

Animace

Matěj Cajthaml

Distanční výuka - 55. - 56. hod
Smíchovská SPŠ | WBA

Abstrakt

Tento dokument vznikl jako podpůrný materiál při **n-té** vlně pandemie CoVID-19 a spojených nemocí. Je určen pouze pro studijní účely. Text byl vytvořen ve školním roce 2020 / 2021 na škole Smíchovská SPŠ. Autorem textu je Matěj Cajthaml jakékoliv připomínky a dotazy buď odesílejte na e-mail matej.cajthaml@ssps.cz nebo na MS Teams.

Obsah

1	Úvod	2
2	Animace & historie	2
3	Typy animací	2
4	Přechodové animace (CSS)	2
5	Pravé animace (CSS)	3
6	Pokročilé animace (JS)	3
7	Úkol	3

1 Úvod

Před námi je poslední oficiální téma z předmětu Webové aplikace - a to animace. V tomto dokumentu si vysvětlíme co je to animace, jak fungují na webových stránkách a jak je používat.

2 Animace & historie

Animace v pravém významu znamená “vytváření zdánlivě pohybujícího objektu”. Když se začaly vytvářet animace tak se používaly tzv. frames - rámec. Rámec je stav animace (jeden snímek) v jednom aktuálním momentu. V obyčejných animacích (filmech, ...) se vytváří okolo 24 snímku pro jednu sekundu záznamu. Tento počet dovolí “ošálit” oko aby si myslelo, že se jedná o plynulý pohyb.

Takto to funguje i na webech (popř. i v jiném software). Při animaci definujeme určité rámce. Jediný rozdíl je však v tom, že nemusíme definovat určitý počet rámců pro danou animaci. Mezi jednotlivými rámci se používá tzv. interpolace, což si můžete představit jako vyplnění informací hodnot mezi dvěma body.

3 Typy animací

Animace lze chápat různými směry. V základech webových aplikací rozdělujeme animace do tří kategorií. První z nich jsou přechodové animace. To jsou animace, které se volají při tom, když se prvek na stránce změní. Základní změna na stránce se provádí například tím, když najedeme na prvek a máme definovanou pseudotřídu hover. Tyto animace jsou nejjednodušší, protože pouze definujeme jak se určité prvky budou proměňovat z jednoho na druhý.

Další je “pravá” animace. Je pravá, protože opravdu definujeme rámce. Tyto animace můžeme nadále upravovat například rychlostí, směrem, typem přechodů. Animace můžeme dokonce vypínat a zapínat při různých stavech. K jednotlivým rámcům definujeme css vlastnosti a ty se podle nastavení prezentují.

Poslední jsou pokročilé animace. Tyto animace se velmi podobají pravým animacím. Rozdíly jsou v tom, že tyto animace jsou velmi pokročilé a používají se pro komplikované animace. Tyto animace jsou přesné a lze je velmi přesně ovládat pomocí skriptovacího jazyk JavaScript.

4 Přechodové animace (CSS)

Tyto animace se nejlépe prezentují na ukázkách. Představte si že máte na stránce čtverec. Při najetí na tento čtverec se změní barva a zaokrouhlení rohů tohoto boxu.

UKÁZKA: bloková animace Tento stav není hezký. Je to způsobeno tím, že není definovaná přechodová animace mezi stavem čtverce bez najetí a po najetí. Lidské oko očekává nějaký přechod místo skokového zobrazení jiné věci.

Proto si nyní definujeme první přechodovou animaci, a to ze stavu bez najetí do stavu najetí. Přechodová animace se definujeme vlastností transition. Vlastnost bere tři parametry, první je jaké vlastnosti se mají pro animaci hlídat. Můžeme taktéž hlídat všechny pomocí hodnoty “all”. Druhá je čas, která tato animace zabere. Třetí je typ přechodové funkce - což je jak se mezi stavy bude interpolací procházet. Základní přechodové funkce jsou např.:

- ease - pomalý přechod na začátku a na konci, uprostřed rychle
- linear - rovný přechod, stejně rychlý v každém momentu
- veškeré funkce naleznete v dokumentaci ZDE

V CSS existují vlastnosti, které nelze interpolovat - jedná se většinou o vlastnosti, které mají přesné stanové možnosti - display, visibility a další. Nejlépe se animují vlastnosti kde je mnoho hodnot - padding, color, background, top, transform a další.

Pro první stav (tj. z not hover na :hover) v :hover definujeme vlastnost “transition: background-color 0.5s ease;”, která říká: vytvoř přechodovou animaci pro vlastnosti background-color, která zabere 0.5s (500ms) s funkcí ease. V tomto stavu vytvoříme další přechodovou animaci a to pro vlastnost border.

UKÁZKA: animace prvního stavu

Když se na ukázkou podíváte, vidíte, že opravdu funguje pouze jedním směrem. Nyní si vytvoříme animaci druhým směrem (tj. z :hover na not hover). Tu budeme definovat v základním stavu.

UKÁZKA: animace obou stavů

5 Pravé animace (CSS)

V CSS se pravé animace definují pomocí tzv. “pravidla” keyframes. V tomto bloku definujeme jednotlivé rámce. Rámce pojmenováváme pomocí “from” (ze stavu), “to” (do stavu) a nebo pomocí procent. Animace vždy musíme pojmenovat.

Znovu si animace ukážeme na ukázce, tentokrát, stejně jako v přechodových animacích, použijeme náš čtverec uprostřed stránky. V CSS si pomocí pravidla keyframes definujeme jednoduchou animaci se třemi stavy: from, 50% a to. V této animaci budeme daný čtverec zvětšovat, zmenšovat, měnit mu barvu a zaokrouhlení rohů.

Animace lze používat na několik prvků najednou (tj. nemusí být na jednom prvku nebo na prvku s určitými vlastnostmi).

Když animaci nastavíme stavu from (0%) a to (100%) bude animace plynulá a nebude poznat že se někde resetovala. Když budou tyto stavy rozdílné, stav se na konci resetuje na začátek.

Animaci (pomocí rámců) máme hotovou. Nyní musíme animaci zapnout na daném prvku. Na to slouží vlastnost animation a další přidružené vlastnosti. Pomocí vlastnosti animation-name nastavíme jméno animace. Vlastnost animation-timing-function nastavíme přechodovou interpolační funkci (stejně jako v předchodových funkcích). Vlastností animation-duration nastavíme čas, jak dlouho bude animace probíhat. Pomocí vlastnosti animation-play-state můžeme nastavit zda animace probíhá (running) nebo je pozastavena (paused). Vlastnost animation-iteration-count určíme kolikrát animace proběhne nebo zda je nekonečná (hraje stále). Animace může mít nekonečně rámců (např. můžeme definovat rámce na procentu 44.1293%).

UKÁZKA: hotová pravá animace

6 Pokročilé animace (JS)

Animace v JS fungují velmi podobně jako pokročilé animace v CSS. Taktéž definují určité rámce, časovou funkci, přechodovou funkci, počet opakování a další.

Velká krása těchto animací je ta, že můžeme pomocí kódu např. zaznamenat že daná animace skončila. Při použití funkce .animate(...) nám funkce vrátí objekt typu Animation. Na tomto objektu můžeme zaznamenat například událost finish a při ní provést akci. Animaci můžeme pomocí metody .cancel() ukončit. Všechny funkce možnosti naleznete v dokumentaci ZDE.

Znovu si animaci ukážeme na příkladu se čtvercem. Tentokrát budeme animaci chtít přehrát několikrát a měnit mu pouze barvu pozadí a vlastnost transform. Přidáme k tomu i přechodovou animaci při tom, když animace pomocí JS skončí.

UKÁZKA: hotová pokročilá animace

7 Úkol

Vypracujte samostatně následující úkol. Bodový základ je 10 bodů, lze získat 15 bodů. Využívejte možnosti internetu, ne vše se ukázalo a je v dokumentu. Odevzdávejte jeden soubor ve formátu (prijmeni)_1.html. Je důležité si všimnout všech změn. Je povinné využít všechny tři typy animací (včetně události finish). Nedodržení podmínek (jména souboru, nevyužití všech tří animací) je hodnoceno 0 body.

Je přísně zakázáno podvádět, ukazovat (či odesílat) svoje řešení či opisovat kódy z internetu. Pokud bude použit internet, vždy uveďte odkaz na zdroj. Při zjištění podvádění bude VŠEM zúčastněným odečteno 40 bodů.

Vytvořte celostránkovou webovou stránku podle videa, které je k dispozici ZDE.