



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v ČB

Pedagogická fakulta

Katedra informatiky

**Akcelerované mobilní webové stránky pomocí
AMP HTML a AMP JS
Accelerated Mobile Websites Using AMP
HTML and AMP JS**

Bakalářská práce

Vypracoval: Martin Mayer

Vedoucí práce: PaedDr. Petr Pexa, Ph.D.

České Budějovice 2021

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Pedagogická fakulta

Akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Martin MAYER**
Osobní číslo: **P17390**
Studijní program: **B7507 Specializace v pedagogice**
Studijní obor: **Informační technologie a e-learning**
Téma práce: **Akcelerované mobilní webové stránky pomocí AMP HTML a AMP JS**
Zadávající katedra: **Katedra informatiky**

Zásady pro vypracování

Cílem bakalářské práce bude představit novou technologii AMP (Accelerated Mobile Pages) pro tvorbu mobilních resp. responzivních aplikací. AMP je technologie vyvinutá firmou Google a je určena pro rychlé načítání statického obsahu na mobilních zařízeních. Výhodou této aktuální technologie je velice jednoduchá HTML struktura s minimem tagů a rychlé načítání dat díky Google AMP Cache, kdy se načte HTML, JavaScript a obrázky do vlastní cache mobilního zařízení. Technologie má zatím i své nevýhody – relativně malou maximální velikost CSS a není možné použít žádný vlastní JavaScript, je ale možné v průběhu zpracování bakalářské práce očekávat výrazné pokroky ve vývoji této užitečné technologie. Teoretická část bude zaměřena na představení samotné AMP technologie, jeho syntaxe, funkci a budou popsány přínosy použití technologie AMP při tvorbě mobilních webových stránek. V praktické části bude vytvořena vlastní webová prezentace pomocí AMP, která bude použita k názorné demonstraci možností této technologie, jejich výhod a nevýhod v porovnání s klasickou tvorbou webových responzivních projektů.

Rozsah pracovní zprávy: **40**
Rozsah grafických prací: **CD ROM**
Forma zpracování bakalářské práce: **tisková**

Seznam doporučené literatury:

1. AMP HTML. Je čas.cz – moderní tvorba webových stránek [online]. Dostupné z: <http://jeas.cz/amp-html>
2. Vzhůru dolů – webová kódřina ze všech stran [online]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/amp-webari-preload#google-amp>
3. Co je AMP a jak může pomoci vašemu webu? – Sunitka.cz. PPC, sociální média, SEO a marketing na internetu od společnosti Sun Marketing – Sunitka.cz [online]. Copyright ? [cit. 04.04.2019]. Dostupné z: <http://www.sunitka.cz/cj/1077-co-je-amp-a-jak-muze-pomoci-vasemu-webu>
4. What is AMP – AMP. Accelerated Mobile Pages Project – AMP [online]. Dostupné z: <https://www.ampproject.org/learn/overview/>
5. Learn AMP by Example. Learn AMP by Example [online]. Dostupné z: <https://ampbyexample.com/>
6. What is AMP – Google Ads Help. Google Help [online]. Copyright ?2019 Google [cit. 04.04.2019]. Dostupné z: <https://support.google.com/google-ads/answer/7336596?hl=en>

Vedoucí bakalářské práce: **PaedDr. Petr Pexa, Ph.D.**
Katedra informatiky

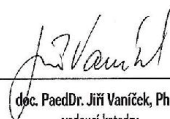
Datum zadání bakalářské práce: 15. dubna 2019
Termín odevzdání bakalářské práce: 30. dubna 2020

V Českých Budějovicích dne 15. dubna 2019



doc. RNDr. Helena Koldová, Ph.D.
děkanka

L.S.



doc. PaedDr. Jiří Vaniček, Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 2. července 2021.

Martin Mayer

Anotace

Cílem bakalářské práce je zpracovat a popsat technologii AMP (Accelerated Mobile Pages). AMP je technologie, s níž přišla společnost Google, a je určena převážně pro rychlé načítání statického obsahu na mobilních zařízeních. Výhodou této technologie je velice jednoduchá HTML struktura s minimem tagů a rychlé načítání díky Google AMP Cache, kdy se načte HTML, JavaScript a obrázky do vlastní cache. Nevýhodou spatřuji v relativně malé maximální velikost CSS a také v tom, že není možné použít téměř žádný vlastní JavaScript.

V teoretické části se zaměřím na představení samotné AMP technologie, jeho syntaxe, funkce, popíši jednotlivé komponenty AMP. Porovnáám ji s klasickým HTML a představím výhody a nevýhody této technologie při tvorbě webových stránek.

V praktické části vytvořím webovou stránku pomocí AMP technologie, která bude použita k názorné demonstraci možností této technologie. Dále vytvořím průzkum, kde budu porovnávat weby vytvořené pomocí AMP a weby vytvořené pomocí klasického HTML u českých a zahraničních webů.

Klíčová slova

AMP, Google AMP Cache, HTML, CSS, webové stránky

Abstract

The objective of the bachelor thesis will be develop and describe AMP (Accelerated Mobile Pages) technology. AMP is a technology invented by Google and it is designed to quickly load static content mostly on mobile devices. The advantage of this technology is a very simple HTML structure with minimal use of tags and fast loading thanks to Google AMP Cache, which loads HTML, JavaScript and images into its own cache. As a disadvantage, i would see a relatively small maximum CSS size and that you can not use almost any custom JavaScript.

In the theoretical part i will focus on the introduction of the AMP technology itself, its syntax, functions, description of the individual AMP components, comparison with classical HTML and description of the advantages and disadvantages of this technology in creating websites.

In the practical part i will create a website using AMP technology, which will be used to demonstrate the possibilities of this technology. I will also create a survey where i will compare websites created by AMP and websites created by classic HTML for Czech and foreign websites.

Keywords

AMP, Google AMP Cache, HTML, CSS, websites

Poděkování

Děkuji vedoucímu mé bakalářské práce, panu PaedDr. Petru Pexovi, Ph.D., za jeho cenné rady a poznatky, které mi pomohly při zpracování této práce. Dále děkuji své rodině za podporu a vstřícnost, která se mi dostávala během celého průběhu studia.

Obsah

1	Úvod	11
1.1	Východiska práce	11
1.2	Cíle práce	11
1.3	Metody práce	12
2	AMP	13
2.1	Historie AMP	13
2.2	Části AMP	14
3	AMP HTML	15
3.1	Kostra AMP HTML	15
3.1.1	Změny v základní kostře oproti běžnému HTML	16
3.2	Základní povinné prvky v AMP HTML	16
3.3	Amp-boilerplate	17
3.4	Zakázané a nahrazené HTML tagy	18
3.5	Lazy loading – odložené načtení	19
3.6	HTML atributy	19
4	CSS v AMP	26
4.1	Zákazy v CSS	26
4.1.1	Zavináčová pravidla	26
4.2	Fonty v AMP	27
5	AMP JS	30
5.1	Amp-script	31
5.2	Akce a události	32
6	AMP Cache	34
6.1	Optimalizace a úpravy	34
6.2	Aktualizace AMP Cache	36

6.3	AMP Viewer	36
6.3.1	Předvykreslení	37
7	Layout v AMP	39
8	Validace AMP	43
9	AMP komponenty	45
9.1	Komponenty pro multimediální obsah	45
9.1.1	Komponenty pro multimediální obsah pro vkládání obsahu třetích stran	50
9.2	Komponenty pro uživatelské rozhraní	57
9.2.1	Komponenty pro přidání vizuálních efektů	66
9.2.2	Observery	69
9.3	Dynamické komponenty	70
9.4	Prezentační komponenty	76
9.5	Komponenty pro reklamu a analytiku	77
9.6	Komponenty pro sociální sítě	77
10	Výhody a nevýhody AMP	81
10.1	Výhody AMP	81
10.2	Nevýhody AMP	81
11	Praktická část	83
11.1	Tvorba webové stránky	83
11.1.1	Ukázka webové stránky	84
11.2	Měření rychlosti webů	86
12	Závěr	91
	Seznam použité literatury a zdrojů	92
A	Příloha	100

1 Úvod

1.1 Východiska práce

Počítačové technologie a společně s nimi webové stránky se vyvíjí velice rychlým tempem. Uživatel je v současné době náročný, tráví většinu svého času na mobilních zařízeních a nechce čekat delší dobu na načtení webové stránky kvůli pomalému připojení. Moderní uživatel vyžaduje rychlý, graficky hezký a přehledně zpracovaný web, který si bude moci prohlížet bez obtíží také na mobilních zařízeních, bez ohledu na rychlost připojení. Přesně toto byl cíl technologie AMP.

Technologií na vytváření webových stránek je v dnešní době mnoho, technologie AMP by však mohla předčít všechny. Především díky své rychlosti, responzivitě, jednoduchosti a také tomu, že i sám Google tyto stránky preferuje před ostatními. v současné době se na internetu začíná objevovat více webových stránek vytvořených touto technologií a aktuální trend nasvědčuje tomu, že budou stále přibývat.

1.2 Cíle práce

Cílem bakalářské práce je popsat jednu z alternativ při tvorbě webu – technologii AMP. Rozebrat její možnosti, využití, funkcionality, případně nastínit potenciaální chyby, které mohou při tvorbě webu nastat.

V teoretické části se zaměřím na popsání technologie AMP, představení její syntaxe, funkce, popsání jednotlivých komponent, porovnání s klasickým HTML a popsání výhod a nevýhod této technologie při tvorbě webových stránek.

V praktické části dojde k vytvoření webové stránky pomocí AMP technologie, která bude použita k názorné demonstraci možností této technologie. Stránku navrhnu tak, aby byl zřetelně vidět rozdíl v responzivitě mezi stránkou vytvořenou pomocí AMP a stránkou vytvořenou HTML. Dále dojde

k vytvoření průzkumu, kde se bude porovnávat rychlost webových stránek vytvořených AMP a stránek vytvořených pomocí klasického HTML u známých českých a zahraničních webů.

1.3 Metody práce

V úvodu práce popíši, co technologie AMP znamená. Bude rozebrána její syntaxe, výhody a nevýhody této technologie, porovnání s klasickým HTML, představím nové, pozměněné nebo zakázané tagy, které jsou součástí AMP technologie. Zaměřím se na manipulaci s Google AMP Cache, použití CSS, případně JavaScriptu v AMP. Celá technologie bude demonstrována na vlastní webové prezentaci s ukázkami použití kódu, porovnáním s klasickým HTML a rozdílu v jejich načtení.

2 AMP

AMP neboli Accelerated Mobile Pages (Akcelerované mobilní webové stránky) je front-endový, open-source framework, který je určený pro co nejrychlejší načtení webových stránek na mobilních zařízeních nebo na počítačích. Základem této technologie je rychlost, které je docíleno tím, že se stránka načte z AMP Cache a případně ještě zobrazí přes AMP Viewer na mobilních zařízeních. V rámci zachování co nejvyšší rychlosti jsou v AMP zavedena nová pravidla a omezení oproti klasickému HTML: například CSS nesmí být v externím souboru a může mít pouze omezenou velikost, použití předvyrobené javascriptové knihovny a velmi omezeného použití vlastního JavaScriptu a v neposlední řadě přidání vlastních AMP komponent.

2.1 Historie AMP

Jedná se o projekt, se kterým přišel Google v říjnu 2015 a byl oficiálně spuštěn v lednu 2016 jako konkurence k již spuštěným projektům Facebook Instant Articles a Apple News. „*V roce 2015 na tom mobilní web nebyl dobře. Analýza více než 10 000 mobilních webových domén ukázala, že průměrná doba načítání mobilních webů byla u 3G připojení 19 sekund. Ve výsledku vznikly nové formáty mobilních webů (...). Vydavatelé, kteří již měli omezené zdroje, nám řekli, že nebudou schopni držet krok s tím, jak každá společnost vymýšlí svůj vlastní nový formát. Po diskusích s evropskými vydavateli jsme vytvořili nový formát, založený na webu, který bude fungovat pro širokou škálu platforem. (...) Byl by open-sourcový a rozšiřitelný, aby mohl nové funkce zavádět kdokoli, platforma i vydavatel. A tak byl vytvořen projekt Accelerated Mobile Pages.*“ [1]

Od konce roku 2018 přešlo AMP do modelu řízení pomocí výborů a pracovních skupin. Ve vedení už není pouze Google. V poradním výboru jsou lidé například z Akamai, Cloudflare, eBay nebo třeba The New York Times a ve výboru pro technické vedení mají zastoupení lidé z Microsoftu, Twitteru, Pantheonu a Pinterestu. O rok později se AMP připojilo k OpenJS Foundation. [1][3]

Na začátcích svého vzniku se AMP, jak název technologie napovídá, zaměřovala především na rychlejší načítání statického obsahu na mobilních zařízeních. To už v dnešní době není úplně přesné. Dnes už lze touto technologií vytvářet plnohodnotné responzivní webové stránky pro všechna zařízení a rozhodně tato technologie není pouze pro statický obsah. Na AMP se v současné době stavějí například e-shopy, včetně nákupního procesu a filtrování produktů. [2]

2.2 Části AMP

Technologii AMP můžeme rozdělit do tří hlavních částí, které se dohromady starají o to, aby webová stránka běžela co nejrychleji.

- AMP HTML – Standard vycházející z HTML 5, který specifikuje, jaké tagy se mohou nebo naopak nemohou používat. [4]
- AMP JS – Knihovna JavaScriptu, která slouží k rychlému načítání AMP HTML. [5]
- AMP Cache – Jedná se o mezipaměť, do které je načten veškerý obsah webu, jako jsou soubory HTML a obrázky. [4]

3 AMP HTML

V podstatě se jedná o obyčejné HTML, které vychází z HTML 5. Oproti běžnému HTML však AMP HTML zavádí nové tagy a některé nové prvky, které musí soubor splňovat, aby prohlížeč věděl, že se jedná o webovou stránku vytvořenou pomocí AMP. Mezi nejviditelnější změnu oproti běžnému HTML patří zákaz tagu `` a jeho nahrazení `<amp-img>`, který je navíc, na rozdíl od ``, párový. Další velký rozdíl je v tom, že AMP stránka musí být validní. Ne podle standardů HTML, ale podle standardů AMP HTML, který je přísnější. Pokud AMP stránka obsahuje chyby, neznamená to, že by se stránka uživateli nezobrazila, znamená to, že přijdete o jednu z výhod AMP, a to je umístění stránky v AMP Cache. [2]

3.1 Kostra AMP HTML

Základní kostra AMP dokumentu se příliš neliší od běžného HTML, ale některé změny jsou na první pohled patrné.

```
1 <!doctype html>
2 <html amp>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width">
6     <script async src="https://cdn.ampproject.org/v0.js">
7   </script>
8   <style amp-custom>
9   </style>
10  <style amp-boilerplate >...</style>
11  <link rel="canonical" href="$SOME_URL">
12  <title >My AMP Page</title >
13 </head>
14 <body>
```

```
15 <h1>Hello World</h1>
16 </body>
17 </html>
```

Příklad 1: Základní kostra AMP HTML

3.1.1 Změny v základní kostře oproti běžnému HTML

- `<html amp>` - Označení že jde o AMP stránku. Místo `<html amp>` se může použít i `<html ⚡>`.
- `<script async src="https://cdn.ampproject.org/v0.js">` - Stahuje povinnou hlavní javascriptovou knihovnu.
- `<style amp-custom>` - Prostor, kam se vloží vlastní CSS styly pro danou stránku.
- `<style amp-boilerplate>` - Obsahuje povinné styly, které pomáhají prohlížeči s vykreslováním stránky.

3.2 Základní povinné prvky v AMP HTML

Aby AMP stránka byla validní a fungovala správně, musí obsahovat následující prvky:

- Stránka musí začít tagem `<!doctype html>`.
- `<html amp>` nebo `<html ⚡>`.
- `<head>` a `<body>`.
- Jako první meta tag pod tagem `<head>` musí být `<meta charset="utf-8">`.
- `<meta name="viewport" content="width=device-width">`. AMP také doporučuje přidat k atributu `content` `minimum-scale=1` a `initial-scale=1`, ale povinné to není.

- `<script async src="https://cdn.ampproject.org/v0.js">`.
- `<style amp-boilerplate>`.
- `<link rel="canonical" href="$SOME_URL">`, což je odkaz na běžnou HTML stránku, případně může odkazovat AMP stránka sama na sebe, pokud běžná HTML stránka neexistuje.

3.3 Amp-boilerplate

Amp-boilerplate patří mezi nové a povinné prvky AMP. Kód tohoto tagu je neměnný a nesmí se do něj zasahovat, aby vše pracovalo, jak má. Celý kód vypadá následovně:

```
1 <style amp-boilerplate>body{-webkit-animation:-amp-start 8s
  steps(1,end) 0s 1 normal both;-moz-animation:-amp-start 8
  s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start
  8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s
  steps(1,end) 0s 1 normal both}@-webkit-keyframes -amp-
  start{from{visibility:hidden}to{visibility:visible}}@-moz-
  -keyframes -amp-start{from{visibility:hidden}to{
  visibility:visible}}@-ms-keyframes -amp-start{from{
  visibility:hidden}to{visibility:visible}}@-o-keyframes -
  amp-start{from{visibility:hidden}to{visibility:visible}}
  @keyframes -amp-start{from{visibility:hidden}to{
  visibility:visible}}</style><noscript><style amp-
  boilerplate>body{-webkit-animation:none;-moz-animation:
  none;-ms-animation:none;animation:none}</style></noscript
  >
```

Příklad 2: AMP-boilerplate

V kódu se nachází CSS animace, která provádí to, že pokud AMP webová stránka čeká příliš dlouho na inicializaci javascriptové knihovny – například proto, že má z nějakého důvodu zpoždění nebo JavaScript selže úplně – amp-boilerplate zařídí, aby se obsah AMP webové stránky zobrazil po osmi sekundách. Pokud nelze v prohlížeči JavaScript pouštět, spustí se pouze část kódu `<noscript>` a zobrazení pomocí animace neproběhne. [2]

3.4 Zakázané a nahrazené HTML tagy

Jak bylo zmíněno v úvodu této práce, AMP řadu HTML tagů nahrazuje nebo úplně zakazuje. Jediným důvodem je rychlost.

Zakázaný tag	Funkce v HTML
<code><base></code>	Specifikuje základní nebo cílenou URL pro všechny odkazy s relativní URL adresou v dokumentu.
<code><frame></code>	Vymezuje prostor pro načtení jiné stránky.
<code><frameset></code>	Rozdělí stránku do více oblastí, do kterých se načítají jiné stránky.
<code><object></object></code>	Dovoluje do stránky vložit libovolný objekt z externích zdrojů (webovou stránku, obrázek, video atd.).
<code><embed></code>	Funkce podobná jako u tagu <code><object></code> .
<code><applet></applet></code>	Používal se pro přidání externího programu psaný v Javě do HTML stránky.

Tabulka 1: Zakázané tagy v AMP HTML [6][7]

Jak je v tabulce znázorněno, jedná se o tagy, které slouží převážně k přidání objektu do HTML stránky, které by v AMP pouze zpomalovaly načtení – proto je také AMP zakázala. Při vykreslování stránky z pohledu výkonu je to zcela pochopitelné. Jedná se také o tagy, které už jsou starší a například tagy `<frameset>` nebo `<applet>` už nejsou podporovány ani v HTML 5.

HTML tag	AMP HTML tag
	<amp-img></amp-img>
<video></video>	<amp-video></amp-video>
<audio></audio>	<amp-audio></amp-audio>
<iframe></iframe>	<amp-iframe></amp-iframe>

Tabulka 2: Nahrazené tagy v AMP HTML

AMP opět nahrazuje tyto značky kvůli výkonu. Na svých speciálních tazích může zajistit to, co v současné době dokáže pouze tag a <iframe>, a to je technika zvaná lazy loading (odložené načtení).

3.5 Lazy loading – odložené načtení

Lazy loading je technika, která rezervuje místo na stránce už při startu jejího vykreslování. Také zamezuje blokování zobrazování stránky externím obsahem. Lazy loading v první řadě zařizuje odložení stažení externích prvků, jako právě třeba obrázku nebo videa, až do momentu, kdy se k nim uživatel přiblíží. Takže pokud se například obrázek nachází v dolní části stránky a není jisté, že se k němu uživatel vůbec přiblíží, obrázek se ze serveru nenačte. Tím se zamezí zbytečnému přenosu dat a zrychlí se načtení webové stránky. [8]

3.6 HTML atributy

HTML tagy mohou obsahovat atributy, které poskytují prohlížeči informace o tom, jak se má daný element zobrazovat nebo chovat. Skládají se z názvu a hodnoty vzájemně oddělené rovnítkem. Hodnota atributu se pak píše do uvozovek. AMP ovšem některé atributy zakazuje. [9]

Zakázané atributy

- Všechny atributy s předponou *on*, jako například onclick nebo onmouseover jsou nahrazené jediným `on=""`. [10]
- XML atributy, jako jsou `xmlns`, `xml:lang`, `xml:base` a `xml:space`. [10]

AMP ovšem přichází i s atributy novými, které slouží k definování layoutu elementu. Jejich cílem je správné zobrazení a rezervace místa v layoutu stránky jednotlivých elementů. Některé se neliší od běžného HTML, ale u elementů, které nějakým způsobem ovlivňují rozvržení stránky, jsou povinné. [2]

Nové atributy

- `layout` – Tento atribut specifikuje, jak se má chovat komponenta v layoutu stránky. Dává vývojáři kontrolu nad tím, jak se budou jednotlivé elementy vykreslovat na stránce a jde to jednodušeji než pomocí CSS. Atribut `layout` se jednoduše přidá k elementu s jednou z možných hodnot. Více v kapitole `Layout v AMP`. [2][11]

```

1 <amp-img src="image.jpg"
2   width="850"
3   height="300"
4   layout="responsive"
5   alt="Popisek">
6 </amp-img>

```

Příklad 3: Příklad atributu `layout`

- `width` a `height` – Tyto atributy jednoduše určují výšku a šířku objektu. Musí obsahovat celočíselnou hodnotu udávající počet pixelů. U některých typů layoutů jsou u AMP komponent oba tyto atributy povinné a u některých pouze jeden z nich. Více se tomuto věnuje v kapitole `Layout v AMP`. [2][11]

- `heights` – Všechny elementy v AMP, které podporují responzivní layout, podporují také atribut `heights`. Tento atribut se vztahuje pouze na výšku prvku a mohou se v něm použít i procenta, např. 70%. Pokud se použijí procenta, indikují procentuální velikost šířky. Pokud je atribut `heights` zadán společně s atributy `width` a `height`, výchozí hodnota layoutu je `"responsive"`. [2][11][12]

```
1 <amp-img src="image.jpg"
2   width="850"
3   height="300"
4   heights="(min-width:500px) 200px, 70%"
5   alt="Popisek">
6 </amp-img>
```

Příklad 4: Příklad atributu `heights`

V tomto případě bude výchozí velikost obrázku 70 % šířky. Pokud však bude viewport větší než 500px, bude výška obrázku nastavena na 200px.

- `sizes` – Sám o sobě je jistým opakem atributu `heights`, pracuje s šířkou objektu a nesmí se v něm použít procenta. Největší přínos tohoto atributu však přichází ve spojení s atributem `srcset`. Pokud je atribut `sizes` zadán společně s atributy `width` a `height`, výchozí hodnota layoutu je `"responsive"`. [2][11][12]

```
1 <amp-img src="image-siroky.jpg"
2   width="850"
3   height="300"
4   srcset="image-siroky.jpg 640w,
5           image-uzky.jpg 320w"
6   sizes="(min-width:650px) 640px, 100vw"
7   alt="Popisek">
8 </amp-img>
```

Příklad 5: Příklad atributu sizes

V tomto případě atribut `srcset` sděluje prohlížeči, že má k dispozici dva obrázky a má je použít podle velikosti viewportu, ve kterém se má obrázek zobrazit. Atribut `size` pak určuje velikost viewportu podle velikosti obrazovky.

- `media` – Tento atribut podporuje responzivitu v AMP. Funguje stejně jako zavináčové pravidlo `@media`, platí ovšem pouze pro daný element. Nastaví podmínku, při níž má být prvek vykreslen. [2][11]

```
1 <amp-img src="image-velky.jpg"
2   media="(min-width: 500px)"
3   width="850"
4   height="300"
5   layout="responsive"
6   alt="Popisek">
7 </amp-img>
8 <amp-img src="image-maly.jpg"
9   media="(max-width: 499px)"
10  width="850"
11  height="300"
12  layout="responsive"
13  alt="Popisek">
14 </amp-img>
```

Příklad 6: Příklad atributu media

V příkladu jsou dva obrázky vzájemně se vylučující podmínkou. V závislosti na velikosti obrazovky se zobrazí buď jeden nebo druhý.

- `on` – Díky tomuto atributu jde za pomoci AMP komponent přidat do AMP stránek interaktivitu. Používá se hlavně pro lightbox, sidebars (hamburgerové menu) a formuláře. Více k atributu `on` v kapitole Akce a události. [2][11]

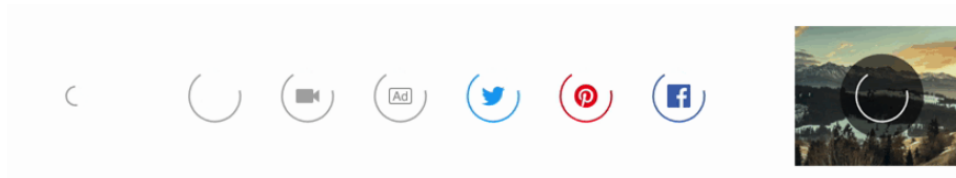
```

1 <amp-image-lightbox id="lightbox1" layout="nodisplay"></
  amp-image-lightbox>
2 <amp-img
3   on="tap:lightbox1"
4   role="button"
5   tabindex="0"
6   src="image.png"
7   width="400"
8   height="184"
9   layout="responsive"
10  alt="Popisek">
11 </amp-img>

```

Příklad 7: Příklad atributu `on`

- `noloading` – Před svým úplným načtením na webové stránce AMP komponenty jako `<amp-img>`, `<amp-video>` nebo `<amp-youtube>` zobrazují místo svého obsahu tzv. *loading indicator*, který ukazuje uživateli, že se obsah teprve načítá. Tímto atributem, se dá tato funkce vypnout. [2][11]



Obrázek 1: "Loading indicators" v AMP [13]

```
1 <amp-img src="image.jpg"  
2   noloading  
3   width="850"  
4   height="300"  
5   layout="responsive"  
6   alt="Popisek">  
7 </amp-img>
```

Příklad 8: Příklad atributu noloading

- placeholder – tzv. zástupný symbol. Tento atribut působí jako zástupce do doby, než se prvek stáhne a vykreslí. Jakmile je prvek stažený, AMP element placeholder schová a zobrazí svůj skutečný obsah. Používá se hlavně u obrázků, videí, zvuků a animací. [2][11]

```
1 <amp-anim src="image.gif "  
2   layout="intrinsic "  
3   width="480"  
4   height="270"  
5   alt="Popisek GIF">  
6 <amp-img placeholder src="image.jpg "  
7   layout="intrinsic "  
8   width="480"  
9   height="270"  
10  alt="Popisek JPG">  
11 </amp-img>  
12 </amp-anim>
```

Příklad 9: Příklad atributu placeholder

- `fallback` – Tento atribut umožňuje použití tzv. náhradního řešení pro danou komponentu. Používá se hlavně u obrázků, videí, zvuků, animací, ale třeba i u sdílení příspěvku ze sociálních sítí. Pokud například prohlížeč nepodporuje používaný formát obrázku nebo původní autor smazal svůj příspěvek na sociálních sítí, atribut `fallback` umožní vypsání chybové hlášky, případně použití jiného formátu např. místo WebP použít jpg. [2][11]

```
1 <amp-youtube
2   width="480"
3   height="270"
4   layout="responsive"
5   data-video-id="H3LoLYZq4h4">
6   <div fallback><p>Video nelze nacist</p></div>
7 </amp-youtube>
```

Příklad 10: Příklad atributu `fallback`

4 CSS v AMP

Stejně jako všechny webové stránky i AMP využívá styly CSS. Jejich použití je stejné jako v běžném HTML, až na pár omezení. Limit velikosti CSS stylů byl původně 50 kB, ale na žádosti vývojářů byl na konci ledna 2020 změněn na 75 kB. Styly CSS se nesmí nacházet v externím souboru, ale musí být v hlavičce webové stránky v tagu `<style amp-custom>`, případně AMP také povoluje použití inline (vložených) stylů. Každý inline styl může být velký maximálně 1 kB a veškeré inline styly se započítávají do celkového limitu CSS stylů 75 kB.

4.1 Zákazy v CSS

- `!important` – Tato klauzule je rezervována pro AMP samotnou. V AMP Cache projde webová stránka úpravami a aby k nim mohla dojít, používá AMP právě `!important`.
- `<link rel="stylesheet">` - Tento zákaz vychází z toho, že CSS se nesmí nacházet v externím souboru. Tento tag je však povolen pro použití externích fontů.
- `"-amp-"` a `"i-amphtml-"` – Názvy tříd začínajících takto jsou opět rezervovány pro AMP technologii, která takto pojmenovává vlastní prvky.

4.1.1 Zavináčová pravidla

Jsou to pravidla, která dále rozšiřují CSS styly o další funkce. Poskytují CSS pokyny, jak a případně kdy se má daný styl provést. Každý takový styl začíná znakem zavináče a následuje ho jedno z několika možných klíčových slov. V AMP ovšem nejsou povolena všechna zavináčová pravidla. [14]

Zavináčové pravidlo	Funkce
@font-face	Umožňuje načíst vlastní písmo na webovou stránku.
@keyframes	Umožňuje vytvářet animace.
@media	Umožňuje, aby se styly uvnitř tohoto pravidla provedla pouze za určitých podmínek. Např. při určité velikosti obrazovky.
@page	Umožňuje definovat styly, která mají jednotlivé stránky při tisku.
@supports	Umožňuje, aby se styl provedl pouze v prohlížečích, který daný styl podporují.

Tabulka 3: Povolená zavináčová pravidla v AMP

Další zavináčová pravidla jako například @import, @charset, @document nebo @namespace povolena nejsou. Samo AMP ale připouští, že časem možná povolí pravidla další. To neplatí pro pravidlo @import, které umožňuje vložit CSS z externího souboru, což je proti základnímu pravidlu AMP.

4.2 Fonty v AMP

V AMP se může použít libovolný typ písma, pokud ovšem chcete použít písmo od externího dodavatele, narazíte na pár omezení. Přidat vlastní písmo na AMP stránku se může dvěma způsoby.

1. Přidání fontu za použití @font-face

Soubory s fonty se umístí na server spolu s webovou stránkou a v CSS se fonty použijí pomocí pravidla @font-face. Pokud je AMP stránka validní, tak se webové fonty nahrají přímo do AMP Cache, díky tomu se fonty nemusí pokaždé stahovat a webová stránka si zachová svoji rychlost. Syntaxe vypadá takto: [2]

```

1 @font - face {
2   font - family = "_muj_nazev_fontu_";
3   src : url (" _cesta_k_souboru_s_fontem_ ");
4 }

```

Příklad 11: Přidání fontu pomocí @font-face

V CSS se pak font zavolá pomocí názvu fontu.

```

1 body {
2   font - family = "_muj_nazev_fontu_";
3 }

```

Příklad 12: Příklad zavolání fontu v CSS

Ne vždy se ale může font uložit do AMP Cache, například z licenčních důvodů. Proto je v AMP další způsob.

2. Přidání fontu od externích dodavatelů pomocí <link>

Použití <link> je stejné jako v běžném HTML. Mohou se ovšem použít pouze fonty od pěti povolených dodavatelů.

Dodavatel	Adresa, ze které je možné font načíst
Typography.com	https://cloud.typography.com
Fonts.com	https://fast.fonts.net
Google Fonts	https://fonts.googleapis.com
Typekit	https://use.typekit.net
Font Awesome	https://maxcdn.bootstrapcdn.com , https://use.fontawesome.com

Tabulka 4: Seznam povolených dodavatelů písma

Tag `<link>` se pak vloží do hlavičky webové stránky. Může vypadat například takto:

```
1 <link href="https://fonts.googleapis.com/css2 family=
  Lato&display=swap" rel="stylesheet">
```

Příklad 13: Ukázka tagu `<link>`

5 AMP JS

Jak bylo zmíněno v úvodu této práce, jedná se o knihovnu JavaScriptu, která slouží k rychlému načítání AMP HTML a k tomu, aby se komponenty zobrazovaly podle AMP standardů. Převážná většina AMP komponent vyžaduje přidání svého specifického skriptu do hlavičky stránky. Bez svého skriptu pak nemusí komponenta správně fungovat nebo nemusí fungovat vůbec a stránka pak nebude validní.

Takže pokud se podíváme zpět na příklad 7, zjistíme, že tento kód by sám o sobě v AMP nefungoval, protože neobsahuje specifický skript pro komponentu `<amp-image-lightbox>`, který vypadá takto:

```
1 <script async custom-element="amp-image-lightbox" src="https
  ://cdn.ampproject.org/v0/amp-image-lightbox-0.1.js"></
  script>
```

Příklad 14: Ukázka skriptu pro komponentu `<amp-image-lightbox>`

Všechny skripty pro AMP komponenty se nacházejí na stránce <https://amp.dev/documentation/components/>, případně se v dnešní době dají přidat i různé AMP rozšíření do programů pro tvorbu webových stránek, která v sobě mají uložené všechny skripty i komponenty.

V začátcích této technologie nebyl povolen žádný vlastní JavaScript a interaktivita na AMP stránkách byla zajištěna pouze pomocí atributu *on* nebo pomocí dynamických komponent AMP. To se ale změnilo koncem srpna 2019, kdy AMP přidalo komponentu `<amp-script>`, která povoluje použití vlastního JavaScriptu, opět se to však neobejde bez omezení.

5.1 Amp-script

`<amp-script>` umožňuje psát a spouštět vlastní JavaScript a zároveň zachovat rychlost a výkon AMP. Spoustu AMP komponent umožňuje běžné webové interakce pomocí své vlastní logiky. S použitím `<amp-script>` se však může použít vlastní logika pro konkrétní případy nebo jedinečné potřeby. Aby byly zachovány všechny výhody AMP, každý `<amp-script>` může být velký maximálně 10 kB, všechny skripty na stránce pak mohou obsahovat až 150 kB a kód je možné spouštět pouze na akci uživatele. [15][16]

Prohlížeče vždy pracují v jednom vlákně. Najednou se zpracovává CSS, JavaScript a další prvky stránky. Zvláště pokud se použije nadměrný JavaScript, stránky jsou pak pomalé a nereagují. Proto JavaScript v AMP běží ve Web Worker, vedlejším vlákně. Dá se říct, že se jedná o zapouzdření JavaScriptu, který pak nemůže blokovat uživatelské rozhraní. [16][17]

Podobně jako moderní javascriptové frameworky, pracuje `<amp-script>` kvůli rychlosti s DOM stromem. Web Worker nemá ale do DOM přístup, proto AMP přichází s open-source knihovnou nazvanou WorkerDOM. Ta pracuje tím způsobem, že zkopíruje DOM do virtuálního DOM a kopii zpřístupní Web Worker. Pokud Web Worker udělá změnu ve virtuálním DOM stromě, WorkerDOM tyto změny vytvoří ve skutečném DOM. To umožňuje provádět změny na stránce pomocí standardních technik. Nicméně, `<amp-script>` nepodporuje celý DOM API nebo Web API, protože by byl pak JavaScript příliš velký. [16][17]

Tato komponenta podporuje populární javascriptové frameworky, jako je například React, Angular, Vue.js nebo jQuery. AMP však doporučuje použít Preact, což je lehčí, alternativní verze Reactu. [16][17]

5.2 Akce a události

V AMP je možné vyvolat událost na základě akce uživatele, bez použití vlastního JavaScriptu pomocí atributu *on*.

```
1 on="eventName : targetId . methodName ( arg1=value , arg2=value )"
```

Příklad 15: Syntaxe atributu on

- `eventName` – Povinné jméno události. Některé AMP komponenty mají své specifické názvy událostí, které se mohou použít. Pak existuje událost `tap` (kliknutí na prvek), kterou je možné použít globálně pro všechny HTML i AMP prvky. [2][18]
- `targetId` – Jedná se o DOM id prvku, který hodláme ovlivňovat. K dispozici jsou také předdefinované speciální cíle jako `scrollTo()`, `navigateTo()`, `setState()`, `goBack` nebo třeba `print`. [2][18]
- `methodName` – Toto je volitelný název metody, kterou chceme provést s `targetId`. Většina AMP komponent má k dispozici nějaké metody. Existují i akce, které se dají použít pro všechny HTML i AMP prvky. Souvisejí se zobrazováním nebo schováním prvku, jako například akce `show`, `hide` nebo `toggleVisibility`. Při vynechání `methodName` se použije výchozí metoda dané komponenty. [2][18]
- `arg=value` – Volitelné parametry metody. [2][18]

```
1 <div id="warning - message">Varovani ... </div>
2 <button on="tap : warning - message . hide">Rozumim</button>
```

Příklad 16: Ukázka atributu on

Může se použít i více událostí na prvku, pokud se události od sebe oddělí středníkem. [18]

```
1 on="submit - success : lightbox1 ; submit - error : lightbox2 "
```

Příklad 17: Ukázka více událostí

Pro jednu událost se dá provést i několik akcí, pokud jsou od sebe akce oddělené čárkou. Jediná výjimka je u předdefinovaných cílů `setState()` a `pushState()`, které se mohou v události použít pouze jednou. [18]

```
1 on="tap : target1 . actionA , target2 . actionB "
```

Příklad 18: Ukázka více akcí

6 AMP Cache

Při normální distribuci HTML stránky je obsah webové stránky skrze konkrétní redakční systém okamžitě k dispozici pro zobrazení v prohlížeči. U AMP je to složitější. Dříve než je AMP stránka k dispozici v prohlížeči, musí se uložit do AMP Cache. Jedná se o mezipaměť, do které se uloží validní AMP stránka a následně jsou na ní provedeny různé optimalizace. Stránka se pak může zobrazit uživateli přes AMP Viewer.

Kopie AMP stránky je stažena do AMP Cache pod cizí doménou. Například:

```
1 https://ukazka--amp-cz.cdn.ampproject.org/c/s/ukazka-amp.cz/
   media
```

Na tuto adresu uživatel běžně nenarazí, ale je to adresa, kde se reálně AMP stránka nachází.

URL pro AMP Viewer pak vypadá například takto:

```
1 https://www.google.com/amp/s/ukazka-amp.cz/
```

6.1 Optimalizace a úpravy

Jak už bylo výše zmíněno, AMP Cache se nespokojí pouze s dodanou AMP stránkou, ale dále stránku upravuje a optimalizuje. Jsou to ovšem úpravy, které nezmění celkový vzhled stránky.

AMP Cache provádí:

- Kromě AMP dokumentů, ukládá do mezipaměti také obrázky a písma. [19]
- Zpracovává obrázky, které se nacházejí ve stránce v tagu `<amp-img>`:
 - Zmenšuje příliš velké obrázky, aby se zabránilo problémům s pamětí prohlížeče a špatné responzivitě.

- Odebírá neviditelná nebo obtížně viditelná data, například meta-data.
 - Konvertuje obrázky typu GIF, JPEG, PNG do formátu WebP v prohlížečích, které formát WebP podporují.
 - Generuje alternativních verze obrázků a přidává srcset pro lepší responzivitu. [19]
- Vkládá různé prefetch značky, které pomáhají prohlížeči s načítáním zdrojů z cizích domén dříve. Pro ilustraci příklad 13, změni na: [2][19]

```

1 <link
2 href="https://fonts.googleapis.com/css2?family=Lato&
   display=swap" rel="stylesheet">
3 <link href="https://fonts.gstatic.com" rel="preconnect">
4 <link href="https://fonts.gstatic.com" rel="dns-prefetch
   ">

```

Příklad 19: Ukázka prefetch značky

- Čistí a upravuje HTML, aby se zabránilo XSS útokům:
 - Hledá neuzavřené HTML tagy.
 - Odstraňuje všechny komentáře.
 - Odstraňuje bílé znaky uvnitř tagů.
 - Názvy všech tagů a atributů jsou psány malými písmeny.
 - Odchozí odkazy jsou vytvořeny jako absolutní, takže stále fungují, i když je stránka poskytována z AMP Cache. [2][19]
- Přednačítá AMP komponenty. [2]
- Může také předrendrovávat webové stránky na pozadí. [2]

Ještě před tím, než AMP Cache provede tyto změny, se stránka validuje. Pokud stránka není validní, do AMP Cache se nedostane a optimalizace ani úpravy neproběhnou.

6.2 Aktualizace AMP Cache

AMP Cache vychází z modelu *stale-while-revalidate*. Pokud uživatel navštíví AMP stránku z AMP Cache, ta automaticky zažádá o aktualizaci a na pozadí se vyvolá stažení nové verze. Další uživatel už tudíž uvidí aktualizovanou verzi stránky. S tímto modelem jsou aktualizace AMP stránek velice rychlé a starou verzi stránky vidí pouze několik uživatelů. [2][19]

Proto, aby nedocházelo k neustálému aktualizování stránky, mezipaměť považuje jakýkoliv dokument za nový po dobu nejméně 15 sekund a ostatní zdroje jako webfonty nebo obrázky za nové po dobu jedné minuty. AMP Cache se řídí kešovacími hlavičkami Max-Age jako náповědou, zda je konkrétní dokument nebo zdroj zastaralý. [2][19]

6.3 AMP Viewer

Jedná se o hybridní prostředí, kde z jedné části platí kód AMP stránky a z druhé části kód majitele AMP Cache (Google, Bing nebo Cloudflare). Cílem AMP Viewer je – kromě zobrazování AMP stránek na mobilních zařízeních – také pomoci prohlížeči s rychlým vykreslováním.

Jak je znázorněno na obrázku, kvůli tomu, že se stránka nachází na cizí URL adrese, která závisí na majiteli cache, se pod lištou prohlížeče nalézá ještě jedna šedá lišta, která se na stránce, která není zobrazena přes AMP Viewer, nenachází. Na této liště jsou dvě tlačítka. První tlačítko vlevo obsahuje informaci o tom, proč se stránka nenachází na původní adrese a druhé tlačítko vpravo obsahuje odkaz na původní adresu stránky a dává uživateli možnost stránku sdílet.

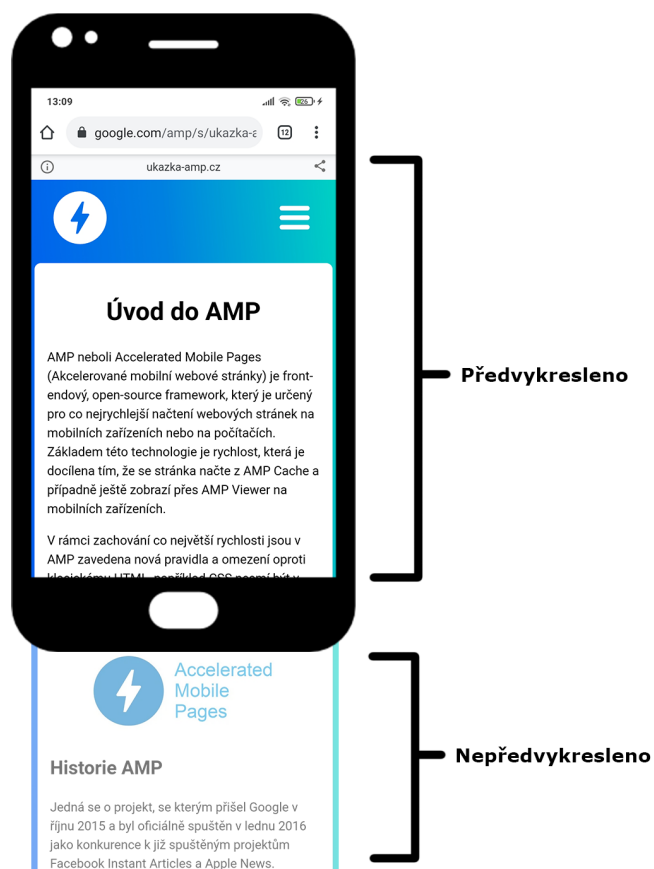


Obrázek 2: Webová stránka zobrazena přes AMP Viewer

6.3.1 Předvykreslení

Jak už bylo zmíněno v předešlé kapitole, AMP Viewer pomáhá prohlížeči s rychlým vykreslením stránky, přesněji pomáhá s předvykreslením (pre-render) stránky. Předvykreslení stránky může často ušetřit sekundy při prvním otevření stránky. V momentě, kdy uživatel projíždí výsledky vyhledávání například na Googlu, AMP Viewer se snaží uhodnout, kam uživatel klikne a danou stránku přednačte a předvykreslí. Takto se přednačte a předvykreslí více stránek, aby se zvýšila šance, že se přednačte stránka, na kterou nakonec uživatel klikne. Ovšem žádná ze stránek o tom, že byly předvykresleny, neví. [2][20][21]

Proto aby mohl AMP Viewer předvykreslit více stránek, nepřednačítává se celá stránka, ale AMP Viewer si stáhne AMP HTML kód stránky z AMP Cache, přednačte skripty z javascriptové knihovny, které jsou na stránce použity, zjistí velikost viewportu a stáhne obrázky, které se ve viewportu nacházejí. AMP Viewer jednoduše stahuje pouze obsah, který se nachází nad zlomem stránky. Díky tomu, je první načtení stránek, které byly přednačteny, téměř okamžité. [2][21]



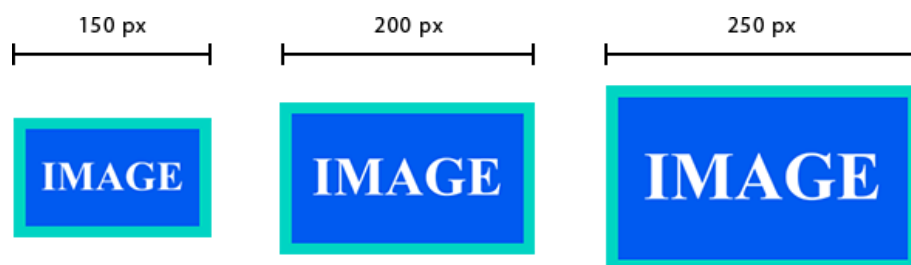
Obrázek 3: Předvykreslení

7 Layout v AMP

Layout v AMP se definuje dvěma způsoby. První je přes atribut `layout`, který slouží k definování layoutu pro nějaký externí obsah, jako například pro komponenty `<amp-img>`, `<amp-video>`, `<amp-iframe>` nebo třeba `<amp-carousel>`. Druhý způsob je přes AMP komponentu `<amp-layout>`, která slouží k definování layoutu i na běžných HTML komponentách. Layout jako takový slouží v AMP k rezervování místa při vykreslování stránky, takže stránka při načítání neposkakuje. Layout může nabývat různých hodnot, díky kterým se určí, jak se má daný prvek v rozložení stránky chovat.

U každé hodnoty se nachází příklad, jak daná hodnota funguje při různých velikostech rodičovského prvku. Zelená barva značí rodičovský prvek a modrá barva reprezentuje obrázek.

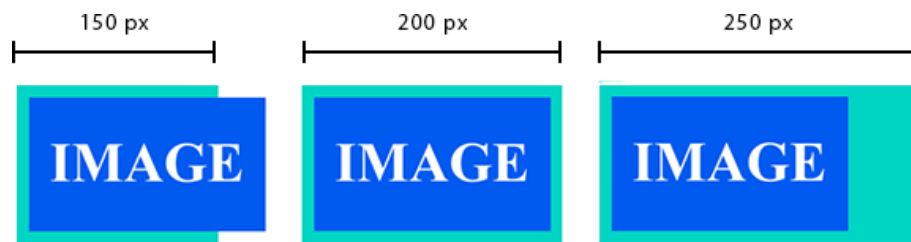
- `responsive` – Jedná se pravděpodobně o nejvíce používanou hodnotu layoutu. Jde o klasický responzivní prvek. Funguje velmi dobře pro většinu AMP komponent, včetně `<amp-img>` a `<amp-video>`. Prvek s touto hodnotou se řídí svým rodičovským prvkem a vždy drží poměr stran, který je určen v povinných attributech `width` a `height`. Takže při zvětšování nebo zmenšování rodičovského prvku se komponenta s hodnotou `layout="responsive"` nezdeformuje. [22]



Obrázek 4: Ukázka `layout="responsive"`

Obrázek se může zvětšit i nad hodnoty, které jsou zadány v attributech `width` a `height`, proto se tato hodnota hodí na obrázky, které jsou ve velkém rozlišení.

- `fixed` – Prvek s touto hodnotou drží fixní šířku a výšku definovanou v povinných atributech `width` a `height`. Takže tato hodnota rozhodně nepodporuje responzivitu. Rodičovský prvek může měnit svou velikost, ale element se zadanou hodnotou `layout="fixed"` zůstane pořád stejný. [22]

Obrázek 5: Ukázka `layout="fixed"`

- `fixed-height` – s touto hodnotou zabírá prvek prostor, který má k dispozici, ale vždy si drží fixní výšku. Povinný je zde pouze atribut `height`. Jediná hodnota, kterou může mít atribut `width`, je hodnota `auto`. [22]

Obrázek 6: Ukázka `layout="fixed-height"`

Jak je na ukázce vidět, tato hodnota layoutu, není příliš vhodná pro obrázky, protože je v závislosti na velikosti rodičovského prvku deformuje. Více se hodí například pro komponenty jako `<amp-carousel>`, kde se obsah umísťuje vodorovně. [2][22]

- `fill` – Prvek s hodnotou `layout="fill"` jednoduše zaplní veškerý dostupný prostor svého rodičovského prvku. Atributy `width` a `height` zde tedy nejsou potřeba. Je ale nutné, aby rodičovský prvek obsahoval `position:relative` nebo `position:absolute`. [2][22]

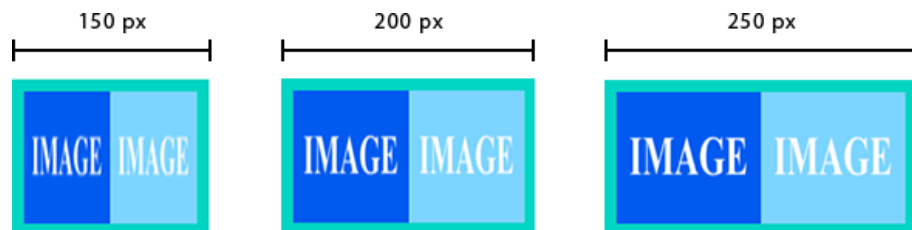
Obrázek 7: Ukázka `layout="fill"`

Sama o sobě tato hodnota není vhodná pro obrázky ze stejného důvodu jako hodnota `fixed-height`, protože dochází k deformaci obrázku. Protože se v této hodnotě neudává atribut `width` a `height`, hodnotě nejsou známe rozměry obrázku. Proto AMP radí použít při použití `layout="fill"` ještě CSS vlastnost `object-fit`, která pomáhá rozměry obrázku udržovat. [2]

- `intrinsic` – Tato hodnota je velice podobná hodnotě `responsive`, pouze s tím rozdílem, že prvky s touto hodnotou se budou zvětšovat, dokud nedosáhnou velikostí definovaných v povinných attributech `width` a `height` nebo nedosáhnou omezení CSS vlastnosti `max-width`. Tato hodnota funguje velmi dobře s většinou AMP komponent, nejlépe ale funguje s prvky, které obsahují obrázky, např. `<amp-img>` nebo `<amp-carousel>`. Nejlépe lze tuto hodnotu použít na obrázky, které nejsou v takové kvalitě a nelze je neustále zvětšovat. [22]

Obrázek 8: Ukázka `layout="intrinsic"`

- flex-item – Prvky s touto hodnotou se chovají stejně jako položky flex-boxu, to znamená, že prvek bude automaticky měnit velikost rovnoměrně s ostatními prvky, které zaplňují prostor svého rodičovského prvku. Atributy width a height zde nejsou povinné. Velikost jednotlivých prvků je určena jejich počtem a velikostí rodičovského elementu. Podmínkou pro správné zobrazení layout="flex-item" je přidání display:flex k rodičovskému prvku. [2][22]



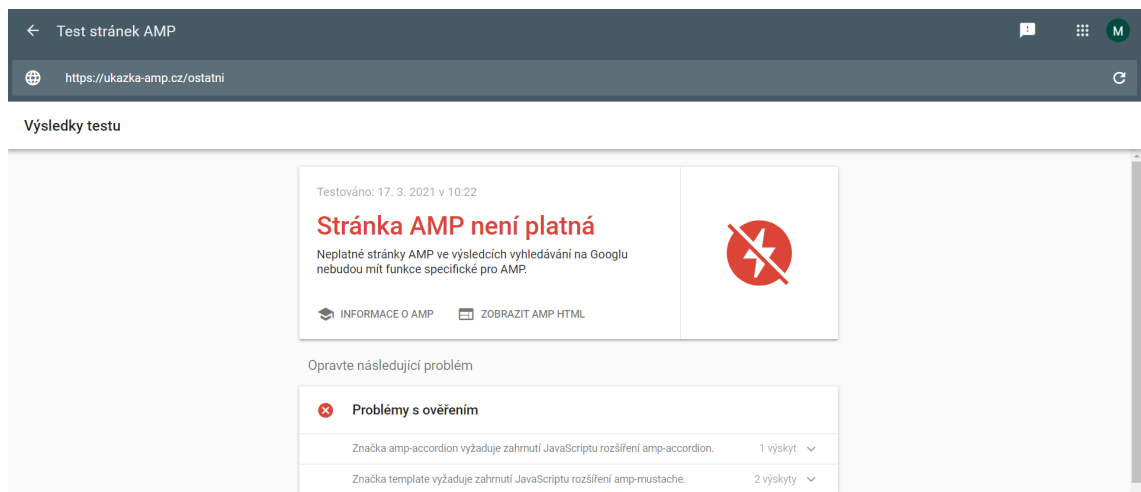
Obrázek 9: Ukázka layout="flex-item"

- container – Jedná se o výchozí hodnotu layoutu. Chová se jako <div> v běžném HTML. U této hodnoty není potřeba uvádět atributy width a height. [2]
- nodisplay – Prvky s touto hodnotou jsou schované, lze ji použít na každý AMP prvek. Prvky s touto hodnotou nezabírají na stránce žádné místo, mohou se ovšem zobrazit na akci uživatele. [22]

8 Validace AMP

Jak už bylo několikrát v této práci zmíněno, AMP stránka musí být validní, aby mohla být uložena do AMP Cache, a tím těžit z výhod této technologie. Pokud stránka validní není, bude se stále uživatelům zobrazovat, ale pomaleji. Ověřit, zda je AMP stránka validní, je možné několika způsoby:

- Pomocí online validátoru, který je dostupný na stránce <https://validator.ampproject.org/>.
- AMP validátor je také dodáván přímo v javascriptové knihovně AMP JS. Na konec URL pak stačí přidat `#development=1` a ve vývojářské konzoli je možné ověřit, zda je stránka validní.
- Validovat je možné rovněž pomocí rozšíření do prohlížeče s názvem AMP Validator. To pak ověřuje každou navštívenou stránku, zda je AMP validní. Pokud není stránka validní – svítí červeně, pokud validní je – svítí zeleně. V případě, že stránka není AMP, ale má k dispozici AMP verzi – svítí modře.
- Validace pomocí NPM balíčku. Výhodou tohoto způsobu je, že je možné takto ověřovat stránku pravidelně.
- Zjistit, zda je stránka validní můžeme také pomocí Google Search Console, kde je pro AMP vyhrazena celá sekce. Hlášení Google Search Console pak chodí majitelům webů přímo do e-mailu.



Obrázek 10: Validace pomocí Google Search Console

9 AMP komponenty

AMP definuje sadu komponent, které lze použít v libovolném AMP dokumentu. Cílem AMP komponent je umožnit autorům zahrnout funkce nad rámec standardního HTML, při zachování požadavků na zabezpečení a výkon. Všechny AMP komponenty se mohou dále stylizovat pomocí CSS stylů a mají pomoci vytvořit pokročilejší a přizpůsobivější uživatelské rozhraní. [23]

AMP komponenty se dají rozdělit na tři základní typy podle vestavěnosti:

- vestavěné (built-in) – To jsou komponenty, které se nacházejí přímo v základní javascriptové knihovně. V době psaní této práce jsou pouze tři, `<amp-img>`, `<amp-layout>` a `<amp-pixel>`.
- rozšiřující (extended) – Jedná se o komponenty, které je potřeba do stránky nainstalovat v samotné knihovně. Do této kategorie patří většina AMP komponent, které se instalují vložením svého specifického skriptu do hlavičky dokumentu. Ukázka skriptu je v příkladu 14.
- experimentální – Do AMP neustále přibývají nové komponenty, některé však ještě nejsou dokončeny a k jejich používání je nutné se přihlásit.

Protože hlavně rozšiřujících AMP komponent je opravdu hodně, jsou AMP komponenty rozděleny do kategorií.

9.1 Komponenty pro multimediální obsah

Tyto komponenty se starají o vkládání externího multimediálního obsahu (obrázky, video a audio).

amp-img

Pomocí této komponenty se vkládají do AMP stránky obrázky. Jedná se zřejmě o nejpoužívanější komponentu.

```
1 <amp-img src="img/Accelerated - Mobile - Pages . jpg "  
2   width="850"  
3   height="300"  
4   layout="responsive "  
5   alt="Logo AMP">  
6 </amp-img>
```

Příklad 20: Ukázka komponenty `<amp-img>`

Další možné atributy:

- `attribution` – Přisouzení autorských práv.
- `srcset` a `sizes` – Umožňují vložit zdroje a rozměry pro responzivní stránky.
Více o těchto attributech v kapitole HTML atributy.

amp-anim

Tato komponenta slouží k vložení animovaného obrázku, například obrázky typu GIF. Chová se velice podobně jako komponenta `<amp-img>`, až na to, že se `<amp-anim>` pokouší zredukovat vytížení procesoru, pokud se animace nachází mimo viditelný výřez stránky. Pro tuto komponentu je i vhodné použít zástupný prvek, pomocí atributu `placeholder`, pokud z nějakého důvodu animace nemůže proběhnout. [2][24]

```
1 <amp-anim src="img/globe . gif "  
2   layout="intrinsic "  
3   height="270"  
4   width="480"  
5   alt="Rotující Zeme GIF">  
6 <amp-img placeholder  
7   src="img/globe . jpg "  
8   layout="intrinsic "  
9   height="270"
```

```

10     width="480"
11     alt="Rotující Zeme JPG">
12 </amp-img>
13 </amp-anim>

```

Příklad 21: Ukázka komponenty <amp-anim>

Další možné atributy:

- attribution, srcset.

amp-audio

Tato komponenta nahrazuje HTML tag <audio> a slouží pro vložení zvukových souborů do webové stránky. Na rozdíl od HTML tagu však komponenta <amp-audio> umožňuje techniku lazy loading.

Komponenta <amp-audio> akceptuje až tři unikátní typy HTML uzlů:

- source – Tento tag umožňuje specifikovat několik zdrojových souborů a prohlížeč si vybere ten, který dokáže přehrát. [25]
- placeholder – Tento atribut vytvoří zástupný symbol, který se zobrazí místo zvuku. Kliknutím kdekoliv uvnitř <amp-audio> kontejneru se pak spustí samotný zvuk.
- fallback – Záložní řešení, které se zobrazí, pokud například prohlížeč nepodporuje HTML 5 audio.

```

1 <amp-audio src="audio/bensound-energy.mp3"
2   layout="fixed-height"
3   height="50"
4   controls controlsList="nodownload">
5 <div fallback>
6   <p>Vas prohlizec nepodporuje HTML5 audio</p>
7 </div>
8 </amp-audio>

```

Příklad 22: Ukázka komponenty `<amp-audio>`

Další možné atributy:

- `preload` – Tento atribut se může použít, pokud si autor stránky myslí, že je potřeba audio soubor načíst okamžitě při načtení stránky.
- `autoplay` – Pokud je použit tento atribut, zvukový soubor začne hrát, jakmile bude načten.
- `loop` – Pokud je použit tento atribut, zvukový soubor se po dosažení svého konce vrátí zase na začátek.
- `muted` – Pokud je použit tento atribut, zvukový soubor je ztlumen.
- `controlsList` – Tento atribut pomáhá určit, které ovládací prvky se nezobrazí, pokud je použit atribut `controls`. Možné hodnoty jsou `nodownload`, `nofullscreen`, a `noremoteplayback`. [26]

Další možné atributy pro práci s médii:

- `artwork` – Odkaz na obrázek, který slouží jako ikona zvukové stopy.
- `artist` – Řetězec, který označuje autora zvukové stopy.
- `album` – Řetězec, který označuje album, ze kterého byla zvuková stopa převzata.
- `title` – Řetězec, který označuje název zvukové stopy.

amp-video

Tato komponenta nahrazuje HTML tag `<video>` a slouží pro vložení videa do AMP stránky. Stejně jako u komponenty `<amp-audio>` je největší rozdíl mezi HTML tagem `<video>` a AMP komponentou `<amp-video>` v možnosti použití techniky lazy loading. Jinak se tato komponenta ovládá stejně jako její nahrazená značka `<video>`.

Komponenta `<amp-video>` akceptuje až čtyři možné unikátní typy HTML uzlů:

- `source` – Tento tag umožňuje specifikovat několik zdrojových souborů a prohlížeč si vybere ten, který dokáže přehrát. [25]
- `track` – Tento tag slouží pro vložení titulků do videa.
- `placeholder` – Zástupný symbol, který se zobrazí před spuštěním videa.
- `fallback` – Pokud atribut nepodporuje HTML 5 video.

```
1 <amp-video controls
2   width="480"
3   height="270"
4   layout="responsive"
5 <source src="video/example.webm"
6   type="video/webm" />
7 <source src="video/example.mp4"
8   type="video/mp4" />
9 <div fallback>
10   <p>Vas prohlizec nepodporuje HTML5 video</p>
11 </div>
12 </amp-video>
```

Příklad 23: Ukázka komponenty `<amp-video>`

Další možné atributy:

- `src` – Je nutný, pokud není použit tag `<source>`.
- `poster` – Označuje obrázek, který se zobrazí před spuštěním videa. Pokud není tento atribut použit, zobrazí se před spuštěním videa první frame videa.

- `autoplay` – Pokud je použit tento atribut, video se spustí automaticky, jakmile bude viditelné.
- `controls` – S tímto atributem je umožněno uživateli ovládat video.
- `controlsList` – Použití je stejné jako u komponenty `<amp-audio>`.
- `dock` – Pro použití tohoto atributu musí být v hlavičce dokumentu použitý skript pro AMP komponentu `<amp-video-docking>`. Pokud je tento atribut použit a video bylo spuštěno uživatelem, je video automaticky zmenšeno a zafixováno na horní roh stránky, když se uživatel posouvá z vizuální oblasti videa.
- `loop` – Po dosažení svého konce se video spustí automaticky znovu.
- `crossorigin` – Tento atribut je povinný, pokud je použit tag `track` a soubor s titulky je z jiného zdroje než video. [27]
- `disableremoteplayback` – Určuje, zda má prvek média povoleno uživatelské rozhraní vzdáleného přehrávání, jako je Chromecast nebo AirPlay. [27]
- `noaudio` – Tento atribut označuje video, které nemá žádný zvuk.
- `rotate-to-fullscreen` – Pokud je video viditelné a uživatel otočí mobil do vodorovné polohy, video se automaticky dá na celou obrazovku. [27]

Další možné atributy pro práci s médii jsou stejné jako u komponenty `<amp-audio>`, `artwork`, `artist`, `album` a `title`.

9.1.1 Komponenty pro multimediální obsah pro vkládání obsahu třetích stran

AMP připravilo řadu komponent speciálně pro vkládání obsahu konkrétních třetích stran. Jedná se o náhražky standardních vkládacích kódů těchto poskytovatelů, protože občas mohou tyto kódy zpomalovat načtení. [2]

amp-youtube

Jedná se asi o nejpoužívanější komponentu pro vkládání obsahu třetích stran. Pomocí této komponenty je možné vložit do stránky video ze stránky youtube.com.

```
1 <amp-youtube data-videoid="H3LoLYZq4h4"  
2   width="480"  
3   height="270"  
4   layout="responsive"  
5   dock>  
6 </amp-youtube>
```

Příklad 24: Ukázka komponenty <amp-youtube>

Možné atributy:

- **autoplay** – Pokud je použit tento atribut a prohlížeč tento atribut podporuje:
 - video se spustí automaticky, jakmile bude viditelné.
 - video je automaticky ztišeno a zvuk se spustí, až když uživatel na video klikne.
 - jakmile je video mimo viditelný výřez stránky, video se automaticky zastaví.
- **loop** – Po dosažení svého konce, se video spustí automaticky znovu.
- **data-videoid** – Id YouTube videa, které je možné najít v URL každého videa. Například úplná URL adresa videa použitého v příkladu je <https://www.youtube.com/watch?v=H3LoLYZq4h4>.
- **data-live-channelid** – Id kanálu YouTube, které poskytuje přímé přenosy. Id kanálu se získá stejně jako u atributu data-videoid. [28]

- `data-param-*` - Všechny takovéto atributy budou přidány do `src` YouTube iframe. Může se použít například k tomu, zda se budou zobrazovat ovládací prvky videa, například: `data-param-controls=1`. [28]
- `dock` – Použití stejné jako u komponenty `<amp-video>`.

amp-dailymotion

Tato komponenta slouží pro vložení videa ze stránky `dailymotion.com` do AMP stránky.

```

1 <amp-dailymotion data-video-id="x7zd2p8 "
2   layout="responsive "
3   width="480"
4   height="270">
5 </amp-dailymotion>
```

Příklad 25: Ukázka komponenty `<amp-dailymotion>`

Možné atributy:

- `autoplay` – Funguje stejně jako u komponenty `<amp-youtube>`.
- `data-video-id` – Id Dailymotion videa, které je možné najít v URL každého videa. Například úplná URL adresa videa použitého v příkladu je `https://www.dailymotion.com/video/x7zd2p8`.
- `data-mute` – Určuje, zda bude video bez zvuku. Možné hodnoty jsou `"true"` a `"false"`. Pokud tento atribut není uveden, je jeho výchozí hodnota `"false"`.
- `data-sharing-enable` – Tento atribut určuje, zda se zobrazí tlačítko pro sdílení videa. Možné hodnoty jsou `"true"` a `"false"`. Pokud není tento atribut uveden, je jeho výchozí hodnota `"true"`.

- `data-start` – Tento atribut určuje čas v sekundách, od kterého se má video spustit. Například `data-start=35`. Pokud tento atribut není uveden, je jeho výchozí hodnota 0 a video se spustí od začátku.
- `data-ui-highlight` – Pomocí tohoto atributu se může změnit barva ukazateli průběhu a menu nastavení videa. Například `data-ui-highlight="00D4C3"`.
- `data-ui-logo` – Určuje, zda se zobrazí logo Dailymotion ve videu. Možné hodnoty jsou `"true"` a `"false"`. Pokud tento atribut není uveden, je jeho výchozí hodnota `"true"`.
- `data-info` – Určuje, jestli se zobrazí informace o videu (titulek a vlastník videa) před spuštěním videa. Možné hodnoty jsou `"true"` a `"false"`. Pokud není tento atribut uveden, je jeho výchozí hodnota `"true"`.
- `data-param-*` a `dock` – Oba tyto atributy fungují stejně jako u komponenty `<amp-youtube>`.

amp-soundcloud

Tato komponenta slouží pro vložení audio stopy z platformy SoundCloud do AMP stránky.

```

1 <amp-soundcloud
2   data-trackid="747747901"
3   height="166"
4   layout="fixed-height">
5 </amp-soundcloud>
```

Příklad 26: Ukázka komponenty `<amp-soundcloud>`

Možné atributy:

- `data-trackid` – Tento atribut je povinný, pokud není použitý atribut `data-playlistid`. Id pro platformu SoundCloud se nedají zjistit tak jednoduše jako například pro komponenty `<amp-youtube>` nebo `<amp-dailymotion>`, kdy se id nachází hned v URL. Pro zjištění id písničky

z platformy SoundCloud se nejprve musí kliknout na tlačítko sdílení dané písně a následně na tlačítko Embed, kde se vygeneruje HTML kód pro vložení do HTML stránky. V tomto HTML kódu se nachází atribut `src`, ve kterém je úplná adresa dané audio stopy. Id se pak nachází na konci této adresy. Úplná adresa audio stopy může vypadat například takto:

```
src=https://w.soundcloud.com/player/?url=https%3A//api.
soundcloud.com/tracks/747747901
```

- `data-playlistid` – Tento atribut je povinný, pokud není použit atribut `data-trackid`. Id playlistu se hledá úplně stejně jako pro atribut `data-trackid`, až na to, že se na začátku neklikne na sdílení písně, ale celého playlistu.
- `data-secret-token` – Tento atribut se používá pro zadání secret tokenu pro video, které je soukromé.
- `data-visual` – Pokud je tento atribut natavený na "true", zobrazí se vizuální režim. Jinak se audio stopa zobrazuje v klasickém režimu. Pokud není tento atribut použit, je jeho výchozí hodnota "false".
- `data-color` – Používá se ke změně barvy přehrávacího tlačítka a ukazatele průběhu audio stopy. Tento atribut se může použít pouze v klasickém režimu. Například `data-color="00D4C3"`.

amp-megaphone

Tato komponenta se používá pro vložení podcastu nebo playlistu podcastů z platformy Megaphone do AMP stránky.

```
1 <amp-megaphone data-episode="HSW9450898480"
2   height="166"
3   layout="fixed-height"
4   data-light >
5 </amp-megaphone>
```

Příklad 27: Ukázka komponenty `<amp-megaphone>`

Možné atributy:

- `data-episode` – Tento atribut je povinný, pokud není použit atribut `data-playlist` a slouží k zadání id podcastu. Id se nachází přímo v URL podcastu. Například úplná URL adresa podcastu použitého v příkladu je <https://cms.megaphone.fm/channel/techstuff?selected=HSW9450898480>.
- `data-playlist` – Tento atribut je povinný, pokud není použit atribut `data-episode` a slouží k zadání id playlistu podcastů.
- `data-start` – Lze použít pouze v případě, že je použit atribut `data-episode`. Slouží k nastavení času, od kterého se má podcast spustit.
- `data-episodes` – Tento atribut se může použít pouze tehdy, pokud je použit i atribut `data-playlist`. Slouží k nastavení počtu podcastů, které se mají zobrazit.
- `data-tile` – Tento atribut se může použít pouze v případě, že je použit i atribut `data-episode`. Pokud je tento atribut použit, části přehrávače podcastu jsou zobrazeny vertikálně.
- `data-light` – Pokud je použit tento atribut, přehrávač se přepne do světlého režimu.

Další komponenty pro multimediální obsah

- `amp-3d-gltf` – Pomocí této komponenty je možné do AMP stránky vložit 3D model, který je v glTF formátu.
- `amp-3q-player` – Vložení videa z platformy 3Q.
- `amp-apester-media` – Pomocí této komponenty je možné na AMP stránce zobrazit například kvíz ze stránky apester.com.

- amp-bodymovin-animation – Pomocí této komponenty je možné na AMP stránce zobrazit animaci přes Airbnb Bodymovin animation player.
- amp-brid-player – Vložení videa ze stránky brid.tv.
- amp-brightcove – Vložení videa ze stránky brightcove.com.
- amp-connatix-player – Vložení videa z platformy Connatix.
- amp-delight-player – Vložení videa z platformy delight-vr.com.
- amp-embedly-card – Pomocí této komponenty je možné vložit do AMP stránky jakýkoliv obsah, jako například článek, příspěvek ze sociálních sítí nebo třeba video z jakékoliv platformy pomocí Embedly card.
- amp-hulu – Vložení videa ze stránky hulu.com.
- amp-ima-video – Pomocí této komponenty je možné vložit videoreklamu před video.
- amp-imgur – Pomocí této komponenty je možné do AMP stránky vložit příspěvek ze stránky imgur.com.
- amp-izlesene – Vložení videa ze stránky izlesene.com.
- amp-jwplayer – Vložení videa ze stránky jwplayer.com.
- amp-kultura-player – Vložení videa ze stránky corp.kultura.com.
- amp-mowplayer – Vložení videa ze stránky mowplayer.com.
- amp-minute-media-player – Vložení videa ze stránky minutemedia.com.
- amp-nexxtv-player – Vložení média z platformy nexxOMNIA.
- amp-ooyala-player – Vložení videa z platformy Ooyala.
- amp-playbuzz – Pomocí této komponenty lze vložit do AMP stránky různé kvízy, dotazníky, průzkumy atd. ze stránky playbuzz.com.

- `amp-reach-player` – Vložení videa z platformy Beachfront Reach.
- `amp-redbull-player` – Vložení videa a jeho zobrazení přes prohlížeč Red bull Video Player.
- `amp-springboard-player` – Vložení videa z platformy Springboard.
- `amp-video-iframe` – Tato komponenta je podobná komponentě `<amp-iframe>`, `<amp-video-iframe>`. Je však určena pouze pro videa a slouží buď k vytvoření vlastního video přehrávače, nebo k vložení videí třetích stran, které nemají vlastní AMP komponentu.
- `amp-vimeo-player` – Vložení videa z platformy Vimeo.
- `amp-wistia-player` – Vložení videa z platformy Wistia.
- `amp-yotpo` – Tato komponenta slouží k vložení widgetů z platformy Yotpo.

9.2 Komponenty pro uživatelské rozhraní

Tyto komponenty pomáhají v AMP hlavně s rozvržením stránky. Do této kategorie spadají ale rovněž prvky uživatelského rozhraní, jako jsou karusely, lightboxy, observery nebo kupříkladu `iframe`.

amp-accordion

Pomocí této komponenty je možné otevírání nebo zavírání sekcí na interakci uživatele.

```
1 <amp-accordion animate>
2   <section>
3     <header>Material a~udrzba</header>
4     <ul>
5       <li>Material svrchni latka: 100% bavlna</li>
6       <li>Konstrukce materialu: Zerzej</li>
```

```
7         <li>Pokyny pro peci: ...</li>
8     </ul>
9 </section>
10 <section>
11     ...
12 </section>
13 </amp-accordion>
```

Příklad 28: Ukázka komponenty `<amp-accordion>`

Možné atributy:

- `animate` – Tento atribut přidá ke komponentě animaci při rozbalování a sbalování obsahu.
- `disable-session-states` – Pomocí tohoto atributu je možná deaktivovat zachování sbaleného/rozbaleného stavu. [29]
- `expanded` – Při použití tohoto atributu do tagu `<section>` se tato sekce rozbalí při načtení stránky. [29]
- `expand-single-section` – Při použití tohoto atributu může být najednou rozbalena pouze jedna sekce. Pokud uživatel klikne na sbalenou sekci, aktuálně rozbalená sekce se sbalí a rozbalí se ta, na kterou uživatel klikl. [29]

amp-carousel

Tato komponenta slouží k vodorovnému zobrazení podobných částí obsahu. Používá se hlavně k zobrazení jakéhokoliv obsahu, který se při nedostatku místa posunuje do stran pomocí šipek nebo přejetí prstem. Může se použít pro zobrazení fotografií nebo například k výpisu položek jednořádkové posuvné navigace, kdy se vypíše takový počet položek, který se vejde do šířky okna. [2]

```
1 <amp;carousel autoplay
2   layout="responsive "
3   width="400"
4   height="240"
5   type="slides">
6   <amp;img src="img/pexels - picjumbocom - 461077.jpg"
7     width="400"
8     height="240"
9     layout="responsive "
10    alt="Prace na pocitaci"
11  ></amp;img>
12  <amp;img src="img/pexels - kevin - ku - 577585.jpg"
13    width="400"
14    height="240"
15    layout="responsive "
16    alt="Pohled na programovani skrze bryle"
17  ></amp;img>
18 </amp;carousel>
```

Příklad 29: Ukázka komponenty <amp;carousel>

Možné atributy:

- type – Tento atribut určuje typ karuselu a může nabývat dvou hodnot:
 - carousel – Výchozí hodnota atributu. Všechny položky karuselu jsou viditelné a je možné mezi nimi posouvat pomocí rolovací lišty. Všechny položky mohou také mít jinou šířku specifikovanou pomocí CSS.
 - slides – Zobrazí vždy jen jednu položku, která se změní poté, co uživatel přejeđe prstem nebo klikne na šipku.

- `controls` – Umožňuje permanentní zobrazení navigačních šipek na mobilních zařízeních. Ve výchozím nastavení navigační šipky zmizí, jakmile uživatel přejde prstem pro přesunutí na další položku. [30]
- `data-next-button-aria-label` a `data-prev-button-aria-label` – Tyto dva atributy slouží ke změně názvů hodnot po najetí myši na šipku. Pokud nejsou tyto atributy použity, výchozí hodnoty jsou `Next item in carousel` a `Previous item in carousel`. [30]
- `data-button-count-format` – Řetězec, který se objeví po najetí myši na šipku a nachází se za hodnotami atributů `data-next-button-aria-label` a `data-prev-button-aria-label`. Výchozí hodnota tohoto atributu je `(%s of %s)`, takže po najetí myši se ukáže například `(1 of 4)`. To slouží hlavně pro uživatele, kteří používají čtečku obrazovky, aby jim poskytovala informace o jejich postupu karuselem. [30]
- `autoplay` – Automaticky projíždí jednotlivé položky bez interakce uživatele. Pokud uživatel změní položky ručně, automatické přehrávání se přeruší. Ve výchozím nastavení se snímek přesune po pěti sekundách. To ale může být přepsáno atributem `delay`. Autoplay je možné nastavit pouze v případě, že jsou v karusele alespoň dvě položky. [30]
- `delay` – Specifikuje čas v milisekundách pro zpoždění postupu na další položku.
- `loop` – Umožňuje se z poslední položky přesunout zpátky na první. Atribut `loop` je možné použít pouze tehdy, pokud jsou v karusele alespoň tři položky.

amp-iframe

Tato komponenta umožňuje na AMP stránce zobrazit iframe. Jedna z velkých výhod této komponenty v kontextu AMP technologie je to, že je v ní možné použít jakýkoliv JavaScript. Může se tak hodit pro vkládání vlastních komponent nebo kódů třetích stran, které nemají vlastní komponentu, například vložení mapy z Google Maps. [2][31]

Aby byl ale `<amp-iframe>` validní, je nutné dodržovat pár pravidel:

- `<amp-iframe>` se nesmí nacházet na začátku dokumentu. Pokud nemá specifikovaný zástupný prvek (placeholder) musí být aspoň 600px daleko od horního okraje stránky nebo se nesmí nacházet v prvních 75 % viewportu. [31]
- Je sandboxovaný, tzn. neběží v něm žádné scripty, neotevřít odkazy mimo vlastní rámec atd., dokud to není povoleno parametrem `sandbox="allow-scripts"`. [2]
- Zdroj musí být stažen přes zabezpečený protokol HTTPS. [2]
- Zdroj se musí nacházet na jiné doméně než AMP stránka, pokud se to ovšem nezmění parametrem `sandbox="allow-same-origin"`. [2][31]

```

1 <amp-iframe width="200"
2   height="100"
3   sandbox="allow-scripts allow-same-origin"
4   layout="responsive"
5   frameborder="0"
6   src="https://www.google.com/maps/embed/v1/place?q=
   place_id:ChIJVxFuTMhPc0cRz3iG41BAPGg&key=
   AIzaSyBDaQzTLFAZFEBnsfDZwQZAatsLRn7X3eM">
7 </amp-iframe>

```

Příklad 30: Ukázka komponenty `<amp-iframe>`

Ačkoliv je možné v `<amp-iframe>` použít vlastní komponenty, sami vývojáři AMP nabádají, aby se, pokud je to možné, dávala přednost AMP komponentám.

Možné atributy:

- `src` – Tento atribut se chová stejně jako u běžného iframu, až na jednu výjimku. `<amp-iframe>` přidává do URL adresy `#amp=1` fragment, aby zdrojové dokumenty věděly, že jsou zakotveny v AMP. [31]
- `sandbox` – `<amp-iframe>` musí vždy obsahovat atribut `sandbox`. Ve výchozím nastavení je prázdný. Možné hodnoty sandboxu jsou `allow-scripts`, `allow-same-origin`, `allow-popups` a `allow-popups-to-escape-sandbox`. Poslední dvě hodnoty slouží k otevření nového okna v prohlížeči přímo z iframu. [31]
- `srcdoc`, `frameborder`, `allowfullscreen`, `allowpaymentrequest`, `allowtransparency` a `referrerpolicy` – Tyto atributy fungují stejně jako v běžném iframu. [31]

amp-image-lightbox

Pomocí této komponenty je možné použít efekt lightboxu na zadaný obrázek. Zobrazí tedy obrázek v plné velikosti.

```

1 <amp-image-lightbox id="lightbox-boilerplate" layout="
  nodisplay"></amp-image-lightbox>
2 <amp-img on="tap:lightbox-boilerplate"
3   role="button"
4   tabindex="0"
5   src="img/amp-boilerplate.png"
6   width="400"
7   height="184"
8   layout="responsive"
9   alt="amp boilerplate">
10 </amp-img>

```

Příklad 31: Ukázka komponenty `<amp-image-lightbox>`

`<amp-image-lightbox>` se používá pomocí atributu `on` přes název události `tap` a `id` `lightboxu`. Jeden `lightbox`, respektive jedno `id` `lightboxu` je možné použít i na více obrázků. V plné velikosti se ale vždy zobrazí pouze jeden obrázek.

K obrázkům je možné přidat i popisek pod obrázek, a to dvěma způsoby:

- Tag `<amp-img>` se obalí tagem `<figure>` a popisek se pak vloží do tagu `<figcaption>`.
- Pokud je u obrázku zadán atribut `aria-describedby`, obsah elementu, který má v sobě specifikovanou hodnotu `aria-describedby` – jako `id` – se pak zobrazí jako popisek.

Možné atributy:

- layout – Nutný atribut, musí být nastavený na `nodisplay`.
- id – Id lightboxu, který je pak použit u události *on*.

amp-image-slider

Tato komponenta slouží k porovnání dvou obrázků pomocí posuvníku. Jeden obrázek se zobrazí nalevo a druhý napravo, pomocí posuvníku je pak možné zobrazit větší část jednoho nebo druhého obrázku. Vedle posuvníku se na každé straně nachází šipky jako nápověda pro uživatele. Při první interakci uživatele šipky zmizí, mohou se ovšem znovu objevit, pokud se komponenta dostane mimo viditelný výřez stránky a znovu se do něj vrátí. Tyto výchozí šipky mohou být upraveny nebo změněny pomocí CSS. Do komponenty mohou být také přidány dva tagy `<div>`, které pak slouží jako popisky obrázků. Tag `<div>` pro levý obrázek musí mít atribut `first` a pro pravý obrázek atribut `second`. Pokud se popisky neupraví pomocí CSS, zobrazí se v levém horním rohu obrázků.

```

1 <amp-image-slider layout="responsive" width="400" height
  ="267">
2 <amp-img src="img/clothes/blue-t-shirt.jpg" alt="Modre
  tricko" layout="fill"></amp-img>
3 <amp-img src="img/clothes/green-t-shirt.jpg" alt="Zelene
  tricko" layout="fill"></amp-img>
4 </amp-image-slider>

```

Příklad 32: Ukázka komponenty `<amp-image-slider>`

Možné atributy:

- disable-hint-reappear – Tento atribut zamezuje opětovnému zobrazení šipek. Šipky jsou pak viditelné pouze jednou při načtení obrázků.

- `initial-slider-position` – Ve výchozím nastavení se posuvník při načtení obrázků nachází vždy uprostřed. Pomocí tohoto atributu se dá počáteční pozice posuvníku změnit. Možné hodnoty se nacházejí mezi 0 a 1, které odpovídají hodnotám 0 % až 100 % od levé strany obrázků. [32]
- `step-size` – K použití tohoto atributu je nejprve nutné definovat na komponentě `tabindex`. Pak je možné, pokud je komponenta zaměřena, změnit polohu posuvníku pomocí šipek klávesnice doleva nebo doprava. Pomocí tohoto atributu je možné nastavit o kolik se posuvník posune na jeden klik. Možné hodnoty se nacházejí mezi 0 a 1 a odpovídají hodnotám 0 % až 100 %. [32]

Další AMP komponenty pro uživatelské rozhraní:

- `amp-app-banner` – Tato komponenta poskytuje možnosti vložení reklamy na mobilní aplikaci s pevnou pozicí a zobrazující výzvu k instalaci aplikace. Tato reklama se ovšem zobrazí pouze v AMP Vieweru. [2]
- `amp-lightbox-gallery` – Tato komponenta slouží pro zobrazení více obrázků v lightboxu najednou.
- `amp-lightbox` – Tento lightbox neslouží jako předešlé lightboxy ke zobrazení obrázků v plném rozlišení, nýbrž pro zobrazení HTML obsahu. [2]
- `amp-sidebar` – Tato komponenta slouží pro vložení bočního panelu. Nejvíce je využívána k vytvoření navigace na mobilních zařízeních.

9.2.1 Komponenty pro přidání vizuálních efektů

V této kategorii se nachází komponenty, které v sobě mají dané různé animace v kombinaci s dalšími styly prvků.

amp-fx-collection

V této komponentě je zabudována kolekce vizuálních efektů a animací. Nejvíce se tato komponenta používá k přidání efektů k obrázkům, ale je ji možné použít například i na text. Tato komponenta se použije tak, že se přidá specifický skript do hlavičky stránky a pak už se jen přidá atribut amp-fx s názvem možného efektu k elementu, na kterém chceme efekt použít.

```

1 <amp-img amp-fx="fly - in - left "
2   data-duration="2000ms"
3   src="img/pexels - soumil - kumar - 735911 . jpg "
4   width="1200"
5   height="720"
6   layout="responsive "
7   alt="Ruka na klavesnici">
8 </amp-img>
```

Příklad 33: Ukázka komponenty <amp-fx-collection>

Podporované vizuální efekty jsou:

- parallax – Efekt, který zařizuje animaci objektu při posunu stránky. Jak uživatel posouvá stránku, prvek se posouvá rychleji nebo pomaleji v závislosti na hodnotě zadané v atributu data-parallax-faktor. [33]
 - data-parallax-faktor – Atribut, který je nutné přidat k objektu, pokud je použit efekt parallax a určuje, jak rychle se bude objekt pohybovat vzhledem k rychlosti posouvání stránky. [33]
 - * Hodnota větší než 1 posouvá prvek nahoru, tzn. prvek se posouvá rychleji, když uživatel posouvá stránku dolů.

- * Hodnota menší než 1 posouvá prvek dolů, tzn. prvek se posouvá pomaleji, když uživatel posouvá stránku dolů.
 - * Hodnota 1 objekt neposouvá, objekt se chová normálně jako by na něm žádný efekt použit nebyl.
 - * Hodnota 0 posouvá objekt na fixní pozici spolu s posunem stránky.
- fade-in – Postupné, animované zobrazení objektu, když se dostane do viditelného výřezu stránky. Možné atributy při použití efektu fade-in jsou:
 - data-duration – Určuje, jak rychle animace zobrazení proběhne. Výchozí hodnota je 1000ms.
 - data-easing – Tento atribut umožňuje měnit rychlost animace v průběhu jejího trvání. Výchozí hodnota ease-out odpovídá cubic-bezier(0.40, 0.00, 0.40, 1.00). Je možné vybrat jednu z možných přednastavených hodnot, jako je linear, ease-in-out, ease-in, nebo je možné nastavit hodnoty manuálně pomocí custom-bezier(). [33]
 - data-margin-start – Tento atribut určuje, kdy se má animace spustit. Hodnota tohoto atributu určuje, že animace proběhne, pokud objekt překročil procento viditelného výřezu stránky. Výchozí hodnota je 5 %. [33]
 - fade-in-scroll – Efekt umožňuje měnit průhlednost prvku, který je ovlivňovaný posunem stránky. Možné atributy při použití efektu fade-in-scroll jsou:
 - data-margin-start – Funkce stejná jako u efektu fade-in. Výchozí hodnota je 0 %.
 - data-margin-end – Tento atribut určuje, kdy se má animace zastavit. Hodnota tohoto atributu určuje, že by animace měla být hotová, když je zadané procento prvku. Výchozí hodnota je 50 %. [33]

- data-repeat – Ve výchozím nastavení je, že pokud je objekt plně viditelný, průhlednost už se znovu neanimuje. Pomocí tohoto atributu je možné funkci vypnout a animace průhlednosti při pohybu stránky bude aktivní po celou dobu. [33]
- float-in-top a float-in-bottom – Tyto efekty se používají hlavně na fixní hlavičku a patičku dokumentu. Objevují se a mizí podle toho, jestli uživatel posouvá dokument nahoru nebo dolů. Objekty, u kterých jsou použity tyto efekty musí mít nastaveny v CSS vlastnosti position="fixed", overflow="hidden", pokud je použit efekt *float-in-top* tak vlastnost top=0 a pokud je použit efekt *float-in-bottom* tak bottom=0. [33]
- fly-in-bottom, fly-in-top, fly-in-right a fly-in-left – Tyto efekty umožňují příjezd objektu z daného směru. Možné atributy při použití těchto efektů jsou:
 - data-duration – Pomocí tohoto atributu je možné upravit trvání animace. Výchozí hodnota se liší podle velikosti obrazovky. Mezi velikostmi obrazovky 480px až 1000px je výchozí rychlost animace 400ms až 600ms. Pokud je tato hodnota nastavena, bude rychlost platit na všech zařízeních. [33]
 - data-easing – Funkce stejná jako u efektu fade-in.
 - data-margin-start – Funkce stejná jako u efektu fade-in. Výchozí hodnota je 5 %.

amp-fx-flying-carpet

Tato komponenta umožňuje vložit prvek do posouvacího kontejneru, který zobrazí prvek na celou obrazovku, aniž by zabral celý výřez stránky. Prvek se tváří, jako kdyby byl fixně na pozadí a stránka přes něj přejíždí. Nejvíce se tato komponenta používá pro vložení reklam. [2]

```
1 <amp-fx - flying - carpet height="350px">
2   <amp-img src="img/pexels - andy - vu - 3244513 . jpg "
3     layout="responsive "
4     width="300 "
5     height="500 "
6     alt="Monument Valley">
7 </amp - img >
8 </amp - fx - flying - carpet >
```

Příklad 34: Ukázka komponenty `<amp-fx-flying-carpet>`

Možné atributy:

- `height` – Tento atribut určuje výšku výřezu okna.

9.2.2 Observery

Tyto komponenty umožňují spuštění animací a efektů na základě změny orientace zařízení nebo na základě pozice viewportu na stránce. Do této kategorie patří dvě komponenty:

- `amp-orientation-observer` – Tato komponenta hlídá orientaci zařízení a podle toho vyvolá zadanou akci. Díky této komponentě je možné vyvolat animace na základě změny orientace zařízení nebo například v kombinaci z různými přehrávači videa, kdy je například možné při otočení zařízení do vertikální polohy zvětšit video na celou obrazovku. [2]
- `amp-position-observer` – Hlídá pozici prvku na stránce. Pomocí této komponenty je možné spouštět různé animace podle toho, kde se zrovna uživatel nachází. Je taky možné například zobrazit různé objekty – například známou šipku, díky které se dostane uživatel zpět na začátek stránky, až poté, co se uživatel dostane na stránce níže.

9.3 Dynamické komponenty

Tyto komponenty tvoří důležitou část pro tvorbu plnohodnotného AMP webu. Díky nim je možné přidat do AMP stránky interaktivní prvky, notifikace, formuláře platební brány atd.

amp-bind

Jedná se o velice důležitou komponentu, která slouží k přidání interaktivity do AMP stránky. Po kliknutí na tlačítko umožňuje změnit text nebo třeba i obrázek. To se dá využít například u e-shopů, kdy je možné pomocí této komponenty změnit obrázek podle toho, jaká je zrovna zvolená barva.

`<amp-bind>` má tři hlavní koncepty:

- Stav (state) – Zpočátku je stav prázdný. Stavové proměnné se mění až na akci uživatele. Vizuální prvky by však měly mít definovaný výchozí stav. [34][35]
- Výrazy (expressions) – Výrazy podobné JavaScriptu, které mohou odkazovat na stav. [34]
- Vazby (bindings) – Speciální atributy, které propojují vlastnost prvku se stavem pomocí výrazu. Vlastnost je propojena tím, že se vloží do hranatých závorek. Atributy mohou být `[text]`, `[class]`, `[hidden]`, `[width]`, `[height]`. [34][35]

```

1 <p [text]=' 'Hello ' + foo">Hello World</p>
2
3 <button on="tap:AMP.setState({foo: 'Interactivity'})">
4   Change text
5 </button>
```

Příklad 35: Ukázka komponenty `<amp-bind>` [34]

Na stránce je ve výchozím stavu zobrazen text Hello World a tlačítko Change text. Po kliknutí na tlačítko se změní stav stavové proměnné foo pomocí atributu on a text se změní na Hello Interactivity.

Stav aplikace pomocí amp-bind

„Stav nebo také výchozí datovou strukturu aplikace je možné inicializovat a upravovat. Skoro jako v moderních javascriptových frameworkách. Nejprve výchozí nastavení stavu:

```

1 <amp-state id="myLocalState">
2   <script type="application/json">
3     {
4       "foo": "bar"
5     }
6   </script>
7 </amp-state>

```

Stav je ale možné inicializovat i z externího zdroje, třeba vašeho webu:

```

1 <amp-state id="myRemoteState" src="https://example.com/data.
   json"></amp-state>

```

Nad strukturou dat udržujících stav lze také samozřejmě provádět nějaké operace:“ [2]

- AMP.setState() – Nastavení stavu. Slouží k aktualizaci hodnot definované stavové proměnné. Může se ale použít také k aktualizaci složitější struktury dat zapsaných v JSON formátu. [2][34]
- AMP.pushState() – Funguje velice podobně jako AMP.setState() s tím rozdílem, že AMP.pushState() přidá ještě záznam do historie prohlížení stránek. Takže pokud se uživatel posune pomocí navigace zpět, obnoví se předchozí stav stavové proměnné. [2][34]

- AMP.setState() – Slouží k tisku aktuálního stavu do vývojářské konzole. K tomu je však nutné, aby stránka běžela ve vývojářském módu. Musí se tedy přidat #development=1 do URL adresy.[2][34]

amp-selector

Tato komponenta umožňuje přidat do AMP seznam možností, ze kterých může uživatel vybrat jednu nebo více. Obsah komponenty se neomezuje pouze na text, ale může obsahovat jakékoliv HTML prvky nebo AMP komponenty, například <amp-carousel>, <amp-img> atd. Tuto komponentu je možné použít pro libovolné seznamy, kde je potřeba dynamicky změnit obsah po aktivaci položky seznamu, například záložkové navigace, stránka produktu apod. Pokud se nachází <amp-selector> uvnitř formuláře a obsahuje atribut name, chová se komponenta jako radio button nebo checkbox v závislosti na tom, zda je u komponenty atribut multiple. [2] [36]

```

1 <amp-carousel on="slideChange:ampSelector.toggle(index=event
  .index, value=true)"
2   id="imageSlides"
3   layout="responsive"
4   width="400"
5   height="240"
6   type="slides">
7 <amp-img src="image.jpg"
8   width="400"
9   height="240"
10  layout="responsive"
11  alt="popisek"
12 ></amp-img>
13  ...
14 </amp-carousel>
15 <amp-selector id="ampSelector">
16 <amp-carousel layout="fixed-height" height="78">

```



```
17 <amp-img selected
18   option="0"
19   on="tap:imageSlides.goToSlide(index=0)"
20   role="button"
21   tabindex="1"
22   src="image.jpg"
23   width="120"
24   height="72"
25   layout="fixed"
26   alt="popisek - mini"
27 ></amp-img>
28 ...
29 </amp-carousel>
30 </amp-selector>
```

Příklad 36: Ukázka komponenty <amp-selector>

Příklad je ilustrace toho, jak je možné vytvořit obrázkovou galerii pomocí karuselu a pod galerii vložit miniatury obrázků z galerie. Obrázky v galerii se pak mohou měnit i po kliknutí na jejich miniatury.

Možné atributy:

- `disabled` – Pokud se tento atribut nachází přímo v <amp-selector>, deaktivuje se díky němu celá komponenta. Pokud se nachází u prvku, který má definován atribut `option`, je deaktivována pouze tato možnost.
- `form` – Definuje, k jakému formuláři komponenta patří.
- `multiple` – Umožňuje výběr více možností najednou.
- `name` – Určuje název komponenty.
- `keyboard-select-mode` – Určuje jak a jestli bude možné ovládat komponentu pomocí klávesnice.

- `option` – Určuje, zda se jedná o možnost, kterou je možné vybrat. Tento atribut je nutné přidat ke každému prvku uvnitř komponenty.
- `selected` – Určuje, která možnost bude předvybrána při načtení stránky.

Další dynamické komponenty:

- `amp-access` – Umožňuje implementaci paywall a podporu předplatného, například po určitém množství článků zobrazených zdarma.
- `amp-access-laterpay` – Paywall implementovaný pomocí mikrotransakční služby LaterPay.
- `amp-access-pool` – Zobrazení Pool paywall.
- `amp-access-scroll` – Umožňuje identifikovat členy scroll.com a poskytnout jim prostředí bez reklam.
- `amp-autocomplete` – Při zadávání do vstupního pole navrhuje dokončené výsledky odpovídající vstupu uživatele.
- `amp-by-side-content` – Umožňuje vložení dynamického obsahu pomocí služby BySide.
- `amp-consent` – Umožňuje do stránky implementovat souhlas uživatele s podmínkami webu, například okno s použitím cookies.
- `amp-data-picker` – Pomocí této komponenty lze do stránky vložit pěkně udělaný kalendář, ze kterého může uživatel vybrat datum. Je možné ji použít například pro rezervační web.
- `amp-form` – Umožňuje do AMP stránky vkládat formuláře.
- `amp-geo` – Tato komponenta poskytuje informace o tom, v jaké zemi se zrovna uživatel nachází.
- `amp-gist` – Vytvoří iframe a vloží do stránky GitHub Gist.

- `amp-google-document-embed` – Zobrazí dokument podporovaný Diskem Google.
- `amp-inputmask` – Pomocí této komponenty je možné vložit podmínku pro vstup ve formulářích. Je například možné psát pouze čísla.
- `amp-install-serviceworker` – Umožňuje nainstalovat ServiceWorker pro danou stránku.
- `amp-layout` – Slouží k definování layoutu na jakýkoliv prvek.
- `amp-link-rewriter` – Pomocí této komponenty mohou majitelé stránek přepsat URL adresu na základě konfigurovatelného vzoru.
- `amp-list` – Dynamicky stahuje data a vytváří položky seznamu podle šablony. Může se použít na seznamy, které se mění v čase, pro aktuální ceny produktů, personalizovaný seznam podobných článků atd.
- `amp-live-list` – Seznam položek aktualizovaný v reálném čase.
- `amp-mustache` – Zobrazování obsahu pomocí šablon Mustache.
- `amp-next-page` – Umožňuje nekonečné rolování stránky.
- `amp-recaptcha-input` – Umožňuje připojit token reCAPTCHA v3 k odeslání formulářů AMP.
- `amp-script` – Více o této komponentě v kapitole Amp-script.
- `amp-smartlinks` – Pomocí této komponenty je možné na stránce spustit Narrativ's Linkmate proces.
- `amp-user-notification` – Pomocí této komponenty je možné přidat do stránky oznámení pro uživatele. AMP radí nekombinovat tuto komponentu s `<amp-consent>`.
- `amp-video-docking` – Umožňuje u videí použít atribut dock.

- amp-web-push – Umožňuje uživateli přihlásit se k push notifikacím ze stránky.

9.4 Prezentační komponenty

V této kategorii se nacházejí všechny možné komponenty. Od těch pro animování až po komponentu pro vložení matematického vzorce.

- amp-animation – Pomocí této komponenty je možné definovat a zobrazit animaci na stránce AMP.
- amp-date-countdown – Tato komponenta umožňuje vložení odpočtu času do zadaného data.
- amp-date-display – Umožňuje zobrazení času v lokálním formátu.
- amp-dynamic-css-classes – Tato komponenta přidá do stránky třídu, díky které je možné zjistit, odkud se uživatel na stránku dostal a podle toho upravit stránku. Například třída amp-viewer oznamuje, že prohlížení stránky probíhá v AMP Vieweru.
- amp-fit-text – Umožňuje zvětšit nebo zmenšit velikost písma tak, aby se vešlo do rodičovského prvku.
- amp-mathml – Zobrazí matematický vzorec.
- amp-pan-zoom – Tato komponenta poskytuje možnost přiblížení nebo posouvání libovolného obsahu.
- amp-timeago – Zobrazení uběhnutého času od zadaného data.

9.5 Komponenty pro reklamu a analytiku

Tyto komponenty slouží k vložení a ovládní reklamních bannerů na AMP stránce. Reklama je důležitým zdrojem příjmů pro většinu obsahových webů, proto AMP přišlo s komponentami pro reklamu a analytiku, které nejsou pomalé ani rušivé.

- amp-ad – Pomocí této komponenty probíhá vložení reklamy do AMP stránky.
- amp-ad-exit – Poskytuje konfigurovatelné chování pro opuštění reklamní plochy.
- amp-auto-ads – Dynamicky vkládá reklamy na určená místa ve stránce. Reklamu pak není třeba definovat komponentou <amp-ad>.
- amp-sticky-ad – Zobrazení reklamy na fixní pozici v dolní části viewportu.
- amp-analytics – Tato komponenta umožňuje jednotný způsob vložení analytických skriptů, například od Google Analytics.
- amp-call-tracking – Poskytuje možnost měření četnosti volání na určité telefonní číslo.
- amp-pixel – Komponenta pro jednoduché měření četnosti zobrazení stránek.

9.6 Komponenty pro sociální sítě

V této poslední kategorii AMP komponent se AMP zaměřuje na vkládání příspěvků, komentářů atd. ze sociálních sítí. Nechybí ani možnost vložení tlačítek pro sdílení webové stránky na různých sociálních sítích.

amp-social-share

Pomocí této komponenty je možné do webové stránky vložit předpřipravená tlačítka pro sdílení na různých sociálních sítích od e-mailu po Facebook, Twitter atd. Možné je i vložit tlačítko pro sdílení na sociální síť, pro které tlačítka připravena nejsou.

```

1 <amp-social-share
2   type="email"
3   width="44"
4   height="44">
5 </amp-social-share>
6 <amp-social-share
7   type="twitter"
8   width="44"
9   height="44">
10 </amp-social-share>

```

Příklad 37: Ukázka komponenty <amp-social-share>

Možné atributy:

- **type** – Tento atribut je povinný pro připravená i nepřipravená tlačítka. Pomocí něj se vybírá druh sociální sítě, na kterou se má stránka sdílet. V době psaní této bakalářské práce jsou možné hodnoty system, facebook, email, linkedin, pinterest, tumblr, twitter, whatsapp, line a sms. [37]
- **data-target** – Určuje, kde se má otevřít stránka pro sdílení. Výchozí hodnota je `_blank` pro všechny typy kromě sms a email v systému iOS, kde je hodnota `_top`. [37]
- **data-share-endpoint** – Tento atribut je povinný pro všechna nepřipravená tlačítka pro sdílení a slouží pro zadání koncového bodu sdílení. [37]

- `data-param-*` – Možné upravení parametrů pro danou sociální síť. Pro každý typ jsou parametry různé. [37]
- `aria-label` – Pokud není tento atribut zadán, jeho výchozí hodnota je `Share by type`, kde `type` je nahrazeno hodnotou v atributu `type`. [37]

amp-instagram

Tato komponenta slouží pro vložení příspěvku z Instagramu do AMP stránky.

```

1 <amp-instagram
2   data-shortcode="B2iqNJlQaj"
3   width="1"
4   height="1"
5   layout="responsive">
6 </amp-instagram>
```

Příklad 38: Ukázka komponenty `<amp-instagram>`

Možné atributy:

- `data-shortcode` – Jedná se o odkaz na daný instagramový příspěvek. Je možné ho najít na konci URL adresy u každého příspěvku.
- `data-captioned` – Pomocí tohoto atributu se objeví u příspěvku i jeho titulek.

Další komponenty pro sociální sítě:

- `amp-addthis` – Umožňuje vložení upravitelného tlačítka pro sdílení na sociálních sítích přes službu AddThis.
- `amp-beopinion` – Umožňuje vložení příspěvku ze služby beOpinion.
- `amp-facebook-comments` – Umožňuje vložení facebookových komentářů.
- `amp-facebook-like` – Vložení tlačítka `To se mi líbí` z Facebooku.

- amp-facebook – Vložení příspěvku z Facebooku.
- amp-gfycat – Vložení příspěvku ze služby GfyCat.
- amp-iframe – Vložení příspěvku ze stránky iframely.com.
- amp-pinterest – Vložení příspěvku ze sociální sítě Pinterest. Pomocí této komponenty je možné vložit například i ukládací tlačítko na Pinterest.
- amp-reddit – Vložení komentáře nebo příspěvku ze sociální sítě Reddit.
- amp-riddle-quiz – Vložení dotazníku, ankety, průzkumu atd. ze stránky riddle.com.
- amp-twitter – Vložení příspěvku nebo momentu ze sociální sítě Twitter.
- amp-vine – Vložení videa ze služby Vine.
- amp-vk – Vložení příspěvku nebo ankety ze sociální sítě VK.

10 Výhody a nevýhody AMP

10.1 Výhody AMP

Jak už bylo mnohokrát zmíněno v této práci, hlavní a největší výhodou AMP technologie je rychlost. Díky uložení AMP stránky do AMP Cache mohou uživatelé AMP stránku zobrazit rychle po celém světě. AMP se zaměřuje na rychlost právě proto, aby nedocházelo k dlouhému načítání obsahu a uživatele nic neodrazovalo od toho, aby na stránce zůstal. Je také více pravděpodobné, že si uživatelé něco koupí, pokud vědí, že proces nákupu bude rychlý a bezproblémový.

Další výhodou je zvýraznění webu ve výsledcích vyhledávání na Google. Samo o sobě AMP technologie nezaručuje lepší pozici ve výsledcích vyhledávání a vyšší návštěvnost, AMP webové stránky jsou však ve výsledcích vyhledávání označeny logem AMP a je pravděpodobné, že tyto stránky uživatelé navštíví. Díky tomu se zvyšuje míra prokliku na stránku, která tak může stoupat ve výsledcích vyhledávání. Všechny tyto výhody mohou mít i pozitivní dopad na SEO.

10.2 Nevýhody AMP

Jako u každé technologie se i u AMP najde několik nevýhod. Mezi největší pravděpodobně patří složitější implementace AMP na již stávající webovou stránku. Pro redakční systémy – jako kupříkladu WordPress, Wix atd – existují různé pluginy, které mají implementaci AMP ulehčit, ale ani tak se proces neobejde zcela bez úprav.

Další nevýhodou je správa dvou verzí webů. Pokud je totiž AMP určena pro prohlížení na mobilních zařízeních, je nutné spravovat obě verze zvlášť, což může být obzvlášť náročné u komplexnějších webů typu e-shop. Je samozřejmě možné vytvořit pouze jednu responzivní AMP verzi webu, a tím tuto nevýhodu ignorovat.

Stále se setkáváme s kritikou, že AMP je pouze pro mobilní zařízení. Toto tvrzení je již dávno vyvráceno a pomocí AMP je možné vytvářet plnohodnotné responzivní webové stránky. Je ovšem pravda, že z výhod AMP je možné čerpat pouze na mobilních zařízeních.

K dalším nevýhodám patří to, že pokud je stránka zobrazena přes AMP Viewer, URL stránky je jiná než skutečná adresa stránky – jak bylo zmíněno v kapitole AMP Viewer. To může mít negativní dopad například pro budování značky.

Další kritika AMP směřuje k tomu, že se v něm nachází málo komponent. Konkrétně komponenty pro vkládání multimediálního obsahu třetích stran působí uzavřeně a exkluzivně pouze pro velké dodavatele. AMP komponent však stále přibývá a snad i časem dosáhne AMP univerzálnosti pro vkládání obsahu třetích stran.

Poslední nevýhody se týkají analytiky. V AMP je těžší získávat data o uživateli. Ačkoliv je do AMP možné vložit Google Analytics, přes který lze sledovat návštěvnost nebo například míru kliků na daný odkaz, další sledování je velice omezené. Navíc, pokud si uživatel prohlídí stránku z AMP Cache, není možné získávat informace ze serverových dat, takže celkově majitel webu získá menší přehled o uživatelském chování.

11 Praktická část

11.1 Tvorba webové stránky

V první části praktické části této bakalářské práce jsem vytvořil webovou stránku pomocí technologie AMP. Stránka je AMP validní a plně responzivní. Na stránce jsou demonstrovány možnosti této technologie spolu s AMP komponentami a ukázkami jejich kódů.

Menu webové stránky je dělané přes klasické hamburgerové menu v pravém horním rohu stránky, skrze které je možné dostat se do dalších částí stránky. V patičce každé stránky se nacházejí tlačítka, přes které je možné stránku sdílet na různých sociálních sítích.

Na úvodní stránce webu se nachází základní informace o AMP z teoretické části této práce. Sdělují tam, co technologie AMP je, jak vznikla a ukazují její základní kostru.

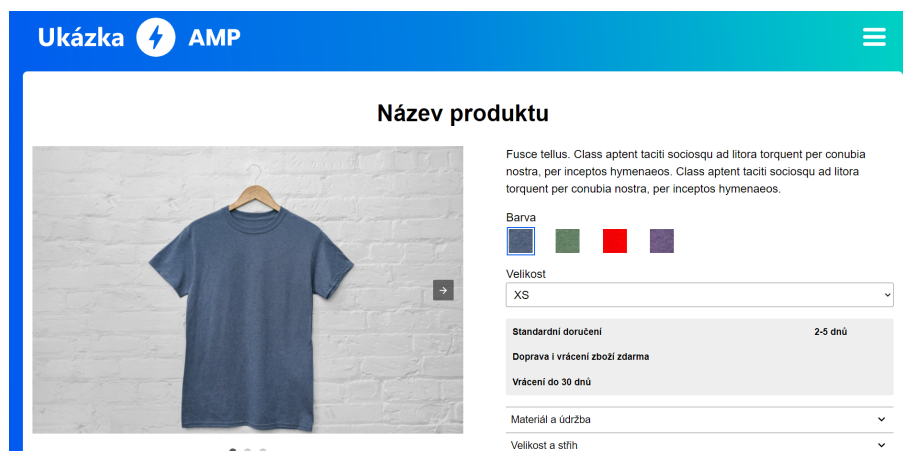
V dalších částech webové stránky – *Multimédia*, *Sociální sítě* a *Ostatní* – ukazují jednotlivé komponenty AMP. Jak vypadají implementované do stránky a jejich možnosti spolu s ukázkami kódu k jednotlivým komponentám.

V poslední části zvané *E-shop* ilustruji, jak by mohly vypadat stránky produktu vytvořené pomocí AMP.

Pro tvorbu webové stránky jsem si zvolil vývojové prostředí Visual Studio Code od společnosti Microsoft. Vybral jsem si jej proto, že už s ním mám jisté zkušenosti a pro tvorbu webových stránek mi vyhovuje. Navíc je možné do tohoto prostředí stáhnout rozšíření pro tvorbu AMP stránek.

Adresa mé webové stránky je: <https://ukazka-amp.cz/>

11.1.1 Ukázka webové stránky



Obrázek 11: Zobrazení webové stránky na počítači



Obrázek 12: Zobrazení webové stránky na mobilním zařízení

Jak už bylo výše zmíněno, menu stránky je dělané přes klasické hamburgerové menu, které jsem vytvořil pomocí amp komponenty `<amp-sidebar>`. Pomocí této komponenty a atributu `on` jsem jednoduše docílil toho, že se menu objeví při kliknutí na *hamburger* a zmizí při kliknutí na křížek v menu nebo při kliknutí mimo menu.

```
1 <amp;side;bar id="sidebar" class="sidebar" layout="nodisplay"
   side="right">
2   <div class="menu" role="button" tabindex="0" on="tap:
     sidebar.toggle"><svg ...></svg></div>
3   <div class="menu-text">Menu</div>
4     <ul class="menu-items">
5       <li><a href=".">Uvod do AMP</a></li>
6       <li><a href="media.html">Multimedia</a></li>
7       <li><a href="social.html">Socialni site</a></li>
8       <li><a href="other.html">Ostatni</a></li>
9       <li>
10        <amp;accordion class="dropdown">
11          <section>
12            <header>E-shop</header>
13            <ul class="menu-items">
14              <li><a href="computer.html">Pocitac</a>
15                </li>
16              <li><a href="clothes.html">Obleceni</a>
17                </li>
18            </ul>
19          </section>
20        </amp;accordion>
21      </li>
22    </ul>
23 </amp;side;bar>
```

Příklad 39: Kód menu použitého ve stránce

11.2 Měření rychlosti webů

V druhém oddílu praktické části této bakalářské práce jsem provedl průzkum, v němž jsem porovnával rozdíl v rychlosti AMP a běžných verzí webových stránek u třech českých a třech zahraničních webů, které používají obě verze. U webů jsem měřil tyto hodnoty:

- TTFB (Time To First Byte) – Okamžik, kdy prohlížeč stáhne první bajt z HTML.
- Index rychlosti (Speed Index) – Moment, kdy je hotové vykreslení stránky nad zlomem obrazovky.
- První vykreslení obsahu (First Contentful Paint – FCP) – Okamžik, kdy dojde k vykreslení prvního textu nebo obrázku.
- Největší vykreslení obsahu (Largest Contentful Paint – LCP) – Okamžik, kdy dojde k vykreslení největšího prvku na stránce (obrázku nebo text-boxu), který je viditelný ve viewportu.
- Úplné načtení (Fully Loaded) – Moment, kdy ustala síťová aktivita.

Jako první jsem vždy měřil původní stránku na hostingu, poté AMP stránku na hostingu, následovala AMP stránka umístěná v AMP Cache a jako poslední AMP stránka zobrazená přes AMP Viewer.

Všechny testy proběhly nástrojem webpagetest.org, při emulaci mobilu Samsung Galaxy S8 a při rychlosti připojení 3GFast.

Reflex

Jako první stránku pro měření jsem si vybral článek ze stránky reflex.cz.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	2,5 s	17,8 s	5,2 s	20,8 s	24,8 s
Hosting/AMP	1,3 s	4,4 s	3,6 s	3,6 s	9,8 s
AMP Cache	0,7 s	3,9 s	2,6 s	2,6 s	8,6 s
AMP Viewer	0,8 s	4,8 s	0,9 s	0,9 s	9,7 s

Tabulka 5: Výsledky měření článku ze stránky reflex.cz

Seznam Zprávy

Dále jsem si vybral také zpravodajský web, který používá AMP, a to článek ze stránky seznamzpravy.cz.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	1,3 s	3,9 s	2,3 s	7,1 s	15,6 s
Hosting/AMP	0,9 s	3,7 s	3,3 s	4,6 s	7,4 s
AMP Cache	0,8 s	3,1 s	2,6 s	4,0 s	6,7 s
AMP Viewer	0,9 s	3,4 s	0,9 s	0,9 s	7,4 s

Tabulka 6: Výsledky měření článku ze stránky seznamzpravy.cz

ProŽeny

Jako další jsem si vybral článek ze stránky prozeny.cz.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	1,0 s	4,9 s	1,9 s	7,4 s	20,8 s
Hosting/AMP	1,0 s	4,3 s	3,1 s	4,7 s	7,6 s
AMP Cache	0,7 s	2,7 s	2,1 s	2,8 s	6,4 s
AMP Viewer	0,8 s	3,6 s	0,9 s	0,9 s	7,2 s

Tabulka 7: Výsledky měření článku ze stránky prozeny.cz

CNN

První zahraniční web – jeden z největších zpravodajských webů vůbec.

Článek ze stránky cnn.com.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	0,7 s	15,7 s	7,8 s	12,5 s	26,1 s
Hosting/AMP	0,7 s	4,3 s	3,1 s	5,4 s	10,7 s
AMP Cache	0,7 s	3,3 s	3,1 s	3,8 s	12,0 s
AMP Viewer	0,8 s	4,0 s	0,9 s	0,9 s	11,8 s

Tabulka 8: Výsledky měření článku ze stránky cnn.com

Greenweez

Jako další testovací web jsem zvolil stránku produktu Greenweez, což je francouzský obchod s biopotravinami.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	0,9 s	4,7 s	3,0 s	3,7 s	10,7 s
Hosting/AMP	0,8 s	4,1 s	3,6 s	4,8 s	7,8 s
AMP Cache	0,7 s	2,8 s	2,1 s	3,6 s	7,9 s
AMP Viewer	0,8 s	3,5 s	0,9 s	0,9 s	7,9 s

Tabulka 9: Výsledky měření produktu ze stránky greenweez.com

Globo

Poslední test proběhl na obrovském zpravodajském webu pro Latinskou Ameriku – článek ze stránky globo.com.

	TTFB	Index rychlosti	První vykreslení obsahu	Největší vykreslení obsahu	Úplné načtení
Hosting	1,3 s	5,3 s	3,3 s	3,7 s	37,3 s
Hosting/AMP	1,3 s	3,1 s	3,1 s	3,1 s	20,5 s
AMP Cache	1,5 s	1,9 s	1,6 s	2,0 s	20,3 s
AMP Viewer	0,9 s	2,4 s	1,0 s	1,0 s	21,0 s

Tabulka 10: Výsledky měření článku ze stránky globo.com

Jak je ze všech měření patrné, AMP je rychlejší než běžný web. Ačkoliv u některých hodnot (například u TTFB) nemusí být na první pohled tak velký rozdíl, u vykreslování obsahu mezi původní stránkou na hostingu a AMP stránkou zobrazenou přes AMP Viewer je diference enormní. U všech hodnot je vždy největší rozdíl mezi původní stránkou na hostingu a AMP stránkou

zobrazenou přes AMP Viewer. Pokud je ovšem AMP stránka vytvořena dobře, uživatel se vždy přes výsledky vyhledávání dostane na stránku zobrazenou přes AMP Viewer. A jak toto měření dokládá, stránky, které se rozhodly využívat AMP, rozhodně neudělaly chybu.

12 Závěr

Bakalářská práce se zabývala technologií Accelerated Mobile Pages na vytváření webových stránek.

V teoretické části jsem představil samotnou technologii AMP, popsal jsem její jednotlivé části, rozdíly oproti běžnému HTML, popsal jsem AMP komponenty a v závěru teoretické části rovněž výhody a nevýhody spojené s webovými stránkami vytvořené touto technologií.

V praktické části jsem pomocí technologie AMP vytvořil responzivní webovou stránku, na níž dokládám možnosti využití této technologie. Názorně v ní ilustruji užití AMP komponent spolu s ukázkami jejich kódů. Zároveň jsem provedl měření rychlosti webových stránek, během kterého jsem porovnával weby vytvořené pomocí AMP a webové stránky vytvořené pomocí běžného HTML u webů, které používají obě verze.

Technologie AMP jde neustále kupředu. Vyvíjí se a vylepšuje, aby splnila požadavky tvůrců webových stránek. Je složité predikovat, jak moc se AMP změní v horizontu několika let. Je však nutné dodat, že v současné době výhody AMP převažují nad jejími nevýhodami. V poslední době zaznamenávám stále častěji logo AMP vedle adresy webu ve výsledcích vyhledávání. Je tedy možné, že v budoucnu k implementaci AMP přistoupí stále více webů.

Seznam použité literatury a zdrojů

- [1] GALFI, Rudy. The benefits of AMP. Google News [online]. 2020 [cit. 2021-03-25]. Dostupné z: <https://blog.google/products/news/benefits-amp/>
- [2] MICHÁLEK, Martin a Robin POKORNÝ. Vzhůru do AMP [online]. Praha: Martin Michálek, 2019 [cit. 2021-03-25]. ISBN 978-80-88253-04-4
- [3] AMP Project's new governance model now in effect [online]. 2018 [cit. 2021-03-25]. Dostupné z: <https://blog.amp.dev/2018/11/30/amp-projects-new-governance-model-now-in-effect/>
- [4] BRENNER, David. Co je AMP a jak může pomoci vašemu webu? Digichef [online]. 2019 [cit. 2021-03-25]. Dostupné z: <https://digichef.cz/co-je-amp-a-jak-muze-pomoci-vasemu-webu>
- [5] LARA, Bryan. What is AMP and why you should use it? [online]. 2017 [cit. 2021-03-25]. Dostupné z: <https://swapps.com/blog/what-its-amp-and-why-you-should-use-it/>
- [6] Objekty. Jakpsatweb [online]. [cit. 2021-6-15]. Dostupné z: <https://www.jakpsatweb.cz/html/objekty.html>
- [7] Rámy. Jakpsatweb [online]. [cit. 2021-6-15]. Dostupné z: <https://www.jakpsatweb.cz/html/ramy.html>
- [8] Lazy loading - odložené načítání obrázků. Jakpsatweb [online]. 2020 [cit. 2021-03-25]. Dostupné z: <https://www.jakpsatweb.cz/clanky/lazy-loading.html>
- [9] HTML Atributy [online]. [cit. 2021-03-25]. Dostupné z: <https://www.klikzone.cz/HTML-navod/HTML-atributy.php>

- [10] AMP HTML Specification. amp.dev [online]. [cit. 2021-03-25]. Dostupné z: <https://amp.dev/documentation/guides-and-tutorials/learn/spec/amphtml/>
- [11] Common element attributes. amp.dev [online]. [cit. 2021-7-1]. Dostupné z: https://amp.dev/documentation/guides-and-tutorials/learn/common_attributes/
- [12] Responsive images with srcset, sizes & heights. amp.dev [online]. [cit. 2021-7-1]. Dostupné z: https://amp.dev/documentation/guides-and-tutorials/develop/style_and_layout/art_direction/
- [13] ANDREW WATTERSON. Making Your Wait a Little More Great: New Loading Indicators in AMP [online]. [cit. 20-04-2021]. Dostupné z: <https://blog.amp.dev/2019/08/26/making-your-wait-a-little-more-great-new-loading-indicators-in-amp/>
- [14] GEOFF, Graham. The At-Rules of CSS [online]. 2015 [cit. 2021-03-25]. Dostupné z: <https://css-tricks.com/the-at-rules-of-css/>
- [15] Amp-script. amp.dev [online]. [cit. 2021-04-02]. Dostupné z: <https://amp.dev/documentation/components/amp-script/>
- [16] Use custom JavaScript in AMP pages. amp.dev [online]. [cit. 2021-04-02]. Dostupné z: <https://amp.dev/documentation/guides-and-tutorials/develop/custom-javascript/>
- [17] MICHÁLEK, Martin. JavaScript v AMP a inovace jménem amp-script [online]. 2019 [cit. 2021-04-02]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/amp-javascript>
- [18] Actions and events. amp.dev [online]. [cit. 2021-04-02]. Dostupné z: <https://amp.dev/documentation/guides-and-tutorials/learn/amp-actions-and-events/>

- [19] Google AMP Cache [online]. 2019 [cit. 2021-04-12]. Dostupné z: <https://developers.google.com/amp/cache/overview>
- [20] UBL, Malte. Why AMP HTML does not take full advantage of the preload scanner. Medium.com [online]. 2015 [cit. 2021-04-16]. Dostupné z: <https://medium.com/@cramforce/why-amp-html-does-not-take-full-advantage-of-the-preload-scanner-7e7f788aa94e>
- [21] BAKAUS, Paul. Why AMP Caches exist. Medium.com [online]. 2017 [cit. 2021-04-16]. Dostupné z: <https://medium.com/@pbakaus/why-amp-caches-exist-cd7938da2456>
- [22] Layout & media queries. amp.dev [online]. [cit. 2021-04-16]. Dostupné z: https://amp.dev/documentation/guides-and-tutorials/develop/style_and_layout/control_layout/
- [23] AMP HTML Components. Github.com [online]. [cit. 2021-5-14]. Dostupné z: <https://github.com/ampproject/amphtml/blob/main/spec/amp-html-components.md>
- [24] Amp-anim. amp.dev [online]. [cit. 2021-5-14]. Dostupné z: <https://amp.dev/documentation/components/amp-anim/>
- [25] MUZIKANT, Tomáš. Multimédia v HTML II. část [online]. 2020 [cit. 2021-5-14]. Dostupné z: <https://www.itnetwork.cz/html-css/html5/multimedia-v-html-ii-cast>
- [26] HTMLMediaElement.controlsList [online]. 2021 [cit. 2021-5-14]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/controlsList>
- [27] Amp-video. amp-dev [online]. [cit. 2021-5-14]. Dostupné z: <https://amp.dev/documentation/components/amp-video/>

- [28] Amp-youtube. amp.dev [online]. [cit. 2021-5-14]. Dostupné z: <https://amp.dev/documentation/components/amp-youtube/>
- [29] Amp-accordion. amp.dev [online]. [cit. 2021-5-26]. Dostupné z: <https://amp.dev/documentation/components/amp-accordion/>
- [30] Amp-carousel. amp.dev [online]. [cit. 2021-5-26]. Dostupné z: <https://amp.dev/documentation/components/amp-carousel/>
- [31] Amp-iframe. amp.dev [online]. [cit. 2021-5-26]. Dostupné z: <https://amp.dev/documentation/components/amp-iframe/>
- [32] Amp-image-slider. amp.dev [online]. [cit. 2021-5-26]. Dostupné z: <https://amp.dev/documentation/components/amp-image-slider/>
- [33] Amp-fx-collection. amp.dev [online]. [cit. 2021-5-26]. Dostupné z: <https://amp.dev/documentation/components/amp-fx-collection/>
- [34] Amp-bind. amp.dev [online]. [cit. 2021-6-4]. Dostupné z: <https://amp.dev/documentation/components/amp-bind/>
- [35] Google AMP - Data Binding. Tutorialspoint [online]. [cit. 2021-6-4]. Dostupné z: https://www.tutorialspoint.com/google_amp/google_amp_data_binding.htm
- [36] Amp-selector. amp.dev [online]. [cit. 2021-6-9]. Dostupné z: <https://amp.dev/documentation/components/amp-selector/>
- [37] Amp-social-share. amp.dev [online]. [cit. 2021-6-9]. Dostupné z: <https://amp.dev/documentation/components/amp-social-share/>

Seznam obrázků

1	"Loading indicators" v AMP [13]	23
2	Webová stránka zobrazena přes AMP Viewer	37
3	Předvykreslení	38
4	Ukázka layout="responsive"	39
5	Ukázka layout="fixed"	40
6	Ukázka layout="fixed-height"	40
7	Ukázka layout="fill"	41
8	Ukázka layout="intrinsic"	41
9	Ukázka layout="flex-item"	42
10	Validace pomocí Google Search Console	44
11	Zobrazení webové stránky na počítači	84
12	Zobrazení webové stránky na mobilním zařízení	84

Seznam tabulek

1	Zakázané tagy v AMP HTML [6][7]	18
2	Nahrazené tagy v AMP HTML	19
3	Povolená zavináčová pravidla v AMP	27
4	Seznam povolených dodavatelů písma	28
5	Výsledky měření článku ze stránky reflex.cz	87
6	Výsledky měření článku ze stránky seznamzpravy.cz	87
7	Výsledky měření článku ze stránky prozeny.cz	88
8	Výsledky měření článku ze stránky cnn.com	88
9	Výsledky měření produktu ze stránky greenweez.com	89
10	Výsledky měření článku ze stránky globo.com	89

Seznam příkladů

1	Základní kostra AMP HTML	15
2	AMP-boilerplate	17
3	Příklad atributu layout	20
4	Příklad atributu heights	21
5	Příklad atributu sizes	21
6	Příklad atributu media	22
7	Příklad atributu on	23
8	Příklad atributu noloading	24
9	Příklad atributu placeholder	24
10	Příklad atributu fallback	25
11	Přidání fontu pomocí @font-face	28
12	Příklad zavolání fontu v CSS	28
13	Ukázka tagu <link>	29
14	Ukázka sriptu pro komponentu <amp-image-lightbox>	30
15	Syntaxe atributu on	32
16	Ukázka atributu on	32
17	Ukázka více událostí	33
18	Ukázka více akcí	33
19	Ukázka prefetch značky	35
20	Ukázka komponenty <amp-img>	46
21	Ukázka komponenty <amp-anim>	46
22	Ukázka komponenty <amp-audio>	47
23	Ukázka komponenty <amp-video>	49
24	Ukázka komponenty <amp-youtube>	51
25	Ukázka komponenty <amp-dailymotion>	52
26	Ukázka komponenty <amp-soundcloud>	53
27	Ukázka komponenty <amp-megaphone>	54
28	Ukázka komponenty <amp-accordion>	57

29	Ukázka komponenty <code><amp-carousel></code>	59
30	Ukázka komponenty <code><amp-iframe></code>	61
31	Ukázka komponenty <code><amp-image-lightbox></code>	63
32	Ukázka komponenty <code><amp-image-slider></code>	64
33	Ukázka komponenty <code><amp-fx-collection></code>	66
34	Ukázka komponenty <code><amp-fx-flying-carpet></code>	69
35	Ukázka komponenty <code><amp-bind></code> [34]	70
36	Ukázka komponenty <code><amp-selector></code>	72
37	Ukázka komponenty <code><amp-social-share></code>	78
38	Ukázka komponenty <code><amp-instagram></code>	79
39	Kód menu použitého ve stránce	85

A Příloha

CD se zdrojovými kódy webové stránky a plné znění BP v PDF.

B Příloha

<https://ukazka-amp.cz/>