Responzivita a přístupnost

Responzivita

- **RESPONZIVNÍ WEBDESIGN:**
  - Přístup k tvorbě webových stránek, který zajišťuje optimální zobrazení a interakci napříč širokou škálou zařízení od mobilních telefonů po stolní počítače
  - **CÍL:**
    - Poskytnout optimální uživatelskou zkušenost bez ohledu na velikost obrazovky, platformu nebo orientaci zařízení
- **ROZLIŠUJEME 3 TYPY:**
  1. **Fluidní mřížka (Fluid Grid)**
  2. **Flexibilní obrázky a média**
  3. **Media Queries v CSS**

- **FLUIDNÍ MŘÍŽKY:**
  - Využití relativních jednotek (%) namísto pevných pixelů
  - Umožňuje prvkům přizpůsobit se velikosti obrazovky

Příklad CSS pro fluidní mřížku

```css
.container {
  width: 100%;
}

.column {
  float: left;
  /* Přizpůsobí se šířce kontejneru */
  width: 50%;
}
```

- **FLEXIBILNÍ OBRÁZKY A MÉDIA:**
  - Obrázky a média se přizpůsobují velikosti kontejneru
  - Zabraňuje přetečení obsahu mimo rodičovský prvek
  - **Udržuje poměr stran pomocí:**
    - height: auto

Použití vlastnosti max-width: 100%

```
img, video {
  max-width: 100%;
  height: auto;
}
```

- **MEDIA QUERIES:**
  - **Umožňují aplikovat různé styly podle vlastnosti zařízení jako:**
    - Výška, šířka prohlížeče
    - Orientace obrazovky
    - Rozlišení
  - **Syntaxe v CSS:**
    - @media (podmínka) { … }
    - Lze kombinovat více podmínek pomocí klíčových slov **and**, **or**, **not**
  - **Typy Media Queries:**
    - **Podle šířky obrazovky:**
      - min-width, max-width
    - **Podle výšky obrazovky:**
      - min-height, max-height

- **Orientace obrazovky:**
  - orientation: portrait nebo landscape
- **Rozlišení obrazovky:**
  - min-resolution, max-resolution

Příklad Media Query podle šířky

```
@media (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Příklad kombinace podmínek

```
@media (min-width: 600px)
and (orientation: landscape) {
  body {
    background-color: lightgreen;
  }
}
```

- **MOBILNÍ PRVNÍ PŘÍSTUP (MOBILE FIRST)**
  - Strategie, kdy se nejprve definují styly pro mobilní zařízení a poté se přidávají úpravy pro větší obrazovky
  - **Výhody:**
    - Lepší výkon na mobilních zařízeních
    - Jednodušší přidávání variant pro větší obrazovky

Příklad Mobile First

```
/* Základní styly pro mobily */
.container {
  display: flex;
  flex-direction: column;
}

/* Pro větší zařízení */
@media (min-width: 768px) {
  .container {
    flex-direction: row;
  }
}
```

- **BREAKPOINTS (PŘECHODOVÉ BODY)**
  - Hodnoty, při kterých se design přizpůsobuje nové velikosti obrazovky
  - Často se vychází z nejběžnějších šířek zařízení
  - **Přístup k volbě:**
    - Zařízení-based (podle konkrétních zařízení)
    - Content-based (podle obsahu a jeho potřeby změnit layout)

Příklad Mobile First

```
/* Základní styly pro mobily */
.container {
  display: flex;
  flex-direction: column;
}

/* Pro větší zařízení */
@media (min-width: 768px) {
  .container {
    flex-direction: row;
  }
}
```

- **PŘÍKLADY KOMBINACE MEDIA QUERIES:**

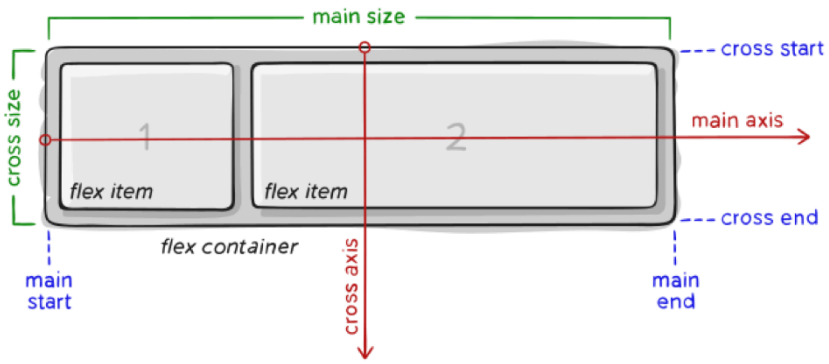
Kombinace podmínek

```
@media (min-width: 768px)
and (orientation: landscape) {
  /* Styly pro tablety na šířku */
}
```

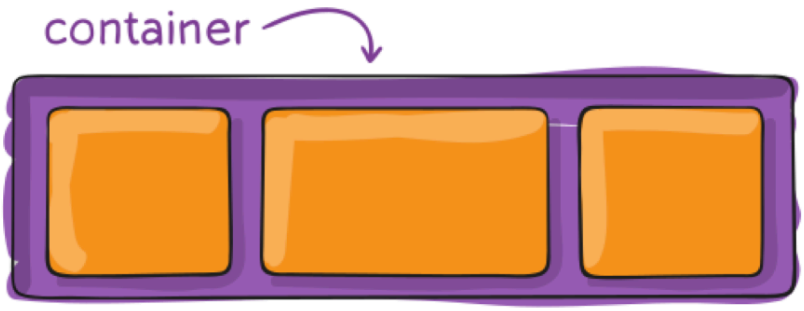
Použití not

```
@media not all and (min-width: 768px) {
  /* Styly pro zařízení menší než 768px */
}
```

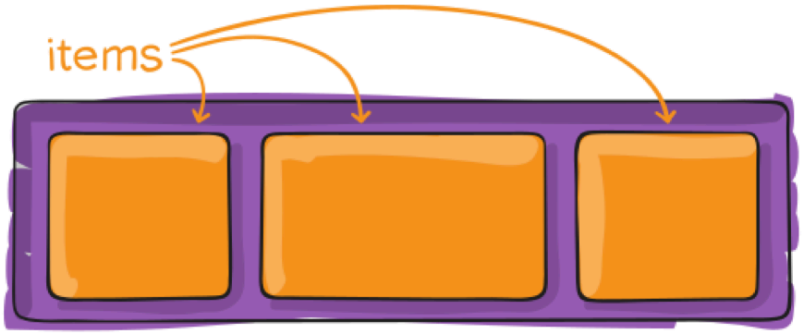
- **CSS Flexbox:**
  - Flexbox (Flexible box Layout) je CSS modul určený pro usnadnění uspořádání prvků v jednom směru (řádku nebo sloupci)
  - Umožňuje jednoduše zarovnávat, rozdělovat prostor a měnit pořadí prvků
  - Ideální pro tvorbu flexibilních a responzivních layoutů
  - **Osy flexboxu:**
    - Hlavní osa (Main Axis) a Křížová osa (Cross Axis)
    - flex-direction určuje směr hlavní osy
    - row (výchozí) nebo column



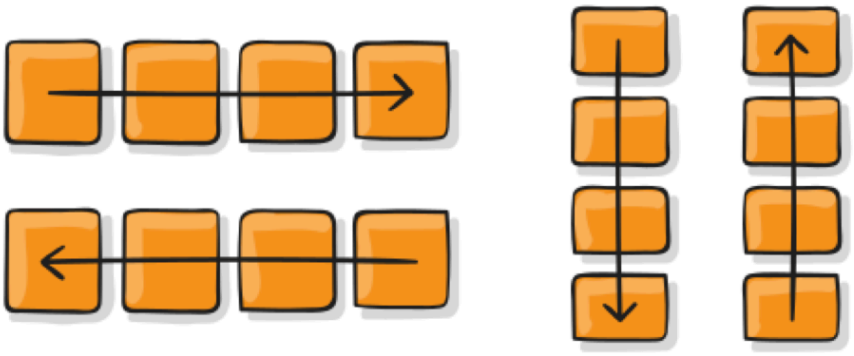
  - **Flexbox container:**
    - Rodičovský prvek definovaný pomocí display: flex;, který obsahuje flex položky



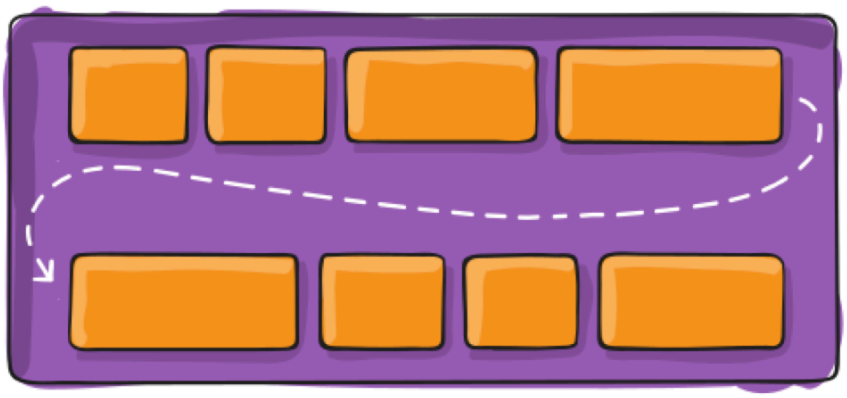
- **Flexbox items:**
  - Přímí potomci flex kontejneru, kteří jsou uspořádáni podle pravidel flexboxu



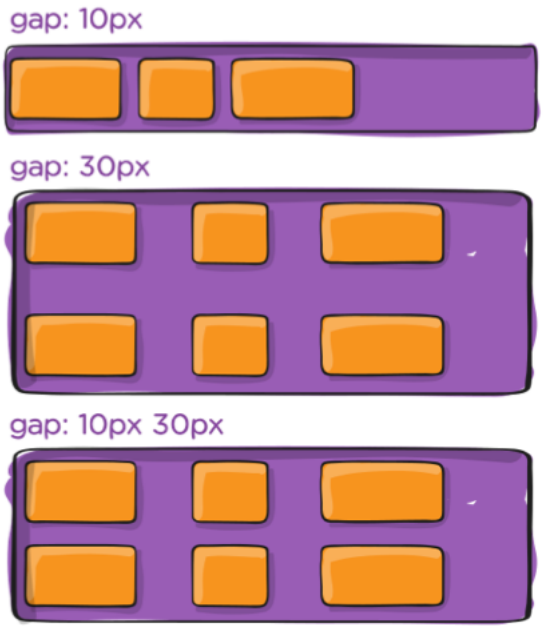
- **Flexbox direction:**
  - Určuje směr hlavní osy (řádek nebo sloupec) pro uspořádání položek (row, column)



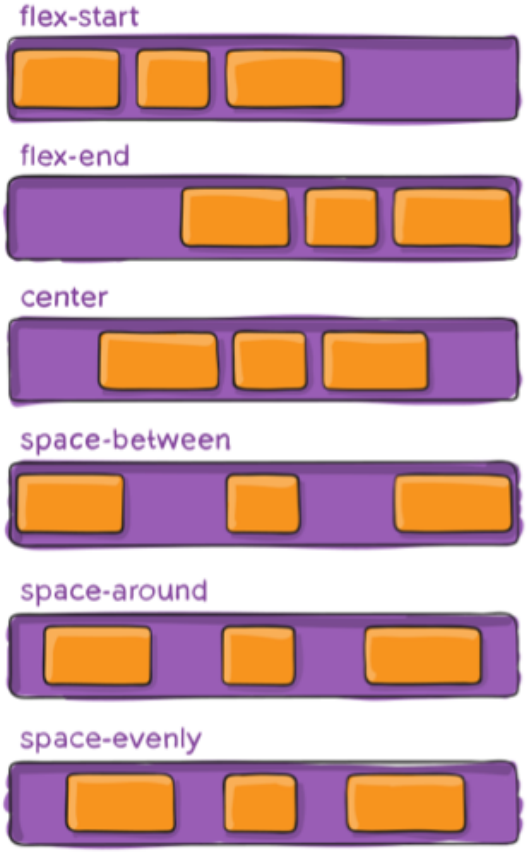
- **Flexbox wrap:**
  - Určuje, zda se položky zalamují na další řádky, pokud není dostatek místa (nowrap, wrap)



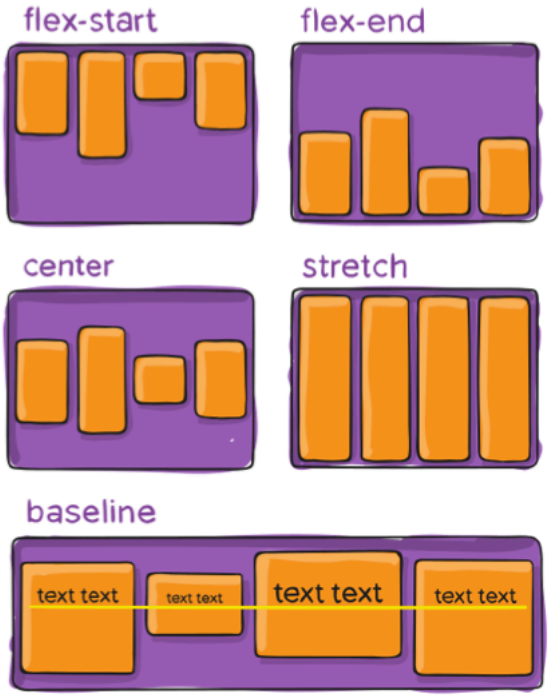
- **Flexbox gap:**
  - Nastavuje mezeru mezi flex box položkami, usnadňuje rozestupy bez použití marginů



- **Flexbox justify-content:**
  - zarovnává položky podél hlavní osy (flex-start, center, space-between)



- **Flexbox align-items:**
  - Zarovnává položky podél křížové osy (stretch, flex-start, center)



- **Příklad:**

HTML kód

index.html

```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
```

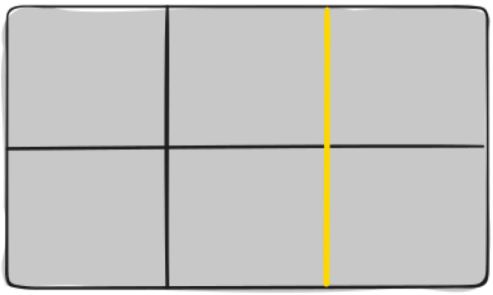
CSS kód

style.css

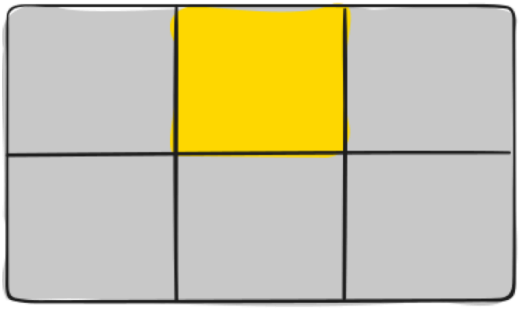
```
.container {
  display: flex;
  flex-direction: row;
  justify-content: space-around;
}

.item {
  background-color: lightcoral;
  padding: 20px;
}
```

1 2 3

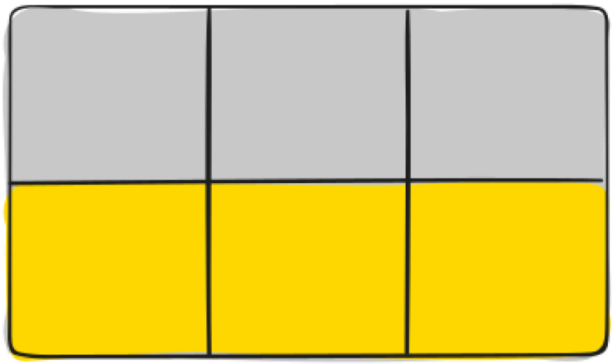
- **CSS Grid**
  - Je mřížka pro vytváření složitých layoutů na webových stránkách
  - Umožňuje uspořádat prvky do řádků a sloupců
  - Ideální pro komplexní rozvržení, která vyžadují přesné umístění prvků v obou osách
  - Definuje se pomocí display: grid | inline-grid
  - **Grid container:**
    - Je obsah mřížky, ve kterém se skládají prvky
  - **Grid item:**
    - Je přímý potom Grid containeru => potomci potomků už nejsou
  - **Grid line:**
    - Je dělicí čára prvků mřížky



  - **Grid cell:**
    - Je buňka mřížky



- **Grid Track:**
  - Je oblast mezi dvěma dělicími čárami (vertikálně i horizontálně)

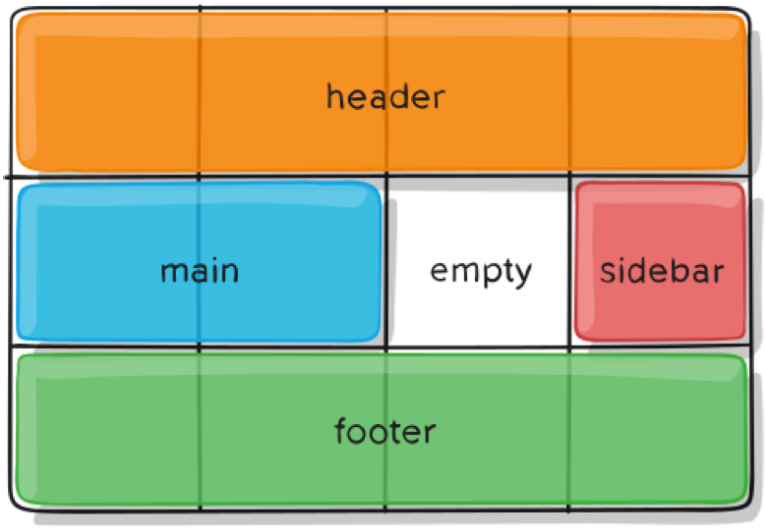


- **Grid Area:**
  - Je oblast mezi čtyřmi dělicími čárami (právě dvě vertikálně, právě dvě horizontálně)

- **Grid templates:**
  - Lze definovat, jak bude mřížka vypadat a jaké bude mít tagy

style.css

```
.item-a {
  grid-area: header;
}
...
.container {
  display: grid;
  grid-template-columns: 50px 50px 50px 50px;
  grid-template-rows: auto;
  grid-template-areas:
    "header header header header"
    "main main . sidebar"
    "footer footer footer footer";
}
```



- **Relativní jednotky v CSS**
  - Umožňují dynamicky přizpůsobený design podle velikosti okna, obrazovky nebo jiných prvků
  - **Zlepšují přístupnost** = uživatelé mohou zvětšit písmo ve svém prohlížeči a design se přizpůsobí
  - Umožňují responzivní design bez Media Queries
  - Usnadňují údržbu kódu
  - **Mezi nejčastější používané relativní jednotky patří:**
    - em, rem
    - vw, vh, vmin, vmax
  - **Jednotky em a rem:**
    - **em** je relativní k velikost písma rodičovského prvku
    - Pokud rodič má font-size: 16px, pak: 1em = 16px
    - **Použití:**
      - Umožňuje dědit velikosti písma a přizpůsobit ji hierarchii dokumentu
    - **rem** je relativní k velikosti písma kořenového prvku (html)
    - Výchozí velikosti písma pro html je obvykle 16px: 1rem = velikost písma kořene
    - **Použití:**
      - Určeno pro změny velikostí vzhledem k písmu (optická čitelnost)
      - Snadná změna globální velikosti písma
  - **Jednotky viewportu: vw, vh**
    - **vw** (viewport width): 1% šířky viewportu
    - **vh** (viewport height): 1% výšky viewportu
    - **viewport** = viditelná oblast webové stránky v prohlížeči
  - **Jednotky vmin a vmax**
    - **vmin:** Menší z hodnot vw a vh

    - **vmax:** Větší z hodnot vw a vh
    - Užitečné pro zachování poměru stran na různých zařízeních

- **Frameworky a knihovny**
    - Použití frameworks může urychlit vývoj, ale může také přidat režii
    - **Bootsrap:**
        - Populární CSS framework s předdefinovanými komponentami a grid systémem
    - **Tailwind CSS:**
        - Utility-first framework umožňující rychlý vývoj
    - **Material UI:**
        - Knihovna komponent založená na Material Designu od Google

---

Přístupnost

- Design a vývoj webu tak, aby jej mohly plnohodnotně používat i osoby se zdravotním postižením: ztáta sluchu, zraku, barvoslepost, pohybová postižení
- Uživatelé musí být schopni web vnímat, ovládat, rozumět mu a přistupovat k obsahu
- **Zahrnuje různé aspekty:**
    - Textové alternativ pro obrázky
    - Klávesová ovladatelnost
    - Srozumitelnost textu
    - Barevné kontrasty …….

- **Proč?:**
  - **15% populace má nějakou formu postižení**
  - **Stárnoucí populace**
  - **Dočasné a situační omezení**
  - **Obchodní a etické důvody**
- **Typy postižení:**
  - **Zrakové postižení**
  - **Sluchové postižení**
  - **Pohybové postižení**
  - **Kognitivní postižení a jiná**
- **Základní principy WCAG (P.O.U.R.)**
  - **Vnímatelnost (Perceivable):**
    - Informace a UI komponenty musí být vnímatelné
  - **Ovladatelnost (Operable):**
    - Prvky uživatelského rozhraní a navigace musí být ovladatelné
  - **Srozumitelnost (Understandable):**
    - Obsah i ovládání musí být srozumitelné (jasný a konzistentní jazyk, předvídatelné chování, instrukce a validace u formulářů)
  - **Robustnost (Robust):**
    - Obsah musí být dostatečně robustní, aby fungoval s různými uživatelskými agenty, včetně asistivních technologií (validní HTML, správně použití ARIA, kompatibilita napříč prohlížeči)

- **Technické základy:**
  - Používejte moderní **HTML5** elementy

ŠPATNĚ

```
<div onclick="submitForm()">
Odešli
</div>
```

DOBŘE

```
<button type="submit">
Odešli
</button>
```

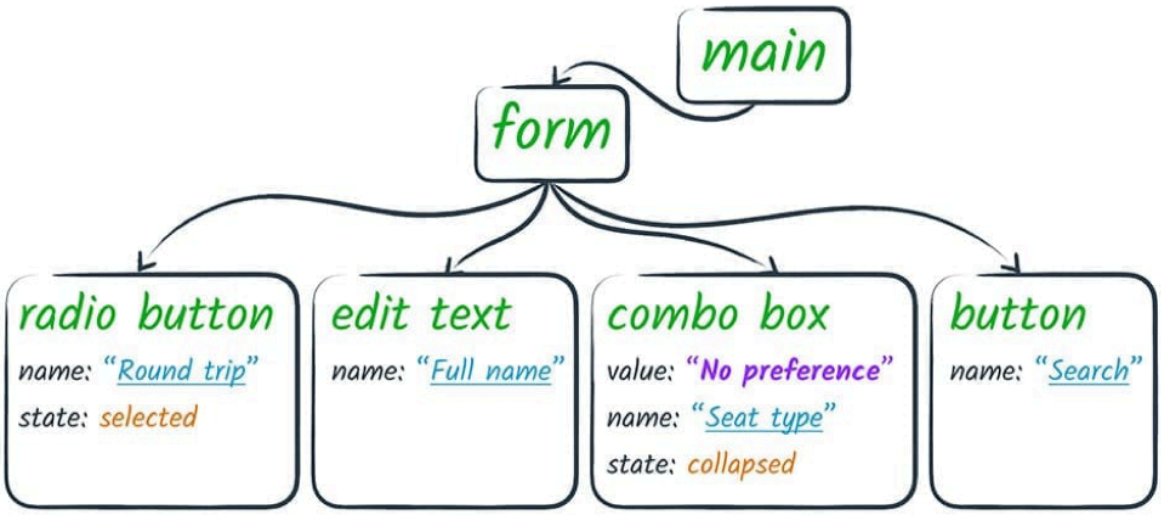
  - **WAI-ARIA:**
    - **ARIA** (Accessible Rich Internet Applications
      - Sada atributů role, aria-* pro doplnění významu u nestandardních komponent (dynamické widgety, JS aplikace)
    - **Příklad rolí:**
      - **role=„navigation"**
      - **role=„main“**
      - **role=„dialog“**
      - **role=„button“**
    - **Pravidla:**
      - Nepoužívat pokud můžeme použít HTML
      - Špatně použita ARIA může přístupnost zhoršit
  - **Struktura stránky a landmarky:**
    - **Landmarky:**
      - Orientační body stránky definované buď HTML5 elementy, nebo ARIA rolemi
      - Umožňují rychlé přeskočení do části obsahu
      - **Hlavní typy:**
        - **Header / Banner:** hlavička stránky | <header> = role=„banner“
        - **Navigace:** menu odkazů | <nav> = role=„navigation“

- **Main:** hlavní obsah stránky | <main> = role=„main“ (má být jen jeden)
- **Komplementární obsah:** doplňující obsah, sidebar | <aside> = role =„complementary“
- **Obsahové informace / patička:** <footer> = role=„contentinfo"
- **Formulářová oblast:** skupina formulářových prvků | role=„form“
- **Vyhledávání:** vyhledávací formulář | role=„search“

- **Přístupné formuláře:**
  - Každý prvek musí mít viditelný popisek (<label> spojený s <input> přes for/id)
  - Skupiny polí seskupit pod <fieldset> s <legend>
  - Chybové hlášky
  - Pokyny (instrukce, formát vstupu nebo příklady)

- **Ovládání z klávesnice:**
  - Celý web musí být ovladatelný pouze z klávesnice
  - Logická posloupnost zaměření
  - Viditelné zaměření (aktivní prvek musí být vizuálně zobrazen)
  - Bez pastí (žádný prvek nesmí zablokovat zaměření klávesou Tab
  - Správné prvky
  - Klávesové zkratky

- **Asistivní technologie**
  - **Screen reader:**
    - Čtečka obrazovky (převádí informace z obrazovky na hlas nebo braillský výstup
    - Interpretuje strom přístupnosti (accessibility tree)



Strom přístupnosti pro webovou stránku

    - Populární čtečky:
      - **JAWS (Windows, komerční)**
      - **NVDA (Windows, open-source)**
      - **VoiceOver (macOS/IOS, součást systému)**
      - **TalkBack (Android)**

- **Praktické příklady:**

Formulářový prvek bez popisku vs. s popiskem

ŠPATNĚ

```
<input type="text" placeholder="Jméno">
```

(uživatel pouze vidí světle šedý text uvnitř pole, čtečka tento placeholder může či nemusí přečíst jako popis pole)

DOBŘE

```
<label for="fname">Jméno:</label>
<input id="fname" type="text"
    placeholder="Celé jméno">
```

(čtečka čte: "Jméno, editační text, prázdné". Placeholder "Celé jméno"může přečíst jako doplňující informaci.)

Alternativní text obrázku

ŠPATNĚ

```
<img src="graf.png" alt="">
```

(chybí popis obrázku. Čtečka může oznámit "obrázek, graf.png"nebo ho úplně přeskočit)

DOBŘE

```
<img src="graf.png"
    alt="Graf vývoje prodejů
        za roky 2019-2024">
```

(čtečka přečte: "Graf vývoje prodejů za roky 2019–2024, obrázek")

Přeskočit navigaci

Bez odkazu pro přeskočení:

Uživatel "tabují"navigaci na stránce.

Tab 1: zaměření jde na první odkaz v menu (např. "Domů"). Čtečka čte **"Domů, odkaz"**.
Tab 2..N: prochází všechny položky horní navigace, než se dostane k obsahu.
⇒ Každá stránka vyžaduje proklikání celého menu znovu.

S odkazem "Přeskočit":

```
<a href="#main" class="skip-link">
Přeskočit na obsah
</a>
<nav>... menu ...</nav>
<main id="main"> ... </main>
```

Tab 1: fokus na skrytý odkaz "Přeskočit na obsah". Uživatel stiskne Enter.
Fokus přeskočí přímo na `<main>` (hlavní obsah). Čtečka čte začátek obsahu stránky.
⇒ Uživatel může hned číst obsah, aniž by musel projít celé menu.

- **Testování přístupnosti**

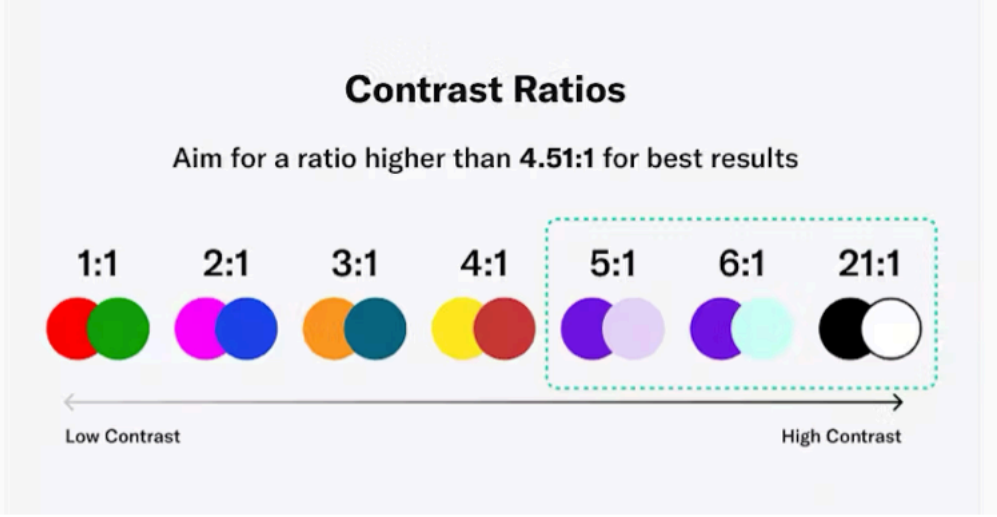
  - **Automatizované nástroje pro testování**

    - Validátory a analyzátory = nástroje jako **WAVE** (WebAIM), **axe** (Duque) nebo **Lighthouse**
    - Integrace do prohlížeče: WAVE, axe DevTools, nebo vestavěná Audits (Lighthouse) v Chrome
    - **Co automat zjistí?:**
      - Chybějící atributy alt u obrázků, chybějící label u inputů, nevalidní pořadí nadpisů atd.
      - cca 30-40% problémů se dá zachytit automaticky
      - Limity = neodhalí vše - např jestli alt dává smysl u obrázků

  - **Testování kontrastu a vizuálních aspektů**

    - Kontrast barev: poměr kontrastu textu vůči pozadí
    - Minimální kontrast pro běžný text: 4.5:1
    - Pro velký text: min 3:1
    - Nástroje: **WebAIM Contrast Checker**, **Accessible Colors**
    - **Barvy nesmí být jediným nosičem informace**

Příklad kontrastového poměru



- **Právní rámec a standardy**
  - **Evropská směrnice 2016/2102/EU**
    - Směrnice o přístupnosti webových stránek a mobilních aplikací veřejného sektoru
    - Ukládá povinnost veřejným institucím zajistit, aby jejich weby a aplikace splňovaly požadavky na přístupnost dle harmonizované normy
    - Směrnice se týká pouze subjektů veřejného sektoru
  - **Zákon č. 99/2019 Sb. (ČR)**
  - **Norma EN 301 549 a WCAG 2.1**