pomocou plôch. Potreba vzniku plošných systémov sa prejavila predovšetkým v oblastiach leteckého, lodiarskeho a automobilového priemyslu, kde sú na spracovanie zložitých plôch kladené veľké nároky

Začíname vytvárať vrcholy, hrany a nakoniec definujeme plochu. Je možný aj opačný postup cez preddefinované plochy tzv. primitíva a vybrať napr. plášť valca, kužela a pod. Oproti drôtovému modelu vieme získať plošný obsah modelu Plošný geometrický model je tvorený bodmi (vrcholmi), hranami a stenami.



Objemový model

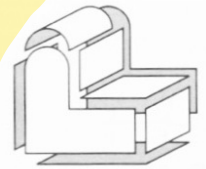
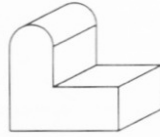
Poznáme dva hlavné koncepty objemového modelovania :

- **Reprezentácia modelu pomocou hraníc** B-rep model (Boundary representation)
- **Reprezentácia modelu pomocou geometrických telies** CSG model (Constructive Solid Geometry).

Obidve koncepcie sú založené na modelovaní súčastok pomocou základných geometrických objektov (entitami, primitívami), na počítačovej transformácii a na využívaní booleanovských operácií.

Reprezentácia modelu pomocou hraníc

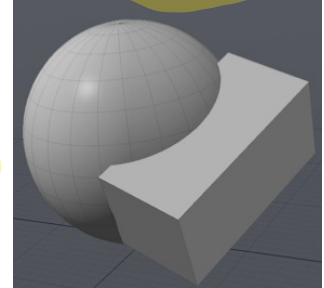
Model je vyjadriteľný svojimi hranicami. Týmto hranicami modelu sú steny - plochy, hranicami stien sú hrany - krivky a hranice pri hranách predstavujú body. Výhody B-rep modelu oproti CSG modelu spočívajú napr. v jednoduchšom spracovaní informácií o súčiastke potrebných napr. pre generovanie dráhy nástroja pri jej výrobe.



v

Reprezentácia modelu pomocou geometrických telies

Táto reprezentácia využíva množinové operácie a to sčítanie, rozdiel alebo prienik. Teleso sa vytvára pomocou stromu množinových operácií aplikovaných na primitíva.

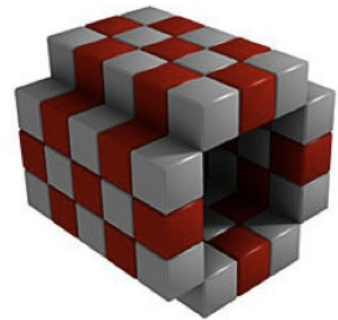


Hybridný model

Modely definované pomocou objektov (CSG) a modely definované pomocou hraníc (B-rep) majú svoje výhody a tiež obmedzenia. Žiadny z nich nie je vhodný pre všetky aplikácie. Riešenie sa ponúka v ich spojení. Potom hovoríme o tzv. hybridných modeloch.

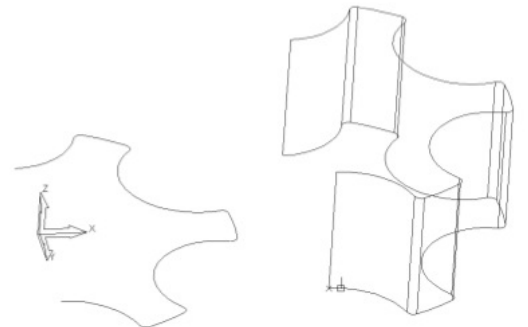
Voxelový model

alebo tiež **dekompozičný model**. Voxelové modely sa najviac podobajú bitmapovej grafike. Rovnako ako v nej je jedným bodom obrázku štvorček, tu sa modely skladajú z malých kociek. Pri použití dekompozičného modelu sa objekt rozloží na elementárne objemové jednotky – kocky. Ide o diskrétny popis modelu, pri ktorom sa ako základná jednotka popisu používa Voxel, čo je objemová jednotka reprezentujúca hodnotu bunky v sieti trojrozmerného priestoru.



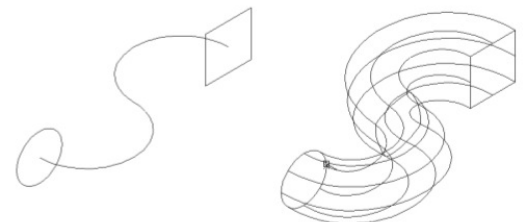
Modelovanie pomocou tvoriacich kriviek

Tvoriacou nazývame krivku (najčastejšie je to 2D krivka ale môže byť aj 3D krivka), ktorá tvorí východisko pre vytvorenie 3D modelu buď jej rotáciou (vo zvolenej osi súradnicového súradného systému) alebo jej posunutím (v jednom alebo súčasne vo viacerých smeroch súradnicového systému). Na obr. je ukážka vytvorenia 3D modelu posunutím tvoriacej krivky pozdĺž jednej osi súradnicového systému.



Model z troch kriviek

V tomto prípade sa model vytvára z troch samostatne nakreslených kriviek. Na obr. je príklad prvotného usporiadania troch kriviek, kde kružnicu nazývame štartovacou krivkou (profilom), štvorec - konečnou krivkou (profilom) a krivku tvaru "S" nazývame vodičom.



Model zo štyroch kriviek

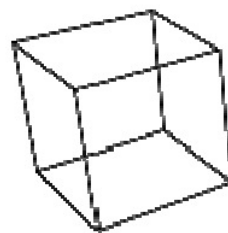
Tento prípad vytvorenia modelu je analogický predchádzajúcemu spôsobu len krivky sú 4.

Modelovanie pomocou kombinácie predošlých spôsobov

Primitíva

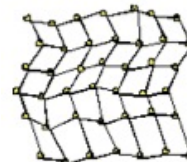
Plošné primitíva

sú plášte základných telies – valca, gule, kužeľa ...



Objemové primitíva

Sú tvorené celým objemom objektu, pričom je možné uvažovať aj o materiálovo – povrchových a materiálovo – objemových a fyzikálnych vlastnostiach objektu - valec, guľa, kužeľ

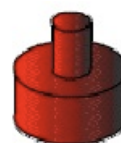


Boolean operácie

Boolean operácie sa dajú prirovnať ku matematickým množinovým operáciám zjednotenia, odpočítania a prieniku, ibaže v 3D priestore. V praxi to znamená, že môžeme dve a viac už existujúcich telies, spojiť do jedného nového telesa, alebo od jedného telesa odpočítať iné teleso, alebo telesá a tým nám opäť vznikne nové teleso, alebo zostrojíme prienik všetkých vybraných telies.

Podmienky uplatnenia

Vo virtuálnom priestore musia existovať najmenej dva priestorové objekty a tieto telesá sa musia vzájomne aspoň dotýkať (alebo cez seba prenikať).



- **Zjednotenie** - ak majú aspoň jeden spoločný bod, pričom výsledný objekt sa správa ako jediný logický celok. Teda pri otáčaní, posúvaní (ale aj napríklad pri kopírovaní a zmene mierky) sa bude podľa zadaného príkazu meniť obe zjednotené objekty.
- **Odpočítanie** - z počiatočného objektu sa odpočíta objem odpočítavaného telesa, pričom odpočítané teleso je „spotrebované“ na vytvorenie výsledného objektu, ktorý je potom zložený kombináciou povrchov oboch objektov. Odpočítavané objekty musia mať spoločný minimálne jeden bod.
- **Prienik** - je opakom odpočítavania. Výsledný 3D objekt vznikne spojením dvoch telies, pričom výsledné teleso je tvorené len bodmi a povrchmi, ktoré majú obe telesá spoločné.

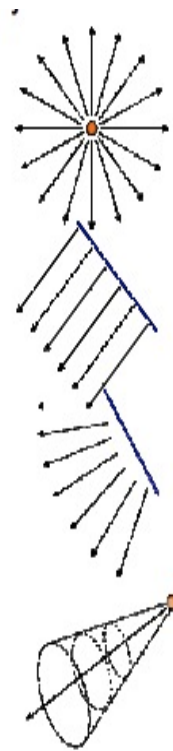


Svetlo a zdroj svetla

Svetelný zdroj

je charakteristický tým, že emituje svetelné žiarenie. Druhy:

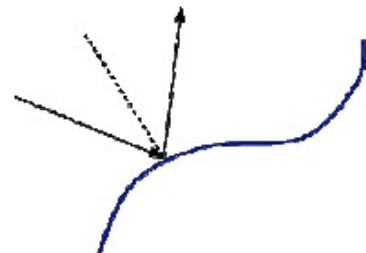
1. **Bodový zdroj** – svetlo sa z neho šíri rovnomerne a s rovnakou intenzitou do všetkých smerov.
2. **Zdroj rovnobežného svetla** - môže byť chápaný ako bodový zdroj ležiaci v nekonečne, alebo ako nekonečne veľký rovinný zdroj ležiaci v konečnej vzdialenosti. Pre tento zdroj svetla je charakteristické, že lúče z neho emitované sú rovnobežné.
3. **Plošný zdroj svetla** - tento typ svetla sa najviac podobá reálnym zdrojom, akými je napríklad žiarivka, alebo okno, ktorým prechádza denné svetlo.
4. **Reflektor alebo smerovo závislý zdroj** svetla, ktorý je určený polohou a orientáciou, t.j. smerom, ktorým žiari. Jeho svetelná intenzita je maximálna v smere, ktorým žiari, a kolmo k tomuto smeru klesá exponenciálne.
5. **Obloha** - je najkomplikovanejší zdroj svetla - je popísaná ako zdroj rovnobežného svetla v tvare pologule s nekonečným polomerom



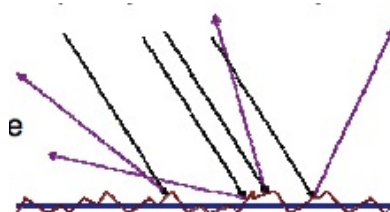
Zložky osvetlenia

Ak dopadá na objekt svetlo určitej farby, tak časť z neho je pohltená a časť odrazená. Odrazená časť svetla určuje farbu objektu, ktorú vníma naše oko. Pri určovaní osvetlenia daného bodu musíme uvažovať tri zložky svetla:

1. **Zrkadlová (specular) zložka** Je to svetlo, ktoré sa od povrchu priamo odrazí. Takéto svetlo vytvára odlesky a zrkadlové efekty. Jeho farba môže byť iná ako farba povrchu objektu. Táto zložka je smerová – smer odrazeného lúča je presne daný a vnímanie odrazu sa mení aj s meniacou sa pozíciou pozorovateľa.



2. **Difúzna zložka** je prítomná vždy, ak povrch telesa nie je dokonale hladký. Ak sú na povrchu drobné nerovnosti, dopadajúce svetlo sa odrazí aj v smeroch, ktoré sú akoby náhodné. Preto táto zložka nie je smerová – bod sa javí rovnako nezávisle od pozície pozorovateľa. Difúzna zložka určuje farbu povrchu telesa. Čím drsnejší (matnejší) je povrch, tým je táto zložka väčšia.



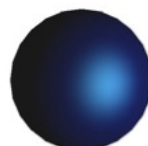
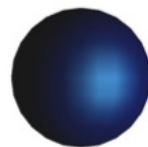
3. **Ambientná zložka** V reálnych scénach býva viac svetelných zdrojov alebo rôzne objekty, ktoré svetlo odrážajú. Dochádza k mnohonásobným odrazom a k skladaniu svetla zo všetkých zdrojov, svetlo môže byť navyše rozptýlené molekulami vzduchu a pod., čo má za následok, že je prítomné tzv. ambientné (okolité) svetlo. Toto svetlo prichádza zo všetkých strán a rovnako osvetľuje všetky časti objektov. Prítomnosť ambientného svetla má za následok, že nijaká plocha nebude celkom čierna a čím je ambientné svetlo silnejšie, tým svetlejší je výsledný obraz. Ambientné svetlo býva väčšinou biele, najmä pri zložitejších scénach.

Tieňovanie

(shading) je určenie výslednej farby povrchu zobrazovaného osvetleného objektu.

3 Typy:

1. **Konštantné tieňovanie (flat shading).**
2. **Gouraudovo tieňovanie**
3. **Phongovo tieňovanie** - toto tieňovanie je najrealistickejšie, ale výpočtovo najnáročnejšie.



Tiene

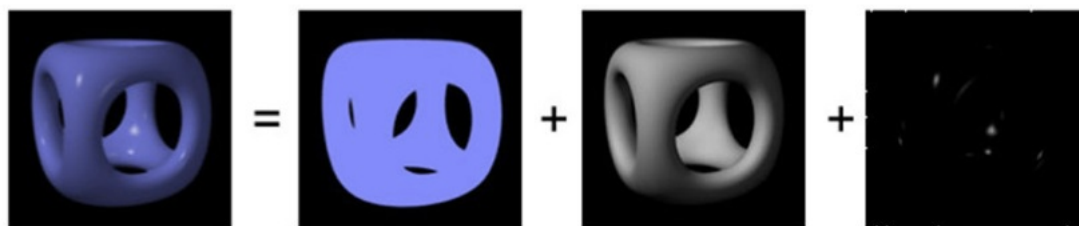
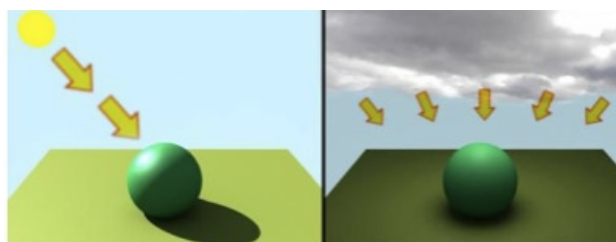
Okrem svetelných zdrojov musíme v 3D grafike uvažovať aj s tieňami

1. **tvrdé tieň**, lineárne ostré, kontrastné, výrazné tieň ...
2. **mäkké tieň**, rozľahané, jemnejšie, rozplývavé až miznúce, s jemnými okrajmi...
3. **krvácanie farieb** - objekty čiastočne odrážajú osvetlenie ktoré prijmu a tento odraz vidno na povrchu objektov okolo.
4. **spektakularita** – lesk na objekte pri jeho priamom osvetlení

Môžeme vidieť takmer rovnaké podmienky, ale nie v rovnakej intenzite, tvrdé tieň zmizli. To sa stalo preto, lebo osvetlenie generujú mraky a nie slnko

Naľavo je slnečný deň. To znamená, že je jeden hlavný bod zdroja svetla, ktoré tvorí svetlo len z jedného smeru, s tvorbou tvrdých tieňov. Spektularita (lesk na objekte) k takémuto typu osvetlenia patrí.

Napravo je zamračený deň, nikde nie je žiadny konkrétny zdroj svetla. Svetlo na objekty vrhá celá obloha, pričom sa vytvárajú mäkké tieň. Svetlo je vrhané rozptýlene, takže tieň sa nakoniec vytvorí plynule okolo celého objektu. Spektularita je rozmazaná.



3D objekt

=

farba a okolité svetlo

+

difúzna zložka

+

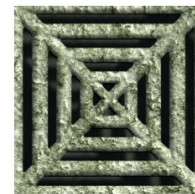
spektularita

Textúry

Povrch reálnych predmetov sa líši - môže byť zvráskavený, priehľadný či lesklý, môže obsahovať farebné prechody a zmeny. Technika, ktorá tieto vlastnosti v počítačovej grafike postihuje, sa nazýva textúra. Zatiaľ čo geometria popisuje tvar telesa, textúra je popisom vlastností jeho povrchu.

Textúra je obraz z reálneho sveta (povrchy - múr, omietka, koža, látka, kovový povrch, piesok, celé budovy s oknami, tváre ľudí a zvierat, architektonické detaily ...) ktorým "obalíme" vygenerovaný trojrozmerný objekt tak aby sme dosiahli jeho maximálnu podobnosť so skutočnosťou.

Textúra je uložená vo forme bitmapy, alebo dvojrozmerného (alebo viacrozmerného) poľa.



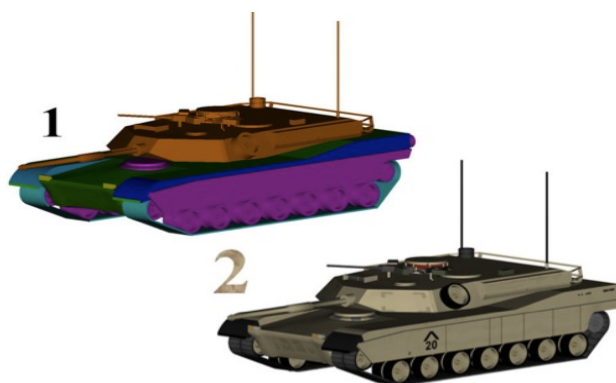
Textúry môžeme rozdeliť podľa toho akú vlastnosť popisujú:

1. Farba povrchu

2. **Odrážanie svetla** sa môže meniť s miestom povrchu. Prejavom takejto vlastnosti je odrážajúce sa okolie od povrchu objektu. Z tohto dôvodu sa táto textúra tiež označuje ako mapovanie okolitého sveta (environmental mapping).

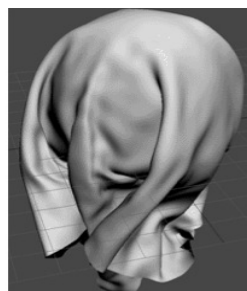
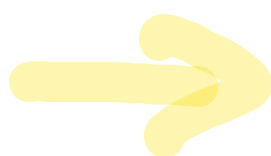
3. **Zmena normálového vektora** opticky mení tvar povrchu telesa bez toho, aby sa zmenila jeho geometria. Výsledkom je povrch, ktorý vyzerá poprehýbaný alebo inak geometricky zmenený.

4. **Priehľadnosť** telesa nemusí byť po celom jeho povrchu konštantná, ale môže sa tiež meniť s miestom.



Mapovanie

UV mapovanie je 3D modelovací proces umožňujúci prenos 2D obrazu na 3D model. Je to spôsob, ktorým sú textúry priradované 3D modelom, resp. metóda, ktorou je materiál "nabalený" na objekt tak, že sa stáva jeho neoddeliteľnou súčasťou a môže byť deformovaný alebo transformovaný spoločne s objektom.



Modifikátory

Modifikátory si je možné predstaviť ako nástroje na editáciu alebo vylepšenie objektu, alebo jeho častí, čím im špecifikujeme nové vlastnosti bez toho, aby sa zmenil pôvodný objekt (je možná cesta späť). Model je možné totiž stále opraviť a vrátiť sa na nižšiu úroveň. Modifikátory aplikované na objekt sú usporiadané v zásobníku modifikátorov.



3D animácie

3D počítačová animácia môžeme definovať ako techniku, v ktorej je ilúzia pohybu vytvorená sériou jednotlivých stavov scény, zobrazovaných na obrazovke alebo nahrávaných na nahrávacie zariadenie.

Typická 3D scéna sa skladá z: objektov, kamier a svetiel. Každý objekt má svoje charakteristické vlastnosti, ktoré sa môžu v čase meniť:

- Objekty: poloha, orientácia, veľkosť, tvar, farba, priehľadnosť
- Kamery: poloha objektívu, bod záujmu, zorný uhol (priblíženie, vzdialenie)
- Svetlá: intenzita, poloha

Techniky môžeme zoskupiť do dvoch základných skupín:

- dvojrozmerné (2D) majú sklon zameriavať sa na manipuláciu a spracovanie obrázkov
- trojrozmerné (3D) obyčajne vytvárajú virtuálne svety, v ktorých sa postavy a objekty pohybujú a vzájomne na seba vplývajú.

Výroba obrázkov pre animáciu pomocou virtuálneho 3D sveta zahŕňa tri kroky:

1. **modelovanie** - proces prípravy a vytvárania elementov scény
2. **animovanie** - roz Pohybovanie
3. **rendering** - konvertuje vlastnosti prvkov scény a ich pohyb do obrázkov

Rendering

Rendering je proces tvorby obrazu z modelu. Proces tvorby obrazu funguje spôsobom, kde kamera (kamerou je myslený pohľad výrezu) akoby vysiela lúče, ktoré prehľadávajú scénu. Pokiaľ narazí na objekt, je jeho povrch v tomto bode analyzovaný, tj. sú analyzované parametre materiálu, úroveň osvetlenia, uhol medzi povrchom a zdrojom svetla a ďalšie vlastnosti. Informácie sa vracajú ku kamere a sú uložené do podoby pixelu konečného obrázku.

Animácia

Zabezpečuje dynamický pohyb v obraze, kde sa objekt hýbe v rôznych smeroch, uhloch alebo mení svoju štruktúru alebo tvar. Za animáciu považujeme zmenu súradníc, teda polohy parametrov objektu za presne stanovený čas. Čas je určený časovou lištou, ktorej dĺžka sa dá prednastaviť alebo meniť podľa potreby a celkovej dĺžky animácie.

Na časovej lište sa nachádzajú kľúčové snímky (key frames), ktoré sú na obrázku farebne zvýraznené. Znárodňujú novú polohu, orientáciu alebo zmenu parametra objektu alebo skupiny objektov za špecifikovanú dobu.

Každý objekt má vlastnú časovú lištu, takže sa dá ľahko editovať zmena každého z nich osobitne.

Metódy animácie:

1. **Kľúčovanie** poskytuje výbornú kontrolu nad pohybom a kreslia sa iba najdôležitejšie – kľúčové snímky (keyframes). Zvyšné medzismímky (in-betweens) kreslí počítač.
2. **Procedurálne metódy** vytvárajú pohyb čisto automatizovaným spôsobom
3. **Snímanie pohybu** je metóda, ktorá používa špeciálne senzory a snímače na záznam pohybu človeka. Nahrané dáta sa potom použijú na vytvorenie pohybu v samotnej animácii. Snímanie pohybu je veľmi populárna metóda. Umožňuje relatívne ľahko nahráť veľké množstvo pohybov, aj keď tiež sa stretáva s určitými problémami. Napríklad keď sa senzory neudržia na pôvodnom mieste, keď rozmery objektu nie sú rovnaké ako rozmery snímanej postavy,